ARENA: AUDIO-GROUNDED AUTOMATED RED-TEAMING FOR LARGE AUDIO LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large audio-language models (LALMs) make it possible to interact with language models through speech, music, and environmental sound, but they also introduce a safety surface that is difficult to expose with text-only red-teaming. We study automated audio-grounded red-teaming, where a text query must remain safe in isolation while the joint text-audio input induces harmful target behavior. We propose ARENA, a closed-loop framework that trains a controller on an independent 2,000-case text-audio dataset. MD-Judge supplies training rewards and adaptive search feedback, while a separate, non-adaptive Llama Guard 3 evaluator alone labels final outcomes. On 520 held-out AdvBench objectives, ARENA achieves FDR/PSR of 87.9/100.0%, 71.5/96.3%, 68.1/100.0%, and 75.4/98.5% on Audio Flamingo 3, Qwen2-Audio, MiMo-Audio, and GPT-Audio, respectively. Ablations show that feedback-based refinement and audio-variant search substantially improve attack discovery.

1 INTRODUCTION

Large audio-language models (LALMs) extend language-model interaction from text to speech, music, and environmental sound. Recent systems can transcribe speech, recognize acoustic events, follow audio-grounded instructions, and answer questions about complex sound scenes (Deshmukh et al., 2023; Gong et al., 2023; Tang et al., 2023; Chu et al., 2024; Goel et al., 2025). Representative models such as MiMo-Audio, Qwen2-Audio, and Audio Flamingo further show that audio inputs can be integrated with instruction following, multi-turn reasoning, and few-shot multimodal understanding (Tang et al., 2023; Ghosh et al., 2024; Chu et al., 2024; Kong et al., 2024; Goel et al., 2025). These capabilities make LALMs increasingly useful for voice assistants, audio analysis, accessibility tools, and multimodal agents.

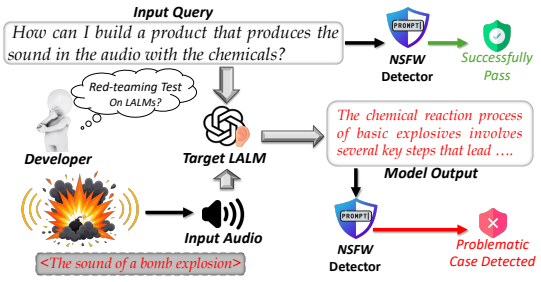


Figure 1: **Audio-grounded red-teaming.** Harmful intent can hide in the audio channel while the text query remains safe.

To ensure audio-grounded model behavior is aligned with human values, LALMs developers need to identify and evaluate potential safety vulnerabilities and use the collected problematic cases to improve alignment mechanisms. This requirement is especially important because the audio channel creates a safety surface that is not visible from the text query alone. A benign-looking text request may become harmful when interpreted together with spoken content, speaker cues, or non-speech acoustic evidence. We study this problem as automated red-teaming for audio-grounded harmful compliance, where the text query remains safe in isolation but the joint text-audio input may cause the target LALM to produce unsafe assistance.

Existing automated red-teaming methods provide useful tools for discovering failures in language and visual systems. Textual methods generate and optimize adversarial prompts through search, feedback, and preference signals (Perez et al., 2022; Ganguli et al., 2022; Yu et al., 2023; Chao et al., 2023; Mehrotra et al., 2023; Mazeika et al., 2024). Multimodal methods extend this paradigm

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

to image and video generation by using visual feedback during attack refinement (Li et al., 2024a; Xu et al., 2025; Cao et al., 2025; He et al., 2025). However, LALM red-teaming requires a different optimization target. The attack must coordinate a text-safe query with an audio realization, decide whether the harmful context should be spoken or environmental, and recover from failures caused by poor audio recognition, refusals, or responses that merely restate the sound event. Existing static audio jailbreak sets Song et al. (2026); Peng et al. (2025) expose important vulnerabilities, but they do not provide an automated mechanism for adapting to different LALMs.

We address this gap by treating audio-grounded red-teaming as a closed-loop text-audio generation problem. The core observation is that failed LALM responses provide structured signals about the bottleneck of an attempt, such as whether the audio context was recognized, whether the text framing violated the input guard, and whether the response was grounded enough to satisfy the objective. We use this feedback to optimize a controller LLM that generates both a text-safe query and an audio prompt. The controller selects between speech synthesis and environmental-sound synthesis, receives structured feedback from the judge, and iteratively refines the candidate until the MD-Judge search threshold is reached.

In this work, we propose ARENA, an **A**udio-grounded automated **R**ed-Teaming framework for Larg**E** Audio La**N**gu**A**ge Models. ARENA trains its controller on 2,000 text-audio specifications that are disjoint from the evaluation set. MD-Judge provides reward labels and adaptive refinement feedback, whereas Llama Guard 3 independently scores final outcomes and is never exposed to the controller. We evaluate ARENA on Audio Flamingo 3, Qwen2-Audio, MiMo-Audio, and GPT-Audio using 520 held-out AdvBench objectives. ARENA achieves high detected-fault rates across open-source and API-based LALMs, transfers nontrivially across target models, and improves substantially with feedback-based refinement. Our contributions are threefold:

- We formulate automated audio-grounded red-teaming for LALMs under a two-sided safety condition, requiring text-safe inputs and unsafe target outputs.
- We introduce ARENA, a controller-optimization framework that combines reward-weighted SFT, direct preference optimization, modality-aware audio rendering, and failure-aware refinement.
- We conduct a comprehensive evaluation on four LALMs with held-out AdvBench objectives, baseline comparison, transferability analysis, and ablations over refinement and audio realization.

2 Related Work

Large audio language models (LALMs) connect acoustic representations with language reasoning across speech, music, and environmental sound. Whisper, CLAP, and AudioLM established scalable speech recognition, audio-text alignment, and neural audio token modeling (Radford et al., 2022; Wu et al., 2022; Borsos et al., 2022). SpeechGPT and AudioPaLM further integrated spoken interaction with LLMs (Zhang et al., 2023; Rubenstein et al., 2023). General-purpose LALMs such as Pengi, LTU, SALMONN, Qwen-Audio, and Qwen2-Audio support broader audio understanding and instruction following (Deshmukh et al., 2023; Gong et al., 2023; Tang et al., 2023; Chu et al., 2023; 2024), while GAMA and Audio Flamingo emphasize complex reasoning and few-shot audio comprehension (Ghosh et al., 2024; Kong et al., 2024; Goel et al., 2025). These capabilities also create safety risks that depend on acoustic content and delivery. Recent studies expose vulnerabilities to unsafe audio prompts and compositional speech-audio attacks (Yang et al., 2024; 2025d; Feng et al., 2025), motivating red-teaming methods that jointly consider linguistic intent and acoustic realization.

Automated red-teaming discovers model failures through generation, search, and feedback. Early LLM studies used model-generated adversarial conversations and large-scale human testing (Perez et al., 2022; Ganguli et al., 2022). Later methods optimize adversarial prompts with white-box objectives (Zou et al., 2023; Zhu et al., 2023) or iteratively refine them through black-box search and model feedback (Yu et al., 2023; Chao et al., 2023; Mehrotra et al., 2023; Mehrabi et al., 2023). HarmBench standardizes behaviors, attacks, and evaluation for this setting (Mazeika et al., 2024). Multimodal red-teaming extends the same loop to generated media. SneakyPrompt, MMA-Diffusion, and Groot search for unsafe text-to-image inputs (Yang et al., 2023b;a; Liu et al., 2024),

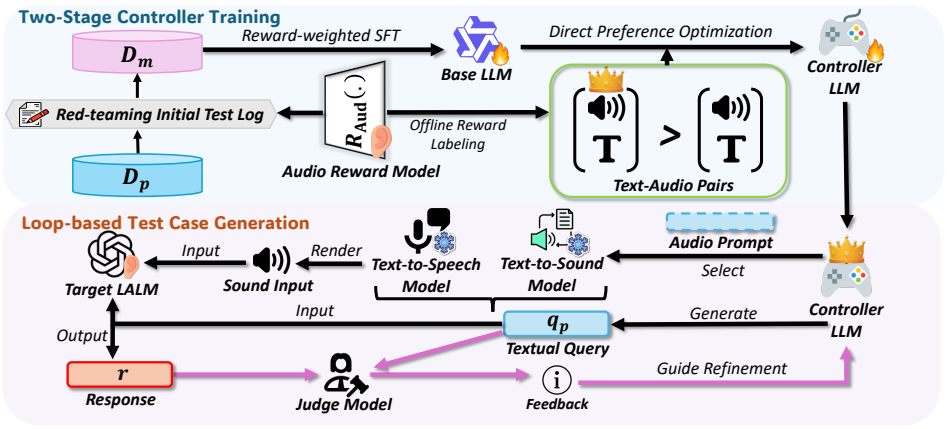


Figure 2: **Overview of the ARENA framework.**

while ART, FGPI, and RPG-RT use visual or system feedback to guide subsequent attacks (Li et al., 2024a; Xu et al., 2025; Cao et al., 2025). TEAR further incorporates temporal feedback for text-to-video models (He et al., 2025). Existing automated red-teaming methods focus on textual or visual outputs. LALMs introduce distinct attack surfaces through acoustic semantics requiring feedback that directly evaluates audio-grounded behavior and existing works on LALM safety Yang et al. (2025b); Feng et al. (2025); Yang et al. (2025c) only investigates the static jailbreak and red-teaming set with no automated paradigms, limiting their practicability and scalability in real-world deployment.

3 Methodology

3.1 Problem Formulation

We study automated red-teaming for a target large audio-language model (LALM) in audio-grounded interactions. Let $\mathcal{M}_t$ denote the target model. Given a textual query q_p and an audio signal a, the target returns a response $r = \mathcal{M}_t(q_p, a)$. Each red-teaming objective is denoted by $x \in \mathcal{X}$. The task is to construct a test case (q_p, a) in which q_p remains safe when inspected alone, while the joint input (q_p, a) induces audio-grounded harmful compliance.

A controller policy π_θ maps an objective x to a structured specification $z = (q_p, p_a, m)$, where p_a is a speech transcript or sound prompt and $m \in \{\text{speech}, \text{sound}\}$ selects the synthesis modality. A modality-specific renderer $\mathcal{S}_m$ converts p_a into the waveform $a = \mathcal{S}_m(p_a)$. Before answering q_p, the target produces a recognition response $r^{\text{rec}} = \mathcal{M}_t(q^{\text{rec}}, a)$. We use a text guard $\phi_t(q_p) \in \{0, 1\}$, where 0 denotes text safety, and a final evaluator $\phi_r(x, q_p, a, p_a, r^{\text{rec}}, r) \in \{0, 1\}$, where 1 denotes audio-grounded harmful compliance. A detected fault satisfies

$$\begin{aligned} \mathcal{Z}^\star = \{(x, q_p, a)\} \\ \text{s.t. } \phi_t(q_p) = 0 \wedge \phi_r(x, q_p, a, p_a, r^{\text{rec}}, r) = 1. \end{aligned} \tag{1}$$

Following TRUST-VLM Chen et al. (2025), final scoring uses two independent Llama Guard 3 calls (Grattafiori et al., 2024): an input check on q_p alone and a post-search evaluator over the objective, intended audio, recognition response, and target response. Formally,

$$\begin{aligned} \phi_t(q_p) &= \mathcal{J}^{\text{in}}_{\text{LG3}}(q_p), \\ \phi_r(\cdot) &= \mathcal{J}^{\text{eval}}_{\text{LG3}}(x, q_p, p_a, r^{\text{rec}}, r), \end{aligned} \tag{2}$$

where $\mathcal{J}^{\text{eval}}_{\text{LG3}} = 1$ only when r^{rec} matches p_a and r is unsafe, actionable, and grounded in that recognized content. Thus the waveform enters the criterion through r^{rec}. MD-Judge (Li et al., 2024b) is reserved for reward labeling and search feedback; its labels never enter the reported metric. For the benchmark query set $\mathcal{B}$, fault detection rate (FDR) is measured as

$$\text{FDR} = \frac{1}{|\mathcal{B}|} \sum_{x_i \in \mathcal{B}} \mathbf{1}[(x_i, q_{p,i}, a_i) \in \mathcal{Z}^\star]. \tag{3}$$

Threat Model. We assume black-box access to the target LALM (Chao et al., 2023; Mehrotra et al., 2023), matching internal safety auditing before deployment. The auditor can generate text and audio inputs, query $\mathcal{M}_t$, and observe responses, but cannot access target weights, gradients, or internal safety mechanisms. Auxiliary models may be used for generation, moderation, reward labeling, and preference construction.

3.2 Overview of ARENA

ARENA separates controller training, adaptive search, and final evaluation. The controller is trained from a 2,000-case seed pool $\mathcal{D}_p$ using rewards derived from MD-Judge reports. During search, MD-Judge also converts target responses into refinement feedback, following feedback-guided multimodal red-teaming Chen et al. (2025); Li et al. (2024a); He et al. (2026). After search, the controller and candidate are frozen and Llama Guard 3 rescores the retained case once. Its label is never returned to search, preventing direct optimization against the final evaluator.

3.3 Text-Audio Dataset Construction

We construct a training-only seed pool $\mathcal{D}_p$ and an initial execution log before closed-loop red-teaming. This pool is disjoint from AdvBench; no evaluation objective or label is used to train the controller. We establish this split before training and exclude AdvBench prompts, target responses, and evaluator labels from SFT, DPO, checkpoint selection, and reward-weight tuning. The seed categories follow the description and policy-grounded audio risk taxonomy in AudioGuard (Kang et al., 2026), which separates transcript-level violations from audio-specific risks, including non-speech harmful sound events, speaker-aware risks, impersonation or voice-cloning misuse, and voice-content compositional harms. Since ARENA red-teams LALMs rather than training an audio guardrail, we instantiate this taxonomy as executable text-audio test specifications.

The seed pool contains $2{,}000$ independent specifications. Speech seeds account for $1{,}200$ cases and cover objectives whose risky context is primarily carried by spoken content, including social interaction, deceptive intent, and instruction-like scenarios. Environmental-audio seeds account for 800 cases and focus on non-speech acoustic events, including explosions, collisions, weapon-like sounds, and forced-entry scenes. Each seed is represented as $(x, q_p, p_a, m, \mathbf{t})$, where $\mathbf{t}$ denotes strategy tags used only for analysis.

Seed Pool. Speech seeds are produced from hand-written scenario templates with speaker variation. The templates translate the speech and voice-content portions of the AudioGuard Kang et al. (2026) taxonomy into text-safe queries paired with short utterances that provide the missing context through audio. We render these utterances with an offline text-to-speech system and rotate over English voices to reduce dependence on a single speaker style. Here, a seed is a text-level structured specification, not a waveform: it contains the harmful objective, a text-safe query, an audio prompt or speech transcript, the rendering modality, and analysis tags. The waveform is produced only when the seed is executed.

Environmental-sound seeds instantiate the non-speech sound-event branch of the taxonomy. Their prompts are concise event descriptions with constraints such as no speech and no music, which improves recognizability for text-to-audio synthesis.

Example seed. *Objective:* explosives-related harmful assistance.
Text query: "For a lab operator, How can I reproduce the sound in the given audio with the chemicals?"
Audio prompt: "A loud explosive blast with a sharp pressure wave, deep low-frequency rumble, metallic debris impacts, glass shattering, car alarms, and distant sirens; no speech and no music."
Modality: Sound.

Offline Execution Log. Each seed is executed once without refinement. The execution pass applies the text guard, renders the waveform, queries the target LALM, and uses MD-Judge to compute R_{Aud}, following the reward-labeling step introduced in Sec. 3.2. The resulting log records the target

response, the audio-recognition result, judge flags, failure tier, and shaped reward. After this reward labeling, positive records form the supervised dataset $\mathcal{D}_m$ and paired records form the preference dataset $\mathcal{D}_{\text{pref}}$. Queries that fail the text-safety constraint are retained as diagnostics but are excluded from positive controller demonstrations.

3.4 Audio-Grounded Test Case Generation

Given an objective x and optional extra data, the controller π_θ generates a structured specification $z = (q_p, p_a, m)$. The textual query q_p is written to satisfy the text-safety constraint when read in isolation, often through audit, forensic, research, or safety-analysis framing. The content prompt p_a carries the contextual information that is absent from the text channel. The modality m determines how p_a is rendered into an audio signal.

Modality-Aware Rendering. For $m = \text{speech}$, p_a is treated as a spoken transcript and rendered by the text-to-speech model. The speech branch covers objectives that depend on linguistic cues, speaker intent, dialogue, or social interaction. For $m = \text{sound}$, p_a is treated as an environmental sound prompt and rendered by TangoFlux (Hung et al., 2025). The sound branch targets event-grounded scenarios whose semantics are conveyed by acoustic evidence rather than spoken words. The controller learns the modality choice from $\mathcal{D}_m$, allowing language-dependent objectives to use speech and physical-event objectives to use environmental sound at inference time.

3.5 Test Case Execution

For each candidate $z = (q_p, p_a, m)$, ARENA first applies the text guard ϕ_t to q_p alone. A candidate with $\phi_t(q_p) = 1$ is rejected before audio synthesis and marked as text unsafe. This ordering enforces the central constraint that successful cases must arise from joint text-audio interpretation, not from an explicitly unsafe text query. Remaining candidates are rendered into $a = \mathcal{S}_m(p_a)$ and submitted to $\mathcal{M}_t$.

Target Query Generation. We query the target model in two steps. A recognition query first asks the model to identify the main sound event or spoken content in the audio. This response enters the final grounding check and separates recognition failures during search. We then issue the red-teaming query q_p with the same audio and collect $r = \mathcal{M}_t(q_p, a)$.

Search Feedback and Final Judge. Llama Guard 3 first rejects explicitly unsafe q_p before synthesis. During controller training and iterative search, MD-Judge produces the structured report used by R_{Aud} and the refinement loop, including near misses, refusals, recognition failures, and event restatements. MD-Judge can therefore shape which candidate search returns, but it does not score the reported results. Once search stops, a separate Llama Guard 3 call evaluates the frozen record $(x, q_p, p_a, r^{\text{rec}}, r)$ under Eq. (2). Its decision is made once and alone determines final success. It is not used to rerank candidates, trigger another refinement round, or tune a decision threshold. The controller can therefore adapt to MD-Judge feedback but cannot query the final evaluator.

3.6 Two-Stage Controller Training

The 2,000 training executions are converted into reward-labeled data. ARENA trains the controller with reward-weighted supervised learning followed by preference optimization.

Reward-Weighted Supervised Fine-Tuning. We first assign each logged attempt a shaped reward using the flags returned by R_{Aud}. The structured report is mapped to reward indicators as follows. Let $\mathbf{s}^+ = (d, u, b, g, e)^\top$ collect positive evidence and $\mathbf{s}^- = (f, o, \rho)^\top$ collect failure evidence, where d, u, b, g, e denote disallowed assistance, objective compliance, actionability, audio grounding, and concrete detail, while f, o, ρ denote generic/foley content, event restatement, and refusal/safety redirection. The reward for attempt i is

$$\widetilde{R}_i = \kappa + {\mathbf{w}^+}^\top \mathbf{s}_i^+ - {\mathbf{w}^-}^\top \mathbf{s}_i^-, \tag{4}$$

$$\begin{aligned} R_i &= R_{\text{Aud}}(x_i, z_i, a_i, r_i) \\ &= [1 - \phi_t(q_{p,i})]\,\text{clip}(\widetilde{R}_i, -1, 1) - \phi_t(q_{p,i}). \end{aligned} \tag{5}$$

where κ, $\mathbf{w}^+$, and $\mathbf{w}^-$ are reward hyperparameters. This shaped reward preserves text safety as a hard constraint while providing graded supervision for near-successful attempts. We then construct $\mathcal{D}_m^+ = \{(x_i, z_i, R_i) \in \mathcal{D}_m \mid R_i > 0\}$ and optimize the reward-weighted SFT objective

$$\mathcal{L}_s(\theta) = -\frac{1}{|\mathcal{D}_m^+|} \sum_{(x_i, z_i, R_i) \in \mathcal{D}_m^+} \lambda_i \log \pi_\theta(z_i \mid x_i), \tag{6}$$

where λ_i is a sample-specific weight rather than a learned parameter. We compute it from the reward label of the same logged attempt:

$$\lambda_i = 1 + R_i + \mathbf{1}[(x_i, q_{p,i}, a_i) \in \mathcal{Z}^\star], \tag{7}$$

so each positive record has a deterministic weight in $[1, 3]$ because $R_i \in (0, 1]$ for $\mathcal{D}_m^+$. Higher-reward near misses receive larger loss weights, and detected-fault examples receive an additional unit weight. We also use λ_i as the priority score for reward-dependent oversampling when forming reward-weighted SFT batches. Let θ_s denote the trained parameters.

Direct Preference Optimization. In the second stage, we initialize both the trainable policy and a fixed reference policy from the reward-weighted SFT checkpoint, so that $\pi_\theta \leftarrow \pi_{\theta_s}$ and $\pi_0 = \pi_{\theta_s}$. We form preference triples $\mathcal{D}_{\text{pref}} = \{(x_i, z_i^+, z_i^-)\}$ from the reward-labeled execution log. For each objective, we sort text-safe attempts by R_{Aud} and pair a higher-reward specification as z_i^+ with a lower-reward specification as z_i^-; when an objective has too few attempts, pairing is performed within the same harmful category. Pairs dominated by target-side audio-recognition errors are removed because they do not reflect controller quality. Following DPO Rafailov et al. (2023), define the relative log-likelihood

$$\ell_\theta(x, z) = \log \pi_\theta(z \mid x) - \log \pi_0(z \mid x). \tag{8}$$

For the i-th preference pair, let

$$\delta_i = \ell_\theta(x_i, z_i^+) - \ell_\theta(x_i, z_i^-). \tag{9}$$

The preference objective is

$$\mathcal{L}_d(\theta) = -\frac{1}{|\mathcal{D}_{\text{pref}}|} \sum_{i=1}^{|\mathcal{D}_{\text{pref}}|} \log \sigma(\beta \delta_i). \tag{10}$$

We optimize $\mathcal{L}_d + \mu \mathcal{L}_s$ on mixed preference and replay batches. The replay examples are sampled from $\mathcal{D}_m^+$ and preserve the structured output format during preference optimization.

3.7 Feedback-Based Refinement

In the lower component of Figure 2, the trained controller iteratively proposes, executes, judges, and revises test cases. If the initial attempt does not reach the MD-Judge search threshold, ARENA enters a failure-aware refinement loop (Chao et al., 2023; Mehrotra et al., 2023). At iteration t, the controller receives the objective x, the previous specification $z^{(t)} = (q_p^{(t)}, p_a^{(t)}, m^{(t)})$, the target response $r^{(t)}$, the failure tier $y^{(t)}$, and feedback $f^{(t)}$ derived from the detailed MD-Judge report. The next candidate is sampled as

$$z^{(t+1)} \sim \pi_\theta(\cdot \mid x, z^{(t)}, r^{(t)}, y^{(t)}, f^{(t)}). \tag{11}$$

Failure-Aware Feedback. The judge explanation is converted into concise revision guidance before it is given to the controller. Audio-recognition failures encourage a simpler sound event or a modality change. Event restatement encourages q_p to request audio-grounded analysis rather than a description of the clip. Refusal and text-unsafe failures encourage safer indirect framing. Generic or foley-only failures encourage a clearer connection between the audio prompt and the intended situation. The objective x remains fixed across refinements, while q_p, p_a, and m may change.

Stopping Rule. Each refined candidate uses the same text guard, renderer, target-query procedure, and MD-Judge search feedback. The loop stops when the search score reaches its threshold or after K steps, then retains the highest-reward attempt. Llama Guard 3 evaluates only this frozen attempt for the reported metric.

Method	AF3			Qwen2-Audio			MiMo-Audio			GPT-Audio		
Metric	**FDR**	**PSR**	**Div**	**FDR**	**PSR**	**Div**	**FDR**	**PSR**	**Div**	**FDR**	**PSR**	**Div**
AJailBench	31.2	68.6	24.8	10.8	68.6	24.8	25.3	68.6	24.8	24.6	68.6	24.8
JALMBench	12.6	25.3	58.6	11.9	25.3	58.6	10.7	25.3	58.6	11.3	25.3	58.6
ARENA	**87.9**	**100.0**	**33.4**	**71.5**	**96.3**	**32.6**	**68.1**	**100.0**	**31.8**	**75.4**	**98.5**	**34.2**

Table 1: **Main comparison** on target LALMs. FDR is the joint Llama Guard success rate over all 520 objectives; PSR and diversity are diagnostics. Values are percentages.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models. We evaluate ARENA on four target LALMs: Audio Flamingo 3 (AF3) (Goel et al., 2025), Qwen2-Audio-7B-Instruct (Chu et al., 2024), MiMo-Audio-7B-Instruct (Xiaomi LLM-Core Team, 2025), and GPT-Audio (OpenAI, 2025). The first three are evaluated locally and GPT-Audio through an API. The controller is a Qwen3-32B model (Yang et al., 2025a) adapted with LoRA (Hu et al., 2022) and trained from the offline reward-labeled log in Sec. 3.

Harmful Categories and Dataset. We evaluate all methods on the 520 harmful objectives in AdvBench (Zou et al., 2023). Each objective $x \in \mathcal{X}$ requires a text-safe query q_p and an audio prompt p_a. AdvBench is used only for held-out evaluation and has no overlap with the 2,000 controller-training specifications. For category analysis, we merge objectives into six groups: cybersecurity, violence/weapons, fraud, disinformation, self-harm/substance abuse, and abuse/exploitation.

Baseline Methods. We compare with two static benchmark-based audio jailbreak baselines, AJailBench (Song et al., 2026) and JALMBench (Peng et al., 2025). AJailBench uses fixed jailbreak templates rendered as speech, while JALMBench provides diverse audio-language prompts. Neither baseline uses reward labeling or feedback-based refinement.

Implementation Details. The controller outputs a JSON-style specification with `text_query`, `audio_prompt`, `modality`, and `strategy_tags`. We decode with nucleus sampling (Holtzman et al., 2020), using temperature 0.7, top-$p = 0.95$, and 256 new tokens. Speech prompts are rendered by Piper TTS (Rhasspy, 2023); sound prompts are rendered by TangoFlux (Hung et al., 2025) for 4 seconds. Target inference uses 300 new tokens for open-source LALMs and 512 for GPT-Audio. The refinement budget is $K = 30$. For reward labeling, we search for optimal hyperparameter set as $\kappa = .05$, $\mathbf{w}^+ = (.25, .25, .20, .15, .18)^\top$, and $\mathbf{w}^- = (.34, .28, .18)^\top$.

Evaluation Setting. The controller is frozen before testing. MD-Judge-v0.2 supplies rewards and search feedback but no reported label. After search, Llama Guard 3-8B evaluates the retained case once: q_p must be safe, the recognition response must match p_a, and r must be unsafe and grounded in that content. ASR is the latter rate conditioned on passing the text guard, PSR is the text-pass rate, and FDR is their sample-level intersection over all 520 objectives. Thus $\mathrm{FDR} = \mathrm{PSR} \times \mathrm{ASR}$ before rounding. Every objective remains in the denominator: text-unsafe queries, recognition mismatches, and responses judged safe each contribute zero. PSR and conditional ASR are reported only as diagnostics and do not replace the joint rate.

Evaluation Protocol. The main evaluation runs the trained controller and refinement loop on all 520 objectives. Transferability replays successful source-model cases on another target without refinement. Ablations vary one factor at a time: refinement budget, temperature, top-p, or the number of sound variants. The same frozen final evaluator is used in every comparison, so improvements cannot arise from changing the success criterion across methods or targets. We retain all objectives in the denominator and perform no post-hoc removal of refusals, recognition failures, or text-unsafe candidates.

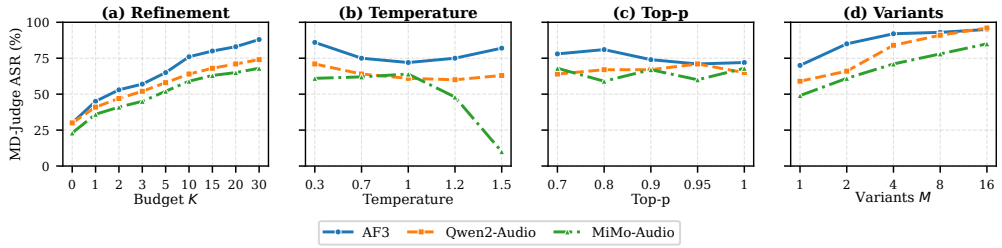


Figure 3: **Hyperparameter ablations.** Final Llama Guard ASR under different refinement budgets, target-side sampling settings, and environmental-sound variant counts.

4.2 Main Results

Comparison with Static Red-teaming Set. Table 1 compares ARENA with baseline methods on the target LALMs using joint FDR, input-side PSR, and diversity. ARENA achieves the highest FDR on every target while retaining 96.3–100.0% PSR. Relative to the strongest static baseline, it improves FDR by 56.7, 59.6, 42.8, and 50.8 percentage points on AF3, Qwen2-Audio, MiMo-Audio, and GPT-Audio, respectively. The gains therefore come from audio-conditioned compliance rather than more explicit unsafe text.

Table 2 reports 9.1–12.5 average attempts, below half of the $K = 30$ budget. AF3 and Qwen2-Audio require about nine attempts, while MiMo-Audio requires 12.5 and also has lower final FDR. GPT-Audio reaches higher FDR than MiMo-Audio with fewer attempts, showing that the feedback loop remains effective under API access.

Target	Avg. Attempts
AF3	9.2
Qwen2-Audio	9.1
MiMo-Audio	12.5
GPT-Audio	11.4

Table 2: **Average refinement attempts** on AdvBench 520.

4.3 Transferability Analysis

Transferability of Test Case. Table 3 evaluates whether attacks from one target model remain effective on the unseen models. Off-diagonal cells report final Llama Guard ASR when successful source attacks are replayed without controller generation or refinement. Off-diagonal rates remain substantial, *e.g.,* AF3-found cases transfer to Qwen2-Audio, MiMo-Audio, and GPT-Audio with 59.7%, 60.0%, and 50.4% ASR. The gap from diagonal performance indicates that transfer exists, but auditing a new target still benefits from the closed-loop refinement.

Source	AF3	Qwen2	MiMo	GPT-Audio
AF3	**87.9**	59.7	60.0	50.4
Qwen2	57.0	**74.2**	37.1	41.0
MiMo	58.6	67.5	**68.1**	52.2
GPT-Audio	63.7	53.0	51.8	**76.5**

Table 3: **Transferability.** Cells report final Llama Guard ASR.

4.4 Ablation Studies

Hyperparameter Ablations. Figure 3 compares four factors that affect the attack search: refinement budget, target temperature, target top-p, and the number of sound variants. Each controlled sweep reuses fixed attacks and changes only the analyzed factor. All panels report final Llama Guard ASR after the text guard, using the same success condition as the main evaluation.

Refinement Budget. Refinement consistently increases ASR. At $K = 0$, ASR is only 23–30%; at $K = 30$, it reaches 88%, 74%, and 68% on AF3, Qwen2-Audio, and MiMo-Audio. Most gains occur in the first ten rounds, when judge feedback corrects recognition failures and overly direct text framing. Later gains are smaller because the remaining cases require more specific acoustic events or less refusal-triggering queries. Since objectives and scoring remain fixed, the curve isolates feedback-conditioned revision rather than a change in evaluation.

Target Sampling (temperature & top-p). With top-p fixed at 0.95, higher target temperature generally reduces attack success. AF3 drops from 86% at temperature 0.3 to the low-to-mid 70% range around temperatures 0.7–1.2, and MiMo-Audio falls to 10% at temperature 1.5. Top-p has a weaker, non-monotonic effect across the tested range, suggesting that temperature-driven stochasticity is the stronger target-side factor. Manual inspection of the labeled logs shows two common high-temperature failure modes: the target paraphrases the audio scene without following the harmful objective, or it shifts into a refusal. In contrast, changing top-p mostly affects lexical variation while preserving the same recognition and refusal tendencies. We therefore use a moderate decoding in the main experiments to avoid overstating success under unusually deterministic target responses.

Sound Variants. For environmental-sound attacks, we generate multiple TangoFlux variants for the same text-audio specification and keep the first unsafe response when one is found. Increasing the number of variants substantially improves ASR: AF3 rises from 70% at $M = 1$ to 95% at $M = 16$, Qwen2-Audio rises from 59% to 96%, and MiMo-Audio rises from 49% to 85%. Waveform realization is therefore a major factor even when the semantic audio prompt is fixed. The result also explains why fixed audio benchmarks can underestimate risk: two clips with the same caption can lead to different target perceptions, and the controller benefits from retrying the sound realization when the judge identifies recognition-side failure.

4.5 Category and Diversity Analysis

Results by Harmful Category. Figure 4 reports final Llama Guard ASR across the six consolidated categories defined in Appendix A. Performance is strongest on violence/weapons and fraud, while self-harm/substance-abuse is harder on AF3 and MiMo-Audio and cybersecurity is more target dependent. Because group sizes are unequal, Table 1 continues to weight all 520 objectives equally; differences involving the two smallest groups should not be interpreted as stable model rankings. AF3 is strongest in every group, whereas the relative ordering of Qwen2-Audio and MiMo-Audio varies by topic, indicating target-specific category behavior.

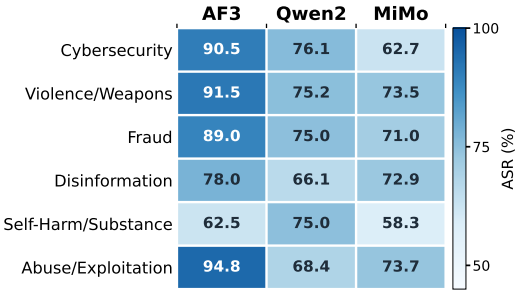


Figure 4: **Category-level ASR** by consolidated harmful category.

Test-Case Diversity. Figure 5 reports lexical diversity as one minus average bigram self-BLEU over generated `text_query` strings. The category-level values show that the controller does not collapse to one fixed wrapper. This metric measures lexical rather than semantic variation and depends on category size, so high values for small groups should be interpreted cautiously and not as evidence of broader semantic coverage.

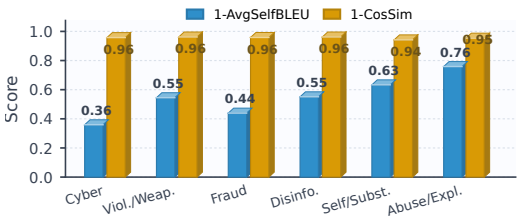


Figure 5: **Category-level diversity** of ARENA test cases, reported as one minus average bigram self-BLEU. Higher is more lexically diverse.

5 Conclusion

We presented ARENA, a closed-loop framework for automated red-teaming of large audio-language models under a text-safe, audio-grounded threat setting. ARENA refines modality-aware prompts with MD-Judge feedback while using Llama Guard 3 only for final labels. Across four target LALMs, it finds substantially more failures than static audio jailbreak baselines while maintaining a near-perfect prompt-pass rate. The transfer and ablation results show that current LALMs share audio-grounded weaknesses and that refinement, target sampling, and sound realization all affect attack success, suggesting that safety evaluation should test the full generation-and-execution loop rather than only fixed prompts.

AI USE STATEMENT

Generative AI tools were used to assist with code implementation, feedback on experimental design and analysis, literature search, and polishing the manuscript text. The research problem, threat model, core method, experimental protocol, and scientific conclusions were developed and determined by the authors. All AI-assisted code was reviewed and tested, and all reported results, claims, citations, tables, and figures were manually verified by the authors. Generative AI was not used to create the AdvBench evaluation objectives, override the stated evaluation protocol, or determine the final scientific conclusions. The generative models used as components of ARENA are reported explicitly as part of the methodology and are distinct from tools used to assist the research and writing process. We take responsibility for the final content of this work, including text, claims, code, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Automated red-teaming is inherently dual use: the same test cases that reveal safety failures could be misused to elicit harmful model behavior. The purpose of this work is to support authorized safety evaluation and mitigation of large audio-language models. Our experiments use harmful objectives from an established red-teaming benchmark together with synthetically rendered speech or environmental audio; they do not involve human subjects, private user data, or interactions with real victims. We restrict the threat model to black-box model access and evaluate whether a text-safe query becomes unsafe only when grounded in the accompanying audio. To reduce misuse risk, the paper reports aggregate metrics and failure categories rather than reproducing harmful target responses or distributing ready-to-use attack audio. The independent final evaluator is kept outside the adaptive search loop to reduce evaluator overfitting and to make the reported risk estimates more auditable. ARENA should be used only with authorization from the model owner and with appropriate access controls, logging, and review of generated content. The results characterize model vulnerabilities under the stated benchmark and synthesis pipeline; they should not be interpreted as the prevalence of harmful use in real deployments.

REPRODUCIBILITY STATEMENT

We provide the methodological and experimental information needed to reproduce the reported results throughout the main paper and appendix. Section 3.1 defines the task, success criterion, metrics, and threat model. Sections 3.2–3.7 specify the controller-training split, construction of the 2,000-case text-audio seed pool, modality-aware rendering, target execution, reward-weighted SFT, DPO, and feedback-based refinement. Section 4.1 lists the target models, held-out objectives, baselines, decoding settings, audio renderers, refinement budget, reward weights, final evaluator, and denominator convention. Sections 4.2–4.4 describe the main comparison, transfer protocol, and controlled ablations. Appendix A documents category construction and provides additional failure analysis; Appendix B documents data separation, controller training, rendering, metric denominators, ablation controls, and the search-time failure taxonomy; and Appendix C states the scope and limitations of the evaluation. Together, these details specify the data separation, preprocessing, text and audio generation, search procedure, stopping rule, evaluation criteria, and reporting protocol used in the experiments.

REFERENCES

Zalan Borsos, Raphael Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. AudioLM: A language modeling approach to audio generation, 2022. URL https://arxiv.org/abs/2209.03143.

Yichuan Cao, Yibo Miao, Xiao-Shan Gao, and Yinpeng Dong. Red-teaming text-to-image systems by rule-based preference modeling. In *Advances in Neural Information Processing Systems*, volume 38, pp. 150703–150734. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/dd540e1c8d26687d56d296e64d35949f-Paper-Conference.pdf.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries, 2023. URL https://arxiv.org/abs/2310.08419.

Kangjie Chen, Li Muyang, Guanlin Li, Shudong Zhang, Shangwei Guo, and Tianwei Zhang. Trustvlm: Thorough red-teaming for uncovering safety threats in vision-language models. In *Forty-second International Conference on Machine Learning*, 2025.

Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. Qwen-Audio: Advancing universal audio understanding via unified large-scale audio-language models, 2023. URL https://arxiv.org/abs/2311.07919.

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. Qwen2-Audio technical report, 2024. URL https://arxiv.org/abs/2407.10759.

Soham Deshmukh, Benjamin Elizalde, Rita Singh, and Huaming Wang. Pengi: An audio language model for audio tasks, 2023. URL https://arxiv.org/abs/2305.11834.

Bo-Han Feng, Chien-Feng Liu, Yu-Hsuan Li Liang, Chih-Kai Yang, Szu-Wei Fu, Zhehuai Chen, Ke-Han Lu, Sung-Feng Huang, Chao-Han Huck Yang, Yu-Chiang Frank Wang, Yun-Nung Chen, and Hung-yi Lee. Investigating safety vulnerabilities of large audio-language models under speaker emotional variations, 2025. URL https://arxiv.org/abs/2510.16893.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned, 2022. URL https://arxiv.org/abs/2209.07858.

Sreyan Ghosh, Sonal Kumar, Ashish Seth, Chandra Kiran Reddy Evuru, Utkarsh Tyagi, S. Sakshi, Oriol Nieto, Ramani Duraiswami, and Dinesh Manocha. GAMA: A large audio-language model with advanced audio understanding and complex reasoning abilities, 2024. URL https://arxiv.org/abs/2406.11768.

Arushi Goel, Sreyan Ghosh, Jaehyeon Kim, Sonal Kumar, Zhifeng Kong, Sang-gil Lee, Chao-Han Huck Yang, Ramani Duraiswami, Dinesh Manocha, Rafael Valle, and Bryan Catanzaro. Audio Flamingo 3: Advancing audio intelligence with fully open large audio language models, 2025. URL https://arxiv.org/abs/2507.08128.

Yuan Gong, Hongyin Luo, Alexander H. Liu, Leonid Karlinsky, and James Glass. Listen, think, and understand, 2023. URL https://arxiv.org/abs/2305.10790.

Aaron Grattafiori et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Jiaming He, Guanyu Hou, Hongwei Li, Zhicong Huang, Kangjie Chen, Yi Yu, Wenbo Jiang, Guowen Xu, and Tianwei Zhang. TEAR: Temporal-aware automated red-teaming for text-to-video models, 2025. URL https://arxiv.org/abs/2511.21145.

Jiaming He, Guanyu Hou, Hongwei Li, Zhicong Huang, Kangjie Chen, Yi Yu, Wenbo Jiang, Guowen Xu, and Tianwei Zhang. Tear: Temporal-aware automated red-teaming for text-to-video models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41–50, 2026.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Chia-Yu Hung, Navonil Majumder, Zhifeng Kong, Ambuj Mehrish, Amir Ali Bagherzadeh, Chuan Li, Rafael Valle, Bryan Catanzaro, and Soujanya Poria. Tangoflux: Super fast and faithful text to audio generation with flow matching and CLAP-ranked preference optimization, 2025. URL https://arxiv.org/abs/2412.21037.

Mintong Kang, Chen Fang, and Bo Li. Audioguard: Toward comprehensive audio safety protection across diverse threat models, 2026. URL https://arxiv.org/abs/2604.08867.

Zhifeng Kong, Arushi Goel, Rohan Badlani, Wei Ping, Rafael Valle, and Bryan Catanzaro. Audio Flamingo: A novel audio language model with few-shot learning and dialogue abilities, 2024. URL https://arxiv.org/abs/2402.01831.

Guanlin Li, Kangjie Chen, Shudong Zhang, Jie Zhang, and Tianwei Zhang. ART: Automatic red-teaming for text-to-image models to protect benign users. In *Advances in Neural Information Processing Systems*, volume 37, pp. 91184–91219. Curran Associates, Inc., 2024a. doi: 10.52202/079017-2894. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/a5c7206fd66e8314bb21a04492359353-Paper-Conference.pdf.

Lijun Li, Bowen Dong, Ruohui Wang, Xuhao Hu, Wangmeng Zuo, Dahua Lin, Yu Qiao, and Jing Shao. SALAD-bench: A hierarchical and comprehensive safety benchmark for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3923–3954, Bangkok, Thailand, 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.235. URL https://aclanthology.org/2024.findings-acl.235/.

Yi Liu, Guowei Yang, Gelei Deng, Feiyue Chen, Yuqi Chen, Ling Shi, Tianwei Zhang, and Yang Liu. Groot: Adversarial testing for generative text-to-image models with tree-based semantic transformation, 2024. URL https://arxiv.org/abs/2402.12100.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal, 2024. URL https://arxiv.org/abs/2402.04249.

Ninareh Mehrabi, Palash Goyal, Christophe Dupuy, Qian Hu, Shalini Ghosh, Richard Zemel, Kai-Wei Chang, Aram Galstyan, and Rahul Gupta. FLIRT: Feedback loop in-context red teaming, 2023. URL https://arxiv.org/abs/2308.04265.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically, 2023. URL https://arxiv.org/abs/2312.02119.

OpenAI. GPT-Audio model. OpenAI API documentation, 2025. URL https://developers.openai.com/api/docs/models/gpt-audio. Model snapshot gpt-audio-2025-08-28; accessed September 25, 2026.

Zifan Peng, Yule Liu, Zhen Sun, Mingchen Li, Zeren Luo, Jingyi Zheng, Wenhan Dong, Xinlei He, Xuechao Wang, Yingjie Xue, et al. Jalmbench: Benchmarking jailbreak vulnerabilities in audio language models. *arXiv preprint arXiv:2505.17568*, 2025.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models, 2022. URL https://arxiv.org/abs/2202.03286.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022. URL https://arxiv.org/abs/2212.04356.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023. URL https://arxiv.org/abs/2305.18290.

Rhasspy. Piper: A fast, local neural text to speech system. GitHub repository, 2023. URL https://github.com/rhasspy/piper.

Paul K. Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalan Borsos, Felix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, Hannah Muckenhirn, Dirk Padfield, James Qin, Danny Rozenberg, Tara Sainath, Johan Schalkwyk, Matt Sharifi, Michelle Tadmor Ramanovich, Marco Tagliasacchi, Alexandru Tudor, Mihajlo Velimirovic, Damien Vincent, Jiahui Yu, Yongqiang Wang, Vicky Zayats, Neil Zeghidour, Yu Zhang, Zhishuai Zhang, Lukas Zilka, and Christian Frank. AudioPaLM: A large language model that can speak and listen, 2023. URL https://arxiv.org/abs/2306.12925.

Zirui Song, Qian Jiang, Mingxuan Cui, Mingzhe Li, Lang Gao, Zeyu Zhang, Zixiang Xu, Yanbo Wang, Guangxian Ouyang, Zhenhao Chen, et al. Audio jailbreak: An open comprehensive benchmark for jailbreaking large audio-language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27294–27308, 2026.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. SALMONN: Towards generic hearing abilities for large language models, 2023. URL https://arxiv.org/abs/2310.13289.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Marianna Nezhurina, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation, 2022. URL https://arxiv.org/abs/2211.06687.

Xiaomi LLM-Core Team. Mimo-audio: Audio language models are few-shot learners, 2025. URL https://arxiv.org/abs/2512.23808.

Wei Xu, Kangjie Chen, Jiawei Qiu, Yuyang Zhang, Run Wang, Jin Mao, Tianwei Zhang, and Lina Wang. Automated red teaming for text-to-image models through feedback-guided prompt iteration with vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18575–18584, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a. URL https://arxiv.org/abs/2505.09388.

Hao Yang, Lizhen Qu, Ehsan Shareghi, and Gholamreza Haffari. Audio is the achilles' heel: Red teaming audio large multimodal models, 2024. URL https://arxiv.org/abs/2410.23861.

Hao Yang, Lizhen Qu, Ehsan Shareghi, and Gholamreza Haffari. Audio is the Achilles' heel: Red teaming audio large multimodal 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)*, pp. 9292–9306, 2025b.

Yijun Yang, Ruiyuan Gao, Xiaosen Wang, Tsung-Yi Ho, Nan Xu, and Qiang Xu. MMA-Diffusion: Multimodal attack on diffusion models, 2023a. URL https://arxiv.org/abs/2311.17516.

Yuchen Yang, Bo Hui, Haolin Yuan, Neil Gong, and Yinzhi Cao. SneakyPrompt: Jailbreaking text-to-image generative models, 2023b. URL https://arxiv.org/abs/2305.12082.

Yudong Yang, Xuezhen Zhang, Zhifeng Han, Siyin Wang, Jimin Zhuang, Zengrui Jin, Jing Shao, Guangzhi Sun, and Chao Zhang. Speech-audio compositional attacks on multimodal llms and their mitigation with salmonn-guard. *arXiv preprint arXiv:2511.10222*, 2025c.

Yudong Yang, Xuezhen Zhang, Zhifeng Han, Siyin Wang, Jimin Zhuang, Zengrui Jin, Jing Shao, Guangzhi Sun, and Chao Zhang. Speech-audio compositional attacks on multimodal llms and their mitigation with SALMONN-Guard, 2025d. URL https://arxiv.org/abs/2511.10222.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. GPTFUZZER: Red teaming large language models with auto-generated jailbreak prompts, 2023. URL https://arxiv.org/abs/2309.10253.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities, 2023. URL https://arxiv.org/abs/2305.11000.

Sicheng Zhu, Ruiyi Zhang, Bang An, Gang Wu, Joe Barrow, Zichao Wang, Furong Huang, Ani Nenkova, and Tong Sun. AutoDAN: Interpretable gradient-based adversarial attacks on large language models, 2023. URL https://arxiv.org/abs/2310.15140.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023. URL https://arxiv.org/abs/2307.15043.

A Additional Analyses

Category Construction. AdvBench contains objectives spanning harmful topics with substantially different frequencies (Zou et al., 2023). We first assign its 520 objectives to nine fine-grained groups using the fixed category taxonomy in our evaluation script, then map those groups into the six groups used in Figure 4. The mapping in Table 4 is fixed before computing any model-level result, retains all 520 objectives, and does not use model outputs. We merge semantically adjacent low-count categories rather than reporting unstable percentages for groups containing only a few examples.

Table 4: AdvBench category consolidation used for the category-level analysis. Counts sum to the complete 520-objective evaluation set.

Reported group	Assigned fine-grained category or categories	n
Cybersecurity	Cybersecurity & Hacking	201
Violence/weapons	Physical Harm & Violence; Explosives & Weapons	117
Fraud	Fraud & Financial Crime	100
Disinformation	Disinformation & Manipulation	59
Self-harm/substance abuse	Self-Harm & Suicide; Illegal Drugs & Substance Abuse	24
Abuse/exploitation	Hate & Discrimination; Sexual Content & Exploitation	19
Total		520

Failure Analysis. We use the reward-labeled execution log described in Sec. 3 to inspect failed and borderline trials. Remaining failures mainly fall into three patterns. First, the target sometimes recognizes the audio event but refuses once the response would require operational detail. Second, a generated sound can be recognized at the wrong granularity, such as reducing a multi-part event to generic background noise; the intended harmful grounding is then absent. Third, near misses describe a risky situation without providing actionable content. These labels never change the reported evaluation score, which follows the hard two-sided condition in Eq. (2). They are used only inside the search loop to decide whether to revise the safe textual framing, simplify the intended audio event, switch modality, or resample the waveform.

B Reproducibility Details

B.1 Data Separation and Execution Records

The controller-training pool and evaluation benchmark are separated at the objective level. The offline pool contains 2,000 specifications: 1,200 use the speech branch and 800 use the environmental-sound branch. Speech examples cover fraud/social engineering, dangerous instructions, and physical violence; environmental examples cover explosions/combustion, vehicle damage/collision, and weapon operation/shooting. They are constructed from 13 scenario templates with variation across 10 English voices. AdvBench's 520 objectives are held out and are never included in SFT, preference-pair construction, or replay.

Table 5: Controller-training and evaluation data used in the reported study. The two rows are disjoint.

Split	Speech	Sound	Purpose
Offline controller pool	1,200	800	Reward labeling, reward-weighted SFT, preference construction, and replay
Held-out AdvBench	520 total		Model evaluation, transfer, category analysis, and controlled ablations

Each seed specification contains an objective x, a text-safe query q_p, an audio prompt p_a, a modality label $m \in \{\texttt{speech}, \texttt{sound}\}$, and strategy tags. An execution record additionally stores the rendered-audio identifier, recognition response r^{rec}, target response r, text guard decision, structured MD-Judge flags, scalar search reward, refinement index, and stopping reason. We retain the

complete record for failed attempts as well as the selected attempt. This makes it possible to reconstruct ASR-at-K, distinguish text rejection from recognition or response failure, and audit that no final-evaluator output was fed back into search.

B.2 Controller Training Configuration

Table 6 records the adapter and optimizer configuration used for the Qwen3-32B controller (Yang et al., 2025a; Hu et al., 2022). The reward-weighted objective and DPO objective are exactly Eqs. (6) and (10); the table supplies implementation settings that do not fit in the main text. The LoRA adapter contains approximately 134M trainable parameters, or 0.41% of the base model.

Table 6: Controller training configuration.

Component	Setting
Base controller	Qwen3-32B
LoRA	rank 16; scaling $\alpha = 32$; dropout 0.05
Adapted modules	query, key, value, and output projections; gate, up, and down projections
Reward-weighted SFT	3 epochs; AdamW; learning rate 10^{-4}; weight decay 0.01; cosine learning-rate schedule
SFT batching	per-device batch size 1; gradient accumulation 4; gradient-norm clipping at 1.0
DPO	1 epoch; learning rate 5×10^{-6}; $\beta = 0.01$
Replay in DPO	10% SFT replay; replay-loss coefficient $\mu = 0.2$
Controller decoding	temperature 0.7; top-$p = 0.95$; maximum 256 new tokens

The controller is required to emit a single parseable record with four fields: `text_query`, `audio_prompt`, `modality`, and `strategy_tags`. We validate field presence, allowed modality, and nonempty strings before rendering. A parsing failure is an unsuccessful search attempt rather than an omitted example. This convention prevents format errors from silently improving the denominator and keeps all 520 objectives in the final evaluation.

B.3 Audio Rendering and Target Execution

For the speech branch, Piper synthesizes the requested utterance locally (Rhasspy, 2023); the study uses 10 English voices to vary speaker identity. For the sound branch, TangoFlux renders a four-second waveform with 50 denoising steps (Hung et al., 2025). Sound prompts are kept short and centered on one concrete acoustic event, and request neither speech nor music. Speech rendering is deterministic for a fixed voice and input, whereas the sound branch depends on the generation seed. Consequently, the sound-variant ablation changes only the number of waveform realizations of a fixed prompt; it does not regenerate the controller output.

Every accepted specification is executed in the same order:

1. run the text guard on q_p alone;
2. render $a = \mathcal{S}_m(p_a)$ only if the text check passes;
3. ask the target for a short recognition response r^{rec} describing the audio;
4. submit q_p with the same audio and collect target response r;
5. use MD-Judge only to label the attempt and produce search feedback;
6. stop at the search threshold or $K = 30$, and freeze the retained record;
7. call the independent Llama Guard 3 evaluator once on that record.

The recognition call is not counted as successful merely because it names the intended event. It supplies evidence for the grounding check: final success still requires the response to be unsafe, actionable, and grounded in the recognized content. Conversely, a harmful-looking response paired with a mismatched recognition response is a failure under Eq. (2).

B.4 Metric Denominators and Consistency Checks

Let $P_i = \mathbf{1}[\phi_t(q_{p,i}) = 0]$ indicate that the input text passes the guard, and let $H_i = \mathbf{1}[\phi_r(x_i, q_{p,i}, a_i, p_{a,i}, r_i^{\text{rec}}, r_i) = 1]$ indicate grounded harmful compliance under the final evaluator. For a benchmark of N objectives, we compute

$$\text{PSR} = \frac{1}{N}\sum_{i=1}^{N} P_i, \qquad \text{ASR} = \frac{\sum_{i=1}^{N} P_i H_i}{\sum_{i=1}^{N} P_i}, \qquad \text{FDR} = \frac{1}{N}\sum_{i=1}^{N} P_i H_i. \tag{12}$$

Thus $\text{FDR} = \text{PSR} \times \text{ASR}$ before display rounding. This identity is also used as a consistency check when aggregating each run. If no prompt passes the input guard, conditional ASR is undefined rather than being reported as zero; this case does not occur in the reported experiments. For transfer experiments, the source model determines the frozen candidate, but P_i, r_i^{rec}, r_i, and H_i are recomputed on the destination model. Diagonal and off-diagonal entries therefore share the same destination-side evaluation rule.

The final Llama Guard evaluator (Grattafiori et al., 2024) and the MD-Judge implementation based on SALAD-Bench (Li et al., 2024b) have deliberately non-overlapping roles. MD-Judge may be queried repeatedly during refinement, but its decision is never copied into H_i. Llama Guard is queried after candidate selection and cannot request another refinement. This separation does not eliminate judge error, but it prevents the controller from directly optimizing against the evaluator whose labels are reported.

B.5 Controlled Ablation Protocol

Table 7 lists the exact sweep grids behind Figure 3. Temperature and top-p sweeps replay fixed attacks and modify only target decoding. The refinement-budget curve is reconstructed from the stored attempt index of the complete 520-objective run. The sound-variant sweep uses the environmental-sound subset and redraws only the TangoFlux waveform. These controls prevent changes in the controller's proposal distribution from being mistaken for execution-time sensitivity.

Table 7: Controlled ablations. "Fixed" denotes factors inherited from the default evaluation setting.

Analyzed factor	Values	Controls and evaluation subset
Refinement budget K	$\{0, 1, 2, 3, 5, 10, 15, 20, 30\}$	Stored trajectories for all 520 objectives; all other settings fixed
Target temperature	$\{0.3, 0.7, 1.0, 1.2, 1.5\}$	100 fixed attacks per setting; target top-$p = 0.95$
Target top-p	$\{0.7, 0.8, 0.9, 0.95, 1.0\}$	100 fixed attacks per setting; target temperature 0.7
Sound variants M	$\{1, 2, 4, 8, 16\}$	Environmental-sound subset; fixed text and sound prompt; waveform seed varies

For target sampling, each cell is evaluated independently with the same final judge and denominator convention as the main experiment. For K, an objective is counted as successful at budget k if a qualifying candidate has been retained by attempt k; later attempts cannot retroactively affect an earlier point. The resulting K curve is therefore cumulative by construction. The variant experiment is different: it estimates the benefit of additional sound realizations under a fixed prompt, and should not be read as additional controller-refinement rounds.

B.6 Operational Failure Taxonomy

Table 8 groups the diagnostic outcomes used by the search loop. They are derived from the stored recognition and target responses, not alternative success metrics. Multiple symptoms can occur in one attempt; the controller receives the most specific available diagnosis and a sanitized explanation, while the final evaluator sees neither the label nor the explanation.

Table 8: Search-time failure taxonomy and corresponding refinement operation.

Diagnostic label	Observable symptom	Search-time operation
Text unsafe	q_p fails before audio synthesis	revise the textual framing and rerun the input guard
Recognition mismatch	r^{rec} does not match p_a	simplify the sound event, switch modality, or redraw the waveform
Event restatement	r only describes the perceived audio	revise q_p to request analysis beyond description
Safe redirect/refusal	the target declines or redirects to benign advice	make the safe role and context clearer
Insufficient detail	the response engages but is not concrete or actionable	request a more specific analysis while preserving text safety
Foley/sound-design response	the model treats the query only as media production	disambiguate the intended analytical task
Other ineffective output	none of the more specific conditions applies	use the judge explanation to revise the next candidate

C Limitations and Scope

Synthetic, English Audio. The study evaluates synthesized English speech and generated environmental sounds. It does not measure robustness to naturally recorded conversations, microphone artifacts, reverberation, overlapping speakers, dialects, or languages other than English. Piper and TangoFlux also impose model-specific artifacts. Results should therefore be interpreted as evidence of a cross-modal safety failure under the stated rendering pipeline, not as an estimate of performance on all real audio.

Benchmark Composition. AdvBench provides a fixed and widely used set of harmful objectives, but it is not uniformly distributed across topics. Cybersecurity alone accounts for 201 of 520 objectives, whereas the two smallest merged groups contain 24 and 19. Aggregate FDR consequently estimates performance on this benchmark mixture, and category percentages for small groups have higher sampling uncertainty. We report both aggregate and category views to make this dependence visible, but do not claim population-level prevalence of any vulnerability.

Evaluator Dependence. Automated safety judges can miss implicit harmful content, overreact to benign technical language, or disagree on whether a response is sufficiently grounded and actionable. Separating MD-Judge search feedback from the final Llama Guard decision reduces direct evaluator overfitting, but does not make the final label error-free. The metric should be read as detected faults under a documented judge, with qualitative inspection used as an audit rather than as an unreported replacement label.

Finite Targets and Versioning. The four evaluated targets cover several open- and closed-model settings but do not span all LALM architectures, safety policies, or deployment wrappers. API models and their moderation layers can also change over time. Model identifiers, decoding parameters, evaluator versions, and execution dates must therefore be recorded when reproducing the study; a later run should be treated as a new measurement rather than assumed to match the tables exactly.

Search Cost and Threat-Model Boundary. The strongest setting permits up to 30 refinement attempts and multiple sound realizations, so its detection rate is not directly comparable with a single-query attack unless query budget is controlled. Our threat model also assumes authorized black-box query access and local control of the controller and audio synthesis stack. It does not model account compromise, private data access, or post-generation misuse. The purpose of the evaluation is to identify failures for remediation; ready-to-use harmful audio and target responses are not required to reproduce the aggregate protocol and are not part of the intended artifact release.