SEE: Signal Embedding Energy for Quantifying Noise Interference in Large Audio Language Models

Anonymous ACL submission

Abstract

Large Audio Language Models (LALMs) have been widely applied in real-time scenarios, such as in-car assistants and online meeting comprehension. In practice, audio inputs are often corrupted by device and environmental noise, leading to performance degradation. However, existing LALM studies on noise lack quantitative analysis and rely mainly on intuition and empirical observation, thus failing to understand practical robustness. To address this issue, we introduce **S**ignal **E**mbedding **E**nergy (**SEE**), a method for quantifying the impact of noise intensity on LALM inputs, enabling the differentiation of LALM robustness in real-world deployments. SEE introduces a perspective based on structured activation subspaces derived from the model's internal representations, which more accurately captures its perception of noise than raw audio features. Across experiments, SEE exhibits a strong correlation with LALM performance, achieving a correlation of 0.98. Surprisingly, traditional audio denoising methods are only marginally effective for LALMs, and, in some cases, even increase SEE and impair performance. This suggests a mismatch between speech-centric denoising objectives and the noise sensitivity of modern LALMs. Therefore, we propose a mitigation strategy derived from SEE to denoise LALM inputs, outperforming existing denoising methods. This paper introduces a novel metric for noise quantification in LALMs, providing guidance for robustness improvements in real-world deployments. Our code is publicly available at `https://anonymous.4open.science/r/SEEN-954F`.

1 Introduction

Large Audio Language Models (LALMs) are increasingly deployed in real-time applications, such as in-car assistants, where audio serves as the primary interface for language understanding and interaction (Rubenstein et al., 2023; Radford et al., 2023; Li et al., 2025c). In these real-world environments, device sampling imperfections and environmental noise corrupt the input waveform, turning a clean request into a signal with interference (Gelbart and Morgan, 2002; Deliyski et al., 2005). This corrupted signal is then fed into the LALMs, where the same noise that distorts the waveform also degrades the model's generation quality(Lin et al., 2025; Kumar and Mishra, 2025). While quantifying this effect in LALMs remains challenging, existing studies typically rely on task performance (Shen et al., 2024), a proxy that requires large-scale evaluation and is difficult to leverage for mitigation.

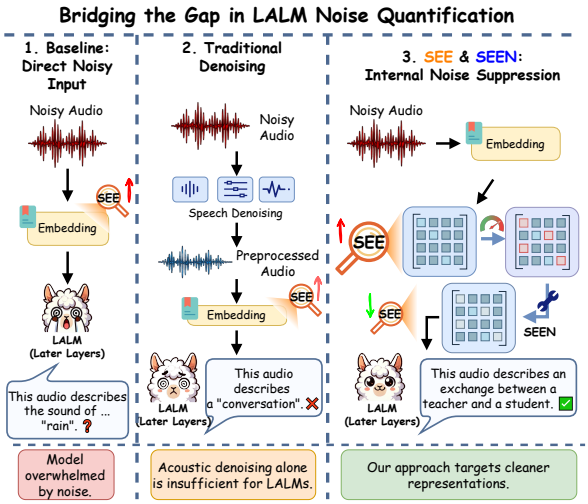

Figure 1: Motivation and overview of representation-level noise robustness in LALMs. Waveform-level denoising improves acoustic quality but may introduce semantic interference, which is quantified by Signal Embedding Energy (SEE) and mitigated by SEEN.

To maximize performance on benchmarks, many LALMs are trained and validated under comparatively idealized conditions (Chu et al., 2024b; Zhang et al., 2023). In real-world deployments, however, input audio is frequently corrupted by device artifacts and environmental noise, and the resulting impact on LALMs is still difficult to quan-

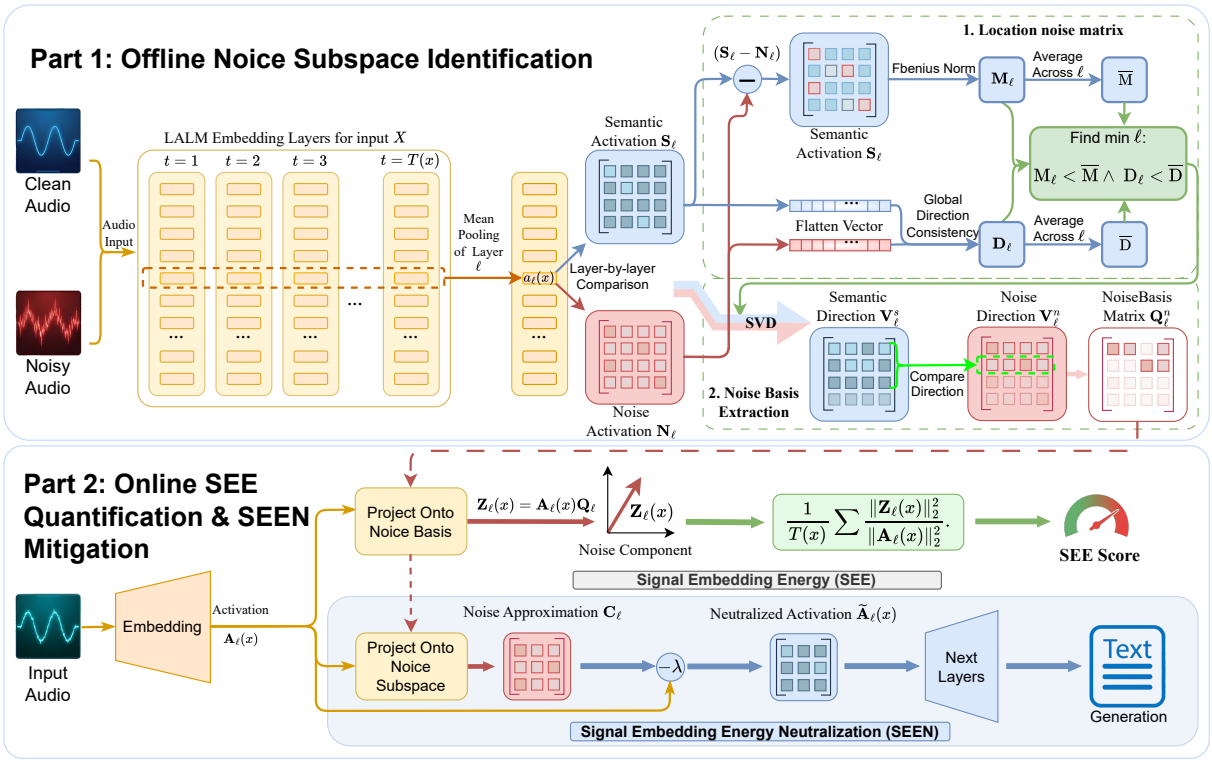

Figure 2: Offline, we construct noise activation matrices, localize noise dominant layers, and extract principal noise directions to form a noise subspace. Online, SEE quantifies activation energy projected onto this subspace, and SEEN mitigates noise by removing the projected components at the embedding layers.

tify in a principled way (Li et al., 2025b; Kumar and Mishra, 2025; Hou et al., 2025). Although incorporating additional real-world data or applying speech enhancement may improve robustness, the lack of quantitative evaluation makes it challenging to assess these interventions and compare models reliably (Goel et al., 2021; Hu et al., 2024). Existing studies often rely on outcome-level metrics such as performance on diverse tasks, which require large-scale benchmarking and provide limited guidance for diagnosing noise effects or informing mitigation (Yang et al., 2024; Wang et al., 2025). These limitations motivate a model-aware criterion that directly quantifies noise interference within LALMs.

To quantify semantic interference, we introduce **S**ignal **E**mbedding **E**nergy (**SEE**), a metric that measures noise intensity in the LALM embedding space. SEE leverages the activation space of LALMs to disentangle the semantic and noise components of the input embedding. By analyzing directional and magnitude discrepancies across layers, we identify the primary points at which noise distorts semantic processing. We then measure that captures the cumulative strength of noise activations, providing a quantification of semantic interference during generation. Because SEE captures embedding-level inference bias, it serves as a direct proxy for the impact of noise on model generation. Using SEE as a probe, we show that SEE increases monotonically with noise intensity and strongly correlates with generation quality, with a Pearson correlation (Benesty et al., 2009) coefficient of 0.98. Moreover, mainstream LALMs are vulnerable to real-world noise, and standard denoising pipelines often fail to reduce SEE and may even increase it, suggesting that acoustic denoising does not necessarily mitigate semantic interference.

Guided by SEE, we propose a defense that mitigates noise without any retraining. Concretely, we introduce **S**ignal **E**mbedding **E**nergy **N**eutralization (**SEEN**), which operates directly on the model's audio embeddings to minimize noise components. Because SEEN targets embedding interference rather than waveform energy, it aligns the mitigation objective with SEE. In experiments, SEEN improves accuracy by 6.7% over existing denoisings.

Our contributions are summarized as follows:

- We introduce **SEE**, a metric that quantifies the perturbation intensity in LALMs beyond traditional acoustic measures.

- We propose **SEEN**, a training-free gating that optimizes audio embedding and reduces misjudgments under noise conditions.
- We conduct extensive experiments to analyze the risks of acoustic mitigation and evaluate the effectiveness of our study.

2 Related Work

2.1 Large Audio Language Models

LALMs extend the reasoning capabilities of Large Language Models(LLMs) to the auditory domain by integrating pre-trained audio encoders with textual backbones (Radford et al., 2023; Chen et al., 2023). Technically, while LLMs operate on discrete text tokens, LALMs typically align high-dimensional continuous signals with semantic spaces through discrete tokenization (Lakhotia et al., 2021; Zhang et al., 2023) or cross-modal adapters (Tang et al., 2023; Chu et al., 2023). This architectural shift allows LALMs to capture paralinguistic cues and environmental contexts often inaccessible to text-only models, expanding application scenarios to sophisticated audio-centric reasoning. Representative frameworks include AudioPaLM (Rubenstein et al., 2023), Qwen-Audio (Chu et al., 2024a), MiniCPM-o (Yao et al., 2024), and StepAudio (Wu et al., 2025). Unlike cascaded pipelines, LALMs' end-to-end nature facilitates unified instruction following across diverse acoustic tasks (Gong et al., 2024; Peng et al., 2024).

2.2 Noise Robustness

Traditional speech robustness primarily focuses on mitigating acoustic mismatch (Li et al., 2015) via multi-condition training, data augmentation (Ko et al., 2015; Park et al., 2019), or enhancement front-ends (Wang and Chen, 2018) to minimize Word Error Rate (Watanabe et al., 2018). Conversely, the robustness of Large Audio Language Models (LALMs) centers on semantic reasoning stability and instruction-following integrity (Li et al., 2025a; Hou et al., 2025). Given that non-stationary noise frequently induces "semantic hallucinations" (Hou et al., 2025), research has pivoted from acoustic fidelity toward logical consistency and task reliability under environmental perturbations (Wang et al., 2025; Xiong et al., 2025; Gopal et al., 2025).

2.3 Speech Enhancement and Denoising Front-Ends

The paradigm of speech enhancement has shifted from statistical signal processing to data-driven deep learning. Early classical methods, such as Wiener Filtering (Lim, 1978), Spectral Subtraction (Boll, 2003), and MMSE-STSA (Ephraim and Malah, 2003), primarily targeted stationary noise through spectral estimation. Modern neural approaches have advanced this via spectral mapping (Xu et al., 2014), end-to-end time-domain modeling like Conv-TasNet (Luo and Mesgarani, 2019), and generative frameworks including GANs (Pascual et al., 2017) and diffusion models (Lu et al., 2022). However, traditional Speech Enhancement prioritizes acoustic fidelity for task performance, often neglecting the semantic coherence required by LALMs. Consequently, denoising artifacts may improve signal metrics while undermining multimodal alignment and reasoning.

3 Method

In this section, we formally present a framework for analyzing and mitigating noise disturbances in LALMs at the level of internal embedding activations. We first introduce Signal Embedding Energy (SEE), a metric that quantifies noise interference through semantic direction decomposition. Then, we present Signal Embedding Energy Neutralization (SEEN), a training-free strategy that subtracts the noise components and improves downstream generation quality.

3.1 Noise Substance Identification

Notation. We denote a LALM with L observable activation function layer, indexed by $\ell \in \{1, \ldots, L\}$. For an audio input $x \in X$, the activation at block ℓ is a matrix $\mathbf{A}_\ell(x) \in \mathbb{R}^{T(x) \times d_\ell}$, where $T(x)$ is the number of time steps and d_ℓ is the hidden width at layer ℓ.

We use two aligned input sets, containing a semantic (clean) request set $X^s = \{x_1^s, x_2^s, \cdots, x_m^s\}$ and a pure noise set $X^n = \{x_1^n, x_2^n, \cdots, x_m^n\}$, where m denotes the number of signal frames collected in the downstream application environment.

We keep dominant singular directions using threshold α; treat a direction as "noise-only" if its cosine similarity to any dominant semantic direction is below $\delta = 0.1$; and use $\varepsilon > 0$ as a small constant for numerical stability.

Separate the Noise Direction. Since audio inputs are represented as frame-level token sequences with different sequence lengths $T(x)$. To balance the differences in sample lengths, we apply mean pooling over the time dimension at layer ℓ:

$$\mathbf{a}_\ell(x) = \frac{1}{T(x)} \sum_{t=1}^{T(x)} \mathbf{A}_\ell(x)_{t,:} \in \mathbb{R}^{d_\ell}, \quad (1)$$

where $\mathbf{A}_\ell(x)_{t,:}$ denotes activation vector of the t-th frame token.

Next, we stack the pooled vectors across M samples to form a joint representation space:

$$\begin{aligned} \mathbf{S}_\ell &= \begin{bmatrix}\mathbf{a}_\ell(x_1^s) & \cdots & \mathbf{a}_\ell(x_M^s)\end{bmatrix}^\top \in \mathbb{R}^{M\times d_\ell}, \\ \mathbf{N}_\ell &= \begin{bmatrix}\mathbf{a}_\ell(x_1^n) & \cdots & \mathbf{a}_\ell(x_M^n)\end{bmatrix}^\top \in \mathbb{R}^{M\times d_\ell}. \end{aligned} \quad (2)$$

Here $\mathbf{S}_\ell$ is the semantic (clean) activation matrix and $\mathbf{N}_\ell$ is the noise activation matrix at layer ℓ.

Location Noise Matrix. To pinpoint the location where noise dominantly affects the encoding tendency, we characterize interference in two complementary aspects: magnitude and direction.

To identify where noise begins to alter semantic processing, we compute Frobenius Norm (Böttcher and Wenzel, 2008) as the overall discrepancy energy using the difference matrix:

$$\mathrm{M}_\ell = \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{d_\ell}(\mathbf{S}_\ell(i,j) - \mathbf{N}_\ell(i,j))^2}, \quad (3)$$

here $\mathbf{S}_\ell(i,j)$ (resp. $\mathbf{N}_\ell(i,j)$) denotes the value of the pooled activation vector at feature dimension j for the i-th clean (resp. noise) input at layer ℓ. We then use $\mathrm{vec}(\cdot)$ to flatten the matrix into a vector in $\mathbb{R}^{Md_\ell\times 1}$ and compute the global direction consistency:

$$\mathrm{D}_\ell = \frac{\|\mathrm{vec}(\mathbf{S}_\ell)^\top \mathrm{vec}(\mathbf{N}_\ell)\|_2}{\|\mathrm{vec}(\mathbf{S}_\ell)\|_2 \, \|\mathrm{vec}(\mathbf{N}_\ell)\|_2 + \varepsilon}, \quad (4)$$

we use $\|\cdot\|_2$ to denote the Euclidean (L2) norm of a vector. And its average value $\overline{\mathrm{M}} = \frac{1}{L}\sum_{\ell=1}^{L}\mathrm{M}_\ell$ and $\overline{\mathrm{D}} = \frac{1}{L}\sum_{\ell=1}^{L}\mathrm{D}_\ell$ can be obtained.

Using these two indicators, the primary locations for noise monitoring are pinpointed:

$$\ell^\star = \min\{\ell \mid \mathrm{M}_\ell > \overline{\mathrm{M}} \wedge \mathrm{D}_\ell > \overline{\mathrm{D}}\}, \quad (5)$$

$\ell^\star$ typically occurs in the later layers of the model. Let $\mathcal{L}^\star = \{\ell^\star, \ell^\star+1, \ldots, L\}$ denote the set of layers at which the basis is retained and used for identification.

Noise Basis Extraction. For each $\ell \in \mathcal{L}^\star$, the semantic matrix $\mathbf{S}_\ell$ and noise matrix $\mathbf{N}_\ell$ are decomposed via singular value decomposition (SVD) (Stewart, 1993) as $\mathbf{S}_\ell = \mathbf{U}_\ell^s \mathbf{\Sigma}_\ell^s (\mathbf{V}_\ell^s)^\top$ and $\mathbf{N}_\ell = \mathbf{U}_\ell^n \mathbf{\Sigma}_\ell^n (\mathbf{V}_\ell^n)^\top$, respectively.

Here $\mathbf{U}_\ell^s, \mathbf{U}_\ell^n \in \mathbb{R}^{M\times M}$ span the sample space, $\mathbf{V}_\ell^s, \mathbf{V}_\ell^n \in \mathbb{R}^{d_\ell\times d_\ell}$ span the hidden space, and $\mathbf{\Sigma}_\ell^s, \mathbf{\Sigma}_\ell^n \in \mathbb{R}^{M\times d_\ell}$ store singular values in descending order. Importantly, because our goal is to identify directions in the hidden space, we focus on the right singular vectors in V. Each column $\mathbf{v}_{\ell,j}^s \in \mathbb{R}^{d_\ell}$ (resp. $\mathbf{v}_{\ell,j}^n$) represents a principal direction along which semantic (resp. noise) independent activations.

Let $\sigma_{\ell,j}^s$ (resp. $\sigma_{\ell,j}^n$) denote the j-th singular value associated with $\mathbf{v}_{\ell,j}^s$ (resp. $\mathbf{v}_{\ell,j}^n$). Larger singular values indicate directions that contain stronger similarity information at layer ℓ. We retain dominant semantic $\mathcal{I}_\ell^s = \{j \mid \sigma_{\ell,j}^s > \alpha\}$ and noise information $\mathcal{I}_\ell^n = \{j \mid \sigma_{\ell,j}^n > \alpha\}$ via threshold α. In subsequent steps, we only use the corresponding right singular vectors $\{\mathbf{v}_{\ell,j}^s\}_{j\in\mathcal{I}_\ell^s}$ and $\{\mathbf{v}_{\ell,j}^n\}_{j\in\mathcal{I}_\ell^n}$ to construct the noise-only subspace for SEE.

For every noise direction index $j \in \mathcal{I}_\ell^n$, we compute its cosine similarity with each semantic direction $k \in \mathcal{I}_\ell^s$, and aggregate them by the maximum absolute similarity:

$$\mathbf{m}_{\ell,j} = \max_{k\in\mathcal{I}_\ell^s}(\cos(\mathbf{v}_{\ell,j}^n, \mathbf{v}_{\ell,k}^s)). \quad (6)$$

A dominant noise direction is retained if it is nearly orthogonal to all dominant semantic directions: $\mathcal{J}_\ell = \left\{ j \in \mathcal{I}_\ell^n \;\middle|\; \mathbf{m}_{\ell,j} < \delta \right\}$.

Define a binary mask vector $s_\ell \in \{0,1\}^{d_t}$ with $s_\ell[j] = 1$ if $j \in \mathcal{J}_\ell$, and a diagonal mask matrix $\mathbf{M}_\ell = \mathrm{diag}(s_\ell) \in \mathbb{R}^{d_\ell\times d_\ell}$. Then we get noise basis matrix:

$$\mathbf{Q}_\ell = \mathbf{V}_\ell^n \mathbf{M}_\ell \in \mathbb{R}^{d_\ell\times r_\ell}. \quad (7)$$

3.2 SEE Calculation

For each audio input x, we project token activations onto the noise directions for each $\ell \in \mathcal{L}^\star$:

$$\mathbf{Z}_\ell(x) = \mathbf{A}_\ell(x)\mathbf{Q}_\ell \in \mathbb{R}^{T(x)\times r_\ell}. \quad (8)$$

For each frame token t, we define the layer SEE component as the energy:

$$\mathrm{SEE}_\ell(x) = \frac{1}{T(x)} \sum_{t=1}^{T(x)} \frac{\|\mathbf{Z}_\ell(x)_{t,:}\|_2^2}{\|\mathbf{A}_\ell(x)_{t,:}\|_2^2 + \varepsilon}. \quad (9)$$

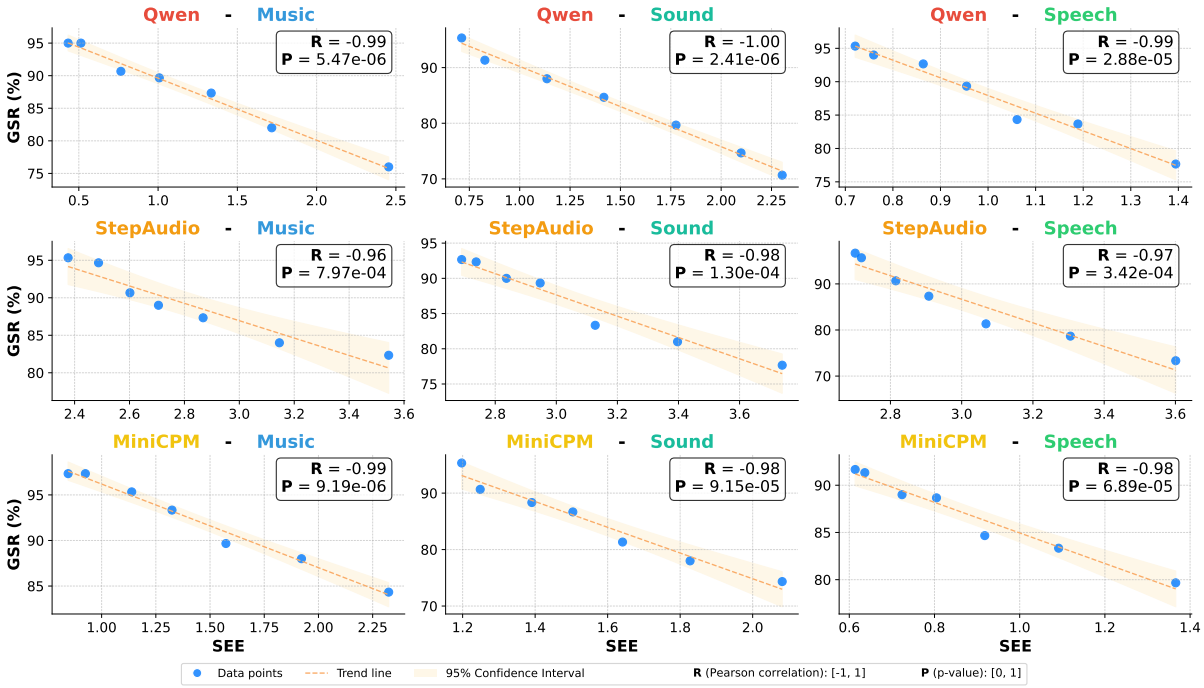

Figure 3: SEE is negatively correlated with GSR, with a consistent trend observed across different task types.

Finally, we aggregate over retained layers to form a single score for the input:

$$\mathrm{SEE}(x) = \frac{1}{|\mathcal{L}^\star|} \sum_{\ell \in \mathcal{L}^\star} \mathrm{SEE}_\ell(x), \tag{10}$$

$\mathrm{SEE}(x)$ measures the embedding energy aligned with the noise subspace, and thus directly quantifies noise interference inside the model.

3.3 Signal Embedding Energy Neutralization (SEEN) Construction

While SEE quantifies how strongly an input activates noise directions, SEEN uses the same noise subspace to remove the corresponding components from intermediate activations, aiming to reduce noise bias without changing model parameters or requiring additional training data.

Given the retained layer set $\mathcal{L}^\star$ and the noise matrices $\{\mathbf{Q}_\ell\}_{\ell \in \mathcal{L}^\star}$, we manipulate the activation in the audio embedding. For an input x and layer $\ell \in \mathcal{L}^\star$, we first reconstruct the noise component in the original hidden space by projecting $\mathbf{A}_\ell(x) \in \mathbb{R}^{T(x) \times d_\ell}$ onto the noise subspace:

$$\mathbf{C}_\ell(x) = \mathbf{A}_\ell(x) \mathbf{Q}_\ell \mathbf{Q}_\ell^\top \in \mathbb{R}^{T(x) \times d_\ell}. \tag{11}$$

SEEN then neutralizes the activation by subtracting the projected component:

$$\widetilde{\mathbf{A}}_\ell(x) = \mathrm{SEEN}(x) = \mathbf{A}_\ell(x) - \lambda \mathbf{C}_\ell(x), \quad \lambda \in [0, 1],\ \ell \in \mathcal{L}^\star. \tag{12}$$

Here λ controls the neutralization strength (the default setting is $\lambda = 1$).

The wthen continues the forward pass using $\widetilde{\mathbf{A}}_\ell(x)$. In this way, SEEN provides an effective and lightweight means to align noise mitigation with SEE, thereby reducing internal perturbation while preserving the remaining semantic components needed for generation.

4 Experiments

4.1 Setups

Models. We conduct experiments across 3 models, including Qwen (`Qwen-2.5-omni-7b`) (Chu et al., 2024a), MiniCPM (`Minicpm-o-2.6`) (Yao et al., 2024), and StepAudio (`Step-Audio-2-mini`) (Wu et al., 2025). Additional experimental details are provided in the Appendix A

Datasets. In the experiments, we primarily evaluate our method on the MMAU (Kumar et al., 2025) and Librispeech (Panayotov et al., 2015) datasets. Based on the underlying audio modality, the tasks are grouped into four categories: speech-to-text (STT), Speech, Sound, and Music. STT evaluates word-level recognition. Speech focuses on world-knowledge QA. Sound assesses environmental sound perception, while music examines multicultural music reasoning.

For noise settings, we employ randomly generated white Gaussian noise (Gauss) (Ko et al.,

Noise	Gauss	Crowd	Traffic	Machine	Animal	Shower	Wind
Clean	1.212	1.212	1.212	1.212	1.212	1.212	1.212
10db	1.767	1.709	1.633	2.106	1.462	1.503	1.402
0db	2.052	2.287	2.324	2.808	1.544	1.536	1.551
-10db	2.436	3.101	3.481	3.306	1.797	1.586	1.921

Table 1: The effect of SEE generated with mixed noise types. The underlined items are known noise types.

Dataset	Method	Gauss	Crowd	Machine	Traffic
Music	**Clean**	0.35	1.32	1.30	1.17
	Noise	2.45	5.97	11.15	5.84
Sound	**Clean**	0.67	2.12	2.47	1.85
	Noise	2.72	5.71	10.87	5.30
Speech	**Clean**	0.66	3.12	3.54	3.28
	Noise	1.63	4.79	6.74	4.14

Table 2: SEE separation across noise types.

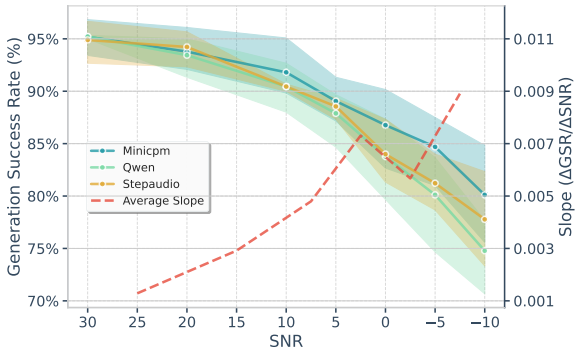

Figure 4: Generation success rate (GSR) versus SNR, showing a sharp degradation below SNR $= 10$.

2015), together with noise categories from the PNL dataset (Hu and Wang, 2010), including crowd noise (Crowd), mechanical noise (Machine), and vehicle noise (Traffic).

Baselines. We measure the deviation between model outputs under normal (Clean) and noisy (Noise) conditions as our metric, rather than ground-truth accuracy.

For traditional audio denoising, we select two frequency-based and two model-based methods as baselines. Specifically, STFT (Ephraim and Malah, 2003) and WT (Donoho, 2002) represent frequency-based approaches, while Segan (Pascual et al., 2017) and DFL (Purwins et al., 2019) are adopted as model-based baselines.

Metrics. Noise intensity is standardized using SNR, set to -10 dB for unlabeled experiments. We use Generation Success Rate (GSR) as the primary metric, measuring noisy sample accuracy relative to clean inputs. The SEE index's validity is assessed via Pearson correlation (Benesty et al., 2009) with GSR and associated p-values.

4.2 Quantifying Noise Intensity with SEE

We evaluate Signal Embedding Energy (SEE) across three LALM frameworks to assess their robustness to noise. As shown in Figure 3, GSR degrades consistently as SEE increases, with strong Pearson correlations ($R \in [-0.96, -1.00], P \ll 0.001$), indicating that SEE is tightly coupled with performance drop under corruption. This confirms SEE as a model-centric metric that captures embedding-space interference to predict generation failure. Additional empirical evidence is provided in the Appendix B

The above analysis establishes SEE as an internal indicator; we also evaluate its robustness against a standard physical SNR scale to validate the selected GSR interval. Figure 4 shows robustness versus SNR. Across all LALMs, GSR degrades monotonically as SNR decreases, falling more steeply below $\approx$ 10dB, which marks an SNR sensitivity regime. The slope curve confirms this transition via larger marginal losses at low SNR. This physical turning point aligns with the GSR range where SEE rises in our previous analysis, confirming that our selected interval effectively captures noise-induced failures.

4.3 Effectiveness of SEE Across Noise Types

We evaluated SEE across noise types to verify generalizability. Results on Qwen (Table 2) show a clear gap between clean and noisy conditions. Specifically, clean and noisy inputs exhibit non-overlapping SEE ranges across request types, with SEE increasing for all noise categories.

Furthermore, we evaluate the applicability of SEE across diverse noise categories and optimize SEE under four representative noise types (Gauss, Crowd, Machine, and Traffic). Table 1 shows consistent SEE values across these categories at the same SNR, suggesting SEE is driven by noise in-

Dataset	Method	MiniCPM		Qwen		StepAudio	
		Noise	Clean	Noise	Clean	Noise	Clean
Music	STFT	84.33%	93.33%	71.67%	90.33%	79.00%	92.67%
	WT	82.00%	94.67%	67.67%	91.00%	77.67%	94.00%
	Segan	82.67%	89.67%	71.00%	86.33%	78.00%	87.33%
	DFL	82.00%	88.33%	67.33%	76.67%	75.67%	85.33%
	SEEN	**85.00%**	**99.00%**	**76.99%**	**97.67%**	**82.67%**	**98.00%**
Sound	STFT	72.00%	86.67%	64.33%	88.33%	76.00%	87.67%
	WT	70.33%	92.33%	63.00%	91.67%	71.33%	92.33%
	Segan	73.00%	87.00%	58.67%	83.33%	73.67%	87.00%
	DFL	73.67%	81.33%	60.00%	79.33%	72.33%	83.00%
	SEEN	**75.27%**	**98.67%**	**72.00%**	**99.67%**	**78.33%**	**99.00%**
Speech	STFT	75.33%	94.00%	68.00%	97.00%	66.33%	93.33%
	WT	70.67%	93.00%	68.00%	97.00%	55.67%	94.67%
	Segan	69.00%	93.00%	64.67%	91.33%	60.67%	90.00%
	DFL	72.67%	93.00%	65.33%	92.00%	62.00%	92.33%
	SEEN	**80.00%**	**98.00%**	**78.33%**	**99.00%**	**74.00%**	**96.33%**

Table 3: Generation success rate under noisy conditions with different denoising strategies.

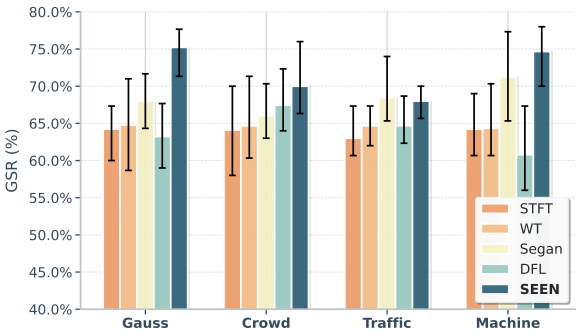

Figure 5: SEEN improves generation quality across noise types without consistent performance degradation.

Model	Dataset	Early	Middle	ALL	SEEN
MiniCPM	Clean	0.002	0.004	0.004	0.763
	Noise	0.126	0.270	0.456	1.777
Qwen	Clean	0.001	0.000	0.015	0.633
	Noise	0.024	0.005	1.751	1.972
StepAudio	Clean	0.005	0.022	0.092	2.489
	Noise	0.312	0.661	5.233	3.878

Table 4: Ablation on layer selection for SEE.

tensity rather than specific acoustic traits.

To further examine the generalization capability, we introduce three additional noise categories that are not included in the optimization process. Table 1 reported slightly lower SEE values compared to those obtained on directly optimized noise types, likely due to differing spectral and temporal structures. However, SEE still increases monotonically as SNR drops. This preserved monotonicity confirms that SEE remains sensitive to noise intensity even under unseen conditions. More detailed results are reported in Appendix E

4.4 SEE on traditional denoising methods

We use SEE to analyze preprocessed samples. As shown in Figure 6, denoised inputs generally exhibit higher SEE than raw inputs. This contrasts with waveform-level evidence that these methods reduce acoustic noise, highlighting a mismatch between denoising objectives and LALM semantic encoding. Quantitatively, the SEE of denoised audio is comparable to inputs with an additional SNR $= 0$, suggesting that denoising artifacts act like extra noise within the model's internal representations.

Moreover, denoising causes unique waveform misalignments with clean embeddings. While SEE captures residual interference, these distortions extend beyond the original noise. This indicates added variability in the encoding space that may hinder robustness-oriented training. More results are reported in Appendices C and F

4.5 Suppressing Noise with SEEN

Table 3 evaluates SEEN's impact on generation quality. In clean or low-noise settings, SEEN causes negligible performance drops and preserves semantic fidelity, confirming that suppressing noise-related directions does not harm normal requests. Under noise, SEEN consistently improves GSR across all datasets and models, outper-

Model	MiniCPM	Qwen	StepAudio
None	84.33%	76.00%	82.33%
0.25	84.67%	76.33%	82.67%
0.5	84.33%	76.33%	82.67%
0.75	85.00%	76.67%	82.67%
1	**85.00%**	**77.00%**	**82.67%**
1.2	85.00%	76.67%	82.67%

Table 5: Sensitivity of SEEN to the neutralization strength λ.

Model	MiniCPM	Qwen	StepAudio
None	84.33%	76.00%	82.33%
0.1	**85.00%**	**77.00%**	**82.67%**
0.3	84.33%	77.00%	82.33%
0.5	83.33%	77.33%	82.33%
0.7	84.33%	77.33%	82.33%
1	83.67%	77.33%	82.33%

Table 6: Sensitivity of SEEN to the cosine threshold δ.

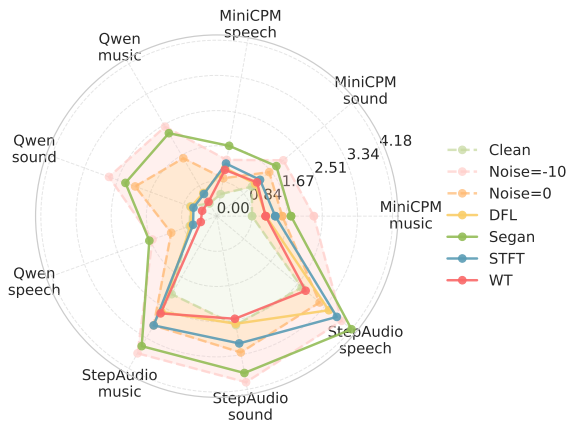

Figure 6: Effect of waveform denoising measured by SEE. Many conventional denoisers fail to reduce SEE and can even increase it. Lower SEE indicates less noise-aligned interference.

forming traditional and learned denoisers by 6.7% on average. These results suggest that embedding-space mitigation is more effective than acoustic suppression, as it avoids new distortions while targeting noise-induced semantic bias. Appendix H provides more analysis.

We also assess SEEN's robustness across noise types on three MMAU datasets. As shown in Table 5, SEEN maintains effective noise suppression across all tested conditions. Additional results of the SEEN method under various SNR settings are reported in Appendix D.

4.6 Ablation Analysis

Layer Selection for Noise Monitoring. We first ablate the layer-selection strategy used in SEE. To evaluate its necessity, we construct three alternatives: (i) applying the method to the first third of layers (Early), (ii) only to the middle third of layers (Middle), and (iii) to all layers(ALL).

Table 4 shows that restricting SEE to the early layers or middle layers does not produce significant results. This is because screening at these layers produces few valid directions, leading to an unstable detection. Applying the method to all layers is feasible. However, in different models, the variation magnitude becomes unstable and can be influenced by interference from early and middle layers. Our method is more closely aligned with the extracted noise subspace and better preserves cross-modal embedding consistency.

Sensitivity to Hyperparameters. We next study the sensitivity of SEEN to two key hyperparameters: the suppression strength λ and the cosine similarity threshold δ. Table 5 shows that larger values of λ lead to more effective suppression, and that strengthening noise-direction suppression improves generation quality. However, when the suppression ratio exceeds 1, components opposite to the noise direction are introduced, causing additional semantic perturbations. Table 6 reports the effect of cosine similarity. In most cases, amplifying the δ interferes with the main semantic direction, ultimately degrading performance.

5 Conclusion

We introduce Signal Embedding Energy (SEE), a metric to quantify real-world noise impact. SEE is based on structured activation subspaces derived from the model's internal representations, enabling a more accurate characterization of noise perception than clean audio features. Using SEE as a probe, we show that state-of-the-art LALMs are highly sensitive to noise. More critically, conventional speech denoising techniques often fail to reduce SEE, revealing a semantic misalignment between acoustic suppression and LALM embedding-level reasoning. Driven by SEE, we propose Signal Embedding Energy Neutralization (SEEN), a training-free method that removes noise subspaces from embedding activations. SEEN consistently reduces SEE and outperforms existing denoising methods. Overall, our study highlights the need for semantic-level criteria and offering a principled direction for future LALM design.

Limitations

Our approach assumes access to aligned clean requests and pure-noise recordings from the target deployment environment to estimate a stable noise subspace. In practice, such "noise-only" collections cannot be directly acquired during training; they require collecting noise from the actual application environment. Downstream application environments with relatively single or few categories of noise are easier to collect and are suitable for SEE measurement. Highly variable environments can increase the difficulty of obtaining pure-noise recordings.

Methodologically, our current SEE compresses variable length via mean pooling, which may underestimate temporally corrupted and paralinguistic cues. While effective in our tested regimes, overly aggressive neutralization could remove task-relevant information under certain acoustic conditions. A better approach would not be to eliminate noise semantic components, but rather to enhance the model's understanding of incomplete task semantics.

A promising direction is to use SEE as a training robustness signal rather than only a mitigation tool. This can be achieved by incorporating SEE as a regularizer under noise augmentation to explicitly penalize noise-aligned energy in embeddings, or by using SEE to select or learn embedding-space enhancement modules that reduce semantic interference without relying solely on acoustic fidelity. We leave this direction for future work.

References

Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer.

Steven Boll. 2003. Suppression of acoustic noise in speech using spectral subtraction. *IEEE Transactions on acoustics, speech, and signal processing*, 27(2):113–120.

Albrecht Böttcher and David Wenzel. 2008. The frobenius norm and the commutator. *Linear algebra and its applications*, 429(8-9):1864–1885.

Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, Wanxiang Che, Xiangzhan Yu, and Furu Wei. 2023. Beats: Audio pre-training with acoustic tokenizers. In *International Conference on Machine Learning*, pages 5178–5193. PMLR.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, Chang Zhou, and Jingren Zhou. 2024a. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, and 1 others. 2024b. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. 2023. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*.

Dimitar D Deliyski, Heather S Shaw, and Maegan K Evans. 2005. Adverse effects of environmental noise on acoustic voice quality measurements. *Journal of Voice*, 19(1):15–28.

David L Donoho. 2002. De-noising by soft-thresholding. *IEEE transactions on information theory*, 41(3):613–627.

Yariv Ephraim and David Malah. 2003. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. *IEEE Transactions on acoustics, speech, and signal processing*, 32(6):1109–1121.

David Gelbart and Nelson Morgan. 2002. Double the trouble: handling noise and reverberation in far-field automatic speech recognition. In *Interspeech*, pages 2185–2188.

Karan Goel, Nazneen Rajani, Jesse Vig, Zachary Taschdjian, Mohit Bansal, and Christopher Ré. 2021. Robustness gym: Unifying the nlp evaluation landscape. *NAACL-HLT 2021*, page 42.

Yuan Gong, Hongyin Luo, Alexander H Liu, Leonid Karlinsky, and James Glass. 2024. Listen, think, and understand. In *International Conference on Learning Representations*.

Shreyas Gopal, Ashutosh Anshul, Haoyang Li, Yue Heng Yeo, Hexin Liu, and Eng Siong Chng. 2025. Explainable disentanglement on discrete speech representations for noise-robust asr. *arXiv preprint arXiv:2510.25150*.

Guanyu Hou, Jiaming He, Yinhang Zhou, Ji Guo, Yitong Qiao, Rui Zhang, and Wenbo Jiang. 2025. Evaluating robustness of large audio language models to audio injection: An empirical study. *arXiv preprint arXiv:2505.19598*.

Guoning Hu and DeLiang Wang. 2010. A tandem algorithm for pitch estimation and voiced speech segregation. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(8):2067–2079.

Yuchen Hu, Chen Chen, Chao-han Huck Yang, Ruizhe Li, Chao Zhang, Pin-Yu Chen, and Ensiong Chng. 2024. Large language models are efficient learners of noise-robust speech recognition. In *International Conference on Learning Representations*.

Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. 2015. Audio augmentation for speech recognition. In *Proc. Interspeech 2015*, pages 3586–3589.

Pankaj Kumar and Subhankar Mishra. 2025. Robustness in large language models: A survey of mitigation strategies and evaluation metrics. *arXiv preprint arXiv:2505.18658*.

Sonal Kumar, Šimon Sedláček, Vaibhavi Lokegaonkar, Fernando López, Wenyi Yu, Nishit Anand, Hyeonggon Ryu, Lichang Chen, Maxim Plička, Miroslav Hlaváček, and 1 others. 2025. Mmau-pro: A challenging and comprehensive benchmark for holistic evaluation of audio general intelligence. *arXiv preprint arXiv:2508.13992*.

Kushal Lakhotia, Eugene Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Abdelrahman Mohamed, and 1 others. 2021. On generative spoken language modeling from raw audio. *Transactions of the Association for Computational Linguistics*, 9:1336–1354.

Bohan Li, Wenbin Huang, Yuhang Qiu, Yiwei Guo, Hankun Wang, Zhihan Li, Jing Peng, Ziyang Ma, Xie Chen, and Kai Yu. 2025a. Isa-bench: Benchmarking instruction sensitivity for large audio language models. *arXiv preprint arXiv:2510.23558*.

Chen-An Li, Tzu-Han Lin, and Hung-yi Lee. 2025b. When silence matters: The impact of irrelevant audio on text reasoning in large audio-language models. *arXiv preprint arXiv:2510.00626*.

Jing Li, Jingyuan Li, Guo Yang, Lie Yang, Haozhuang Chi, and Lichao Yang. 2025c. Applications of large language models and multimodal large models in autonomous driving: A comprehensive review. *Drones*, 9(4):238.

Jinyu Li, Li Deng, Reinhold Haeb-Umbach, and Yifan Gong. 2015. *Robust automatic speech recognition: a bridge to practical applications*. Academic Press.

Jae Lim. 1978. All-pole modeling of degraded speech. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(3):197–210.

Liang Lin, Miao Yu, Kaiwen Luo, Yibo Zhang, Lilan Peng, Dexian Wang, Xuehai Tang, Yuanhe Zhang, Xikang Yang, Zhenhong Zhou, and 1 others. 2025. Hidden in the noise: Unveiling backdoors in audio llms alignment through latent acoustic pattern triggers. *arXiv preprint arXiv:2508.02175*.

Yen-Ju Lu, Zhong-Qiu Wang, Shinji Watanabe, Alexander Richard, Cheng Yu, and Yu Tsao. 2022. Conditional diffusion probabilistic model for speech enhancement. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7402–7406. Ieee.

Yi Luo and Nima Mesgarani. 2019. Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, 27(8):1256–1266.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 5206–5210. IEEE.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. In *Interspeech*, pages 2613–2617.

Santiago Pascual, Antonio Bonafonte, and Joan Serrà. 2017. Segan: Speech enhancement generative adversarial network. In *Proc. Interspeech 2017*, pages 3642–3646.

Jing Peng, Yucheng Wang, Yangui Fang, Yu Xi, Xu Li, Xizhuo Zhang, and Kai Yu. 2024. A survey on speech large language models. *arXiv preprint arXiv:2410.18908*.

Hendrik Purwins, Bo Li, Tuomas Virtanen, Jan Schlüter, Shuo-Yiin Chang, and Tara Sainath. 2019. Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):206–219.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, and 1 others. 2023. Audiopalm: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925*.

Yongliang Shen, Kaitao Song, Xu Tan, Wenqi Zhang, Kan Ren, Siyu Yuan, Weiming Lu, Dongsheng Li, and Yueting Zhuang. 2024. Taskbench: Benchmarking large language models for task automation. *Advances in Neural Information Processing Systems*, 37:4540–4574.

Gilbert W Stewart. 1993. On the early history of the singular value decomposition. *SIAM review*, 35(4):551–566.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2023. Salmonn: Towards generic hearing abilities for large language models. *arXiv preprint arXiv:2310.13289*.

Bin Wang, Xunlong Zou, Geyu Lin, Shuo Sun, Zhuohan Liu, Wenyu Zhang, Zhengyuan Liu, AiTi Aw, and Nancy Chen. 2025. Audiobench: A universal benchmark for audio large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4297–4316.

DeLiang Wang and Jitong Chen. 2018. Supervised speech separation based on deep learning: An overview. *IEEE/ACM transactions on audio, speech, and language processing*, 26(10):1702–1726.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, and 1 others. 2018. Espnet: End-to-end speech processing toolkit. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, volume 2018, pages 2207–2211.

Boyong Wu, Chao Yan, Chen Hu, Cheng Yi, Chengli Feng, Fei Tian, Feiyu Shen, Gang Yu, Haoyang Zhang, Jingbei Li, Mingrui Chen, Peng Liu, Wang You, Xiangyu Tony Zhang, Xingyuan Li, Xuerui Yang, Yayue Deng, Yechang Huang, Yuxin Li, and 90 others. 2025. Step-audio 2 technical report. *Preprint*, arXiv:2507.16632.

Zhen Xiong, Yujun Cai, Zhecheng Li, Junsong Yuan, and Yiwei Wang. 2025. Thinking with sound: Audio chain-of-thought enables multimodal reasoning in large audio-language models. *arXiv preprint arXiv:2509.21749*.

Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee. 2014. A regression approach to speech enhancement based on deep neural networks. *IEEE/ACM transactions on audio, speech, and language processing*, 23(1):7–19.

Qian Yang, Jin Xu, Wenrui Liu, Yunfei Chu, Ziyue Jiang, Xiaohuan Zhou, Yichong Leng, Yuanjun Lv, Zhou Zhao, Chang Zhou, and 1 others. 2024. Air-bench: Benchmarking large audio-language models via generative comprehension. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1979–1998.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, and 1 others. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2023. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15757–15773.

A More Experimental Setups

In this section, we provide a comprehensive description of the experimental configurations used in our study, covering both the classification and transcription tasks.

A.1 Model Architecture and Audio Processing

All experiments utilize an omni-modal Large Language Model (LLM) loaded in `bfloat16` precision. We leverage the `sdpa` (Scaled Dot Product Attention) implementation for efficient computation. For acoustic input, all raw audio signals are resampled to 16,000 Hz and converted to mono-channel using the `librosa` library. All experiments are conducted on a server equipped with an NVIDIA RTX 5090 GPU.

A.2 Task Definitions and Prompting

We evaluate our method on two distinct types of tasks:

1. **Multiple-Choice Question Answering (MCQA)**: Applied to *Music*, *Sound*, and *Speech* datasets. The model is prompted to answer multiple-choice questions with a strict constraint: *"Answer the multiple-choice question by outputting ONLY one letter."* A regex pattern `r'([A-Za-z])'` is used to parse the final answer.

2. **Speech-to-Text(STT)**: Applied to the *LibriSpeech* dataset. The model is tasked with transcription using the prompt: *"Please listen to the audio snippet carefully and transcribe the content."*

A.3 Noise Synthesis and Dataset Scale

To simulate real-world interference, we inject various types of environmental noise into the clean audio samples at seven SNR levels: $\{-10, -5, 0, 5, 10, 20, 30\}$. The sample sizes are set to $N = 300$ for MCQA task and $N = 100$ for STT task to ensure statistically significant results.

A.4 SEE Intervention and Hyperparameters

The SEE is implemented by registering forward hooks on the target layers. The set of target layers for intervention is identified through the diagnostic procedure outlined in Algorithm 1.

- **Target Layers**: The suppression is applied to the encoder layers of models. For Qwen, target layer is from layer 23 to the final layer (`model.thinker.audio_tower.layers[23:]`). For MiniCPM, target layer is from layer 18 to the final layer (`model.apm.layers[18:]`). For StepAudio, target layer is from layer 27 to the final layer (`model.llm.encoder.blocks[27:]`)

- **Suppression Factor**: The intervention strength $\alpha = 1.0$. Cosine similarity threshold $\delta = 0.1$. The energy threshold ratio used to determine the number of retained singular values $\rho = 0.95$.

- **Noise Subspace (V_{noise})**: The noise singular vectors are derived during an offline calibration phase using a representative set of 50 pure-noise segments sampled from the speech dataset, with the intensity standardized to a reference level equivalent to 0 dB SNR.

A.5 About Baseline Methods

To ensure a fair and rigorous comparison, all traditional baseline methods included in this study were implemented and configured in strict accordance with the parameter settings and experimental protocols specified in their original publications. No additional hyperparameter tuning was performed on these baselines beyond the recommendations provided by the respective authors to maintain the integrity of the comparative analysis.

B Supplementary Analysis of SEE

SEE Is Stable on Clean Audio. We evaluate Signal Embedding Energy (SEE) across three LALM frameworks. Figure 7 shows that,for clean audio, SEE remains stable within each model and exhibits only minor variation across categories, suggesting that the metric is largely insensitive to content differences when the input is clean. The clean SEE embeddings are concentrated in a relatively compact region, with limited activation along the noise-aligned subspace.

SEE Tracks Noise-induced Degradation. As the signal-to-noise ratio (SNR) decreases, the injected noise becomes stronger and SEE increases monotonically, reflecting progressively larger projection onto the noise directions. The corresponding impact on generation is summarized in Figure 4: once SNR falls below 10 dB, task performance drops by more than 10%, and under severe corruption SNR $= -10$ accuracy degrades by up to 25%. This degradation aligns closely with the behavior of SEE, which rises sharply in the same SNR

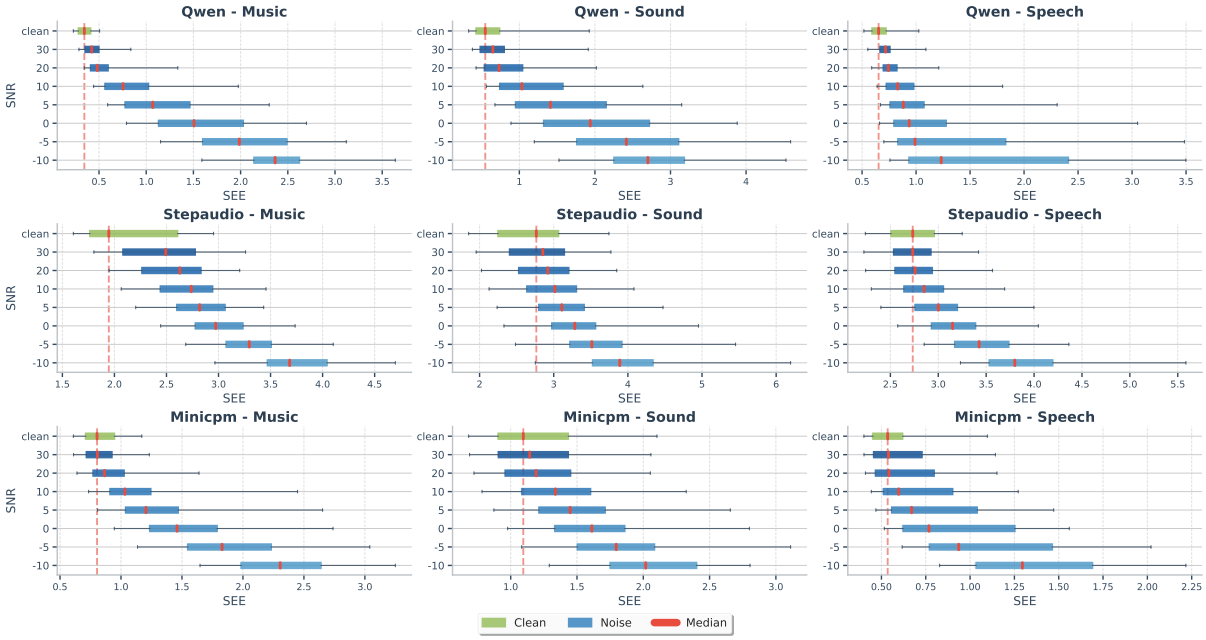

Figure 7: SEE under varying noise intensity across three LALMs. Clean inputs yield stable SEE, while SEE increases monotonically as SNR decreases. Boxplots report the interquartile range and median.

Method	MiniCPM				Qwen				StepAudio			
	STT	Music	Sound	Speech	STT	Music	Sound	Speech	STT	Music	Sound	Speech
BASE	0.481	0.834	1.176	0.561	0.854	0.354	0.669	0.657	2.409	2.137	2.676	2.735
Noise	1.378	2.306	2.064	1.357	1.082	2.455	2.725	1.625	3.872	3.763	4.007	3.871
STFT	1.416	1.399	1.347	1.318	0.607	0.604	0.599	0.600	4.442	3.049	3.025	3.658
WT	1.188	1.213	1.255	1.153	0.442	0.391	0.369	0.404	3.001	2.667	2.476	2.756
Segan	1.906	1.816	1.910	1.682	2.051	2.330	2.270	1.675	5.078	3.829	3.698	4.014
DFL	1.237	1.194	1.206	1.171	0.604	0.614	0.647	0.683	3.628	2.703	2.714	3.406
SEEN	**0.008**	**0.017**	**0.016**	**0.009**	**0.007**	**0.015**	**0.017**	**0.010**	**0.040**	**0.044**	**0.047**	**0.039**

Table 7: SEE scores under different settings. SEEN consistently reduces SEE, whereas most waveform denoisers increase SEE to varying degrees (underlined).

regime and becomes clearly separated from the clean-audio baseline. Overall, these results indicate that SEE reliably tracks noise intensity and captures how increasing acoustic corruption translates into interference.

C Evaluate SEEN by SEE

We apply SEEN as an embedding-space denoising module and report its effect on SEE in Table 7. In contrast to waveform-level denoisers, SEEN yields a substantial decrease in SEE. Because SEEN and SEE share similar computational structures, SEEN primarily suppresses components that are directly captured by SEE, which limits the interpretability of direct evaluation.

D Summary of the Performance of SEEN under Various SNRs

Compared with raw noise corruption, SEEN improves task performance across multiple datasets, although the gains are limited, as shown in Table 8 and Table 9. Regarding this phenomenon, we have the following conjectures. SEEN operates by suppressing activations within the noise-aligned subspace, and thus can only remove noise-related components from the representation. However, when noise corrupts the semantic content of the input itself, the resulting task information becomes incomplete or ambiguous, which cannot be recovered through representation-level suppression alone. Consequently, if a model lacks robustness to incomplete or degraded inputs, applying SEEN in isolation is unlikely to yield substantial per-

Dataset	Method	MiniCPM		Qwen		StepAudio	
		Noise	Clean	Noise	Clean	Noise	Clean
Music	None	84.33%	100.00%	76.00%	100.00%	82.33%	100.00%
	STFT	84.33%	93.33%	71.67%	90.33%	79.00%	92.67%
	WT	82.00%	94.67%	67.67%	91.00%	77.67%	94.00%
	Segan	82.67%	89.67%	71.00%	86.33%	78.00%	87.33%
	DFL	82.00%	88.33%	67.33%	76.67%	75.67%	85.33%
	SEEN	**85.00%**	**99.00%**	**76.99%**	**97.67%**	**82.67%**	**98.00%**
Sound	None	74.33%	100.00%	70.67%	100.00%	77.67%	100.00%
	STFT	72.00%	86.67%	64.33%	88.33%	76.00%	87.67%
	WT	70.33%	92.33%	63.00%	91.67%	71.33%	92.33%
	Segan	73.00%	87.00%	58.67%	83.33%	73.67%	87.00%
	DFL	73.67%	81.33%	60.00%	79.33%	72.33%	83.00%
	SEEN	**75.27%**	**98.67%**	**72.00%**	**99.67%**	**78.33%**	**99.00%**
Speech	None	79.67%	100.00%	77.67%	100.00%	73.33%	100.00%
	STFT	75.33%	94.00%	68.00%	97.00%	66.33%	93.33%
	WT	70.67%	93.00%	68.00%	97.00%	55.67%	94.67%
	Segan	69.00%	93.00%	64.67%	91.33%	60.67%	90.00%
	DFL	72.67%	93.00%	65.33%	92.00%	62.00%	92.33%
	SEEN	**80.00%**	**98.00%**	**78.33%**	**99.00%**	**74.00%**	**96.33%**

Table 8: Generation success rate under noisy (SNR=-10) and clean conditions with different denoising strategies.

Method	Gauss	Crowd	Traffic	Machine
None	74.78%	69.67%	67.89%	73.45%
STFT	68.00%	66.00%	68.44%	71.22%
WT	63.22%	67.44%	64.67%	60.78%
Segan	64.78%	64.66%	64.67%	64.33%
DFL	64.22%	64.11%	63.00%	64.22%
SEEN	**75.22%**	**70.00%**	**68.00%**	**74.67%**

Table 9: SEEN improves generation quality across noise types.

formance improvements. Despite this constraint, SEEN consistently yields modest improvements, indicating that reducing noise-induced bias provides measurable benefits even when information loss dominates.

Method	STFT	WT	Segan	DFL
Time (s)	23.11	2.21	4.67	1156.15
Complexity	$O(N \log N)$	$O(N)$	$O(N \cdot L \cdot C)$	$O(N \cdot L \cdot K^2)$

Table 10: Efficiency Analysis: Execution Time and Time Complexity for 100 Audio Samples

E The transferability of SEE metric

To evaluate the transferability and generality of the proposed SEE metric, we conducted extensive cross-model evaluations across three distinct ALLMs: MiniCPM, Qwen, and StepAudio. The experimental protocol involved extracting directional vectors using a limited calibration set of 50 samples per noise category. These vectors were subsequently applied to a larger, unseen test suite comprising 300 samples per noise type.

The experimental results, summarized in table 11, consistently demonstrate that across all three models and various acoustic environments (Gauss, Crowd, Traffic, and Machine), the SEE metric exhibits a stable monotonic relationship with the Signal-to-Noise Ratio (SNR). Specifically, as the SNR decreases from 10dB to -10dB, the SEE values increase progressively. This consistent trend across diverse model architectures and noise conditions underscores the strong transferability and reliability of the SEE metric in characterizing audio degradation.

F Comparison of Representation Similarity across Enhancement Methods

Tables 12–18 reports the cosine similarity between clean speech and its noisy or enhanced counterparts under different noise levels and enhancement methods. Specifically, `Clean_vs_Noisy` measures the similarity between clean and noisy speech, while `Clean_vs_Enhanced` measures the similarity between clean speech and speech processed by different enhancement methods.

Dataset	SNR	MiniCPM				Qwen				StepAudio			
		Gauss	Crowd	Traffic	Machine	Gauss	Crowd	Traffic	Machine	Gauss	Crowd	Traffic	Machine
Music	**Clean**	3.136	3.136	3.136	3.136	0.891	0.891	0.891	0.891	5.430	5.430	5.430	5.430
	10	3.016	3.290	3.140	3.323	1.578	1.498	1.414	1.868	7.623	6.780	6.884	7.132
	0	3.228	3.579	3.369	3.440	1.895	2.217	2.283	2.643	8.163	8.458	8.310	8.370
	-10	3.567	3.569	3.562	3.455	2.453	3.159	3.616	3.27	8.612	9.654	9.291	9.770
Sound	**Clean**	3.145	3.145	3.145	3.145	1.533	1.533	1.533	1.533	7.884	7.884	7.884	7.884
	10	3.060	3.161	3.056	3.235	1.956	1.919	1.853	2.344	8.416	8.337	8.426	8.363
	0	3.223	3.377	3.279	3.308	2.209	2.358	2.365	2.972	8.449	8.603	8.516	9.025
	-10	3.549	3.545	3.634	3.350	2.420	3.044	3.346	3.342	8.168	8.967	8.351	9.906
Speech	**Clean**	2.461	2.461	2.461	2.461	1.839	1.839	1.839	1.839	7.322	7.322	7.322	7.322
	10	2.764	2.747	2.641	2.871	2.007	2.026	1.945	2.028	7.749	8.014	7.859	7.834
	0	3.085	3.067	2.985	3.089	1.798	2.034	1.904	2.093	8.453	9.004	8.309	8.194
	-10	3.703	3.706	3.862	3.391	1.768	2.353	2.681	2.354	8.583	9.703	8.890	8.559

Table 11: Cross-Noise Transferability of the SEE Metric across Different Noise Conditions

Dataset	Method	MiniCPM		Qwen		StepAudio	
		Clean_vs_Noisy	Clean_vs_Enhanced	Clean_vs_Noisy	Clean_vs_Enhanced	Clean_vs_Noisy	Clean_vs_Enhanced
Music	**STFT**	0.7138	0.6121	0.7549	0.7048	0.7121	0.6897
	WT	0.7138	0.6108	0.7549	0.6335	0.7121	0.6543
	Segan	0.7138	0.6273	0.7549	0.6304	0.7121	0.6481
	DFL	0.7138	0.6775	0.7549	0.6944	0.7121	0.6736
	SEEN	**0.7138**	**0.7341**	**0.7549**	**0.7628**	**0.7121**	**0.7142**
Sound	**STFT**	0.7446	0.6339	0.7754	0.6938	0.8627	0.7924
	WT	0.7446	0.6423	0.7754	0.6410	0.8627	0.7263
	Segan	0.7446	0.6880	0.7754	0.7098	0.8627	0.8015
	DFL	0.7446	0.7284	0.7754	0.6880	0.8627	0.7804
	SEEN	**0.7446**	**0.7858**	**0.7754**	**0.7827**	**0.8627**	**0.8631**
Speech	**STFT**	0.5837	0.4708	0.7271	0.6880	0.8399	0.8119
	WT	0.5837	0.3683	0.7271	0.5345	0.8399	0.6943
	Segan	0.5837	0.3997	0.7271	0.6303	0.8399	0.7831
	DFL	0.5837	0.5439	0.7271	0.6588	0.8399	0.7986
	SEEN	**0.5837**	**0.6579**	**0.7271**	**0.7345**	**0.8399**	**0.8476**

Table 12: Cosine Similarity Between Clean Speech and Noisy or Enhanced Speech Under Different Enhancement Methods and Noise Conditions(SNR=-5)

Dataset	Method	MiniCPM		Qwen		StepAudio	
		Clean_vs_Noisy	Clean_vs_Enhanced	Clean_vs_Noisy	Clean_vs_Enhanced	Clean_vs_Noisy	Clean_vs_Enhanced
Music	**STFT**	0.6523	0.5965	0.6701	0.6551	0.7031	0.6912
	WT	0.6523	0.5752	0.6701	0.6390	0.7031	0.6566
	Segan	0.6523	0.6078	0.6701	0.6012	0.7031	0.6459
	DFL	0.6523	0.6525	0.6701	0.6648	0.7031	0.6656
	SEEN	**0.6523**	**0.6509**	**0.6701**	**0.6823**	**0.7031**	**0.7002**
Sound	**STFT**	0.7129	0.6188	0.7034	0.6460	0.8291	0.7641
	WT	0.7129	0.6031	0.7034	0.6297	0.8291	0.6921
	Segan	0.7129	0.6693	0.7034	0.6557	0.8291	0.7713
	DFL	0.7129	0.6951	0.7034	0.6512	0.8291	0.7510
	SEEN	**0.7129**	**0.7124**	**0.7034**	**0.7125**	**0.8291**	**0.8114**
Speech	**STFT**	0.5314	0.4141	0.6452	0.6058	0.8060	0.7692
	WT	0.5314	0.3365	0.6452	0.4967	0.8069	0.6932
	Segan	0.5314	0.3760	0.6452	0.5359	0.8060	0.7410
	DFL	0.5314	0.4517	0.6452	0.5867	0.8060	0.7560
	SEEN	**0.5314**	**0.5292**	**0.6452**	**0.6474**	**0.8069**	**0.7868**

Table 13: Cosine Similarity Between Clean Speech and Noisy or Enhanced Speech Under Different Enhancement Methods and Noise Conditions(SNR=-10)

Dataset	Method	MiniCPM		Qwen		StepAudio	
		Clean_vs_Noisy	Clean_vs_Enhanced	Clean_vs_Noisy	Clean_vs_Enhanced	Clean_vs_Noisy	Clean_vs_Enhanced
Music	STFT	0.7967	0.6714	0.8280	0.7631	0.7435	0.6994
	WT	0.7967	0.6394	0.8280	0.6539	0.7435	0.6553
	Segan	0.7967	0.6861	0.8280	0.6895	0.7435	0.6629
	DFL	0.7967	0.7084	0.8280	0.7275	0.7435	0.6847
	SEEN	**0.7967**	**0.8099**	**0.8280**	**0.8312**	**0.7435**	**0.7371**
Sound	STFT	0.8129	0.6849	0.8418	0.7420	0.8928	0.8170
	WT	0.8129	0.6762	0.8418	0.6675	0.8928	0.7460
	Segan	0.8129	0.7488	0.8418	0.7669	0.8928	0.8267
	DFL	0.8129	0.7609	0.8418	0.7235	0.8928	0.8068
	SEEN	**0.8129**	**0.8486**	**0.8180**	**0.8462**	**0.8928**	**0.8718**
Speech	STFT	0.7042	0.6155	0.8097	0.7703	0.8495	0.8446
	WT	0.7042	0.4365	0.8097	0.6083	0.8495	0.7077
	Segan	0.7042	0.5320	0.8097	0.7206	0.8495	0.8186
	DFL	0.7042	0.6295	0.8097	0.7253	0.8495	0.8348
	SEEN	**0.7042**	**0.7257**	**0.8097**	**0.8082**	**0.8495**	**0.8486**

Table 14: Cosine Similarity Between Clean Speech and Noisy or Enhanced Speech Under Different Enhancement Methods and Noise Conditions(SNR=0)

Dataset	Method	MiniCPM		Qwen		StepAudio	
		Clean_vs_Noisy	Clean_vs_Enhanced	Clean_vs_Noisy	Clean_vs_Enhanced	Clean_vs_Noisy	Clean_vs_Enhanced
Music	STFT	0.8566	0.7610	0.8750	0.8083	0.7702	0.7220
	WT	0.8566	0.6815	0.8750	0.6955	0.7702	0.6724
	Segan	0.8566	0.7659	0.8750	0.7599	0.7702	0.6887
	DFL	0.8566	0.7326	0.8750	0.7562	0.7702	0.6880
	SEEN	**0.8566**	**0.8636**	**0.8750**	**0.8749**	**0.7702**	**0.7822**
Sound	STFT	0.8681	0.7491	0.8909	0.7817	0.9184	0.8390
	WT	0.8681	0.7268	0.8909	0.7104	0.9184	0.7899
	Segan	0.8681	0.8075	0.8909	0.8134	0.9184	0.8455
	DFL	0.8681	0.7864	0.8909	0.7556	0.9184	0.8290
	SEEN	**0.8681**	**0.8948**	**0.8909**	**0.8923**	**0.9184**	**0.9067**
Speech	STFT	0.7990	0.7449	0.8683	0.8297	0.9002	0.8774
	WT	0.7990	0.5480	0.8683	0.6902	0.9002	0.7549
	Segan	0.7990	0.6618	0.8683	0.8002	0.9002	0.8392
	DFL	0.7990	0.7105	0.8683	0.7909	0.9002	0.8617
	SEEN	**0.7990**	**0.8213**	**0.8683**	**0.8757**	**0.9002**	**0.8944**

Table 15: Cosine Similarity Between Clean Speech and Noisy or Enhanced Speech Under Different Enhancement Methods and Noise Conditions(SNR=5)

Dataset	Method	MiniCPM		Qwen		StepAudio	
		Clean_vs_Noisy	Clean_vs_Enhanced	Clean_vs_Noisy	Clean_vs_Enhanced	Clean_vs_Noisy	Clean_vs_Enhanced
Music	STFT	0.8979	0.8319	0.9066	0.8404	0.7984	0.7468
	WT	0.8979	0.7395	0.9066	0.7493	0.7984	0.6942
	Segan	0.8979	0.8209	0.9066	0.8200	0.7984	0.7153
	DFL	0.8979	0.7522	0.9066	0.7782	0.7984	0.6917
	SEEN	**0.8979**	**0.9035**	**0.9066**	**0.9048**	**0.7984**	**0.7891**
Sound	STFT	0.9076	0.7966	0.9243	0.8095	0.9393	0.8495
	WT	0.9076	0.7864	0.9243	0.7646	0.9393	0.8384
	Segan	0.9076	0.8489	0.9243	0.8442	0.9393	0.8534
	DFL	0.9076	0.8032	0.9243	0.7779	0.9393	0.8445
	SEEN	**0.9076**	**0.9274**	**0.9243**	**0.9234**	**0.9303**	**0.9172**
Speech	STFT	0.8635	0.8304	0.9018	0.8615	0.9221	0.9010
	WT	0.8635	0.6704	0.9018	0.7695	0.9221	0.8111
	Segan	0.8635	0.7550	0.9018	0.8514	0.9221	0.8613
	DFL	0.8635	0.7745	0.9018	0.8360	0.9221	0.8801
	SEEN	**0.8635**	**0.8893**	**0.9018**	**0.8983**	**0.9221**	**0.9050**

Table 16: Cosine Similarity Between Clean Speech and Noisy or Enhanced Speech Under Different Enhancement Methods and Noise Conditions(SNR=10)

Dataset	Method	MiniCPM		Qwen		StepAudio	
		Clean_vs_Noisy	Clean_vs_Enhanced	Clean_vs_Noisy	Clean_vs_Enhanced	Clean_vs_Noisy	Clean_vs_Enhanced
Music	**STFT**	0.9530	0.8985	0.9444	0.8886	0.8480	0.7906
	WT	0.9530	0.8340	0.9444	0.8427	0.8480	0.7551
	Segan	0.9530	0.8854	0.9444	0.8749	0.8480	0.7570
	DFL	0.9530	0.7716	0.9444	0.7988	0.8480	0.7039
	SEEN	**0.9530**	**0.9492**	**0.9444**	**0.9413**	**0.8480**	**0.8361**
Sound	**STFT**	0.9687	0.8667	0.9621	0.8483	0.9661	0.8804
	WT	0.9687	0.8819	0.9621	0.8596	0.9661	0.9046
	Segan	0.9687	0.8978	0.9621	0.8719	0.9661	0.8618
	DFL	0.9687	0.8215	0.9621	0.8004	0.9661	0.8588
	SEEN	**0.9687**	**0.9650**	**0.9621**	**0.9590**	**0.9661**	**0.9436**
Speech	**STFT**	0.9404	0.9186	0.9345	0.9016	0.9590	0.9292
	WT	0.9404	0.8445	0.9345	0.8928	0.9590	0.8782
	Segan	0.9404	0.8698	0.9345	0.8863	0.9590	0.8784
	DFL	0.9404	0.8391	0.9345	0.8699	0.9590	0.9025
	SEEN	**0.9404**	**0.9379**	**0.9343**	**0.9307**	**0.9590**	**0.9497**

Table 17: Cosine Similarity Between Clean Speech and Noisy or Enhanced Speech Under Different Enhancement Methods and Noise Conditions(SNR=20)

Dataset	Method	MiniCPM		Qwen		StepAudio	
		Clean_vs_Noisy	Clean_vs_Enhanced	Clean_vs_Noisy	Clean_vs_Enhanced	Clean_vs_Noisy	Clean_vs_Enhanced
Music	**STFT**	0.9783	0.9137	0.9671	0.9092	0.8967	0.8196
	WT	0.9783	0.8818	0.9671	0.8892	0.8967	0.7968
	Segan	0.9783	0.9070	0.9671	0.8938	0.8967	0.7832
	DFL	0.9783	0.7840	0.9671	0.8121	0.8967	0.7109
	SEEN	**0.9783**	**0.9743**	**0.9671**	**0.9637**	**0.8967**	**0.8822**
Sound	**STFT**	0.9873	0.8835	0.9811	0.8672	0.9809	0.8960
	WT	0.9873	0.9204	0.9811	0.9027	0.9809	0.8912
	Segan	0.9873	0.9092	0.9811	0.8827	0.9809	0.8637
	DFL	0.9873	0.8320	0.9811	0.8123	0.9809	0.8670
	SEEN	**0.9873**	**0.9833**	**0.9811**	**0.9772**	**0.9809**	**0.9582**
Speech	**STFT**	0.9605	0.9371	0.9527	**0.9253**	0.9816	0.9418
	WT	0.9605	0.9186	0.9527	0.9403	0.9816	0.9038
	Segan	0.9605	0.8880	0.9527	0.8946	0.9816	0.8833
	DFL	0.9605	0.8726	0.9527	0.8878	0.9816	0.9173
	SEEN	**0.9605**	**0.9580**	**0.9527**	**0.9588**	**0.9816**	**0.9612**

Table 18: Cosine Similarity Between Clean Speech and Noisy or Enhanced Speech Under Different Enhancement Methods and Noise Conditions(SNR=30)

As shown in the results, conventional speech enhancement methods often fail to reduce the representation gap between clean and noisy speech. In many cases, they even enlarge this gap, resulting in lower cosine similarity compared to the original noisy input. In contrast, our proposed method consistently narrows the similarity gap between clean and noisy speech, indicating its superior ability to preserve semantic and acoustic consistency under noise perturbations. This auxiliary experiment provides additional evidence supporting the superiority of our proposed method over traditional approaches.

G Efficiency Analysis About Baseline Methods

To evaluate computational efficiency, we conducted a benchmark on 100 audio samples from the Speech dataset with SNR=0 dB. As shown in Table 10, external denoising methods inevitably introduce additional processing time: the WT and STFT methods require 2.21s and 23.11s respectively, while the DFL model reaches 1156.15s. Owing to the limited CUDA version support of the DFL implementation, we performed the evaluation on the CPU. While applying traditional or external deep learning denoising to ALLMs typically incurs such "bottleneck" delays, our method introduces zero additional processing time, enabling more seamless integration for real-time applications.

The time complexity of the evaluated methods in inference phase is also summarized in Table 10. Here, N represents the total number of audio samples in a signal. For the deep learning models (Segan and DFL), L denotes the number of hidden layers, C represents the number of feature channels, and K indicates the size of the convolutional kernels.

H Validity Statement of SEEN

In our experiments, SEE consistently increases under stronger corruption and remains well separated from the clean-input range, indicating that it captures a stable noise-aligned component in the embedding space. However, applying SEEN to suppress this component yields only a modest but consistent improvement in generation quality, rather than a large robustness recovery. This gap between measurability and recoverability suggests that the interference identified by SEE is one of core component of the mechanism driving downstream failures.

A plausible explanation is that the subspace isolated by SEE represents directions along which noise reliably perturbs the model's internal coding tendency, so removing these directions can reduce embedding bias and mitigate spurious activation. Yet generation errors are not solely caused by biased representations; when noise is strong, it can also erase or distort task-relevant acoustic cues at the source, producing genuine information loss that cannot be recovered by subtracting a projection term. In such cases, SEEN can remove interference but cannot reconstruct missing content, so the attainable improvement is bounded by the amount of recoverable information rather than by the clean–noisy performance gap. This effect may be particularly pronounced for audio inputs. Frame-level tokenization typically produces long token sequences with relatively low per-token information density, meaning that corruption can distribute small distortions across many tokens and accumulate into semantic degradation that is difficult to reverse through a single subspace suppression.

These observations point to a natural direction for further improvement. While SEE provides a useful criterion for diagnosing and comparing noise-induced interference, stronger robustness may require training-time mechanisms that explicitly enhance the model's ability to infer semantics under partial or degraded acoustic evidence. For example, one could consider incorporating SEE-guided objectives into robustness-oriented training, or adopting augmentation curricula that emphasize semantic preservation under frame-level corruption. We view these as promising hypotheses rather than confirmed solutions, and leave a systematic exploration of training-side enhancements as an important next step.

I Algorithm Flow

In this appendix, we present the formal procedures for our proposed framework SEEN. The process is divided into three main components: algorithm 1: layer localization and noise extraction (Algorithm 1), algorithm 2: energy-based evaluation (Algorithm 2), and algorithm 3: the final robust inference (Algorithm 3). Together, these algorithms form a complete pipeline for diagnosing and neutralizing noise-aligned activations in Large Audio Language Models.

J AI Writing Assistance Disclosure

We used AI tools solely for language polishing to improve clarity and readability. The AI tools did not contribute to the scientific content, ideas, analyses, or conclusions of this work.

Algorithm 1 SEE Setup: Layer Localization and Noise Extraction

Require: Semantic set $X^s = \{x_i^s\}_{i=1}^m$, noise set $X^n = \{x_i^n\}_{i=1}^m$, thresholds α, δ, stabilizer ε
Ensure: Retained layers $\mathcal{L}^\star$, noise bases $\{\mathbf{Q}_\ell\}_{\ell \in \mathcal{L}^\star}$

for $\ell = 1$ **to** L **do**
  for $i = 1$ **to** m **do**
    Extract $\mathbf{A}_\ell(x_i^s) \in \mathbb{R}^{T(x_i^s) \times d_\ell}$ and $\mathbf{A}_\ell(x_i^n) \in \mathbb{R}^{T(x_i^n) \times d_\ell}$
    $\mathbf{a}_\ell(x_i^s) \leftarrow \frac{1}{T(x_i^s)} \sum_{t=1}^{T(x_i^s)} \mathbf{A}_\ell(x_i^s)_{t,:} \in \mathbb{R}^{d_\ell}$
    $\mathbf{a}_\ell(x_i^n) \leftarrow \frac{1}{T(x_i^n)} \sum_{t=1}^{T(x_i^n)} \mathbf{A}_\ell(x_i^n)_{t,:} \in \mathbb{R}^{d_\ell}$
  end for
  Stack $\{\mathbf{a}_\ell(x_i^s)\}_{i=1}^m$ to form $\mathbf{S}_\ell \in \mathbb{R}^{m \times d_\ell}$; stack $\{\mathbf{a}_\ell(x_i^n)\}_{i=1}^m$ to form $\mathbf{N}_\ell \in \mathbb{R}^{m \times d_\ell}$
  Magnitude: $\mathrm{M}_\ell \leftarrow \|\mathbf{S}_\ell - \mathbf{N}_\ell\|_F$
  Direction: $\mathrm{D}_\ell \leftarrow \frac{\left|\mathrm{vec}(\mathbf{S}_\ell)^\top \mathrm{vec}(\mathbf{N}_\ell)\right|}{\|\mathrm{vec}(\mathbf{S}_\ell)\|_2 \, \|\mathrm{vec}(\mathbf{N}_\ell)\|_2 + \varepsilon}$
end for
$\overline{\mathrm{M}} \leftarrow \frac{1}{L} \sum_{\ell=1}^L \mathrm{M}_\ell, \quad \overline{\mathrm{D}} \leftarrow \frac{1}{L} \sum_{\ell=1}^L \mathrm{D}_\ell$
$\ell^\star \leftarrow \min\{\ell \mid \mathrm{M}_\ell > \overline{\mathrm{M}} \ \wedge \ \mathrm{D}_\ell > \overline{\mathrm{D}}\}$
$\mathcal{L}^\star \leftarrow \{\ell^\star, \ell^\star + 1, \ldots, L\}$
for each $\ell \in \mathcal{L}^\star$ **do**
  SVD: $\mathbf{S}_\ell = \mathbf{U}_\ell^s \mathbf{\Sigma}_\ell^s (\mathbf{V}_\ell^s)^\top; \quad \mathbf{N}_\ell = \mathbf{U}_\ell^n \mathbf{\Sigma}_\ell^n (\mathbf{V}_\ell^n)^\top$
  $\mathcal{I}_\ell^s \leftarrow \{j \mid \sigma_{\ell,j}^s > \alpha\}; \quad \mathcal{I}_\ell^n \leftarrow \{j \mid \sigma_{\ell,j}^n > \alpha\}$
  for each $j \in \mathcal{I}_\ell^n$ **do**
    $\mathbf{m}_{\ell,j} \leftarrow \max_{k \in \mathcal{I}_\ell^s} \left|\cos(\mathbf{v}_{\ell,j}^n, \mathbf{v}_{\ell,k}^s)\right|$
  end for
  $\mathcal{J}_\ell \leftarrow \{j \in \mathcal{I}_\ell^n \mid \mathbf{m}_{\ell,j} < \delta\}; \quad r_\ell \leftarrow |\mathcal{J}_\ell|$
  Build mask $s_\ell \in \{0,1\}^{d_\ell}$ with $s_\ell[j] = 1$ iff $j \in \mathcal{J}_\ell$; $\quad \mathbf{M}_\ell \leftarrow \mathrm{diag}(s_\ell)$
  $\mathbf{Q}_\ell \leftarrow \mathrm{NonZeroCols}(\mathbf{V}_\ell^n \mathbf{M}_\ell) \in \mathbb{R}^{d_\ell \times r_\ell}$ {equivalently keep columns $\mathcal{J}_\ell$ of $\mathbf{V}_\ell^n$}
end for

Algorithm 2 SEE Eval: Computing Signal Embedding Energy for an Input

Require: Test input x, retained layers $\mathcal{L}^\star$, noise bases $\{\mathbf{Q}_\ell\}_{\ell \in \mathcal{L}^\star}$, stabilizer ε
Ensure: $\mathrm{SEE}(x)$

Forward to obtain $\mathbf{A}_\ell(x) \in \mathbb{R}^{T(x) \times d_\ell}$ for all $\ell \in \mathcal{L}^\star$
for each $\ell \in \mathcal{L}^\star$ **do**
  Project: $\mathbf{Z}_\ell(x) \leftarrow \mathbf{A}_\ell(x) \mathbf{Q}_\ell \in \mathbb{R}^{T(x) \times r_\ell}$
  $\mathrm{SEE}_\ell(x) \leftarrow \frac{1}{T(x)} \sum_{t=1}^{T(x)} \frac{\|\mathbf{Z}_\ell(x)_{t,:}\|_2^2}{\|\mathbf{A}_\ell(x)_{t,:}\|_2^2 + \varepsilon}$
end for
$\mathrm{SEE}(x) \leftarrow \frac{1}{|\mathcal{L}^\star|} \sum_{\ell \in \mathcal{L}^\star} \mathrm{SEE}_\ell(x)$

Algorithm 3 SEEN Inference: Neutralizing Noise-aligned Activations

Require: Test input x, retained layers $\mathcal{L}^\star$, noise bases $\{\mathbf{Q}_\ell\}_{\ell \in \mathcal{L}^\star}$, neutralization strengths $\{\lambda\}_{\ell \in \mathcal{L}^\star}$
Ensure: Neutralized activations $\{\widetilde{\mathbf{A}}_\ell(x)\}$

Forward to obtain $\mathbf{A}_\ell(x) \in \mathbb{R}^{T(x) \times d_\ell}$ for all $\ell \in \mathcal{L}^\star$
for each $\ell \in \mathcal{L}^\star$ **do**
  Reconstruct noise component: $\mathbf{C}_\ell(x) \leftarrow \mathbf{A}_\ell(x) \mathbf{Q}_\ell \mathbf{Q}_\ell^\top \in \mathbb{R}^{T(x) \times d_\ell}$
  Neutralize: $\widetilde{\mathbf{A}}_\ell(x) \leftarrow \mathbf{A}_\ell(x) - \lambda \, \mathbf{C}_\ell(x)$
end for
Continue forward pass using $\widetilde{\mathbf{A}}_\ell(x)$ to obtain the final generation output