Misspoken Whispers: Attacking ASR Models to Mistranscribe Toxic Audios

Dan Kienast, Jiaojiao Jiang
School of Computer Science and Engineering
UNSW
Sydney, Australia
{d.kienast, jiaojiao.jiang}@unsw.edu.au

***Abstract*—Automatic speech recognition (ASR) is increasingly used as the front end of content moderation pipelines, where ASR transcripts are passed to downstream text-based toxicity classifiers. This design implicitly assumes that ASR outputs are semantically faithful and robust to benign acoustic variation. In this paper, we show that this assumption can be systematically violated by small, structured, and *speaker-specific* acoustic perturbations that preserve transcript coherence while selectively suppressing toxic lexical content. We introduce a compositional attack that applies a fixed sequence of simple acoustic transformations, with parameters selected via a speaker-level search using a small number of utterances. Once learned, the perturbation generalises to unseen utterances from the same speaker, enabling practical misuse without per-sample optimisation. We evaluate our attack on Whisper-base and eight additional ASR models. For toxic speech, the attack alters more than 92% of transcripts and affects more than 93% of speakers, yielding substantial reductions in text-based toxicity scores. On non-toxic speech, the average change in toxicity remains near zero, indicating minimal collateral effects. We further demonstrate partial transferability across ASR architectures while maintaining intelligible, syntactically plausible transcripts. Our findings reveal a systemic weakness in ASR-dependent moderation pipelines and demonstrate that text-only, transcript-based toxicity detection is insufficient to counter structured audio-level manipulation. We argue that effective defences must incorporate audio-aware robustness mechanisms rather than relying solely on downstream text classifiers.**

***Index Terms*—ASR, adversarial audio, Whisper, toxicity detection, safety, moderation**

Content Warning. The paper includes illustrative examples of hateful content.

I. Introduction

ASR systems have become a core component of modern content moderation pipelines. On many short-form video platforms (such as TikTok, Instagram, and YouTube), spoken content is typically transcribed by an ASR model—often a Whisper variant [1]—before the resulting text is passed to downstream classifiers for toxicity or harm detection[1]. This architecture implicitly assumes that ASR transcripts are semantically faithful and robust to benign acoustic variations, allowing the moderation system to rely almost exclusively on lexical cues extracted from text. However, this design introduces a critical and underexplored vulnerability. Because ASR outputs serve as the sole semantic interface between audio content and moderation systems, any systematic distortion of key lexical items—while preserving overall transcript coherence—can directly undermine downstream toxicity detection. In contrast to overt transcript corruption or unintelligible noise, such manipulations may remain invisible to both automated classifiers and human reviewers who rely on the generated text.

Prior work has demonstrated that ASR models are vulnerable to adversarial perturbations, including targeted transcription manipulation [2], optimisation-based perturbations [3]–[5], and universal disruptions that mute or derail decoding in modern ASR architectures [6]–[8]. While these attacks reveal important robustness limitations, many are impractical for real-world misuse: they often require per-sample optimisation, produce unintelligible or semantically nonsensical transcripts,, or aim to globally disrupt transcription rather than selectively altering harmful content. As a result, existing attacks do not fully capture how ASR vulnerabilities can be exploited to bypass text-based toxicity moderation in realistic deployment settings.

In this work, we study a more practical and targeted threat: *speaker-specific acoustic perturbations* that consistently suppress toxic lexical content in ASR transcripts while preserving intelligibility and syntactic plausibility. Our method composes simple, low-level audio transformations—such as pitch shifts, temporal stretching, and spectral filtering—and selects their parameters at the speaker level; an adversary can learn a deterministic perturbation profile for a given speaker. Once learned, these perturbations generalise to unseen utterances from the same speaker, enabling scalable attacks without per-sample optimisation or access to moderation internals.

We evaluate our attack across multiple ASR architectures, including Whisper variants [1] and non-Whisper models such as wav2vec 2.0 and Speech-to-Text (S2T) on the Civil Comments dataset [2]. Our results show that these speaker-specific perturbations reliably alter toxic transcripts, substantially reducing text-based toxicity scores as measured by Detoxify [9]. Crucially, the resulting transcripts often remain coherent and readable, meaning that moderation pipelines relying solely on transcript-based classifiers fail to detect the manipulation. These findings reveal a systemic weakness in ASR-mediated moderation and challenge the assumption that robust text-based toxicity detection can be achieved without considering audio-level

[1]https://www.tiktok.com/transparency/en/dsa-transparency/

[2]https://www.kaggle.com/competitions/jigsaw-unintended-bias-in-toxicity-classification/data

adversarial behaviour.

This work makes the following contributions:

- We identify a practical vulnerability in ASR-based content moderation, showing that small, structured, speaker-specific acoustic perturbations can selectively suppress toxic lexical content while preserving transcript coherence.
- We propose a deterministic, speaker-level attack framework that learns reusable perturbation profiles without per-sample optimisation or access to moderation internals.
- We demonstrate strong empirical effectiveness and robustness, achieving high attack success rates and substantial toxicity reduction on toxic speech with negligible average impact on non-toxic speech, and showing that these effects persist across multiple Whisper and non-Whisper ASR architectures.

II. Related Work

A. Adversarial Robustness of ASR Systems

A substantial body of work has demonstrated that ASR systems are vulnerable to adversarial manipulation. Early research showed that optimisation-based, targeted attacks on end-to-end models such as DeepSpeech showed that carefully crafted perturbations can force the system to output an attacker-chosen transcript [2]. Subsequent efforts extended classical adversarial techniques, such as Fast Gradient Signed Method (FGSM) [3], Momentum Iterative Fast Gradient Sign Method (MI-FGSM) [4], and Projected Gradient Descent (PGD) [5], from other domains, such as visual to the audio domain, enabling targeted attacks on ASR pipelines.

Beyond transcript manipulation, adversarial audio has also been used to inject hidden or unintelligible voice commands into commercial systems [10]. Other studies examined robustness across architectures and model sizes, highlighting limited transferability and emphasising the role of self-supervised pretraining in improving resilience [11]. Recent work has also analysed adversarial vulnerabilities in multimodal ASR–LLM models [12]. Overall, most existing attacks are targeted, requiring the ASR model to output a pre-specified sentence, and often involve per-sample optimisation that limits scalability.

B. ASR Vulnerabilities in Content Moderation

Modern ASR foundation models such as Whisper [1] have prompted renewed investigation into model robustness. Raina et al. showed that Whisper can be controlled or muted using a universal 0.64-second acoustic perturbation corresponding to the <|endoftext|> token, effectively terminating transcription across datasets and languages [6]. Other work has demonstrated that Whisper can be induced to produce systematic mistranscription or hallucination under carefully constructed distortions [7], typically by substantially increasing Word Error Rate (WER). Since Whisper also supports speech translation, additional research has explored attacks that selectively disrupt translation output [8]. In contrast to these global, model-wide attacks, our work focuses on selectively inducing mistranscriptions for toxic content, thereby affecting downstream toxicity classification.

C. Defences Against Adversarial Audio

Defensive techniques typically fall into two categories: removing adversarial noise or detecting it. Reconstruction-based defences—such as denoising or purification—seek to recover clean speech from adversarially perturbed audio [13]. A related line of work uses diffusion models to purify audio by adding and subsequently removing noise; if the Character Error Rate (CER) increases after purification, the sample is flagged as adversarial [14]. Other studies investigate whether adversarial training can improve the robustness of downstream tasks, such as fake audio detection [15]. While these defences demonstrate progress, they largely target high-distortion or gradient-based attacks; their effectiveness against low-level, compositional perturbations such as those used in our work remains unclear.

D. Social Defence Against Content Moderation

"Algospeak"—intentional lexical substitutions to evade text moderation—is well-documented in social media (e.g., "corn"→porn; "unalive"→suicide) [16]. Our attack achieves a similar level of evasion without altering the speaker's lexical choice: perturbations cause the ASR to output benign substitutions, whereas human listeners hear the original toxic word. This makes the evasion compatible with platforms where speakers cannot or will not change wording, and where moderation is transcript-only.

III. Threat Model

In this section, we specify the threat model that underpins our proposed attack. We consider a large social media or content-sharing platform where users can upload short-form videos or audio clips. The platform employs one or more ASR systems to generate transcripts for two main purposes: (1) subtitle generation, and (2) downstream content moderation via text-based toxicity classifiers. Our attack targets this ASR-mediated moderation pipeline. Fig. 1 illustrates the standard moderation workflow and shows how our adversarial perturbations alter the ASR output to bypass toxicity detection.

A. ASR-Mediated Moderation Pipeline

Let a denote an input audio clip uploaded by a user, and let $\mathsf{ASR}(\cdot)$ be the ASR model deployed by the platform. The ASR model produces a transcript $t = \mathsf{ASR}(a)$ which is then passed to a toxicity classifier $\tau(\cdot)$ that outputs a toxicity score $\tau(t) \in [0, 1]$. Content moderation decisions (e.g., blocking, flagging, or deprioritising content) are made based on this score.

The pipeline can be summarised as:

$$a \xrightarrow{\mathsf{ASR}} t \xrightarrow{\tau} \tau(t) \xrightarrow{\text{policy}} \text{moderation decision.}$$

We assume that all downstream moderation logic operates solely on text and has no access to the raw audio signal beyond the ASR output.

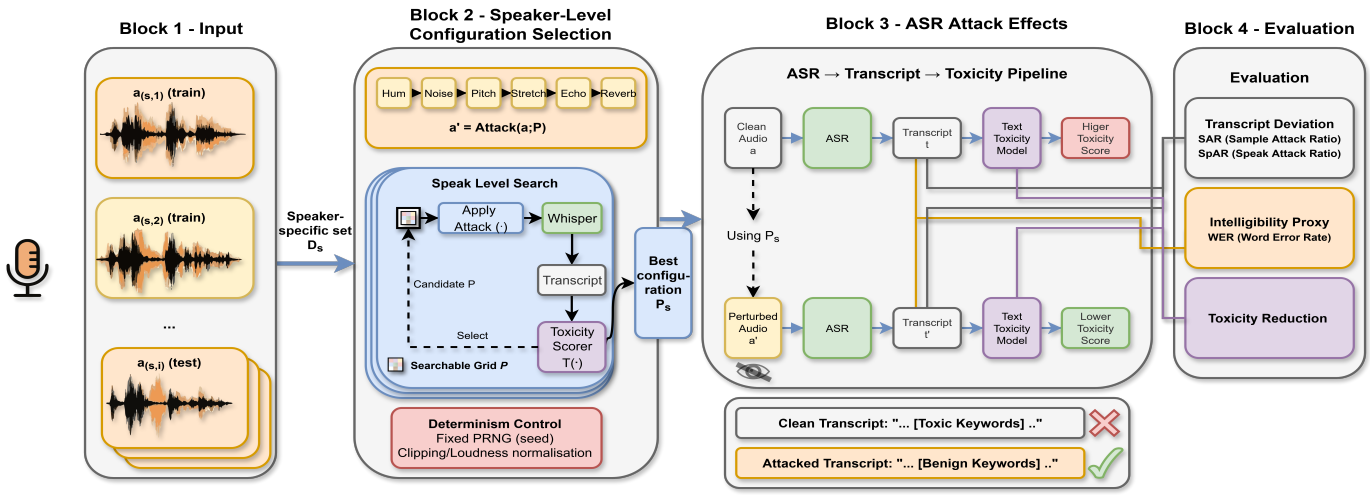

Fig. 1: Overview of the Misspoken Whispers attack on ASR-mediated moderation.

B. Adversary Capabilities and Knowledge

We assume the adversary has the following capabilities:

1) **Control over input audio.** The adversary can arbitrarily process and modify the audio waveform before upload. In particular, they can apply a composition of simple and more complex audio transformations (e.g., hum, noise, pitch shift, time stretch, echo, and reverb) to generate a'.
2) **Access to multiple utterances from the same speaker.** The adversary can obtain a small set of utterances from the target speaker to learn a speaker-specific perturbation configuration. This learned configuration can then be used in testing.
3) **Offline access to a source ASR model during attack construction.** During an *offline* attack-construction phase, the adversary can query a source ASR model (e.g., through open-source weights or via an API) sufficiently many times to evaluate candidate perturbation parameters. This access is used to learn *speaker-specific* perturbation configurations. Once this phase is complete, the adversary no longer requires online access to the ASR model.
4) **Black-box deployment setting.** At deployment time (i.e., when uploading adversarial content to the platform), the adversary treats the platform's ASR system as a black box. They do not know the exact architecture or parameters. Still, they may infer which ASR family is used (e.g., Whisper vs. other models) by experimenting on public content and observing aggregate ban/flag rates.
5) **No control over moderation internals.** The adversary cannot modify or query the internal state of the platform's toxicity classifier or its moderation policies beyond the coarse feedback provided by whether content is accepted, flagged, or removed.

We also assume that the manipulated audio remains understandable for typical users, i.e., perturbations are bounded so that a human listener can still recognise the intended speech content. The adversary may selectively apply the attack only to utterances they deem toxic, leaving non-toxic content unmodified.

C. Attack Surface

The primary attack surface is the ASR model deployed in the moderation pipeline. Since *all* text consumed by the toxicity classifier is produced by $\mathsf{ASR}(\cdot)$, any systematic bias or failure mode in the ASR model can be exploited to alter downstream toxicity scores.

Our attack operates by *poisoning the audio input* rather than manipulating model parameters or network traffic. As a result, the perturbations are *model-agnostic* at deployment time. Once a speaker-specific perturbation profile is learned, it can be reused on future utterances from that speaker. It may persist in its effect even if the platform later switches to a different ASR model (subject to transferability).

D. Success Conditions

We deem an attack on a given audio sample successful if the following conditions are satisfied:

1) **Transcript deviation:** The ASR-generated transcript changes after the attack, i.e.,

$$\mathsf{Norm}(t) \neq \mathsf{Norm}(\mathsf{ASR}(a')),$$

where a' is the perturbed audio, $t = \mathsf{ASR}(a)$, and $\mathsf{Norm}(\cdot)$ denotes a normalisation function (e.g., case-folding, whitespace normalisation) used to abstract away trivial formatting differences. The transcript should remain syntactically plausible.
2) **Toxicity reduction:** The toxicity score assigned to the attacked transcript is significantly lower than that of the ground-truth transcript:

$$\tau(\mathsf{ASR}(a')) \ll \tau(\mathsf{ASR}(a)).$$

In our experiments, we quantify this gap using absolute score differences and threshold-based metrics.

E. Defender Assumptions

We model a defender as follows. A platform that trusts ASR outputs, and uses ASR-generated transcripts as semantically faithful representations of the spoken content and treats them as ground truth for moderation purposes. The moderation is performed solely on the text output $\mathsf{ASR}(a)$ via $\tau(\cdot)$, without cross-checking with alternative ASR models, audio-based detectors, or human review in the general case. The deployed ASR models are not adversarially trained against structured audio perturbations and do not incorporate signal-level integrity verification (e.g., audio fingerprinting, speaker-level consistency checks) as part of the standard pipeline. We also assume that the attack time horizon and the overall moderation architecture remain fixed (or change slowly), so that learned speaker-specific perturbations remain useful.

Under these assumptions, we show that an adversary with access only to the audio input channel can reliably bypass toxicity-based moderation by exploiting vulnerabilities in the ASR layer.

IV. Proposed Attack Method

Having specified the adversarial setting and threat model, we now present our proposed attack method. The attack operates at the speaker level and applies a fixed, compositional sequence of audio perturbations to induce targeted mistranscription while preserving perceptual intelligibility.

A. Perturbation Operations and Configuration

Let $\mathcal{O}$ denote the ordered list of audio operations used by the attack:

$$\mathcal{O} = \langle \mathtt{hum},\ \mathtt{noise},\ \mathtt{pitch},\ \mathtt{stretch},\ \mathtt{echo},\ \mathtt{reverb} \rangle.$$

Each operation $op \in \mathcal{O}$ is associated with a discrete set of strength parameters $\mathcal{V}_{op}$, as specified by the grid in Table I.

A perturbation configuration P specifies one strength value for each operation:

$$P = \{v_{op}\}_{op \in \mathcal{O}}, \qquad \text{with } v_{op} \in \mathcal{V}_{op}.$$

Together, these parameters fully specify a single attack configuration, determining the extent to which each operation is applied to the audio signal.

The whole configuration space induced by the discrete parameter grids is finite and given by the Cartesian product

$$\mathcal{P} = \prod_{op \in \mathcal{O}} \mathcal{V}_{op}, \qquad |\mathcal{P}| = K,$$

where K denotes the total number of possible perturbation configurations.

B. Attack Transformation

Given an input audio signal and a perturbation configuration P, we define the attack as a deterministic transformation that maps clean audio to perturbed audio.

Let $a_{(s,i)}$ denote the i-th utterance of speaker s. The attack applies the perturbation operations specified by P in a fixed order to produce the perturbed audio $a'_{(s,i)}$:

$$a'_{(s,i)} = \mathsf{Attack}(a_{(s,i)}; P) = \mathsf{clip}\Big(\Psi_{\text{hum}} \circ \Psi_{\text{noise}} \circ \Psi_{\text{pitch}} \circ \Psi_{\text{stretch}} \circ \Psi_{\text{echo}} \circ \Psi_{\text{reverb}}(a),\ -1, 1\Big),$$

where each $\Psi_{op}(\cdot; v_{op})$ denotes the operation-specific transformation parameterised by v_{op}, and $\mathsf{clip}(\cdot, -1, 1)$ bounds the signal amplitude.

With the search space $\mathcal{P}$ consisting of all possible perturbation configurations formed by taking the Cartesian product of the discrete parameter grids in Table I. To ensure determinism during training, we fix the pseudorandom number generator (PRNG) seed for all stochastic operations. This design yields a deterministic, compositional attack that operates model-agnostically, making it compatible with black-box ASR systems.

TABLE I: Operation grid for perturbation search

Operation (op)	**Values** (v_{op})
Hum frequency	0, 60, 120 (Hz)
Hum amplitude	0.0, 0.05, 0.1
Noise level	0, 0.02
Echo delay (ms)	0, 25, 50
Reverb time	0, 0.05, 0.1
Pitch	-1, 0, 1
Stretch	0.9, 1.0, 1.1

C. Speaker-Level Attack Objective

Let D denote a dataset, partitioned by speaker, such that each speaker s is associated with a set of audio–transcript pairs:

$$D_s = \{(a_{(s,i)},\ t_{(s,i)}) \mid i = 1, \ldots, |D_s|\},$$

where $a_{(s,i)}$ is the i-th utterance of speaker s and $t_{(s,i)}$ is its corresponding ground-truth (golden) transcript.

The adversary seeks a *speaker-specific* perturbation configuration P_s that consistently reduces the toxicity of the ASR transcripts across that speaker's utterances. Formally, the goal is to minimise the average transcript toxicity:

$$P_s = \arg\min_{P \in \mathcal{P}} \frac{1}{|D_s|} \sum_{i=1}^{|D_s|} \tau\big(t'_{(s,i)}\big), \tag{1}$$

Equivalently, we can express the objective as *maximising toxicity reduction* relative to the clean baseline transcripts:

$$P_s = \arg\max_{P \in \mathcal{P}} \frac{1}{|D_s|} \sum_{i=1}^{|D_s|} \Big[\tau\big(t_{(s,i)}\big) - \tau\big(t'_{(s,i)}\big)\Big]. \tag{2}$$

Both Eq. (1) and Eq. (2) capture the same optimisation goal: find a perturbation configuration that systematically induces mistranscriptions which yield lower toxicity scores.

Our objective is to learn, for each speaker s, a perturbation configuration P_s that consistently reduces the toxicity of ASR-generated transcripts while preserving transcript intelligibility.

The attack is entirely audio-based and requires no target transcript. It operates by composing a fixed sequence of low- and mid-level audio transformations, with parameters selected via a structured search procedure.

D. Speaker-Level Configuration Selection

This exhaustive grid search is computationally and practically feasible in our setting for three reasons. (i) The perturbation strengths are discrete, and the resulting search space is modest[3] and can be evaluated offline during one-time per-speaker calibration. (ii) All transformations are deterministic, and each candidate configuration can be evaluated and generated independently, making the search trivially parallelisable across GPUs/CPUs. (iii) Optimisation is performed at the speaker level rather than at the sample level. This means that once a configuration is selected for a speaker, it can be reused for all of that speaker's future utterances, amortising the offline cost over many samples.

V. Experiments

We now present a comprehensive evaluation of the proposed attack. Experiments quantify its effectiveness on toxic and non-toxic speech, its robustness across speakers, its performance relative to strong baseline attacks, the contribution of individual perturbation families, and its transferability to other ASR systems.

A. Experimental Settings

All experiments used a fixed random seed of 42 and a numerical stabiliser of ε of 1e-12. All results, unless specified, were evaluated on Whisper-base, with other models tested on the evaluation dataset described in Sec. V-F. Training was conducted on an 8-GPU server cluster comprising NVIDIA A100 and A200 GPUs, using mixed-precision (FP16). Testing was performed on a 2-GPU setup with either A100 or A200 GPUs. High GPU utilisation was employed to accelerate testing across all models. Tests were run on a batch size of 128.

1) Datasets: We use the English portion of Mozilla Common Voice v21[4]. Each audio sample $a_{(s,i)}$ is associated with a ground-truth transcript $t_{(s,i)}$. Because Common Voice does not provide toxicity labels, we use Detoxify to assign a toxicity score, $\tau(t)$, to each ground-truth transcript and treat it as a proxy for harmful content. To select the toxicity cutoff, we manually reviewed a validation set (50 toxic / 50 non-toxic transcripts) labelled by a human annotator to determine which threshold of the Detoxify toxicity score best matched human judgement. We tested thresholds from 0.6 to 0.9, with 0.8 given the highest F1 score when computed using Detoxify. Using the found threshold, we split the dataset into **Toxic** and **Non-Toxic**:

$$\text{Toxic} = \{(s,i) \mid \tau(t_{(s,i)}) \geq 0.8\}, \quad \text{Non-Toxic} = \mathcal{D} \setminus \text{Toxic}.$$

Then, within Toxic (or Non-Toxic), we take all the samples for each speaker and split them into 80% for training and 20% for evaluation. We also capped the number of samples per non-toxic speaker at 20 to prevent the dataset from being dominated by non-toxic speech. Overall, 127 unique speakers were identified across both the Toxic and Non-Toxic datasets.

2) Evaluation Metrics and Normalisation: During evaluation, the learned speaker-specific perturbation P_s is applied to unseen test samples for the same speaker. For each sample, we compare:

$$\mathsf{Norm}(t_{(s,i)}) \quad \text{vs.} \quad \mathsf{Norm}(t'_{(s,i)}),$$

where $t'_{(s,i)} = \mathsf{ASR}(a'_{(s,i)}))$, and the deterministic text normalisation function $\mathsf{Norm}(\cdot)$ lowercases all text, capitalises the first word, and collapses whitespace. This is done to provide consistently structured input to the toxicity classifier and ensure comparable scoring across models. We then construct the binary mask:

$$m_{(s,i)} = \mathbf{1}\{\mathsf{Norm}(t_{(s,i)}) \neq \mathsf{Norm}(t'_{(s,i)})\},$$

which indicates whether the transcript has changed due to the perturbation.

We evaluate the proposed attack using the following metrics:

- **SAR** (Sample Attack Ratio): the proportion of samples for which $m_{(s,i)} = 1$.
- **SpAR** (Speaker Attack Ratio): the proportion of speakers with at least one attacked sample.
- **WER** (Word Error Rate): the standard WER between $t_{(s,i)}$ and $t'_{(s,i)}$, measuring transcription distortion.

B. Attack on Toxic Audio

We first evaluate the effectiveness of the learned speaker-specific perturbations on the Toxic subset. For each speaker s, the optimal configuration P_s (computed during training) is applied to all unseen toxic samples from that speaker. We assess both transcript alteration (SAR, SpAR) and toxicity reduction.

Table II reports the speaker-level and sample-level effectiveness of the attack. On the training split, the perturbation successfully altered transcripts for every speaker in the dataset, and it modified 93.02% of all toxic samples. When applied to the held-out test split, the attack continued to generalise well: it succeeded for 93.70% of speakers and changed the transcripts of 92.24% of their toxic utterances. This demonstrates that once a perturbation profile is learned for a speaker, it remains effective across new toxic samples from the same speaker.

Beyond transcript alteration, we examine how the perturbations affect toxicity. The second half of Table II breaks down the direction of toxicity change on the test set. A large majority of attacked samples—93.47%—exhibited decreased toxicity. A small portion (4.08%) increased, and 2.45% remained unchanged. These infrequent increases typically occur when a substitution inadvertently introduces a word that Detoxify considers slightly more harmful, reflecting known sensitivities in toxicity classifiers.

The magnitude of toxicity reduction is substantial. As shown in Table III, the average reduction across the test set is -0.503, which corresponds to a reduction of 0.503 points. Nearly

[3]Exact grid size: 1,458 parameter combinations (Cartesian product of the discrete choices in Table I).

[4]https://datacollective.mozillafoundation.org/datasets?q=common+voice

TABLE II: Distribution of toxicity changes and speaker and sample-level attack success on **Toxic**.

Dataset	Toxicity change distribution (%)			Attack success (%)	
	↓	↑	No Change	SpAR	SAR
Training	100.00	0.00	0.00	100.00	93.02
Testing	93.47	4.08	2.45	93.70	92.24

half of the test samples (43.67%) show reductions of at least 0.5, and the majority demonstrate meaningful reductions, with 86.94% decreasing by at least 0.1. These results confirm that the perturbations consistently suppress toxic lexical content in the generated transcripts.

TABLE III: Toxicity reduction performance after perturbation on **Toxic**.

Dataset	Toxicity Change	$\geq$ 0.5	$\geq$ 0.4	$\geq$ 0.3	$\geq$ 0.2	$\geq$ 0.1
Training	-0.587	71.65	95.44	98.86	100.00	100.00
Testing	-0.503	43.67	66.12	77.14	82.45	86.94

Note: negative mean change indicates reduction.

C. Attack on Non-Toxic Audio

We next examine the behaviour of the speaker-specific perturbations on the Non-Toxic subset. This analysis is essential for assessing whether the attack unintentionally introduces harmful content or significantly alters benign speech.

Table IV summarises the direction of toxicity change for all attacked non-toxic samples. On the test split, 40.94% of samples show a decrease in toxicity, while 59.06% show an increase, and none remain unchanged. A similar pattern appears in the training split, though with a smaller proportion of toxicity reductions (11.02%) and a larger proportion of toxicity increases (88.98%). These increases, however, must be interpreted in terms of magnitude rather than frequency.

TABLE IV: Distribution of toxicity increases (↑), decreases (↓), and unchanged cases on **Non-Toxic**.

Dataset	Toxicity change distribution (%)			Mean Toxicity Change
	↓	↑	No Change	
Training	11.02	88.98	0.00	0.009
Testing	40.94	59.06	0.00	0.007

The second part of Table IV provides the corresponding average toxicity change. Although most samples exhibit an increase in toxicity, the magnitude of this increase is minimal, with the mean change remaining near zero (0.007 on test), consistent with Detoxify's sensitivity to minor lexical shifts rather than the introduction of harmful content. Because the original non-toxic transcripts have very low baseline toxicity, even slight substitutions—such as changes in word morphology or the introduction of rare but weakly correlated terms—can shift Detoxify scores upward without introducing genuinely harmful content.

These results indicate that the perturbation does not meaningfully elevate toxicity in benign speech. Unlike the Toxic subset, which systematically removes or alters toxic keywords, the Non-Toxic subset contains few lexical items strongly associated with toxicity. As a result, any small increases in toxicity largely reflect the sensitivity of text-based toxicity classifiers to incidental lexical changes rather than the introduction of new harmful content. Importantly, we observe no cases in which the perturbation generates strongly toxic transcripts from initially benign samples.

D. Comparison with Baseline Attacks

To contextualise the effectiveness of our method, we compare our approach with several standard adversarial attacks: CW (500-step and early-stop), FGSM, MI-FGSM, and PGD. All models were run with the same seed, 42. CW was run with a learning rate of 0.0001 in two ways: early stop, which returns the audio the moment the attack worked, and full, which runs for 500 steps. FGSM, MI-FGSM, and PGD use a $\varepsilon = 0.01$, with MI-FGSM and PGD running for 10 iterations each. Table V show results for the Toxic and Non-Toxic subsets.

TABLE V: Baseline adversarial methods vs. our attack on **Toxic** and **Non-Toxic**.

	Toxic		Non-Toxic	
Method	$\bar{\tau}(Pk) \downarrow$	**WER**	$\bar{\tau}(Pk) \downarrow$	**WER**
CW [2]	0.2920	1.8336	0.0152	0.6960
CW-First [2]	0.7396	0.2760	0.0089	0.2522
MI-FGSM [4]	0.5732	0.4551	0.0122	0.3539
FGSM [3]	0.6313	0.3864	0.0130	0.3078
PGD [5]	0.5794	0.6817	0.0105	0.3473
Ours	**0.3821**	**0.5914**	**0.0154**	**0.5495**

Note: $\bar{\tau}(\cdot)$ is the average toxicity for a given perturbation.

On the **Toxic** subset, CW-500 achieves the lowest toxicity (0.2920) but does so by heavily degrading transcription quality (WER 1.8336), producing outputs that largely fail to preserve the original meaning. The early-stop variant reduces distortion but provides weaker toxicity suppression (0.7396). MI-FGSM and PGD offer moderate reductions (0.5732 and 0.5794, respectively), but they still introduce substantial noise into decoding. Our method achieves a toxicity of 0.3821, which is second only to CW-500, while maintaining a significantly lower WER of 0.5914. Compared to optimization-based CW, our attack sacrifices some toxicity reduction in exchange for substantially lower transcript distortion, consistent with our goal of selective lexical suppression rather than transcript collapse.

On the **Non-Toxic** subset, all methods, including ours, produce only minor changes in toxicity, with average scores remaining close to zero. This behaviour aligns with the previous two subsections and reflects the sensitivity of toxicity classifiers to small lexical shifts rather than the generation of genuinely harmful content.

Overall, while optimisation-based attacks can sometimes achieve stronger toxicity reduction, they often do so by destroying transcript quality. We found that CW-First is low-distortion but weak toxicity suppression, while CW is strong but impractical as it has extremely high WER. Our approach

provides a more practical and targeted degradation of toxic content while preserving coherent ASR output.

E. Ablation Study

To understand the contribution of each perturbation family, we evaluate the six operation groups (`hum`, `noise`, `pitch`, `stretch`, `echo`, and `reverb`) individually and compare them to the full composite perturbation. Table VI summarise the results for the Toxic and Non-Toxic subsets.

TABLE VI: Ablation results showing each perturbation family's contribution on **Toxic** and **Non-Toxic**.

	Toxic			**Non-Toxic**		
Family	$\bar{\tau}(Pk)\downarrow$	**SAR↑**	**WER**	$\bar{\tau}(Pk)\downarrow$	**SAR↑**	**WER**
`hum`	0.7521	69.80%	0.2491	0.0096	65.21%	0.2461
`noise`	0.7474	69.80%	0.2543	0.0115	65.55%	0.6816
`pitch`	0.4325	89.80%	0.9552	0.0186	89.47%	0.6093
`stretch`	0.3933	92.65%	0.6669	0.0159	90.16%	1.0625
`echo`	0.7255	69.39%	0.2690	0.0089	68.50%	0.2642
`reverb`	0.7174	70.00%	0.3674	0.0109	65.35%	0.2475
Ours	**0.3821**	**92.24%**	**0.5914**	**0.0154**	**91.14%**	**0.5495**

On the **Toxic** subset, `pitch` and `stretch` are the two most influential families. Both achieve high attack success rates and noticeably reduce toxicity, although they also introduce higher WER than other single-operation perturbations. `Hum`, `noise`, `echo`, and `reverb` each have a measurable but weaker effect: they modify transcripts less frequently and produce smaller toxicity changes, suggesting that Whisper is comparatively robust to these isolated acoustic distortions.

Our method, which combines all six families, consistently outperforms the individual operations. It achieves the lowest toxicity on the Toxic subset while maintaining a WER that is significantly lower than pitch or stretch alone. This indicates that the full attack benefits from interactions between different types of audio transformations: the phonetic disruptions caused by pitch and stretch are reinforced by the spectral and temporal variations introduced by the remaining operations, producing more targeted lexical drift without overwhelming the ASR system.

A similar pattern appears on the **Non-Toxic** subset. `Pitch` and `stretch` again have the strongest individual impact, whereas `hum`, `noise`, `echo`, and `reverb` produce only small changes. The composite perturbation maintains low toxicity overall, comparable to the individual families, and does not introduce harmful content. This reaffirms that the attack does not generate toxic output when applied to benign speech.

F. Transferability

We evaluate how well the speaker-specific perturbations learned on Whisper-base transfer to other ASR models. The same perturbation configuration, P_s, is applied to eight additional models across both Whisper-derived and non-Whisper architectures. Table VII reports toxicity, SAR, and WER for the Toxic subset.

On the **Toxic** subset, the perturbations transfer strongly in terms of transcript alteration. Most models show very high SAR values—often exceeding 99%—including all S2T variants and wav2vec2-base. This indicates that perturbations alter acoustic patterns in a model-agnostic manner, affecting token-level decisions across a wide range of architectures. The magnitude of toxicity reduction, however, varies across models. Some systems, particularly Whisper-small and Distil-large-v2, show relatively modest decreases, suggesting that the interaction between perturbed phonetic cues and the model's decoding strategy differs across architectures. Even so, most models exhibit non-trivial toxicity suppression, demonstrating that the attack generalises beyond the Whisper-base model.

TABLE VII: Transferability of perturbations across diverse ASR models on **Toxic**.

Model	$\bar{\tau}(Pk_s)\downarrow$	**SAR ↑**	**WER**
Distil-large-v2	0.7990	63.27%	0.1971
S2t-large-librispeech-asr	0.5917	99.59%	0.4925
S2t-medium-librispeech-asr	0.5916	99.59%	0.4768
S2t-small-librispeech-asr	0.5963	99.59%	0.4853
Wav2vec2-base-960h	0.5647	99.59%	0.5387
Wav2vec2-large-960h-lv60	0.6924	99.59%	0.3951
Whisper-small	0.7923	63.67%	0.2112
Whisper-tiny	0.6482	76.73%	0.3399
Whisper-base (source)	**0.3821**	**92.24%**	**0.5914**

Overall, the transferability evaluation shows that the perturbations reliably induce transcript changes across diverse ASR systems, even when those systems differ substantially from the Whisper model used during training. While the exact toxicity reduction is model-dependent, the consistently high SAR values highlight a structural weakness shared across modern ASR architectures: small, structured distortions are sufficient to push the decoding process away from toxic keyphrases, even without tailoring the perturbation to each specific model.

VI. Defences

Robust ASR Training. One natural direction is to strengthen the ASR model itself. Adversarial training—using perturbed audio samples during training—may improve robustness to the specific acoustic distortions examined in this work. However, adversarial training for audio is more challenging than for text or images. The perturbations we study are not additive noise but structured compositions of pitch, temporal stretch, spectral modifications, and reverberation. Training against all combinations would significantly increase computational cost and may still fail to generalise, given the large space of possible distortions. Moreover, as our ablation results show, the attack's effectiveness stems from interactions across multiple transformation families, which may require larger architectural changes rather than simply expanding training data.

Multi-ASR and Cross-Model Consistency Checking. Platforms may improve robustness by using multiple ASR systems and detecting inconsistencies across their transcripts. As our attack is based on a single-source model, running two or more ASR models in parallel may reveal discrepancies that flag potentially manipulated audio. However, as demonstrated in Section V-F, perturbations transfer strongly across models, with many ASR systems exhibiting near-universal transcript

alterations. Although toxicity reduction varies by architecture, transcript inconsistency alone may not provide a reliable signal. More sophisticated approaches may be required, such as comparing confidence distributions or identifying unusual phoneme-level decoding behaviour.

VII. Conclusion

In this paper, we propose a method for adversarial attacks on ASR models and demonstrate its effectiveness against baselines and transferability. It has been shown that ASR models can be attacked to produce mistranscriptions of toxic audio samples while causing minimal change in non-toxic samples. Our speaker-specific permutation approach achieved a 92.24% sample attack success rate and significantly reduced toxicity scores for toxic inputs (by approximately 0.5), with minimal impact on benign speech. We also demonstrated transferability across ASR models, highlighting current limitations in moderation pipelines. We also covered a range of possible defences and future approaches to protect against such speaker-specific attacks, focusing on speaker voice profiles to make them harder to detect, and on how new moderation pipelines could be designed to increase the resilience of current ones.

VIII. Ethical Considerations

This work investigates adversarial attacks on ASR models to highlight the issues they pose for content moderation pipelines. While the techniques presented here could be misused to bypass the toxicity filter, we intended only to highlight the weaknesses in current models. Below are potential harms our work may pose to the non-academic community.

Our attack demonstrates its ability to misclassify toxic content. This creates a clear pathway for malicious threat actors to circumvent current ASR-based moderation systems. Our findings indicate that acoustically induced transcript drift can further widen these gaps, enabling harmful content to pass undetected.

A side effect of this ability to evade detection is the fact that it can allow toxic speech to attack marginalised groups. Studies have found that perceptions of toxicity vary across demographic groups and that classifiers can underdetect content targeting underrepresented groups [17].

We acknowledge the dual-use nature of adversarial research, but believe that these findings outweigh any potential harm and that specific defence techniques can reduce their effectiveness.

IX. AI Acknowledgements

We acknowledge the use of AI in this paper. We used ChatGPT/Copilot to assist in creating table titles and captions. It was also used to improve the clarity of paragraphs in this paper, with human reviewers assessing the content. It was also used to develop Python code and to improve code structure and optimisation, using a human-written codebase as a template. An AI-assisted grammar tool was also used.

X. Acknowledgements

This research was supported by the Commonwealth through an Australian Government Research Training Program Scholarship https://doi.org/10.82133/C42F-K220.

This work was supported by the Australian Research Council (ARC) under the DECRA Fellowship DE240101049.

References

[1] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," 2022.

[2] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," 2018.

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations (ICLR)*, 2015.

[4] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9185–9193.

[5] A. Mądry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *stat*, vol. 1050, p. 9, 2017.

[6] V. Raina, R. Ma, C. McGhee, K. Knill, and M. Gales, "Muting whisper: A universal acoustic adversarial attack on speech foundation models," *arXiv preprint arXiv:2405.06134*, 2024.

[7] W. Jin, Y. Cao, J. Su, D. Wang, Y. Zhang, M. Xue, J. Hao, J. S. Dong, and Y. Yang, "Whispering under the eaves: Protecting user privacy against commercial and llm-powered automatic speech recognition systems," *arXiv preprint arXiv:2504.00858*, 2025.

[8] H. Wu, C. Liu, J. Chen, R. Du, K. He, Y. Zhang, C. Wu, T. Zhang, Q. Guo, and J. Zhang, "When translators refuse to translate: A novel attack to speech translation systems," in *34th USENIX Security Symposium (USENIX Security 25)*, 2025, pp. 4723–4740.

[9] L. Hanu and Unitary team, "Detoxify," Github. https://github.com/unitaryai/detoxify, 2020.

[10] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou, "Hidden voice commands," in *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, Aug. 2016, pp. 513–530.

[11] R. Olivier and B. Raj, "Recent improvements of asr models in the face of adversarial attacks," in *Interspeech 2022*, 2022.

[12] R. Peri, S. M. Jayanthi, S. Ronanki, A. Bhatia, K. Mundnich, S. Dingliwal, N. Das, Z. Hou, G. Huybrechts, S. Vishnubhotla, D. Garcia-Romero, S. Srinivasan, K. J. Han, and K. Kirchhoff, "Speechguard: Exploring the adversarial robustness of multimodal large language models," 2024.

[13] P. Żelasko, S. Joshi, Y. Shao, J. Villalba, J. Trmal, N. Dehak, and S. Khudanpur, "Adversarial attacks and defenses for speech recognition systems," *arXiv preprint arXiv:2103.17122*, 2021.

[14] N. L. Kühne, A. H. F. Kitchena, M. S. Jensen, M. S. L. Brøndt, M. Gonzalez, C. Biscio, and Z.-H. Tan, "Detecting and defending against adversarial attacks on automatic speech recognition via diffusion models," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.

[15] P. Kawa, M. Plata, and P. Syga, "Defense against adversarial attacks on audio deepfake detection," in *Interspeech 2023*, 2023, pp. 5276–5280.

[16] D. Klug, E. Steen, and K. Yurechko, "How algorithm awareness impacts algospeak use on tiktok," in *Companion Proceedings of the ACM Web Conference 2023*, 2023, pp. 234–237.

[17] D. Kumar, P. G. Kelley, S. Consolvo, J. Mason, E. Bursztein, Z. Durumeric, K. Thomas, and M. Bailey, "Designing toxic content classification for a diversity of perspectives," in *Proceedings of the 17th Symposium on Usable Privacy and Security (SOUPS 2021)*. USENIX Association, 2021.