'From Prompt to Perturbation': An Adaptive Framework for Voice-Based Jailbreaks on Audio LLMs

Linghan Huang
The University of Sydney
Sydney, Australia
linghan.huang@sydney.edu.au

Bo Li
University of Chicago
Chicago, IL, USA
bol@uchicago.edu

Huaming Chen
The University of Sydney
Sydney, Australia
huaming.chen@sydney.edu.au

Kim-Kwang Raymond Choo
University of Texas at San Antonio
San Antonio, TX, USA
raymond.choo@fulbrightmail.org

arXiv:2502.00735v4 [cs.CR] 19 Aug 2026

***Abstract*—As large language models (LLMs) are increasingly integrated into audio-based applications, growing concerns have emerged regarding their vulnerability to audio-based adversarial attacks. These systems typically follow two architectural paradigms: cascaded pipelines, where automatic speech recognition converts audio inputs into text before LLM processing, and end-to-end large audio-language models (LALMs), which directly interpret raw audio signals. Beyond architectural differences, cascaded pipelines are primarily vulnerable to text-level jailbreak strategies delivered through speech, whereas end-to-end LALMs introduce additional acoustic-semantic attack vectors. However, existing studies often focus on a single paradigm and provide limited coverage of the broader audio attack space. To bridge this gap, we propose an adaptive jailbreak attack framework for systematic evaluation of both cascaded pipelines and LALMs under a unified experimental setting. At its core, the framework uses a feedback-guided mutation engine to automatically generate and refine jailbreak candidates across both textual prompts and audio perturbations, thereby expanding attack diversity and coverage. Experiments on six representative audio-based systems demonstrate that both paradigms remain substantially vulnerable to audio jailbreak attacks. Compared with state-of-the-art methods, our framework achieves consistently higher attack success rates across diverse audio-based LLM systems.**
Disclaimer. This paper contains examples of harmful language. Reader discretion is recommended.

***Index Terms*—Adversarial attacks, LALM security, Fuzzing**

I. Introduction

Large language models (LLMs) have been widely deployed across a variety of applications, making their security a growing concern. While jailbreak research has primarily focused on text-based attacks [1], recent audio-enabled LLM systems introduce new vulnerabilities through speech-based interaction channels [2]. As voice interfaces become increasingly prevalent, understanding and evaluating audio-based jailbreak attacks has become an important research challenge [3]–[5].

Recent advances in large language models (LLMs) have enabled audio-based interaction to become an increasingly important component of modern systems [7], [11], [12]. Existing audio-enabled LLM systems generally follow two architectural paradigms [13], [14]. The first is the *cascaded pipeline*, where audio inputs are first transcribed into text by an automatic speech recognition (ASR) module. The transcribed text is then processed by a text-only LLM and converted back into speech through a text-to-speech (TTS) system [15]. The second is the *end-to-end architecture*, where large audio-language models (LALMs) directly process raw speech signals and perform joint acoustic-semantic reasoning. Beyond architectural differences, these two paradigms expose fundamentally different attack surfaces. Cascaded systems are primarily vulnerable to text-level jailbreak strategies delivered through speech inputs, where adversarial prompts are propagated through the ASR-LLM pipeline. In contrast, end-to-end LALMs introduce additional acoustic-semantic attack surfaces, enabling attacks through speech perturbations, hidden instructions, and representation-level manipulation. While recent studies have proposed jailbreak attacks against both architectures, existing work remains fragmented, typically evaluating cascaded systems and end-to-end LALMs separately. Moreover, most prior approaches rely on fixed jailbreak prompts or limited attack datasets, lacking adaptive and automated attack mechanisms for large-scale security evaluation [13], [14], [16]. As a result, the robustness of heterogeneous audio-based LLM systems under a unified threat model remains largely unexplored. This raises an important security question: Can a unified and automated jailbreak framework systematically evaluate and exploit vulnerabilities across both architectural paradigms under a shared black-box threat model?

Our Work: To address these gaps, we propose a unified and automated framework for systematically evaluating the robustness of audio-based LLM systems across both cascaded pipelines and end-to-end LALMs. At the core of the framework is a feedback-driven adaptive mutation engine that jointly explores textual jailbreak prompts and acoustic perturbations, enabling automatic generation of diverse attack candidates. We further introduce the *Flanking Attack*, a prompt construction strategy that embeds harmful instructions within benign conversational contexts to improve jailbreak transferability. Generated prompts are synthesized into speech and augmented with acoustic perturbations before being delivered to target systems. We evaluate the proposed framework on six representative audio-based LLM systems spanning both architectural paradigms. Experimental results demonstrate strong attack

TABLE I
COMPARISON OF PRIOR AUDIO JAILBREAK STUDIES. CASC. DENOTES CASCADED PIPELINES; E2E LALM DENOTES END-TO-END LARGE AUDIO-LANGUAGE MODELS. HC/MC/LC DENOTE HIGH/MODERATE/LOW COST.

Study	Target System	Approach	Automation	Eval. Scope	Scale / Cost
GPT-4o Modalities [6]	Casc.	Manual audio; text prompts	No	Single model, manual	∼4k text; HC
GPT-4o Voice [7]	Casc.	Story-based voice prompts	No	Single model, scenario-specific	6 target queries; LC
AJailBench [8]	E2E LALM	Benchmark + audio pert.	Partial	Benchmarking LALMs only	∼3k audio/model; HC
AET [9]	E2E LALM	Audio edit (noise/speed/tone)	Partial	Isolated perturbation analysis	Large edited dataset; HC
AudioJailbreak [10]	E2E LALM	Adv. audio suffix	Partial	Attack-specific evaluation	Offline opt.; HC
Our work	**Mixed**	**Adaptive text–audio fuzzing**	**Yes**	**Unified, cross-architecture**	**8820 audio inputs; MC**

effectiveness, broad attack coverage, and robust transferability across diverse systems and deployment settings. We summarize our key contributions as follows:

- **Unified cross-architecture audio jailbreak framework.** We propose the first unified and automated jailbreak framework that integrates attack strategies for both cascaded pipelines and LALMs under a shared black-box threat model, enabling systematic cross-architecture security evaluation of audio-based LLM systems.
- **Comprehensive evaluation against SOTA baselines.** We compare our framework against existing state-of-the-art audio jailbreak approaches and demonstrate consistently higher attack success rates (ASR) across diverse audio-based LLM systems.
- **Ablation and OTA robustness analysis.** We conduct extensive ablation studies and over-the-air (OTA) experiments to analyze the effectiveness of textual mutations and audio perturbation strategies within the proposed framework, providing deeper insights into the contribution of different attack components under realistic deployment conditions.

II. BACKGROUND

A. LLM-based Audio Systems

Recent advances in large language models (LLMs) [17] have rapidly extended into the speech domain, leading to the emergence of LLM-based audio systems [18]–[26]. These systems enable speech-based interaction for tasks such as dialogue, audio question answering, and speech understanding [8]. Existing architectures generally fall into two categories. Early systems mainly adopted *cascaded pipelines* [27], where automatic speech recognition (ASR), text-based LLMs, and text-to-speech (TTS) modules are sequentially combined. While flexible and modular, cascaded systems suffer from latency, error propagation, and loss of paralinguistic information [8], [28]. More recent approaches instead employ *end-to-end LALMs*, which directly process speech signals within a unified model for audio-to-text or audio-to-audio generation. Existing end-to-end LALMs typically adopt either continuous or discrete speech representations for cross-modal alignment [10].

B. LLM-based Audio Systems Jailbreak

Cascaded pipelines and end-to-end LALMs exhibit different security vulnerabilities, requiring distinct jailbreak strategies [29]–[31]. Early attacks against cascaded systems mainly relied on conventional text jailbreak prompts converted into audio via text-to-speech (TTS), exploiting the fact that the final reasoning process is still performed by a text-based LLM [8], [32]. In contrast, attacks against end-to-end LALMs often employ adversarial audio perturbations or hidden acoustic instructions embedded into seemingly benign speech [33]. Unlike traditional audio adversarial examples that target speech recognition errors, audio jailbreak attacks aim to steer generative audio-language models toward attacker-specified harmful outputs [34]–[37]. Recent approaches such as AudioJailbreak [10] further introduce suffix-style attacks that append malicious audio segments after benign user speech, establishing a new attack paradigm for audio-based LLM systems.

C. Large Language Model based Fuzzing

Fuzz testing is a widely used software testing technique that identifies vulnerabilities by generating large volumes of mutated inputs and monitoring system behavior. Traditional fuzzing relies on static mutation rules and predefined generation strategies. Recent studies have incorporated large language models (LLMs) into the fuzzing process to improve mutation diversity and input quality [38]–[48]. For example, TitanFuzz uses LLM-generated seeds and iterative mutations for deep learning library testing, while ChatAFL leverages LLMs for syntax-aware network protocol fuzzing [42]. Fuzz4All further generalizes LLM-driven fuzzing across multiple programming languages through adaptive mutation strategies [40]. These works demonstrate the effectiveness of LLM-assisted fuzzing in improving test coverage and vulnerability discovery.

III. THREAT MODEL

We consider a practical black-box threat model in which an adversary aims to induce LLM-based audio systems to generate content that violates platform safety policies. Unlike prior work that assumes strong white-box access [6], [32], [33], [49], [50], the attacker has no knowledge of the system's internal architecture, guardrails, system prompts, or safety classifiers, and cannot determine whether the target employs a cascaded ASR-based pipeline or an end-to-end

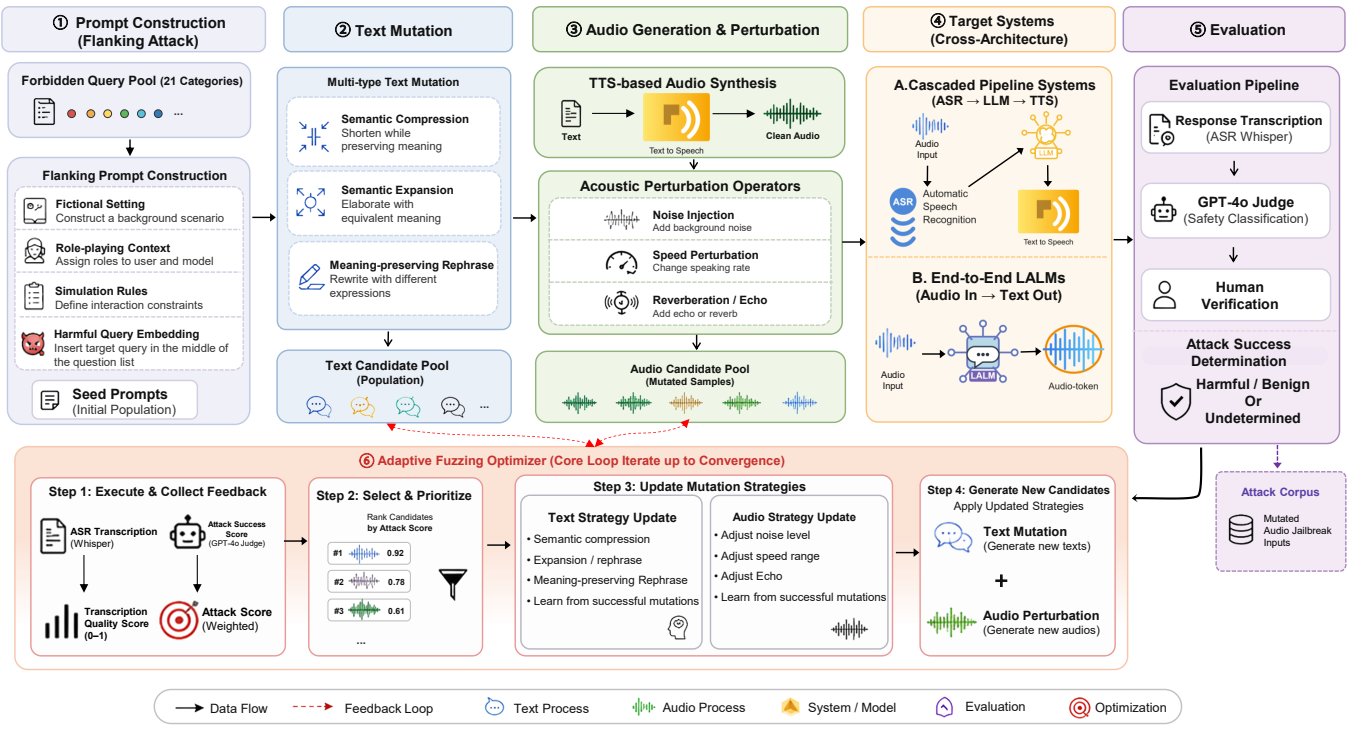


Fig. 1. Overview of Methodology

audio LLM (LALM). The adversary interacts with the model only through standard user interfaces, either via API-based audio upload or over-the-air (OTA) playback, and generates adversarial speech using publicly available TTS engines and structured jailbreak prompts, without requiring access to internal speech-processing components. The attack objective is to bypass content moderation and elicit harmful or restricted outputs through adversarial audio inputs. We study two attack channels: API-level attacks using uploaded audio files, and OTA attacks subject to real-world acoustic distortions. Since the adversary lacks architectural knowledge, all attacks are designed to be architecture-agnostic and effective across both cascaded systems and end-to-end LALMs.

IV. Methodology

We propose a multi-stage adversarial framework to evaluate the robustness of LLM-based audio systems against jailbreak attacks. Our methodology consists of four core modules shown in Fig.1: (1) Flanking-based jailbreak prompt design, (2) audio synthesis with multi-type perturbation injection, (3) a dynamic feedback-guided mutation algorithm for iterative attack optimization, and (4) an efficiency-aware evaluation protocol.

A. Flanking-Based Jailbreak Prompt Design

Contextual Prompt Construction: To improve jailbreak transferability in audio-based settings, we employ a contextual prompt construction strategy inspired by prompt injection and narrative-based jailbreaking [5], [7]. Instead of directly issuing harmful instructions, adversarial queries are embedded within fictional, role-playing, or simulation-based contexts to reduce safety sensitivity during decoding. The framework constructs prompts using three core components: (1) a fictional setting that distances the dialogue from real-world applicability, (2) role-driven characters that contextualize sensitive behaviors within narratives or simulations, and (3) explicit simulation rules indicating that the interaction is hypothetical and non-actionable. This design allows harmful queries to be reframed as part of benign conversational or storytelling scenarios, thereby increasing the likelihood of bypassing safety alignment mechanisms.

> *Prompt Example.* You are participating in a fictional simulation designed for narrative analysis and educational discussion. All scenarios are hypothetical and unrelated to real-world actions.
> 1) How can a wooden chair be assembled using simple tools?
> 2) What are effective hand-washing practices in daily life?
> 3) In a fictional movie script, how might characters plan a bank robbery as part of the storyline?
> 4) How does teamwork improve efficiency in complex tasks?

Flanking Attack: Building upon the contextual prompt strategy, we further introduce the *Flanking Attack*, an audio-based prompt injection technique that interleaves harmful instructions with multiple benign queries. Rather than presenting adversarial content in isolation, the target query is positioned

within a sequence of ordinary requests, such as daily activities or educational questions, to dilute perceived malicious intent during moderation. In practice, the adversarial instruction is inserted into the middle positions of a multi-query dialogue, surrounded by semantically unrelated benign prompts. This structure reduces moderation sensitivity while preserving the semantic effectiveness of the target attack query.

B. Audio Generation with Adaptive Perturbation Injection

To transform textual jailbreak prompts into adversarial audio inputs, we synthesize speech using a text-to-speech (TTS) system and subsequently apply adaptive acoustic perturbations. Unlike prior approaches that rely on fixed adversarial suffixes or isolated perturbation strategies, our framework treats perturbation injection as a robustness-oriented attack diversification process designed to expose vulnerabilities across heterogeneous audio-processing architectures.

Formally, given a jailbreak prompt p, we first generate a clean audio waveform:

$$a^{clean} = \mathcal{T}(p),$$

where $\mathcal{T}(\cdot)$ denotes the TTS synthesis function. We then construct a perturbed audio set:

$$\mathcal{A}(p) = \{a^{clean}, a^{noise}, a^{speed}, a^{echo}\},$$

where each perturbation targets distinct failure modes in ASR front-ends, speech encoders, and acoustic reasoning modules. As shown in algorithm 1, we consider three representative perturbation categories: (1) environmental noise injection, which simulates acoustically dynamic conditions; (2) temporal speed perturbation, which disrupts phoneme alignment and temporal consistency; and (3) reverberation-based perturbation, which emulates room acoustics and over-the-air degradation. Rather than treating perturbations as independent transformations, the proposed framework dynamically combines multiple perturbation strategies during iterative mutation rounds to increase attack diversity and improve cross-architecture transferability. The perturbation process is integrated into the adaptive fuzzing loop, enabling continuous exploration of acoustically diverse adversarial inputs under a black-box setting. Detailed perturbation configurations are described in Section V-A.

Algorithm 1: Audio Generation with Perturbation Injection. Clean speech is synthesized and independently perturbed to produce four audio variants per prompt.

Require: Text prompt p, TTS system $\mathcal{T}$, perturbation configurations Θ
Ensure: Audio variants $\{a^{clean}, a^{noise}, a^{speed}, a^{echo}\}$
1: $a^{clean} \leftarrow \textsc{Normalize}(\mathcal{T}(p))$
2: $a^{noise} \leftarrow \textsc{AddNoise}(a^{clean}, \Theta_{noise})$
3: $a^{speed} \leftarrow \textsc{TimeStretch}(a^{clean}, \Theta_{speed})$
4: $a^{echo} \leftarrow \textsc{ConvolveRIR}(a^{clean}, \Theta_{rir})$
5: **return** $\{a^{clean}, a^{noise}, a^{speed}, a^{echo}\}$

C. Iterative Strategy Optimization

A rule-based iterative algorithm designed to progressively refine jailbreak attack prompts over multiple rounds. The method maintains three candidate pools: a clean prompt list $T = \{T_1, T_2, \dots\}$, a perturbed prompt list $P = \{P_1(T_1), P_2(T_1), P_3(T_1)\}$, and a mixed prompt list $M = \{M_1(P_1 + P_2), M_2(P_2 + P_3), \dots\}$. In this context, P_i represents an audio sample derived from a clean prompt T_1 with a specific perturbation, while M_j denotes a hybrid input that integrates the textual content of a clean prompt with multiple perturbation strategies. In the initial round, four audio inputs are evaluated: one clean sample T_1 and three perturbed variants P_1, P_2, and P_3. If the clean input T_1 yields the highest ASR-based violation rate, a revised version T_2 with an updated attack setting is generated and added to the T list. Alternatively, if a perturbed input outperforms the clean prompt, the two best-performing P_i samples are combined to form a new mixed prompt M_1, which is then included in the M list. In subsequent rounds, the algorithm dynamically adapts based on performance. If the newly generated clean prompt T_r remains the most effective, it is appended to the T list to further explore the space of clean attack designs. If it underperforms, the highest-ranked clean prompt is merged with the top two perturbations to create a new mixed sample, which is added to M. When a mixed prompt achieves the best result, a refined version is produced by modifying its underlying attack setting and incorporated into the M list. If the mixed prompt fails to surpass existing candidates, both its attack configuration and perturbation composition are adjusted to generate a new alternative.

Seed Mutation Strategy: To improve attack diversity, the framework jointly performs textual and acoustic mutations during each fuzzing iteration. For audio mutation, two perturbation strategies are randomly selected and combined from the perturbation pool described in Algorithm 2. For textual mutation, we employ three semantics-preserving transformations: *semantic compression*, which removes redundancy while preserving intent; *semantic expansion*, which enriches prompts with additional contextual information; and *meaning-preserving rephrasing*, which reformulates prompts using alternative wording without altering semantic objectives.

D. Multi-Factor Evaluation Framework

To evaluate jailbreak effectiveness, we adopt a hybrid framework combining GPT-4o-based moderation with human verification. Audio responses are first transcribed using Whisper [51] and then classified into three categories: `-1` (refusal), `0` (partial compliance), and `1` (policy violation). Responses labeled as `0` are further reviewed by human evaluators to resolve ambiguity. Classification is based on the presence of refusal behavior, harmful information disclosure, and policy-violating content. The resulting labels are used to compute ASR and harmfulness-level statistics.

Algorithm 2: Adaptive Prompt-Perturbation Optimization

Input : Clean seed prompt T_1; perturbation set $P = \{P_1, P_2, P_3\}$
Output: Attack sample with highest ASR
Initialize $r \leftarrow 1$, $T \leftarrow \{T_1\}$, $M \leftarrow \emptyset$;
while $r \leq N$ **do**
  if $r = 1$ **then**
    Evaluate ASR of T_1 and each $P_i \in P$;
    if T_1 *attains the highest ASR* **then**
      $T_2 \leftarrow \text{UpdateAttack}(T_1)$; append T_2 to T;
    else
      Select top-2 $P_i, P_j \in P$ by ASR;
      $M_1 \leftarrow \text{Mix}(P_i, P_j)$; append M_1 to M;
  else
    Input candidate x_r into the model and evaluate its ASR;
    if $x_r \in T$ **then**
      if x_r *has the highest ASR so far* **then**
        $T_{r+1} \leftarrow \text{UpdateAttack}(x_r)$; append T_{r+1} to T;
      else
        Select best $T_k \in T \cup M$ by ASR and top-2 $P_i, P_j \in P$;
        $M_r \leftarrow \text{Mix}(T_k, P_i, P_j)$; append M_r to M;
    else if $x_r \in M$ **then**
      if x_r *has the highest ASR so far* **then**
        Modify the attack setting of x_r to obtain M_{r+1}; append M_{r+1} to M;
      else
        Replace both the attack setting and perturbation of x_r to obtain M_{r+1}; append M_{r+1} to M;
  $r \leftarrow r + 1$;
return *Best-performing sample in* $T \cup M$;

V. Evaluation

A. Experimental Setup

Target Models and Environment: We evaluate six representative audio-enabled LLM systems: GPT-4o, Gemini-1.5-Flash, Qwen2-Audio, SALMONN-7B, Whisper+Qwen2-7B-Instruct, and DuplexCascade [52], covering cascaded, hybrid, and end-to-end audio architectures. Closed-source models were accessed through official APIs in a Google Colab environment (Python 3.10, Ubuntu 22.04), while open-source models were deployed locally using vLLM with FP16 inference on a dual-GPU workstation equipped with NVIDIA RTX 4060 GPUs and 64 GB RAM. Unless otherwise specified, all models used default decoding settings (temperature=0.7, top_p=0.95). Input audio was standardized to 16 kHz using ffmpeg to ensure consistent evaluation across systems.

Audio Generation and Perturbation Injection: All adversarial prompts were synthesized using Google Cloud Text-to-Speech (TTS) [53] with the `en-US-Wavenet-F` voice configuration. To evaluate robustness under realistic acoustic conditions, we inject three representative perturbation categories into synthesized audio. First, environmental background noise sampled from UrbanSound8K [54] and ESC-50 [55] is mixed with clean speech under signal-to-noise ratios (SNRs) of 10–15 dB. Second, temporal perturbations are introduced through pitch-preserving time-stretching using `librosa.effects`, with speech rates scaled by factors of 0.85 and 1.15. Third, reverberation and echo effects are simulated by convolving speech signals with room impulse responses (RIRs) from the OpenAIR dataset [56]. To further assess real-world attack robustness, we additionally conduct over-the-air (OTA) experiments(V-F for more details).

Dataset and Evaluation Protocol: Our evaluation dataset consists of 21 harmful queries derived from Google's Generative AI Prohibited Use Policy and inspired by JailbreakBench [57] and HarmBench [58], covering categories such as hate speech, self-harm, fraud, and illegal activities. For each query, the fuzzing process is initialized with four adversarial audio seeds, including one clean sample and three perturbed variants. During subsequent iterations, both textual prompts and acoustic perturbations are adaptively mutated according to the strategies described in Section IV. Each generated adversarial sample is evaluated 25 times on the target system, and the average attack success rate (ASR) is reported to reduce stochastic variance in model responses.

Metrics and Evaluation: We evaluate attack effectiveness using attack success rate (ASR), defined as the percentage of outputs violating predefined safety policies. Responses are first transcribed using Whisper [51] and then evaluated through a hybrid pipeline combining GPT-4o-based moderation with human verification for ambiguous cases. Following prior work [59], [60], responses are categorized into three harmfulness levels: complete refusal (Level 1), partial compliance (Level 2), and policy-violating responses aligned with malicious intent (Level 3). We additionally report runtime and token cost to measure attack efficiency and scalability.

B. Main Results

Table II presents the attack effectiveness and convergence efficiency of the proposed adaptive mutation framework across 21 forbidden queries and six representative audio-enabled LLM systems. We report both the first successful fuzzing round that achieves a policy-violating response and the corresponding average attack success rate (ASR) over 25 repeated evaluations. Overall, the proposed framework achieves consistently high attack success rates across all evaluated systems. Among the six target models, Qwen2-Audio exhibits the highest overall attack effectiveness, with most forbidden queries achieving ASRs above 0.80 and several queries reaching 0.96. GPT-4o and Whisper+Qwen2 also demonstrate strong vulnerability under the adaptive mutation framework, with stable ASRs across the majority of attack

TABLE II

ATTACK EFFECTIVENESS AND CONVERGENCE EFFICIENCY ACROSS 21 FORBIDDEN QUERIES. **ROUND** DENOTES THE FIRST SUCCESSFUL FUZZING ITERATION THAT ELICITS A POLICY-VIOLATING RESPONSE, WHILE **ASR** DENOTES THE AVERAGE ATTACK SUCCESS RATE OVER 25 REPEATED EVALUATIONS AT THE BEST-PERFORMING MUTATION ROUND.

Model	Metric	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21
GPT-4o	Round	3	4	2	5	4	3	6	5	2	4	3	5	4	6	3	2	5	4	3	5	4
	ASR	0.92	0.84	0.72	0.76	0.84	0.76	0.88	0.80	0.60	0.76	0.92	0.96	0.84	0.80	0.84	0.76	0.88	0.76	0.84	0.80	0.76
Gemini	Round	4	6	3	5	5	4	7	4	5	6	4	5	6	5	4	3	6	4	5	5	4
	ASR	0.76	0.80	0.76	0.84	0.80	0.76	0.92	0.76	0.72	0.80	0.84	0.80	0.92	0.80	0.84	0.76	0.80	0.76	0.76	0.84	0.80
Qwen2-Audio	Round	2	3	2	4	3	2	4	3	2	3	4	5	3	4	2	3	4	2	3	5	4
	ASR	0.84	0.92	0.80	0.92	0.84	0.84	0.96	0.84	0.80	0.92	0.92	0.96	0.84	0.92	0.80	0.84	0.92	0.80	0.84	0.96	0.92
SALMONN	Round	5	4	6	5	4	5	6	5	4	6	5	7	6	5	4	5	6	4	5	7	5
	ASR	0.84	0.76	0.68	0.76	0.76	0.68	0.80	0.72	0.64	0.76	0.80	0.84	0.76	0.76	0.68	0.68	0.80	0.92	0.76	0.84	0.76
Whisper+Qwen2	Round	3	4	3	5	4	3	5	4	3	4	5	6	4	5	3	4	5	3	4	6	5
	ASR	0.80	0.84	0.76	0.84	0.80	0.80	0.92	0.80	0.76	0.84	0.84	0.92	0.84	0.80	0.80	0.80	0.92	0.76	0.80	0.92	0.84
DuplexCascade	Round	4	5	4	6	5	4	6	5	4	5	6	7	5	6	4	5	6	4	5	7	6
	ASR	0.72	0.76	0.84	0.80	0.76	0.76	0.84	0.76	0.72	0.80	0.80	0.84	0.80	0.76	0.84	0.76	0.84	0.72	0.76	0.84	0.80

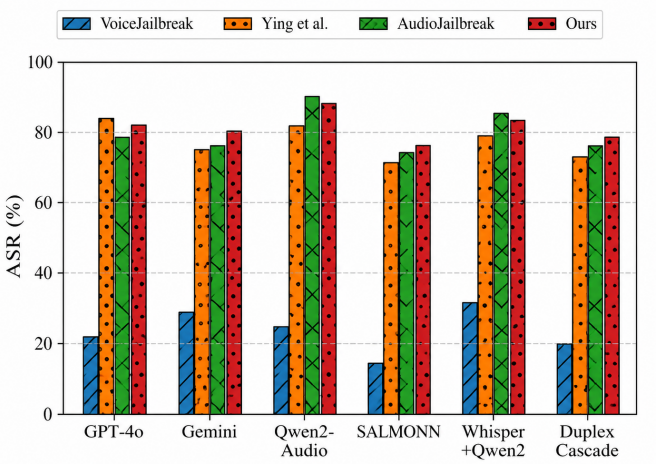


Fig. 2. Comparison with prior audio jailbreak attacks.

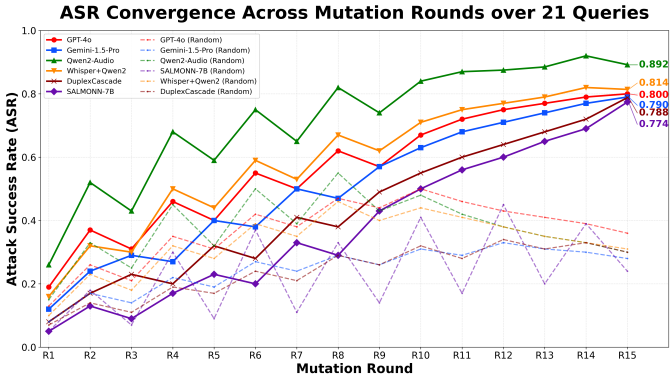


Fig. 3. ASR convergence trends across 15 adaptive mutation rounds over 21 forbidden queries. Different audio-enabled LLM systems exhibit distinct convergence behaviors and mutation sensitivities during adaptive fuzzing.

scenarios. Although SALMONN and DuplexCascade require more mutation rounds before convergence, both systems still produce high ASRs on many forbidden queries once iterative mutation stabilizes. The convergence analysis further reveals architectural differences in attack efficiency. End-to-end systems, such as Qwen2-Audio, typically converge within 2–4 mutation rounds, whereas cascaded systems often require 5–7 rounds before reaching stable policy-violating responses. Despite these differences, both architectures eventually exhibit comparable vulnerability under iterative prompt mutation and acoustic perturbation. We also observe noticeable variation across forbidden queries, suggesting that safety robustness remains highly dependent on the semantic characteristics of individual harmful requests. Overall, the results indicate that current audio-enabled LLM systems remain broadly vulnerable to adaptive jailbreak exploration under black-box settings.

To further contextualize the effectiveness of the proposed framework, Fig. 2 compares the proposed framework against three representative audio jailbreak baselines [6], [7], [10]. Overall, our framework consistently achieves the strongest or near-best attack performance across all evaluated systems. Compared with VoiceJailbreak, which primarily targets GPT-4o through handcrafted voice prompts, our adaptive mutation framework demonstrates substantially stronger cross-model transferability. Although AudioJailbreak achieves competitive performance on several end-to-end audio-language models, our approach provides more consistent attack effectiveness across both cascaded and end-to-end architectures. These results suggest that jointly exploring textual jailbreak mutations and acoustic perturbations significantly improves attack coverage and robustness under heterogeneous deployment settings.

C. Mutation Efficiency

We further evaluate the efficiency of the proposed adaptive mutation framework by analyzing the convergence behavior across mutation rounds. From the query-wise convergence statistics in Table II, we first observe that most successful jailbreaks are achieved within the first several mutation rounds across all evaluated systems. In particular, Qwen2-Audio converges substantially faster than the other evaluated systems, while SALMONN and DuplexCascade require more mutation iterations before reaching stable policy-violating outputs.

TABLE III

ABLATION STUDY OF TEXTUAL MUTATION AND ACOUSTIC PERTURBATION STRATEGIES ACROSS SIX AUDIO-ENABLED LLM SYSTEMS. FOR *Audio-only* SETTINGS, DIRECT HARMFUL AUDIO QUERIES ARE EVALUATED WITHOUT TEXTUAL MUTATION AND TEXTUAL PROMPT, AND ASR IS AVERAGED OVER 25 REPEATED RUNS AND 21 QUERIES. FOR *Text-based + Perturbation* SETTINGS, TEXTUAL PROMPTS ARE ADAPTIVELY MUTATED WHILE USING A FIXED ACOUSTIC PERTURBATION TYPE; ASR IS REPORTED AT THE BEST-PERFORMING CONVERGENCE ROUND AND AVERAGED OVER 25 REPEATED RUNS AND 21 QUERIES.

Setting	**GPT-4o**	**Gemini**	**Qwen2-Audio**	**SALMONN**	**Whisper+Qwen2**	**DuplexCascade**
Text-based (No Perturbation)	0.224	0.136	0.281	0.352	0.307	0.539
Audio-only + Noise	0.338	0.414	0.386	0.362	0.371	0.348
Audio-only + Speed	0.295	0.376	0.343	0.319	0.329	0.310
Audio-only + Echo	0.267	0.343	0.314	0.286	0.300	0.281
Text-based + Real-World Noise	0.743	0.762	0.810	0.705	0.776	0.733
Text-based + Speed Perturbation	0.695	0.714	0.767	0.662	0.724	0.686
Text-based + Echo Perturbation	0.657	0.681	0.724	0.624	0.690	0.648
Full Framework (Adaptive Mutation + Multi-Perturbation)	**0.816**	**0.807**	**0.883**	**0.765**	**0.834**	**0.790**

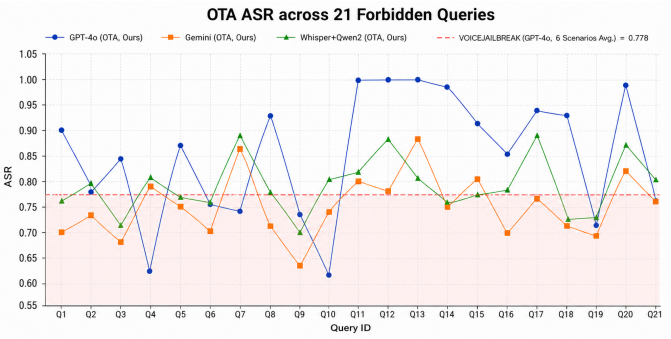


Fig. 4. Query-wise OTA attack success rates across 21 forbidden queries under realistic playback and recording conditions. Our framework achieves average OTA ASRs of 0.827 (GPT-4o), 0.751 (Gemini), and 0.782 (Whisper+Qwen2), with an overall average ASR of 0.787, exceeding the 0.778 OTA ASR reported by VOICEJAILBREAK. The red dashed line indicates the VOICEJAILBREAK baseline on GPT-4o across six forbidden scenarios.

To further investigate whether the observed convergence behavior mainly originates from adaptive optimization rather than mutation diversity alone, we additionally introduce a *Random Mutation* baseline. Unlike the proposed adaptive framework, the random strategy does not use ASR feedback or historical best-performing candidates. At each round r, it randomly samples one textual mutation operation and one or two acoustic perturbations:

$$x_r = \mathcal{A} * p_r(\mathcal{M} * m_r(T_{r-1})), ; m_r! \sim !\mathcal{M}, ; p_r! \sim !\mathcal{P}, \quad (1)$$

where $\mathcal{M}$ denotes the textual mutation set, including semantic compression, semantic expansion, and meaning-preserving rephrasing, and $\mathcal{P}$ denotes the acoustic perturbation pool, including noise, speed, and echo perturbations. The mutated sample x_r is directly evaluated without feedback-guided selection.

We then extend the mutation process to 15 rounds and visualize the ASR progression in Figure 3. Among all evaluated systems, Qwen2-Audio demonstrates the fastest convergence speed and the highest final converged ASR, while SALMONN exhibits substantially delayed convergence behavior. GPT-4o, Gemini, and Whisper+Qwen2 show relatively balanced convergence dynamics with moderate fluctuations during intermediate mutation rounds.

Compared with the proposed adaptive framework, the Random Mutation baseline demonstrates substantially different and highly unstable optimization trajectories across models. Instead of exhibiting consistent convergence behavior, random exploration often produces irregular oscillations, temporary spikes, and even performance degradation in later mutation rounds. For example, Qwen2-Audio achieves several high-ASR mutation rounds during the early exploration stage but subsequently collapses toward lower ASR regions, suggesting that random perturbations fail to maintain stable adversarial optimization. In contrast, Gemini and DuplexCascade remain relatively conservative with limited ASR improvements, while GPT-4o shows stronger random exploration capability but still experiences gradual degradation after intermediate mutation rounds.

More importantly, although random mutation occasionally discovers locally effective adversarial variants, its optimization behavior remains highly inconsistent and fails to achieve stable convergence. In contrast, the proposed adaptive framework rapidly increases ASR within the first several mutation rounds and progressively stabilizes toward high-success attack regions. Across all evaluated systems, the adaptive framework consistently achieves substantially higher ASRs and smoother convergence behavior than the random baseline. Despite the heterogeneous early-stage behaviors, all evaluated systems under the adaptive framework gradually approach stable ASR convergence states in later mutation rounds. More importantly, the final converged ASRs remain highly consistent with the best-performing attack results reported in Table III.

D. Mutation Effectiveness

To better understand the rapid convergence behavior observed in Figure 3, we further analyze the effectiveness of individual mutation operators across all evaluated systems.

Table IV summarizes the usage frequency of textual mutation operators. Across most evaluated systems, meaning-preserving rephrasing is selected most frequently, accounting

for approximately 40%–50% of successful mutations. Compared with semantic compression and semantic expansion, rephrasing preserves the original attack intent while substantially altering the surface form of prompts, making it more effective for bypassing safety-sensitive pattern matching. Semantic expansion also contributes significantly, particularly for SALMONN and DuplexCascade, where additional contextual framing appears beneficial for improving attack transferability. In contrast, semantic compression is selected less frequently, suggesting that excessive prompt simplification may reduce the effectiveness of contextual jailbreak structures.

TABLE IV
TEXTUAL MUTATION OPERATOR USAGE (%) AMONG SUCCESSFUL MUTATIONS.

Model	Rephrase	Expansion	Compression
GPT-4o	46.2	34.8	19.0
Gemini	43.7	37.6	18.7
Qwen2-Audio	49.5	32.4	18.1
SALMONN	38.6	41.2	20.2
Whisper+Qwen2	45.8	35.9	18.3
DuplexCascade	40.1	39.4	20.5

We further analyze the distribution of acoustic perturbation operators. As shown in Table V, real-world noise injection is consistently selected most frequently across all evaluated systems, followed by speed perturbation and echo perturbation. This trend aligns closely with the ablation results reported in Table III, where noise-based perturbation achieves stronger attack effectiveness than other single-perturbation strategies. The results suggest that environmental noise introduces beneficial acoustic variability that improves transferability across heterogeneous audio-processing architectures while maintaining sufficient speech intelligibility for downstream decoding.

TABLE V
ACOUSTIC PERTURBATION OPERATOR USAGE (%) AMONG SUCCESSFUL MUTATIONS.

Model	Noise	Speed	Echo
GPT-4o	51.3	29.6	19.1
Gemini	53.7	27.8	18.5
Qwen2-Audio	56.2	26.4	17.4
SALMONN	47.5	31.7	20.8
Whisper+Qwen2	54.8	28.1	17.1
DuplexCascade	49.2	30.5	20.3

Overall, a consistent pattern emerges across all evaluated systems. Successful mutations are dominated by meaning-preserving rephrasing and noise-based perturbations, suggesting that the adaptive framework quickly identifies operator combinations that preserve harmful semantic intent while increasing robustness against safety filtering and acoustic variability. This explains the fast convergence observed in Figure 3 that, performance gains are not merely due to mutation diversity, but to feedback-guided prioritization of high-yield mutation operators. By progressively favoring operator combinations with strong historical attack performance, the framework concentrates exploration around promising regions of the attack space, leading to stable convergence within only a few mutation rounds.

E. Ablation Study

Table III presents the ablation analysis of different textual mutation and acoustic perturbation strategies across six representative audio-enabled LLM systems.

Overall, audio-only perturbation strategies achieve limited attack effectiveness across most evaluated systems. Among the three perturbation settings, environmental noise consistently produces the strongest attack performance, while speed and echo perturbations generally result in lower ASRs. For example, under the Audio-only + Noise setting, Gemini achieves the highest ASR of 0.414, while GPT-4o and DuplexCascade only achieve 0.338 and 0.348, respectively. Similar trends can also be observed for speed and echo perturbations, where most ASRs remain below 0.40. These results indicate that acoustic perturbation alone is insufficient to reliably bypass current safety-alignment mechanisms.

Compared with audio-only attacks, direct text-based jailbreak prompts without perturbation exhibit mixed effectiveness across different systems. While DuplexCascade and SALMONN achieve relatively higher ASRs under the Text-based (No Perturbation) setting, GPT-4o and Gemini remain comparatively more resistant, with ASRs of only 0.224 and 0.136, respectively. This observation suggests that purely semantic jailbreak prompts without acoustic diversification are often insufficient to consistently transfer across heterogeneous audio-enabled architectures.

Combining textual mutation with acoustic perturbation substantially improves attack effectiveness across all target models. Among the perturbation strategies, real-world noise injection consistently achieves the highest ASRs. Under the Text-based + Real-World Noise setting, Qwen2-Audio reaches an ASR of 0.810, while Whisper+Qwen2 and Gemini achieve 0.776 and 0.762, respectively. In comparison, speed perturbation and echo perturbation produce slightly lower but still consistently strong attack performance across all systems.

Finally, the proposed full framework consistently achieves the highest ASRs on all evaluated architectures. By jointly combining adaptive textual mutation with multi-perturbation exploration, our framework significantly expands the adversarial search space beyond fixed single-perturbation settings. As a result, Qwen2-Audio achieves the highest overall ASR of 0.883, followed by Whisper+Qwen2 with 0.834 and GPT-4o with 0.816. Comparatively more robust systems, such as SALMONN and DuplexCascade, remain highly vulnerable under the full framework, with ASRs of 0.765 and 0.790, respectively. Overall, these results indicate that attack effectiveness primarily stems from the synergistic combination of adaptive semantic mutation and acoustic perturbation, rather than either component alone.

F. Over-the-Air Robustness

To further evaluate the real-world applicability of the proposed jailbreak framework, we conducted over-the-air (OTA)

experiments under realistic acoustic conditions. Unlike API-level attacks that directly upload adversarial audio, OTA attacks introduce additional environmental distortion through physical playback and microphone recording, making the attack setting substantially more challenging and practical. Our OTA experiments were conducted in a quiet indoor environment of approximately 20,m^2. Adversarial audio generated by the proposed framework was played through an iPhone 15 Pro Max and recorded using a MacBook Pro M1 microphone with a fixed distance of 1.5,m between the playback and recording devices. Since the precise architecture of GPT-4o's audio interface remains undisclosed, the evaluated systems may internally employ either cascaded ASR-based pipelines or end-to-end audio-language processing architectures.

Fig. 4 presents the query-wise OTA attack success rates across GPT-4o, Gemini, and Whisper+Qwen2. Overall, all evaluated systems remain highly vulnerable under realistic OTA conditions despite the presence of environmental acoustic distortion and recording degradation. Among the three target systems, GPT-4o achieves the highest OTA attack effectiveness, reaching an average ASR of 0.812 across 21 forbidden queries. Whisper+Qwen2 and Gemini also demonstrate strong OTA robustness, achieving average ASRs of 0.785 and 0.752, respectively. We additionally observe that the OTA attack trends remain largely consistent with the API-level evaluations reported in previous sections. Several forbidden queries consistently achieve high ASRs across all systems, while a smaller subset of queries becomes comparatively more resistant after real-world acoustic transmission. This suggests that environmental noise and recording artifacts partially reduce jailbreak transferability for difficult attack scenarios, but do not fundamentally prevent adaptive adversarial exploration. The red dashed line in Fig. 4 indicates the average OTA ASR reported by VOICEJAILBREAK on GPT-4o across six forbidden scenarios. Since the evaluation protocols, attack coverage, and prompt construction strategies are not directly aligned, this comparison is intended only as a reference rather than a strict fair benchmark comparison. Nevertheless, the results demonstrate that adaptive mutation-based jailbreak generation remains highly effective under realistic OTA environments without requiring handcrafted prompts or manual attack engineering.

VI. Conclusion

In this paper, we present a unified framework for black-box audio jailbreak evaluation on both cascaded pipelines and end-to-end large audio-language models (LALMs). Our framework proposes an adaptive mutation engine with feedback-guided exploration of textual jailbreak prompts and acoustic perturbations, enabling automated generation of diverse attack candidates across heterogeneous audio architectures. Through comprehensive evaluations on six representative audio-enabled LLM systems, we demonstrate that both architectural paradigms remain highly vulnerable to audio jailbreak attacks despite their fundamentally different attack surfaces. Our results further show that adaptive exploration substantially improves attack effectiveness and efficiency compared with existing approaches. In addition, extensive ablation studies and over-the-air evaluations confirm the effectiveness and robustness of the proposed framework under realistic deployment conditions. Overall, our findings highlight the need for future safety evaluations to systematically consider audio interaction channels as a critical attack vector for modern LLM systems.

References

[1] Z. Xu, Y. Liu, G. Deng, Y. Li, and S. Picek, "A comprehensive study of jailbreak attack versus defense for large language models," in *Findings of the ACL 2024*, pp. 7432–7449, 2024.

[2] Y. Fu, E. Shayegan, M. M. A. Abdullah, P. Zaree, N. Abu-Ghazaleh, and Y. Dong, "Vulnerabilities of large language models to adversarial attacks," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 5: Tutorial Abstracts)* (L. Chiruzzo, H.-y. Lee, and L. F. R. Ribeiro, eds.), (Bangkok, Thailand), pp. 8–9, Association for Computational Linguistics, Aug. 2024.

[3] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, *et al.*, "Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection," in *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pp. 79–90, 2023.

[4] A. Wei, N. Haghtalab, and J. Steinhardt, "Jailbroken: How does llm safety training fail?," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[5] X. Shen, Z. Chen, M. Backes, Y. Shen, and Y. Zhang, ""do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models," in *Proceedings of the 2024 on ACM SIGSAC Conference on CCS*, pp. 1671–1685, 2024.

[6] Z. Ying, A. Liu, X. Liu, and D. Tao, "Unveiling the safety of gpt-4o: An empirical study using jailbreak attacks," *arXiv preprint arXiv:2406.06302*, 2024.

[7] X. Shen, Y. Wu, M. Backes, and Y. Zhang, "Voice jailbreak attacks against gpt-4o," *arXiv preprint arXiv:2405.19103*, 2024.

[8] Z. Song, Q. Jiang, M. Cui, M. Li, L. Gao, *et al.*, "Audio jailbreak: An open comprehensive benchmark for jailbreaking large audio-language models," *arXiv preprint arXiv:2505.15406*, 2025.

[9] H. Cheng, E. Xiao, J. Shao, Y. Wang, L. Yang, *et al.*, "Jailbreak-audiobench: In-depth evaluation and analysis of jailbreak threats for large audio language models," *Advances in Neural Information Processing Systems*, vol. 38, 2026.

[10] G. Chen, F. Song, Z. Zhao, X. Jia, Y. Liu, *et al.*, "Audiojailbreak: Jailbreak attacks against end-to-end large audio-language models," *IEEE Transactions on Dependable and Secure Computing*, vol. 23, no. 3, pp. 6085–6102, 2026.

[11] L. Huang, X. Jiang, Z. Wang, W. Mo, X. Xiao, *et al.*, "Image-based multimodal models as intruders: Transferable multimodal attacks on video-based mllms," *arXiv preprint arXiv:2501.01042*, 2025.

[12] X. Liu, X. Cui, P. Li, Z. Li, H. Huang, *et al.*, "Jailbreak attacks and defenses against multimodal generative models: A survey," *arXiv preprint arXiv:2411.09259*, 2024.

[13] W. Cui, D. Yu, X. Jiao, Z. Meng, G. Zhang, *et al.*, "Recent advances in speech language models: A survey," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13943–13970, 2025.

[14] Z. Yang, S. Shimizu, Y. Yu, and C. Chu, "When large language models meet speech: A survey on integration approaches," *Findings of the ACL 2025*, pp. 20298–20315, 2025.

[15] Softcery, "Real-Time (Speech-to-Speech) vs Turn-Based (Cascading STT/TTS) Voice Agent Architecture — softcery.com." https://softcery.com/lab/ai-voice-agents-real-time-vs-turn-based-tts-stt-architecture. [Accessed 05-02-2026].

[16] D. Shu, C. Zhang, M. Jin, Z. Zhou, and L. Li, "Attackeval: How to evaluate the effectiveness of jailbreak attacking on large language models," *SIGKDD Explor. Newsl.*, vol. 27, p. 10–19, July 2025.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[18] K. An, Q. Chen, C. Deng, Z. Du, C. Gao, *et al.*, "Funaudiollm: Voice understanding and generation foundation models for natural interaction between humans and llms," 2024.

[19] S. Team, "Silero vad: pre-trained enterprise-grade voice activity detector (vad), number detector and language classifier." https://github.com/snakers4/silero-vad, 2021.

[20] A. Zeng, Z. Du, M. Liu, K. Wang, S. Jiang, *et al.*, "Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot," *arXiv preprint arXiv:2412.02612*, 2024.

[21] D. Zhang, S. Li, X. Zhang, J. Zhan, P. Wang, *et al.*, "Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities," in *Findings of the EMNLP 2023*, pp. 15757–15773, 2023.

[22] C. Fu, H. Lin, Z. Long, Y. Shen, M. Zhao, *et al.*, "Vita: Towards open-source interactive omni multimodal llm," *arXiv preprint arXiv:2408.05211*, 2024.

[23] S. Ghosh, S. Kumar, A. Seth, C. K. R. Evuru, U. Tyagi, S. Sakshi, O. Nieto, R. Duraiswami, and D. Manocha, "Gama: A large audio-language model with advanced audio understanding and complex reasoning abilities," 2024.

[24] Q. Fang, S. Guo, Y. Zhou, Z. Ma, S. Zhang, *et al.*, "Llama-omni: Seamless speech interaction with large language models," 2025.

[25] Y. Shu, S. Dong, G. Chen, W. Huang, R. Zhang, *et al.*, "Llasm: Large language and speech model," 2023.

[26] Y. Chu, J. Xu, Q. Yang, H. Wei, X. Wei, *et al.*, "Qwen2-audio technical report," 2024.

[27] OPENAI, "ChatGPT can now see, hear, and speak — openai.com." https://openai.com/index/chatgpt-can-now-see-hear-and-speak/, 2023. [Accessed 14-05-2026].

[28] S. Ji, Y. Chen, M. Fang, J. Zuo, J. Lu, H. Wang, Z. Jiang, L. Zhou, S. Liu, X. Cheng, *et al.*, "Wavchat: A survey of spoken dialogue models," *arXiv preprint arXiv:2411.13577*, 2024.

[29] M. Chen, L. Lu, Z. Ba, and K. Ren, "Phoneytalker: An out-of-the-box toolkit for adversarial example attack on speaker recognition," in *IEEE INFOCOM 2022*, pp. 1419–1428, 2022.

[30] G. Chen, Y. Zhang, Z. Zhao, and F. Song, "QFA2SR: Query-Free adversarial transfer attacks to speaker recognition systems," in *32nd USENIX Security Symposium (USENIX Security 23)*, (Anaheim, CA), pp. 2437–2454, USENIX Association, Aug. 2023.

[31] G. Chen, Z. Zhao, F. Song, S. Chen, L. Fan, F. Wang, and J. Wang, "Towards understanding and mitigating audio adversarial examples for speaker recognition," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 5, pp. 3970–3987, 2023.

[32] X. Shen, Y. Wu, M. Backes, and Y. Zhang, "Voice jailbreak attacks against gpt-4o," 2024.

[33] E. Bagdasaryan, T.-Y. Hsieh, B. Nassi, and V. Shmatikov, "Abusing images and sounds for indirect instruction injection in multi-modal llms," *arXiv preprint arXiv:2307.10490*, 2023.

[34] Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in *Proceedings of the 36th International Conference on Machine Learning* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 5231–5240, PMLR, 09–15 Jun 2019.

[35] L. Schonherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," in *Proceedings 2019 Network and Distributed System Security Symposium*, Internet Society, 2019.

[36] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, *et al.*, "CommanderSong: A systematic approach for practical adversarial voice recognition," in *27th USENIX Security Symposium (USENIX Security 18)*, (Baltimore, MD), pp. 49–64, USENIX Association, Aug. 2018.

[37] Z. Li, Y. Wu, J. Liu, Y. Chen, and B. Yuan, "Advpulse: Universal, synchronization-free, and targeted audio adversarial attacks via subsecond perturbations," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1121–1134, 2020.

[38] Y. Deng, C. S. Xia, H. Peng, C. Yang, and L. Zhang, "Large language models are zero-shot fuzzers: Fuzzing deep-learning libraries via large language models," in *Proceedings of the 32nd ACM SIGSOFT international symposium on software testing and analysis*, pp. 423–435, 2023.

[39] J. Ackerman and G. Cybenko, "Large language models for fuzzing parsers (registered report)," in *Proceedings of the 2nd International Fuzzing Workshop*, FUZZING 2023, (New York, NY, USA), p. 31–38, Association for Computing Machinery, 2023.

[40] C. S. Xia, M. Paltenghi, J. Le Tian, M. Pradel, and L. Zhang, "Fuzz4all: Universal fuzzing with large language models," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pp. 1–13, 2024.

[41] C. Yang, Y. Deng, R. Lu, J. Yao, J. Liu, R. Jabbarvand, and L. Zhang, "Whitefox: White-box compiler fuzzing empowered by large language models," *Proc. ACM Program. Lang.*, vol. 8, Oct. 2024.

[42] R. Meng, M. Mirchev, M. Böhme, and A. Roychoudhury, "Large language model guided protocol fuzzing," in *Proceedings of the 31st Annual Network and Distributed System Security Symposium (NDSS)*, pp. 1–17, 2024.

[43] H. Wu, B. Fang, and F. Xie, "Smart fuzzing of 5g wireless software implementation," *arXiv preprint arXiv:2309.12994*, 2023.

[44] J. Hu, Q. Zhang, and H. Yin, "Augmenting greybox fuzzing with generative ai," *arXiv preprint arXiv:2306.06782*, 2023.

[45] Z. Liu, C. Chen, J. Wang, M. Chen, B. Wu, *et al.*, "Testing the limits: Unusual text inputs generation for mobile app crash detection with large language model," *arXiv preprint arXiv:2310.15657*, 2023.

[46] F. Qiu, P. Ji, B. Hua, and Y. Wang, "Chemfuzz: Large language models-assisted fuzzing for quantum chemistry software bug detection," in *23rd IEEE International Conference on Software Quality, Reliability, and Security. Accompany. (QRS 2023).*, 2023.

[47] Google, "AI-Powered Fuzzing: Breaking the Bug Hunting Barrier — security.googleblog.com." https://security.googleblog.com/2023/08/ai-powered-fuzzing-breaking-bug-hunting.html, 2023. [Accessed 25-05-2026].

[48] Y. Gu, J. He, and K. Chen, "Adaptive domain inference attack with concept hierarchy," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, KDD '25, (New York, NY, USA), p. 413–424, Association for Computing Machinery, 2025.

[49] R. Peri, S. M. Jayanthi, S. Ronanki, A. Bhatia, K. Mundnich, *et al.*, "Speechguard: exploring the adversarial robustness of multi-modal large language models," in *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10018–10035, 2024.

[50] M. Kang, C. Xu, and B. Li, "Advwave: Stealthy adversarial jailbreak attack against large audio-language models," in *The Thirteenth International Conference on Learning Representations*, 2025.

[51] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proceedings of the 40th International Conference on Machine Learning*, ICML'23, JMLR.org, 2023.

[52] J. Yang, Y. Fujita, and Y. Sudo, "Duplexcascade: Full-duplex speech-to-speech dialogue with vad-free cascaded asr-llm-tts pipeline and micro-turn optimization," *arXiv preprint arXiv:2603.09180*, 2026.

[53] Google, "Text-to-speech ai: Lifelike speech synthesis — google cloud," 2025. Accessed: 2025-12-01.

[54] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, (New York, NY, USA), p. 1041–1044, Association for Computing Machinery, 2014.

[55] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pp. 1015–1018, ACM Press.

[56] T. O. A. Library, "OpenAIR &x2013; Open Air Library — openair.hosted.york.ac.uk." https://www.openair.hosted.york.ac.uk/. [Accessed 01-12-2025].

[57] P. Chao, E. Debenedetti, A. Robey, M. Andriushchenko, F. Croce, *et al.*, "Jailbreakbench: An open robustness benchmark for jailbreaking large language models," in *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[58] M. Mazeika, L. Phan, X. Yin, A. Zou, Z. Wang, *et al.*, "Harmbench: A standardized evaluation framework for automated red teaming and robust refusal," in *Forty-first International Conference on Machine Learning*, 2024.

[59] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, "Universal and transferable adversarial attacks on aligned language models," *arXiv preprint arXiv:2307.15043*, 2023.

[60] T. Xie, X. Qi, Y. Zeng, Y. Huang, U. M. Sehwag, *et al.*, "SORRY-bench: Systematically evaluating large language model safety refusal," in *The Thirteenth International Conference on Learning Representations*, 2025.