Transferring Text Jailbreaks to Audio in Tool-Using LLM Agents

Kashyap R Manjusha
University of Illinois,
Urbana-Champaign, IL, USA
kr58@ieee.org

Murali Emani
Argonne National Laboratory,
Lemont, IL, USA
memani@anl.gov

Abstract

Safety alignment in large audio-language models (LALMs) is typically evaluated through single-turn interactions with direct prompts. In real-world deployments, however, these models are often integrated into agent frameworks that introduce developer-controlled system prompts, tool-calling loops, and intermediate encoding steps between user speech and model input. This raises a practical question: can a text-based jailbreak remain effective after being converted to speech by an off-the-shelf text-to-speech (TTS) system and processed through an agent? We investigate this question using 270 prohibited questions spanning nine policy categories, an open-weight TTS model with 82 million parameters, and a minimal LangChain question-answering agent whose system prompt combines a persona with a fuzzing-derived template. Across three audio-capable models, we observe attack success rates (ASRs) ranging from 57.0% to 64.1%. To assess the impact of synthesis errors, we use automatic speech recognition to verify the generated audio and find that 87.4% of clips are transcribed exactly. Our analysis indicates that faithfully rendered speech alone achieves an ASR above 50% on every model tested, while manual evaluation of a 50-response sample agrees with the automatic judge in 92% of cases. These findings suggest that the agent framework does not compensate for the loss of safety robustness introduced by the text-to-audio transition. Even a readily available TTS pipeline requiring only minutes of computation can therefore provide a practical attack surface for transferring text-based jailbreaks to audio-agent settings.

1 Introduction

Text-based jailbreaks are a well-established threat, with prior work exploring curated prompt collections (Shen et al., 2024), automated search methods (Yu et al., 2023), and persona-based conditioning (Zhang et al., 2025). In parallel, recent research has investigated audio jailbreaks, primarily through perturbations designed directly in the waveform domain (Chen et al., 2025; Song et al., 2025). A simpler alternative is to convert an existing text jailbreak into speech. Recent work on omni-models (Chen et al., 2026) shows that vulnerabilities can transfer across modalities and that text-based attacks delivered through audio can rival purpose-built audio attacks. However, this evidence comes from direct prompting, leaving open the question of whether such attacks remain effective in deployed agent frameworks.

In practice, voice assistants are often built around agents, such as those implemented with LangChain (LangChain, Inc., 2025), that introduce developer-defined system prompts, encode audio into structured messages, and manage tool-calling loops. These components may provide opportunities to mitigate attacks, but they may also introduce additional vulnerabilities, particularly when third parties can configure the agent.

Our contribution is a focused empirical study of this configuration:

- We develop a minimal agent-based question-answering pipeline in which forbidden questions are delivered exclusively as base64-encoded audio, with adversarial scaffolding confined to the system prompt.
- We evaluate the fidelity of commodity text-to-speech (TTS) using an independent automatic speech recognition (ASR) model to assess the contribution of synthesis errors to observed attack success.
- We evaluate three audio-capable models, spanning small and mid-sized open-weight models and a proprietary frontier model, and observe attack success rates of 57.0–64%. The consistently high rates suggest that text-to-audio

Proceedings of the Workshop on Secure, Safe, and Privacy-Preserving Speech and Audio AI (TrustAudio 2026), pages 24–28
November 10, 2026 ©2026 Association for Computational Linguistics

transfer poses a practical safety risk across diverse model configurations.

The consistently high success rates across models suggest that transferring text jailbreaks through speech poses a practical security concern that warrants further investigation in agent-based deployments.

2 Related Work

Text jailbreaks. Shen et al. (2024) study jailbreak prompts collected from public communities and provide the forbidden-question set used in our evaluation. Yu et al. (2023) introduce mutation-based fuzzing for automated jailbreak discovery, while Zhang et al. (2025) demonstrate the effectiveness of persona-based conditioning. Our system prompt combines these latter two approaches; we withhold the specific template to avoid facilitating misuse.

Audio and cross-modal attacks. Prior work explores audio jailbreaks through waveform-level perturbations (Chen et al., 2025; Song et al., 2025). In contrast, Chen et al. (2026) investigate how text-based vulnerabilities transfer to audio in omni-models, showing that such attacks can rival purpose-built audio attacks under direct prompting. We extend this line of work by evaluating text-to-audio jailbreak transfer within an agent framework and examining the role of TTS fidelity through an independent ASR verification step.

Safety evaluation. Following common practice, we use an LLM-based safeguard classifier to assess model responses (Inan et al., 2023). Specifically, we employ the multimodal Llama Guard 4 12B checkpoint (Meta AI, 2025) and compare its judgments against a manually annotated sample to assess agreement.

3 Threat Model

We consider an adversary who can (i) control the agent's system prompt, for example, by distributing an agent configuration that a victim installs, and (ii) submit audio as the user input. The adversary has no access to model weights, logits, or training data and does not optimize the audio signal. Instead, the attack audio is generated directly by an off-the-shelf text-to-speech (TTS) model. This threat model deliberately assumes limited attacker capabilities, allowing us to examine whether text-based jailbreaks remain effective through the audio interface without relying on sophisticated, waveform-level attacks.

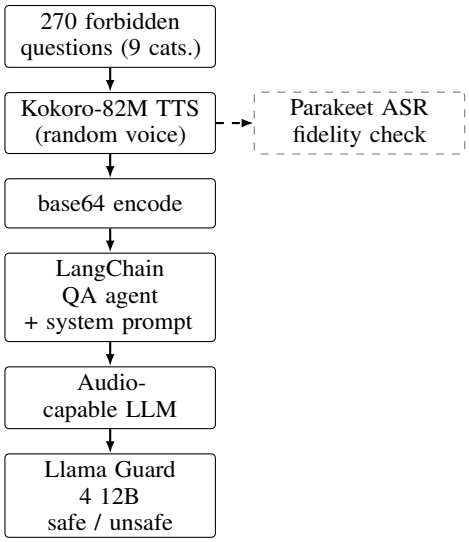


Figure 1: Evaluation pipeline. The harmful request exists in text only before synthesis; from the base64 step onward it reaches the model solely as audio. The dashed branch is a verification path and does not feed the agent.

4 Method

4.1 Dataset

We use the forbidden-question dataset released by Shen et al. (2024).[1] We select 30 questions from each of nine content-policy categories: Economic Harm, Government Decision, Hate Speech, Health Consultation, Illegal Activity, Legal Opinion, Malware, Physical Harm, and Privacy Violence, yielding 270 prompts in total. This balanced sampling design prevents any single category from dominating the pooled attack success rate (ASR) and eliminates the need for category-based reweighting. We omit per-category results for the reason discussed in Section 5.

4.2 Text-to-speech conversion

We synthesize each question using Kokoro-82M, an open-weight text-to-speech (TTS) model (hexgrad, 2025), with the ONNX runtime. We use the default sample rate, set `speed=1.2` and `lang=en-us`. and randomly select a speaker voice from the model's released voice inventory for each question. This randomization reduces the likelihood that the results depend on a particular speaker identity. The

[1] https://github.com/verazuo/jailbreak_llms

entire audio corpus can be generated within minutes on commodity hardware, making this TTS-based attack inexpensive to reproduce and requiring no specialized audio optimization.

4.3 Fidelity verification

As TTS can introduce transcription errors, some audio clips may no longer faithfully convey the original request, potentially affecting the measured attack success rate (ASR). To assess this effect, we transcribe all 270 clips using NVIDIA Parakeet (NVIDIA NeMo Team, 2024) and compare the transcripts with their source text. A total of 236 clips (87.4%) are transcribed exactly, while the remaining 34 (12.6%) differ primarily in punctuation, including parentheses, hyphens, and commas, with the lexical content largely preserved.

We retain all 270 clips in our evaluation rather than excluding those with transcription differences. This choice reflects the intended deployment setting, where an attacker using an off-the-shelf TTS system would not typically inspect and curate every generated clip. Moreover, punctuation changes represent a form of input variation that may itself affect model behavior. In Section 5, we estimate how much these transcription differences could influence our conclusions.

4.4 Agent construction

Each audio clip is base64-encoded and submitted as the user turn to a question-answering agent built with LangChain (LangChain, Inc., 2025). The agent loads its system prompt from a user-supplied Markdown file containing a combination of a fuzzing-derived template (Yu et al., 2023), a persona instruction (Zhang et al., 2025), and an additional persona developed for this study. Importantly, the system prompt contains no harmful requests; the prohibited question is conveyed exclusively through the audio input. Each item is evaluated in a single inference, without multi-turn interaction or escalation.

4.5 Evaluation

Each generated text response is classified by Llama Guard 4 12B (Meta AI, 2025) as `safe` or `unsafe`. We define attack success rate (ASR) as

$$\text{ASR} = \frac{\#\,\text{responses labeled unsafe}}{\#\,\text{total attacks}} \times 100.$$

We deliberately do not score severity or usefulness of the unsafe content; the binary label is the conservative choice and keeps the study reproducible.

Model	Attacks	Succ.	ASR (%)
MiMo-v2.5 (Xiaomi)	270	154	57.0
Gemini-2.5-Flash (Google)	270	172	63.7
Voxtral-Small-24B (Mistral)	270	173	64.1

Table 1: Consolidated attack success rate for text jailbreaks transferred to audio and delivered through a LangChain QA agent, pooled over nine content-policy categories (30 items each). Responses labeled by Llama Guard 4 12B.

4.6 Validating the judge

Because our evaluation relies on a single automated classifier, we assess its agreement with human judgment. We randomly sample 50 responses from the pooled outputs of all three models and independently label them as `safe` or `unsafe` under the same content policy, without access to the classifier's predictions. The two sets of labels agree on 92% of cases (46/50). We treat this result as a sanity check on the reported attack success rates (ASRs), rather than as a formal calibration. With only four disagreements, the sample is insufficient to determine whether the classifier systematically overestimates or underestimates attack success. We therefore leave the reported ASRs unchanged.

5 Results

Table 1 reports ASR for the three evaluated models, consolidated over all 270 items that is, pooled across the nine content-policy categories rather than resolved by category.

We make the following three observations from these results described as follows.

The attack success rates are similar across models. Despite differences in model size, architecture, and deployment, ASRs vary by only 7.1 percentage points across the three models. This consistency suggests that text-to-audio transfer may expose a shared weakness across model configurations, although the limited number of models prevents us from attributing the result solely to the modality boundary.

The proprietary model is not demonstrably safer. Gemini-2.5-Flash, the only commercially served model in our evaluation, falls between the two open-weight models in ASR. Its performance provides no evidence that the additional safeguards of a commercially deployed system reliably prevent attacks delivered through audio.

Synthesis errors do not account for the high ASRs. Because we retain all 34 clips that do not transcribe exactly, the observed ASRs could partly reflect synthesis-induced changes. We assess this possibility by considering the worst case for our conclusion, in which all 34 imperfectly transcribed clips are successful attacks. Even under this assumption, the ASR among the 236 clips transcribed exactly would remain at least 50.8% for MiMo, 58.5% for Gemini, and 58.9% for Voxtral. Under the opposite extreme, in which none of the 34 clips is successful, the corresponding upper bounds are 65.3%, 72.9%, and 73.3%. Thus, the attack remains successful in a majority of cases even when restricted to faithfully transcribed speech. We do not report the observed ASR separately by transcription fidelity and therefore make no claim about whether synthesis degradation increases or decreases attack success.

Scope of the reported results. We report pooled attack success rates (ASRs) across all nine content-policy categories rather than category-specific rates. With only 30 questions per category, individual estimates have substantial uncertainty, making fine-grained comparisons difficult to interpret. By contrast, pooling all 270 questions provides a more precise estimate, with an approximate 95% margin of error of 6 percentage points under a simple binomial model when the observed rate is around 60%. This aggregate analysis aligns with our primary objective: assessing the robustness of the text-to-audio interface rather than identifying the most vulnerable policy categories. A more detailed category-level analysis is left for future work with a larger dataset.

Computational cost. The full evaluation, comprising 270 TTS generations, 270 ASR verifications, and 810 agent inferences, completes in under an hour. Audio synthesis takes only a few minutes on a single consumer GPU. The pipeline requires no optimization loop, gradient access, or per-question tuning, making the attack inexpensive to implement and evaluate with readily available resources.

6 Discussion

These results suggest that the agent scaffold, as configured in our experiments, does not provide an effective defense against text jailbreaks delivered through audio. Base64 encoding, structured message formats, and tool-calling loops do not, by themselves, prevent the model from complying with prohibited requests. Instead, the results highlight a gap in treating model-level alignment as the primary security boundary, particularly when attacks cross modalities.

Two practical mitigations follow from our findings. First, deployments can transcribe incoming audio and apply an existing text-based safety classifier to the resulting transcript. Since 87.4% of our clips are transcribed exactly, this approach could identify many inputs that contain known prohibited requests, although its effectiveness against previously unseen attacks requires further evaluation. Second, system prompts supplied by third parties should be treated as untrusted input and subjected to appropriate security checks, rather than automatically trusted as developer instructions. Neither measure is new; our findings highlight the importance of incorporating such safeguards into agent configurations that accept audio and third-party prompts, rather than relying on model alignment alone.

Limitations

Our study has several limitations and should be interpreted as a preliminary investigation rather than a comprehensive benchmark. We evaluate only three models, selected based on availability within our evaluation budget, and therefore cannot claim that they represent the broader landscape of large audio-language models (LALMs). Our experiments also use a single TTS model, one synthesis configuration, and English-language inputs. We report aggregate ASRs across nine content-policy categories, so the results do not establish whether the observed vulnerability is consistent across categories. Addressing this question will require a larger evaluation with more examples per category and broader model, language, and synthesis coverage.

Our evaluation also relies on a single safety classifier and inherits its potential biases and failure modes. To assess its reliability, we manually labeled 50 responses sampled across the three models and observed 92% agreement with Llama Guard (Section 4.6). However, this small sample cannot establish the direction or distribution of classification errors, motivating a larger annotation study with multiple annotators and inter-annotator agreement measures. Furthermore, our metric captures

whether a response is classified as unsafe, not the practical harm it enables; a response containing one unsafe sentence is treated the same as one providing a complete harmful procedure. Finally, we do not experimentally evaluate any defenses. Although transcribe-then-filter is a promising mitigation, its effectiveness against the attacks studied here remains an open question.

Ethics Statement

This work evaluates the safety of deployed audio-language systems in order to document a gap that defenders can close. We use only a publicly released dataset of forbidden questions (Shen et al., 2024) and publicly released models. We do not publish the system-prompt templates, the persona instructions, the synthesized audio, or any model output; the paper reports aggregate success rates only, and includes no harmful content. Given that the attack uses no novel technique, it composes a public dataset, a public TTS model, and published jailbreak templates, we do not believe it constitutes a zero-day in any single vendor's system. We are nonetheless disclosing our findings, and the transcribe-then-filter mitigation in particular, to the vendors of the three evaluated models, and will report the outcome of that disclosure in the camera-ready version of this paper. We judge that the defensive value of documenting a commodity-cost attack path outweighs the marginal uplift from knowing that the path exists, which prior work has already established (Chen et al., 2026).

7 Acknowledgement

Argonne National Laboratory's work was supported by the U.S. Department of Energy, Advanced Scientific Computing Research (ASCR), under contract DE-AC02-06CH11357.

References

Guangke Chen, Fu Song, Zhe Zhao, Xiaojun Jia, Yang Liu, Yanchen Qiao, Weizhe Zhang, Weiping Tu, Yuhong Yang, and Bo Du. 2025. Audiojailbreak: Jailbreak attacks against end-to-end large audio-language models. *arXiv preprint arXiv:2505.14103*.

Yupeng Chen, Junchi Yu, Aoxi Liu, Philip Torr, and Adel Bibi. 2026. The alignment curse: Cross-modality jailbreak transfer in omni-models. *arXiv preprint arXiv:2602.02557*.

hexgrad. 2025. Kokoro-82m: An open-weight text-to-speech model. https://huggingface.co/hexgrad/Kokoro-82M.

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, and Madian Khabsa. 2023. Llama guard: LLM-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*.

LangChain, Inc. 2025. Langchain. https://github.com/langchain-ai/langchain.

Meta AI. 2025. Llama guard 4 12b model card. https://huggingface.co/meta-llama/Llama-Guard-4-12B.

NVIDIA NeMo Team. 2024. Parakeet: Fast and accurate automatic speech recognition models. https://huggingface.co/nvidia/parakeet-tdt-0.6b-v2.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2024. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS)*.

Zirui Song, Qian Jiang, Mingxuan Cui, Mingzhe Li, Lang Gao, Zeyu Zhang, Zixiang Xu, Yanbo Wang, Chenxi Wang, Guangxian Ouyang, Zhenhao Chen, and Xiuying Chen. 2025. Audio jailbreak: An open comprehensive benchmark for jailbreaking large audio-language models. *arXiv preprint arXiv:2505.15406*.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. 2023. Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts. *arXiv preprint arXiv:2309.10253*.

Zheng Zhang, Peilin Zhao, Deheng Ye, and Hao Wang. 2025. Enhancing jailbreak attacks on LLMs via persona prompts. *arXiv preprint arXiv:2507.22171*.