# ALIF: Low-Cost Adversarial Audio Attacks on Black-Box Speech Platforms using Linguistic Features

*Peng Cheng*[*,#], *Yuwei Wang*[*,#], *Peng Huang*[*,#], *Zhongjie Ba*[*,#,‡], *Xiaodong Lin*[†], *Feng Lin*[*,#], *Li Lu*[*,#], *Kui Ren*[*,#]
[*]*Zhejiang University, Hangzhou, China*
[#]*ZJU-Hangzhou Global Scientific and Technological Innovation Center, Hangzhou, China*
[†]*University of Guelph, Guelph, Canada*
{*peng_cheng, yuwei.wang, penghuang, zhongjieba, flin, li.lu, kuiren*}*@zju.edu.cn, xlin08@uoguelph.ca*

2024 IEEE Symposium on Security and Privacy (SP) | 979-8-3503-3130-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/SP54263.2024.00056

***Abstract*—Extensive research has revealed that adversarial examples (AE) pose a significant threat to voice-controllable smart devices. Recent studies have proposed black-box adversarial attacks that require only the final transcription from an automatic speech recognition (ASR) system. However, these attacks typically involve many queries to the ASR, resulting in substantial costs. Moreover, AE-based adversarial audio samples are susceptible to ASR updates. In this paper, we identify the root cause of these limitations, namely the inability to construct AE attack samples directly around the decision boundary of deep learning (DL) models. Building on this observation, we propose ALIF, the first black-box adversarial linguistic feature-based attack pipeline. We leverage the reciprocal process of text-to-speech (TTS) and ASR models to generate perturbations in the linguistic embedding space where the decision boundary resides. Based on the ALIF pipeline, we present the ALIF-OTL and ALIF-OTA schemes for launching attacks in both the digital domain and the physical playback environment on four commercial ASRs and voice assistants. Extensive evaluations demonstrate that ALIF-OTL and -OTA significantly improve query efficiency by 97.7% and 73.3%, respectively, while achieving competitive performance compared to existing methods. Notably, ALIF-OTL can generate an attack sample with only one query. Furthermore, our test-of-time experiment validates the robustness of our approach against ASR updates.**

## 1. Introduction

Smart devices pervasively integrate voice control functionality. Users are getting used to interacting with smart devices with their voice to enjoy hands-free convenience. As a result, smart devices, such as smartphones, smart speakers, and automobiles, have adopted the voice assistant (VA) function to turn themselves into voice-controllable devices. In point of fact, over 132 million people used VAs in the US in 2021, and the number continues to grow [1]. More than 4.25 billion VAs have been installed globally, projected to reach 8.4 billion by 2024 [2].

‡ *Corresponding Author: Zhongjie Ba*

The prevalence of voice-controllable devices brings security risks. Smart appliances take voice commands as input for performing heterogeneous actions, including security and safety-critical tasks, such as thermal adjustment, online payment, and even autonomous driving [3], [4]. Attackers can exploit the voice interaction to inject malicious speech commands without raising users' suspicions and cause severe security and safety consequences, including economic losses, privacy violations, health issues (e.g., thermometer overheating), bodily harm (e.g., car accidents), etc.

Injecting malicious commands covertly into voice-controllable devices has been realized with adversarial audio techniques [5], [6], [7], [8], [9]. The requirements of an attack audio are twofold: to maintain the covertness of the attack audio, it cannot sound like the intended command; to guarantee the attack's success, it should be correctly recognized as the intended command by a speech recognition model. The adversarial example (AE), which adds minute perturbation to original audios, naturally fits the attack requirements as the method does not affect users significantly and can affect the target model.

The problem of generating AEs to attack commercial voice-controllable devices is formulated as a black-box adversarial attack problem. The speech recognition models applied by commercial products are unknown to the attacker. To generate qualifying AEs, the adversaries usually query the black-box model thousands of times with trial audio examples, get the query results (i.e., transcription), and iteratively optimize the attack examples leveraging the transcriptions as optimization guide [10], [9], [11].

The primary goal of black-box studies is to reduce attack cost, but the state-of-the-art solutions are still lacking in their suitability for practical applications. Considering the generation of each attack sample requires abundant queries and each one has a tangible monetary cost[1], existing works seek to improve the query efficiency and therefore reduce the attack price. Improving the query efficiency can also benefit the covertness of the attack. High-frequency querying with similar audio content may trigger the network defense

1. 300000 queries to the Google Cloud Speech-to-Text service (STT) result in a cost of about 1200 USD [11].



DOI 10.1109/SP54263.2024.00056

system such as intrusion detection [11], [12], [13]. Recent works achieved 1500 queries on a commercial speech recognition model to generate one attack sample [8], [11], and the current level remains considerably distant from the efficient and practical standard, indicating a critical need for further advancement.

In addition to the low efficiency of current state-of-the-art techniques, we have observed that existing attacks based on AEs are susceptible to losing their efficacy when encountering model updates, resulting in highly exacerbated attack costs. Specifically, we found that AEs are sensitive to changes in the deep learning model, such as the automatic speech recognition model (ASR), and fine-tuning the model with new data may cause the previously trained AEs to lose their effectiveness, as demonstrated in Figure 1. This sensitivity represents a common shortcoming of AE-based attacks. For instance, Devil's Whisper [8] discovered that previously workable AE samples no longer worked with later versions of the model. As a result, manufacturers' unpredictable model optimization behavior will make AE-based attacks easily outdated, increasing the chances of attack failure. In case of failure, regenerating adversarial audio from scratch incurs substantial time and financial overhead, doubling or more the cost of the attack.

Reducing the cost of black-box adversarial attacks for command injection encounters significant challenges. Firstly, the limited knowledge of the ASR system. Existing works utilize advanced optimization methods, such as the Revolution Algorithm and Gradient Estimation Algorithm, to reduce query numbers. However, the lack of an explainable guideline and reliance on trial and error remain issues. Approximately 1500 queries are still required to find a satisfactory attack sample in a black-box setting. Secondly, the susceptibility of AEs to model updates lacks sufficient research and solutions. We identify that model fine-tuning changes decision boundaries, rendering previously trained adversarial examples ineffective.

In this paper, we propose a novel low-cost attack on black-box ASR using adversarial linguistic features (ALIF)[2]. The core idea is to generate adversarial audio samples from the decision boundary space of an ASR. Our approach is inspired by two observations: (i) current AE-based methods for adding perturbations result in query inefficiency and susceptibility to model updates. Existing adversarial attacks on ASRs involve adding perturbations to raw inputs (audio waveforms) to search for AEs around the ASR decision boundary. However, mapping raw inputs to lower-dimensional embeddings in the ASR model reduces perturbation efficiency. It fails to guarantee sufficient distance between the embedding and the decision boundary, resulting in susceptibility to model changes. (ii) This limitation can be addressed by constructing adversarial audio from the representation space of the decision boundary. We propose a novel approach that generates adversarial audio samples from a space similar to the one where the ASR decision boundary lies. ASR and text-to-speech (TTS) synthesis are two reciprocal processes. Specifically, the low-dimensional linguistic embeddings extracted from the text analysis module of TTS capture the semantics of the raw input (i.e., text) and largely overlap with the space of the ASR decision boundary. Perturbations on this TTS linguistic feature space can affect ASR recognition. Based on this insight, we generate adversarial perturbations on the linguistic features of a TTS model to construct the adversarial attack audio samples. Compared to existing attacks, our proposed ALIF pipeline significantly improves the query efficiency by approximately an order of magnitude while demonstrating resilience to model updates, resulting in significantly reduced attack costs.

2. The demos are presented in https://taser2023.github.io/. The source code is available in https://github.com/TASER2023/TASER.

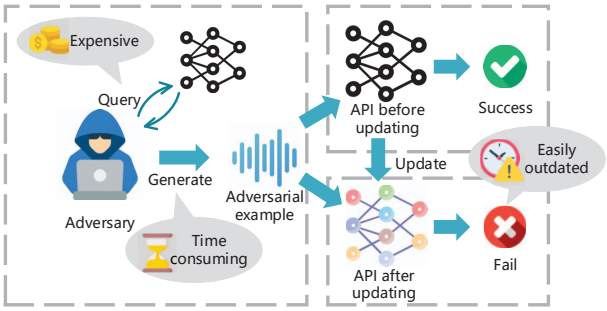


Figure 1. Limitations of existing adversarial attacks based on adversarial examples (AEs). The traditional pipeline for generating black-box audio adversarial examples necessitates the attacker making many API queries, which is both costly and time-consuming. The attacker can then obtain an example capable of successfully attacking the API. However, the example's effectiveness is significantly diminished by model updates.

ALIF can generate audio that is incomprehensible to humans yet interpretable by ASR platforms. The capability opens up various attack scenarios, including hidden command injection in the digital domain, such as deceiving online subtitling services, and physical attacks in the real world to control VAs for unexpected command execution. Specifically, we introduce two attack schemes based on ALIF: ALIF over-the-line (ALIF-OTL) for the adversarial attack in the digital domain and ALIF over-the-air (ALIF-OTA) for the attack in the physical environment.

**Contributions.** Our contributions are as follows.

- To the best of our knowledge, this paper is the first to study the generation of black-box adversarial audio from the linguistic feature space of TTS.
- We propose ALIF, a black-box attack pipeline based on adversarial linguistic features against commercial ASR platforms. Depending on the pipeline, ALIF-OTL and ALIF-OTA are proposed for the digital and over-the-air attack scenarios. A new platform attack scenario that fools the online subtitling service is also proposed.
- We conduct comprehensive experiments against commercial ASRs and well-known VA products to validate the effectiveness and practicality of our adversarial attacks. ALIF-OTL can generate adversarial audio with an average

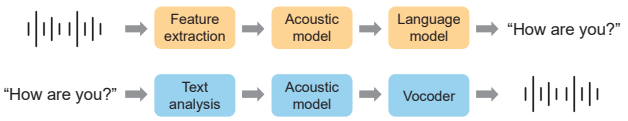


Figure 2. Architectures of an ASR and a TTS system.

of 35 queries for the attack in the digital domain (only one query is also feasible), achieving 97.7% query efficiency improvement over existing works and at least an average success rate of 95.8% on 4 ASR APIs. ALIF-OTA achieves 73.3% query efficiency improvement on state-of-the-art works and obtains an 81.3% success rate on the APIs. Experimental results show our adversarial audio samples are robust against attack environments, attack distance, and model updates. Evaluations of various impact factors verify the practicality of our ALIF attacks.

## 2. Background

This section introduces speech recognition/synthesis and the conventional black-box AE attack method.

### 2.1. Automatic Speech Recognition

An ASR system takes human speech signals as the input and produces the semantic meaning of the spoken content in the form of text. We denote the input waveform as $x$, the ASR system as $f(\cdot)$, and the output transcription as $y$. The function of an ASR is formulated as $y = f(x)$. The upper part of Figure 2 shows a typical ASR system's architecture consisting of feature extraction, acoustic model, and language model. Classical ASR applies a Gaussian Mixture Model-Hidden Markov Model (GMM-HMM) to build the acoustic model. With the rapid development of deep learning, modern ASRs universally apply neural networks as their core and achieve outstanding recognition performance. Depending on the model architecture and neural network variants (e.g., CNN and RNN) used, a wealth of ASR categories exist. Introducing different ASR models in detail is out of the scope of this paper, and we present the representative components instead. The feature extraction component extracts acoustic features essential to the semantic information; then, the acoustic model predicts the most probable linguistic units based on the features. Finally, the language model generates the text sequence with the highest probability.

### 2.2. Speech Synthesis

Speech synthesis, also known as Text-To-Speech (TTS), aims to synthesize natural and intelligible speech signals given an arbitrary sample of text as input [14]. The task can be formulated as $x = f(y)$. The lower part of Figure 2 shows the architecture of a typical TTS model consisting of three components: text analysis, acoustic model, and the vocoder. At the text analysis stage, the system encodes the text to the linguistic feature embedding space. Then the embedding is fed to the acoustic model to obtain the acoustic features (usually a Mel spectrogram). Finally, the vocoder converts the spectrogram to an audio waveform.

### 2.3. Black-Box Adversarial Audio Attacks

**Basic Adversarial Examples.** The goal of an AE-based attack is to deceive the ASR system into transcribing input waveform $x$ incorrectly by adding perturbations $\delta$, resulting in a transcription that does not match the correct transcription $y$. In a non-targeted attack, the requirement is that $SR(x+\delta) \neq y$, where SR denotes the recognition function of the ASR. In contrast, in a targeted attack, the condition is that $SR(x+\delta) = T$, where $T$ represents the malicious command the adversary intends to activate. Since AEs typically require the perturbations to be imperceptible, the amplitude of $\delta$ is usually constrained. The formulation of AE-based attacks can be expressed as:

$$\arg\min_{\delta} L(SR(x+\delta), T) + \alpha \cdot dB_x\delta \tag{1}$$

where $L$ denotes the loss function indicating the similarity between the attack audio transcription and the target text. $\alpha$ is the weight to trade off being covert and adversarial. To solve the problem, a loss value from $L$ is calculated, and gradient descent is applied to find the optimal $\delta$. The gradient calculation requires information about the ASR architecture and parameters, which is impractical in the real world, where the specifics of commercial ASR are unknown.
**Black-Box Adversarial Examples.** In a black-box setting, internal knowledge, such as the weights of the target ASR, is unknown. The attacker can only obtain the final transcription, therefore, cannot calculate $L$, nor can they apply gradient descent to solve for the optimal $\delta$. To ensure the intended command can be activated, the AE training process usually starts from the target command audio sample and gradually modifies the sample towards another unsuspicious signal $x$, aiming to generate an attack audio sample that sounds benign but is recognized as the target command. The problem in Formula 1 is reformed as

$$\arg\min_{\delta} L = \begin{cases} \|\delta\|_p, & SR(x+\delta) = T \\ +\infty, & otherwise \end{cases} \tag{2}$$

where $\|\delta\|_p$ is a norm function to limit the perturbation, and $L$ is the objective function. Given the opaque ASR model, the optimization is based on heuristic techniques, which make no assumption about the model, search a large optimization space, and develop iteratively with experience learned from the previous round. The experience is learned from the query results. Such an iterative process normally requires extensive queries. Related studies have utilized evolution algorithms [15], [9] and gradient estimation methods [11] as the training methodology.

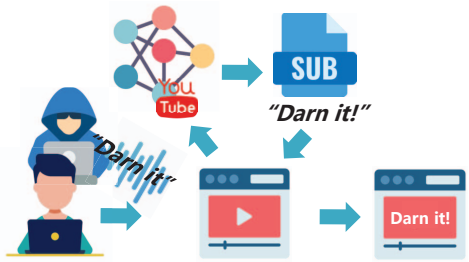


Figure 3. System model of ALIF-OTL. This depicts an attacker incorporating an adversarial audio track into a video. As a result, when the manipulated video is uploaded to the platform, the automatic subtitling service inadvertently generates inappropriate text. The ALIF-OTL attack occurs within the digital domain.

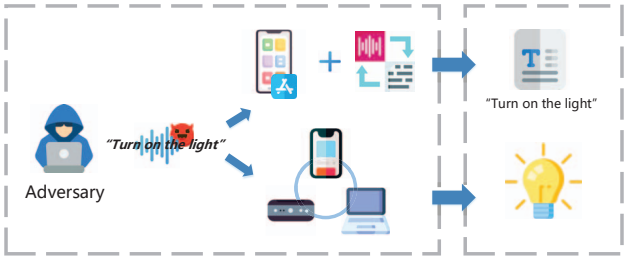


Figure 4. System model of ALIF-OTA. This model illustrates two potential attack scenarios. In the first scenario (upper part), the smartphone apps with voice interaction capabilities record the attack audio signals in the environment and then call online APIs to transcribe audio. In the second scenario (lower part), an attacker plays attack audio samples to activate voice assistants - such as smart speakers - that possess their own ASR backend, thereby executing the commands.

## 3. System and Threat Models

We demonstrate our attack schemes for both the digital domain and the physical environment, namely over-the-line (OTL) and over-the-air (OTA) scenarios. Leveraging the ALIF pipeline, we employ different training methods to generate adversarial audio samples against commercial ASRs in each scenario. The method for generating the adversarial attack in the OTL scenario is called ALIF-OTL. Notably, we propose a new type of digital attack called the *platform attack* in the OTL scenario, where the online subtitle transcribing service is misled. The method for generating the adversarial attack in the OTA scenario is called ALIF-OTA. This section presents the system and threat models of ALIF-OTL and ALIF-OTA.

### 3.1. System and Threat Model of ALIF-OTL

**System Model.** The system model of ALIF-OTL is presented in Figure 3. Online video and multimedia content platforms (e.g., YouTube) have started providing a subtitling service, which supports an automatic transcribing function for on-demand and live media content. These platforms apply an ASR to recognize the semantics of the audio track and generate subtitles. Audiences can opt-in to the service through a corresponding option in the playback agent, which allows subtitles to be shown as the video plays. As Amazon introduces [16], subtitling content helps improve accessibility and engagement. It can also be a compliance requirement for video programming distributors to support hard-of-hearing users. In addition to utilizing the video platform's own service, content creators can use third-party services (e.g., Amazon Transcribe) to add subtitles for their media content.

**Attacker's Goal.** The attacker desires to launch an adversarial audio *platform attack*. The goal is to damage the reputation of media content providers or cause trauma to audiences of particular communities. To achieve this goal, the attacker delivers the adversarial audio as the audio track of a video. The attacker can deceive the ASR of either a third-party subtitling service or the native feature of a video content platform like YouTube, leading to the generation of biased transcription text as the video subtitle. Users will see disinformation created intentionally by the attacker when the subtitle is shown. Since users only hear stuttering sounds and cannot understand the content, they would intuitively think it is caused by *network jitter/packet loss*. However, these inappropriate transcriptions can inflict trauma on viewers and cause them to have negative experiences with the platform, possibly damaging the platform's reputation.

Considering the expense and time overhead of querying commercial online APIs, the attacker would like to keep the cost as little as possible. Meanwhile, he/she would like to ensure the performance stability of the generated attack audio samples, namely reducing the risk that the audio becomes ineffective due to unpredictable ASR model updates by the service provider. A typical failure occurs when: the attacker obtains adversarial audio signals after many queries and time-consuming training but finds them ineffective during the attack.

**Attacker's Capability** The attacker is restricted from launching the attack in the black-box setting. ASRs typically do not provide the confidence score for transcriptions because it does not benefit the user's experience [9]. The attacker also does not know the training dataset of the ASR, and he/she must generate the adversarial audio using a TTS model. The attacker can query the target ASR (e.g., Amazon, Microsoft, iFLYTEK, etc.) and prepare attack audio accordingly.

### 3.2. System and Threat Model of ALIF-OTA

**System Model.** Figure 4 presents the system model of ALIF-OTA, which represents a typical adversarial audio attack scenario. In the OTA scenario, the attack targets can be categorized into two types: commercial online ASR APIs and VAs (such as smart speakers and virtual assistants on computers). Our study specifically includes online ASR APIs to account for practical considerations, as small-scale companies may purchase online APIs rather than develop their own ASR system.

**Attacker's Goal.** The attacker intends to attack the VA running on a voice-controlled smart device, causing the ASR to transcribe the hidden command from the incomprehensible audio, triggering the device to execute unintended actions.
**Attacker's Capability.** We assume the adversary can launch the attack in the physical environment when the owner is not using the VA device, similar to Metamorph [17]. Even if the owner is in proximity and hears the attack audio, he/she cannot recognize the sound as a speech command and would not realize the onset of an adversarial attack, as described in Carlini et al. [18] and Abdullah et al. [7]. The attacker can launch the attack through a covertly placed speaker or a compromised speaker, following the convention in Devil's Whisper [8] and OCCAM [9].

## 4. Motivation and Our Design

Existing black-box attacks suffer from query inefficiency (except for NI-OCCAM [9], a non-interactive attack method). Additionally, they exhibit susceptibility to model updates, as outlined in Section 1. These inefficiencies lead to substantial attack costs. Next, we introduce our proposed design to address these issues.

The vulnerability of deep learning (DL) models to imperceptible perturbations has attracted much attention since the works of Szegedy et al. [19] and Biggio et al. [20]. A recent study by Shamir et al. [21] proposes a dimpled manifold model (DMM), which provides a better explanation for the working mechanism of AE. The effect of AE is closely related to the model's decision boundary. During model training, the initially randomly oriented decision boundary quickly aligns to a low-dimensional manifold that contains the representation embedding of all training samples. In the second training phase, the decision boundary starts dimpling, and *shallow* bulges are generated to move the decision boundary towards the right direction around the data embedding according to the labels. When a service provider fine-tunes its ASR model with newly-collected training data, such as noisy data for robustness improvement, the decision boundary undergoes these two training phases again, resulting in the decision boundary reforming around the new sample embeddings. This reformulation of the decision boundary renders previous AEs invalid with a high probability. Existing AEs add perturbations in the raw input space rather than in the lower-dimensional representation space, which does not guarantee enough distance between the attack sample embedding and the decision boundary. As a result, AEs are easily affected by changes in the decision boundary induced by model updates. To address these challenges, we propose ALIF, a novel attack scheme that reduces inefficient querying and significantly improves reliability.

## 5. ALIF-OTL: Over-the-Line ALIF Attack on ASR APIs

In this section, we present a detailed description of the ALIF-OTL algorithm. It is important to note that ALIF serves as the foundational pipeline for both ALIF-OTL and ALIF-OTA, with the two attack schemes differing primarily in their training methodologies. We introduce the ALIF pipeline in the ALIF-OTL demonstration.

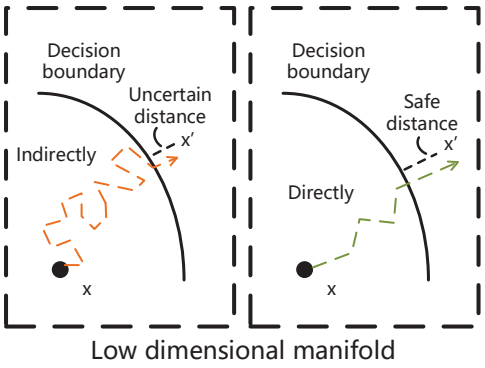


Figure 5. Contrast of AE-based attacks and our work.

### 5.1. Design Intuition

As revealed and verified by a massive number of papers (e.g., Ruderman et al. [22], Pope et al. [23], and Shamir et al. [21]), real-world data distribution can be described by low dimensional structure (i.e., a manifold). The low-dimensional manifold is easier for neural networks to learn and form complex decision boundaries from a relatively small amount of training samples. According to DMM, the decision boundary evolves primarily based on the low dimensional representations of the input sample (i.e., images or text), which heavily affect the establishment and the counterintuitive properties of adversarial examples. As the left of Figure 5 shows, existing black-box AE training adds perturbation to the raw input to shift the low dimensional representation, which is an indirect optimization process. Because the process lacks accurate guidance from the gradient descent method, the indirect search method is inefficient. Regarding the vulnerability to model updates, existing works [11], [9] iteratively optimize attack audio samples in the input space (see Section 2). The process stops when the pre-defined maximum number of queries is reached. Since the raw input space is not where the decision boundary lies, this process cannot ensure a large enough distance between the AE and the decision boundary, which increases the risk of attack audio failure when the decision boundary changes. To address the bottlenecks, our key idea is to construct perturbations directly from the decision boundary space and make the distance between attack samples and the boundary farthest allowed under certain constraints (i.e., right part of Figure 5).

Considering the architectures of ASR and TTS models in Figure 2, we can consider the two models to be reciprocal processes. Both ASR and TTS have a key component called the acoustic model. The purpose of this component in the two models is the same: to map the relationship

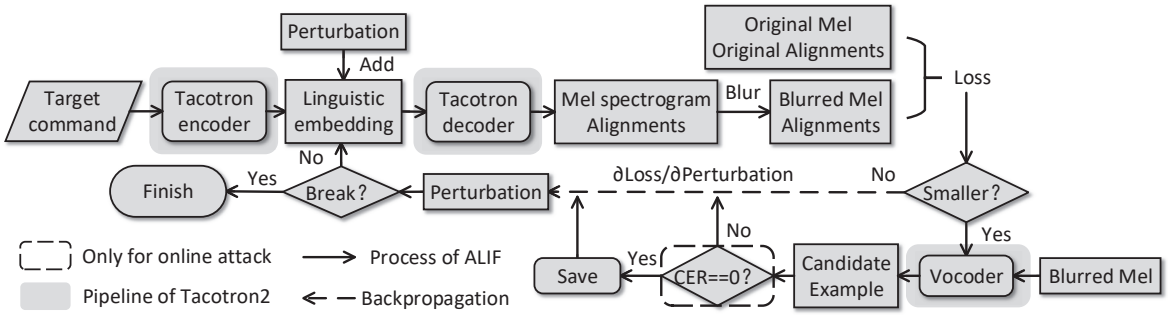


Figure 6. Architecture of the ALIF-OTL scheme. The scheme is based on the ALIF pipeline shown with a solid line.

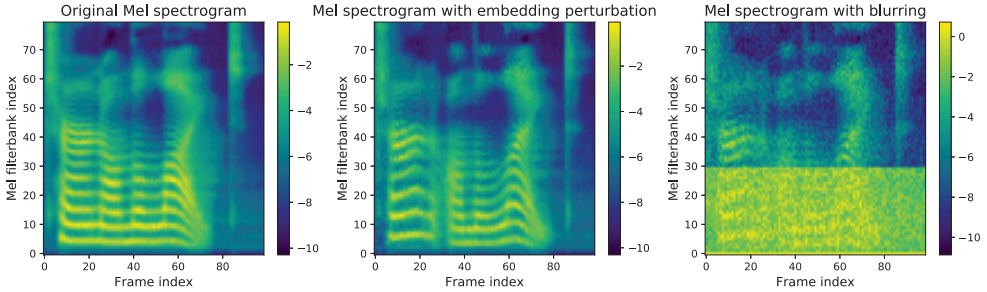


Figure 7. Comparison between the original spectrogram, after perturbing the embedding, and after the blurring operation.

between acoustic features and linguistic features. In TTS, the acoustic model uses linguistic feature as the input, which is the representation embedding of the linguistic feature space [24]. The linguistic embedding captures the essential semantic information of the text. Because the transcription of ASR is also formed based on the linguistic units contained in the output of the acoustic model, the TTS linguistic feature space is closely correlated with its counterpart in the ASR. This correlation between TTS and ASR is the observation concluded from our experiment and existing works. Existing studies have widely adopted TTS to synthesize audio commands [8], [9], [11]. Intuitively, we seek to generate "adversarial examples" from the input space of the acoustic model and design adversarial linguistic feature-based attacks. The generated "adversarial examples" in our scenario can be called "adversarial embeddings".

## 5.2. Our Method

**Design Overview.** ALIF-OTL creates incomprehensible audio to attack the subtitling APIs online. A conventional TTS generates a Mel spectrogram, which represents text content, then applies a vocoder to the spectrogram to synthesize signals representing the intended speech. We add perturbation to the representation of linguistic feature space to achieve a similar goal as hidden command attacks [18], [7], which is to make the attack audio not sound like the command anymore but still recognizable as a command. As a result, users will stay unaware of the occurrence of an attack.

Figure 6 shows the ALIF-OTL attack scheme. Our backbone model for audio generation is based on a well-known TTS called Tacotron2 [25], shown in the shaded part of Figure 6. The output of the Tacotron2 decoder consists of the predicted Mel spectrograms and the alignments. The former is the input of the vocoder, while the latter is used to predict whether the output sequence has been completed. We use a similar loss function to Tacotron2, which includes $L_{Mel}$ and $L_{Gate}$. $L_{Mel}$ is used to measure the difference between the original Mel spectrogram and the adversarial one, while $L_{Gate}$ controls the adversarial audio to have a similar length to the benign one. We will introduce them later.

To make an audio sample highly-distorted that individuals cannot understand, we optimize the perturbations using gradient descent to minimize the loss function. We observe that the magnitude of the perturbation is negatively correlated with the text similarity. The higher the amplitude of the perturbation, the more significant the difference between the transcription of the attack audio and the target content, which aligns with intuition. Based on this observation, we propose two variants of attacks: online and offline. We generate ten candidate adversarial embeddings for a target command and optimize each embedding vector for 50 iterations. Within this process, we query the target ASR API and calculate the character error rate (CER) to check whether the transcription is the same as the target command. During the *online attack*, we only query the target API with audio samples if the loss calculated between the original and perturbed spectrograms is reduced. In contrast, we only

query the API once at the last iteration round for *offline attack*. If the amplitude of the perturbation exceeds the pre-defined threshold during the loop, the process of both attacks is terminated.

**Mel spectrogram blurring.** Prior to computing $L_{Mel}$, we execute a blurring operation on the audio spectrogram to further reduce perceptibility. The key idea is to disturb the spectral structure of the spectrogram while keeping the essential semantic-related part untouched. The output of the Tacotron decoder in Figure 6 is the spectrogram that reflects the energy distribution in different frequencies. The following vocoder generates phase information based on the spectrogram and transforms the spectrogram into an audio waveform. We choose to manipulate the Mel spectrogram rather than the vocoder because the signal energy distribution in a spectrogram presents information about sound types. Vowels and consonants are the two major sound types present in speech, and their distinctiveness contributes to the intelligibility of a sentence [26], [27]. To manipulate the energy distribution while ensuring transcription correctness, we empirically design three blurring techniques: 1) *Decreasing the low-frequency energy.* We multiply the amplitude of frequencies for the first 30 Mel filter banks (80 filter banks in total) by a scalar factor $\alpha$ (e.g., 0.25 or 0.3). 2) *eliminating very low frequencies.* The very low frequencies do not have a significant influence on the recognition results. We set the amplitude of the first $\beta$ Mel filter banks to zero. 3) *superimpose uniform noise.* At last, we add a layer of noise to the Mel spectrogram, which effectively reduces the "sharpness" of the audio. Specifically, we create a noise matrix with values sampling from a uniform distribution ranging from $-\gamma$ to $\gamma$ ($\gamma > 0$) and with the same size as the attack audio, then overlap these two matrices. Figure 7 illustrates the spectrograms of audio before and after the blurring steps.

**ALIF-OTL Formulation.** Our goal is to reduce the intelligibility of the audio while maintaining the correct transcription of the target content to activate the intended command. We calculate the difference between the original audio signal and the perturbed audio sample regarding both the Mel spectrogram (i.e., $L_{Mel}$) and the alignment vectors (i.e., $L_{Gate}$) . Specifically, we use the summed mean squared error (MSE) and BCEWithLogitsLoss [28] for measuring the difference, respectively. As a result, the objective functions are as follows:

$$L_{Mel} = -MSE(Blur(D(Emb{+}\delta)_{Mel}, hp), D(Emb)_{Mel}), \tag{3}$$

$$L_{Gate} = BCE(Sigmoid((D(Emb{+}\delta)_{Gate}), D(Emb)_{Gate}) \tag{4}$$

$$L_{Final} = L_{Mel} + L_{Gate} \tag{5}$$

where $L_{Mel}$ is the spectrogram difference. $D(\cdot)_{Mel}$ referes to the decoder function in Tacotron2 which uses the embedding to predict the corresponding spectrogram [28]. $D(\cdot)_{Gate}$ is vector representing alignment status. $Emb$ means the original representation of the target command in the latent space of linguistic features. $Emb + \delta$ is the adversarial embedding. $hp$ denotes the hyperparameters $\alpha$, $\beta$ and $\gamma$.

Details of ALIF-OTL is shown in Algorithm 1, $Thr$ is the pre-defined upper limit of the perturbation amplitude, which ensures the semantics are not shifted far away from the origin. The perturbations $\delta$ are represented as a matrix of values which are added to the embedding space. $\delta$ is measured by its infinity norm; that is

$$\|\delta\|_p = \max \delta_{ij} \tag{6}$$

where $\delta_{ij}$ is the element of the perturbation matrix $\delta$. $i$ denotes the index of the token (i.e., letter) in the text, and $j$ means the index of the element in the representation embedding of this token.

**Algorithm 1** ALIF-OTL

**Input:** Target command $T$; target API $SR$; encoder $E$ and decoder $D$ of Tacotron2; vocoder V; threshold $Thr$; the blurring parameters $\alpha$, $\beta$, $\gamma$ and the number of iterations for one attack example $EpochMax$.
**Output:** $AttackExample$

$\delta = 0$; $BestLoss = +\infty$; $Epoch = 0$
$Emb = E(T)$
$OriSpec = D(Emb)$
$OriginalAudio = V(OriSpec)$
$AttackExample = OriginalAudio$
**while** $Epoch < EpochMax$ **do**
  $AdvSpec = Blur(D(Emb + \delta), \alpha, \beta, \gamma)$
  Calculate $L_{Final}$ by Eq.3, 4 and 5
  Update $\delta$ by gradient descent
  **if** $L_{Final} < BestLoss$ **then**
    $BestLoss = L_{Final}$
    $CandidateExample = V(AdvSpec)$
    $t = SR(CandidateExample)$
    **if** $t == T$ **then**
      $AttackExample = CandidateExample$
    **end if**
    **if** $\|\delta\|_p > Thr$ **then**
      break
    **end if**
  **end if**
  $Epoch++$
**end while**
**return** $AttackExample$

# 6. ALIF-OTA: Over-the-Air ALIF Attack on Voice Assistants

## 6.1. Technical Challenges

Because ALIF-OTL does not consider the impact factors of physical playback, it cannot be applied to in the OTA scenario. One big challenge is generating attack audio samples that can overcome the distortion brought by the physical playback environment. In traditional AE attacks, the attacker can incorporate OTA impact, such as reverberation, into the training objective function in the perturbation generation stage to achieve a "rehearsal" effect. This processing significantly improves the attack robustness in the physical playback. However, for the ALIF attack, the origin of our optimization is the low-dimensional embedding, which is not the natural waveform domain. It is unknown how to

reflect the reverberation and additional noise impact. Therefore, it is necessary to find an optimization solution that can simultaneously achieve intelligibility reduction and physical inference endurance.

### 6.2. Our Method

The key idea is to guide the optimization direction in a way that physical interference is considered. Metaheuristic algorithms are suitable in this context. Inspired by Xie et al. [29], we solve the problem with Particle Swarm Optimization (PSO).

**Particle Swarm Optimization.** PSO is a kind of metaheuristic algorithm which is designed to solve nonlinear functions [30]. It requires no knowledge and does not need the problem to be differentiable. In more detail, the algorithm first initializes a population of random solutions. Every solution can be regarded as a particle $X_i = \{x_{i1}, x_{i2}, ..., x_{id}\}$ and the value of the solution is the position of the particle in the hyperspace. Meanwhile, each particle has a randomized velocity $V_i = \{v_{i1}, v_{i2}, ..., v_{id}\}$ and then they "fly" iteratively through the hyperspace. Specifically, during iteration, the global best position and the personal best position of each particle are recorded. Then in each iteration, the velocity and position of each particle can be updated as follows:

$$v_{ij} = w \cdot v_{ij} + c_1 \cdot r_1 \cdot (pbest_{ij} - x_{ij}) + c_2 \cdot r_2 \cdot (gbest_j - x_{ij}) \quad (7)$$

$$x_{ij} = x_{ij} + v_{ij} \quad (8)$$

where $v_{ij}$ means the $j^{th}$ dimension of the velocity of $i^{th}$ particle in the current iteration. $pbest$ and $gbest$ are the personal best position and global best position. $r_1$ and $r_2$ are two random uniformly distributed numbers between 0 and 1. $w$ is the inertia weight while $c_1$ and $c_2$ are two acceleration constants. In this work, we regard the noise added to the linguistic embedding as a particle. Unlike the original PSO, we initialize all the particles with a zero value to help them find the initial best position. We randomize $r_1$ and $r_2$ at every iteration for every particle. In order to reduce the computing overhead, each dimension of the same particle shares these two random numbers. We also set a threshold and only update the particle when it does not exceed the threshold. As a result, Eq. 8 becomes:

$$x_{ij} = x_{ij} + v_{ij}, \text{ when } max(x_{ij} + v_{ij}) < threshold \quad (9)$$

**Design Overview.** Our goal in the OTA case is more challenging than the OTL case becuase of additional interferences in the physical playback environment. Based on the same ALIF pipeline shown in Figure 6, we use PSO to search for the optimal perturbation iteratively. The perturbation vector $Emb$ is the particle, and the set of the embedding range consists of the optimization space. The ALIF-OTA scheme differs from the OTL counterpart in an extra noise overlay step. We intentionally create and combine the white noise with the generated waveform. The particles recognized correctly under this constraint have a better chance of succeeding in the OTA scenario. We do not consider the reverberation influence as we empirically find the effect is not severe in a residential room environment. Having obtained the attack audio waveform, we generate and add a white noise segment with the maximum magnitude of $\eta$ (e.g., 0.05 or 0.1) to the normalized audio $x$. The maximum value of the mixed audio becomes $1+\eta$ (1.05/1.1). We then normalize each waveform data point and multiply it by 10000. The processed audio is fed into an online ASR API to obtain the recognition result (i.e., $y$). The above parameter setting is empirically concluded with experiments.

**ALIF-OTA Formulation.** To begin, we initialize 20 particles (perturbations) as zero matrices. Each particle then performs ten iterations to search for its best position. In each iteration, we query the API with and without the noise addition and record the loss $L_{Final}$ of those particles, which results in the correct transcription. Upon reaching the maximum number of iterations, we pick the particle with the minimum loss as the perturbation for the attack. In short, the ALIF-OTA problem is formulated as

$$\begin{aligned} L_{Final} = &\begin{cases} L_{Mel} + L_{Gate}, & \text{if } condition1 \text{ and } condition2 \\ +\infty, & otherwise \end{cases} \\ & while \quad \|\delta\|_p \leq Thr, \\ & AdvSpec = Blur(D(Emb + \delta), \alpha, \beta, \gamma), \\ & condition1 : SR(V(AdvSpec)) = T, \\ & condition2 : SR(V(AdvSpec))) + N) = T. \end{aligned} \quad (10)$$

where $N$ is the white noise. More details are shown in Algorithm 2.

## 7. Experiments

For ALIF-OTL, we first present a large-scale baseline evaluation on industry-grade cloud ASR APIs. Then, we pick the attack audio generated from the baseline study to launch the platform attack against the subtitling services. For ALIF-OTA, we attack the same online APIs in the over-the-line scenario but launch the attack in the physical environment. Additionally, we examine the feasibility of attacking VAs. Lastly, we conduct an impact factor study to investigate the overall performance of the ALIF-OTA attack.

### 7.1. Experimental Setup

**Target Commands and Text.** We aim to generate attack audio samples that can be successfully transcribed as different speech commands[3] and common sentences[4]. We use the same target commands and text for both ALIF-OTL and ALIF-OTA evaluation. In this section, we mainly conduct evaluations based on a dataset consisting of ten commands and two sentences. To validate the efficacy of our method,

3. The commands include *airplane mode on*, *call 123*, *cancel my alarm clock*, *I need help*, *navigate to my office*, *send a message to my mom*, *transfer the payment*, *turn on the light*, *unlock the door*, and *what's the time*.

4. *I can't take it anymore* and *darn it*.

TABLE 1. ATTACK SUCCESS RATES OF ALIF-OTL ON COMMERCIAL ASR APIS.

| parameters | | | Amazon | | | Azure | | | iFLYTEK | | | Tencent | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\gamma$ | $\beta$ | $\alpha$ | online-SR | offline-SR | query | online-SR | offline-SR | query | online-SR | offline-SR | query | online-SR | offline-SR | query |
| 0 | 0 | 0.25 | 12/12 | 6/12 | 32.8 | 9/12 | 3/12 | 34.8 | 10/12 | 4/12 | 33.8 | 12/12 | 5/12 | 33.6 |
| | | 0.3 | 12/12 | 5/12 | 30.8 | 10/12 | 5/12 | 33.6 | 12/12 | 4/12 | 33.9 | 12/12 | 4/12 | 33.1 |
| 1 | 1 | 0.25 | 12/12 | 4/12 | 35.6 | 9/12 | 1/12 | 38.0 | 10/12 | 2/12 | 40.3 | 11/12 | 4/12 | 34.6 |
| | | 0.3 | 12/12 | 3/12 | 31.3 | 10/12 | 5/12 | 35.0 | 11/12 | 4/12 | 36.3 | 11/12 | 4/12 | 34.9 |
| 1 | 2 | 0.25 | 11/12 | 4/12 | 35.6 | 10/12 | 4/12 | 36.8 | 10/12 | 1/12 | 39.2 | 12/12 | 6/12 | 33.4 |
| | | 0.3 | 12/12 | 5/12 | 32.4 | 9/12 | 3/12 | 35.2 | 10/12 | 2/12 | 36.4 | 12/12 | 5/12 | 34.3 |
| 2 | 4 | 1 | 12/12 | 4/12 | 30.4 | 11/12 | 5/12 | 36.6 | 11/12 | 2/12 | 34.6 | 12/12 | 7/12 | 29.8 |
| average | | | 11.9/12 | 4.4/12 | 32.7 | 9.7/12 | 3.7/12 | 35.7 | 10.6/12 | 2.7/12 | 36.4 | 11.7/12 | 5.0/12 | 33.4 |

The scalar factor, represented as $\alpha$, is employed to attenuate the amplitude of frequencies within the initial 30 Mel filter banks. $\beta$ is the Mel filter bank index. We set the amplitude of frequencies within the first $\beta$ Mel filter banks to 0. $\gamma(> 0)$ means the upper limit of the uniform noise amplitude we add to the Mel spectrogram. The terms "online-SR" and "offline-SR" represent the two variants of OTL attacks that require at most 50 and 1 query, respectively.

**Algorithm 2** ALIF-OTA

**Input:** Target command $T$; target API SR; encoder $E$ and decoder $D$ of Tacotron2; vocoder V; threshold $Thr$; the number of particles $k$; the number of iterations $EpochMax$; the amplitude of white noise $\eta$ and the blurring parameters $\alpha$, $\beta$ and $\gamma$.
**Output:** $AttackExample$

Initialize the value of $k$ particles $P = \{\delta_0, \delta_1, ..., \delta_k\}$ to 0, and the velocity is random;
$Epoch = 0; gloss = +\infty; ploss = \{+\infty_0, +\infty_1, ..., +\infty_k\}$
$Emb = E(T)$
$OriSpec = D(Emb)$
$OriginalAudio = V(OriSpec)$
**while** $Epoch¡EpochMax$ **do**
  **for** $\delta_i \in P$ **do**
    $AdvSpec_i = Blur(D(Emb + \delta_i), \alpha, \beta, \gamma)$
    $CandidateExample_i = V(AdvSpec_i)$
    Generate white noise $N$ with an amplitude of $\eta$
    $CandidateExample'_i = CandidateExample_i + N$
    $t_i = SR(CandidateExample_i)$
    $t'_i = SR(CandidateExample'_i)$
    **if** $t_i == T$ & $t'_i == T$ **then**
      $L_{Finali} = L_{Meli} + L_{Gatei}$
    **else**
      $L_{Finali} = +\infty$
    **end if**
    **if** $ploss_i > L_{Finali}$ **then**
      $ploss_i = L_{Finali}$
      $pbest_i = \delta_i$
    **end if**
    **if** $gloss > L_{Finali}$ **then**
      $gloss = L_{Finali}$
      $gbest = \delta_i$
      $AttackExample$ = $CandidateExample_i$
    **end if**
  **end for**
  Update all particles according to $Thr$, $gbest$ and $pbest$ by Eq.7 and 9
  $Epoch++$
**end while**
**return** $AttackExample$

we increase the dataset size for a more comprehensive evaluation, successfully generating attack audio samples using 32 commands and sentences in total. More details are provided in Appendix A.

**Parameter Settings.** The Mel spectrogram blurring and noise addition steps involve numerous hyperparameters: 1) $\gamma$ refers to the maximum value of the uniformly distributed noise. 2) $\beta$ is the Mel filter bank index. We set the amplitude of frequencies within the first $\beta$ Mel filter banks to 0. 3) $\alpha$ is the coefficient of energy attenuation in the low-frequency region of the Mel spectrogram. 4) $\eta$ indicates the magnitude of the white noise relative to the normalized synthesized audio waveform.

**Targets.** *OTL scenario.* We first conduct a baseline study: evaluating the performance of ALIF-OTL on commercial ASR APIs, including Amazon [16], Microsoft Azure [31], iFLYTEK [32], and Tencent [33]. Because Amazon, iFLYTEK [34] and Microsoft [35] also provide the subtitling service, we pick the adversarial audio samples in the baseline study to perform the platform attack. *OTA scenario.* We launch the attack by playing the adversarial audio samples over the air in the environment. The audio samples are recorded and transcribed in three ways: 1) they are recorded by a common USB microphone connected to a laptop and are then transcribed by the four ASR APIs; 2) they are recorded and processed by a laptop running Microsoft's Cortana voice assistant; 3) Amazon Echo processes them.

**Hardware.** In the OTA scenario, we use a Marshall EMBERTON II as the Bluetooth speaker for playing the adversarial audio. The laptop running Cortana is a Lenovo Thinkbook14+, and we use a Xiaomi phone connected to the Marshall speaker for playing audio. When attacking a standalone smart speaker, we use a third generation Amazon Echo Dot and play the attack audio via the Marshall speaker connected to the laptop.

**Evaluation Metrics.** We measure the attack effectiveness using the attack success rate (SR), the proportion of all attack audio successfully transcribed into the target command/text.

## 7.2. Evaluation of ALIF-OTL Attacks on Cloud Speech-to-Text APIs

**Baseline.** To evaluate the effectiveness of the ALIF-OTL attack, we perform the attack against four cloud ASR APIs. Each attack instance is generated after at most 50 queries to the target API. Specifically, for the *online attack*, we only query the target ASR as long as the loss continues to decrease, and terminate after 50 iterations regardless. For

TABLE 2. SUCCESS RATES OF ALIF-OTL ATTACKS ON SUBTITLING SERVICES.

| Amazon | | Microsoft | | iFLYTEK | |
|---|---|---|---|---|---|
| digital-SR | attack-SR | digital-SR | attack-SR | digital-SR | attack-SR |
| 12/12 | 5/12 | 10/12 | 5/10 | 11/12 | 5/11 |

"digital-SR" means the success rates of online attacks on the ASR APIs, which are the same as those in Table 1 ($\alpha = 0.3, \beta = 1, \gamma = 1$). "attack-SR" means the success rates of attacking the subtitling services.

the *offline attack*, we only query the API once after 50 iterations. For both situations, the process also terminates if the perturbation magnitude exceeds the threshold. We generate ten audio instances for each command and select the best one. This number of instances is a parameter that can vary depending on the attack strategy. Table 1 shows the evaluation results. In the online attack, our method can generate attack examples for almost all target commands. The best attack success rate achieved was 95.8% on the four APIs with only 35 queries per attack sample, using the parameter settings of $\alpha = 1$, $\beta = 4$, and $\gamma = 2$. In offline attacks, our method reaches a success rate of 33.3% for Amazon, Azure, and Tencent services with parameter settings of $\alpha = 0.3$, $\beta = 0$, and $\gamma = 0$. Though this seems low compared to the success rate of the online attacks, offline attacks do not require any API queries during the attack sample generation, which can limit the success rate due to the lack of feedback. The experimental results validate the effectiveness of the ALIF-OTL attack on commercial APIs. As a comparison, Devil's whisper [8] failed to attack the Amazon API effectively, and its success rate was only 4/10, even with the confidence score. OCCAM [9] can achieve a 100% success rate on Azure, iFLYTEK, and Tencent but requires a significantly high number of queries (around 30,000). Our work achieves a success rate of 95.8% with very few queries (about 35 for one instance), and the efficiency is 99.9% and 97.7% higher than that of OCCAM and Devil's whisper, respectively.

**Platform Attacks.** With the parameters where $\alpha = 0.3$, $\beta = 1$, and $\gamma = 1$, we pick the attack samples that can successfully attack Amazon, Azure, and iFLYTEK, respectively, then use these adversarial audio samples to attack the online subtitling services. Windows provides the subtitling service via a software named Clipchamp, iFLYTEK only provides website entry without API, and we assume a close relationship between the Amazon ASR model and the subtitling model. Therefore, we perform a transfer attack rather than training attack audio samples tailored for each platform.

The experimental results are shown in Table 2. Almost half of the attack examples are still effective after transferring from speech recognition APIs to automatic subtitling APIs. The results verify the threat of our method to subtitling services. The performance can be further improved by training attack samples specifically targeting the Amazon subtitling API.

## 7.3. Evaluation of ALIF-OTA Attacks on APIs and Voice Assistants.

**Attacks on cloud ASR APIs.** We attack the same APIs of ALIF-OTL but launch the attack by playing attack audio samples out loud, using a speaker. We conduct the evaluation in a normal bedroom using a Marshall EMBERTON II speaker and use a fixed speaker-to-microphone distance of 15cm (the same setting as Zheng et al. [9]). Table 3 demonstrates the attack performance of ALIF-OTA. The column of "digital SR" refers to the success rate of our ALIF-OTA algorithm in the digital domain. In most parameter settings, our method can successfully generate almost all target commands. We play and record all the attack audio samples three times and provide them to the corresponding API. We consider an attack sample successful if at least one of the three recordings is transcribed by the targeted API to the intended target command. The results show that despite enduring interference in the physical domain, our attack can achieve a success rate of more than 50% towards all the APIs under most parameter settings. Specifically, the success rate is 81.2% under $\alpha = 1$, $\beta = 4$ and $\gamma = 2$.

**Attacks on voice assistants.** Assuming the ASR systems behind the online API and the VA of the same company are similar, we use the adversarial audio samples generated from the API attack scenario to attack the Amazon Echo Dot and Microsoft Cortana (i.e., a transfer attack). Since Amazon Echo can't respond to "Darn it!" and "I can't take it anymore!", we generate one attack example for each of the remaining ten commands. Under the same environment settings as the cloud ASR API attack, we regard the attack as successful if the targeted command can be correctly transcribed within ten attempts (i.e., play each attack sample ten times). The results are in Table 4. Under different parameter settings, our attack can achieve an average success rate of up to 69.2% on VAs (80% on Echo and 58.3% on Cortana), which illustrates the effectiveness of our OTA attack. Although Ni-OCCAM can also attack the VAs with a low cost, their success rates (average 50% on Echo and Cortana) are lower.

## 7.4. Impact of Various Factors on ALIF-OTA

**Speaker dependency.** Different speakers have various hardware properties that affect the audio attributions. We intend to understand the sensitivity of audio samples generated by ALIF-OTL to the particular sound profiles of individual playback devices. Table 5 describes the attack success rate of our attack commands in different hardware and environmental settings. We pick all examples generated from ALIF-OTA using the parameters $\gamma = 1$, $\beta = 1$, $\alpha = 0.3$, and $\eta = 0.1$ and test their performance using products from three well-known speaker manufacturers. The results demonstrate that our attack examples exhibit similar performances when played by the three speakers. All speakers achieve a success rate of higher than 57.1% (4/7) on Echo and Cortana at a distance of 15cm. The only exception is using the JBL

TABLE 3. SUCCESS RATES OF ALIF-OTA ATTACKS ON COMMERCIAL ASR APIS.

| parameters | | | | Amazon | | Azure | | iFLYTEK | | Tencent | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\gamma$ | $\beta$ | $\alpha$ | $\eta$ | digital-SR | record-SR | digital-SR | record-SR | digital-SR | record-SR | digital-SR | record-SR |
| 0 | 0 | 0.25 | 0.05 | 9/12 | 5/12 | 6/12 | 4/12 | 6/12 | 5/12 | 9/12 | 5/12 |
| | | | 0.1 | 11/12 | 7/12 | 6/12 | 6/12 | 4/12 | 3/12 | 10/12 | 5/12 |
| | | 0.3 | 0.05 | 12/12 | 8/12 | 7/12 | 5/12 | 7/12 | 6/12 | 9/12 | 8/12 |
| | | | 0.1 | 12/12 | 7/12 | 5/12 | 5/12 | 6/12 | 5/12 | 11/12 | 6/12 |
| 1 | 1 | 0.25 | 0.05 | 8/12 | 6/12 | 5/12 | 4/12 | 6/12 | 6/12 | 11/12 | 5/12 |
| | | | 0.1 | 8/12 | 6/12 | 5/12 | 4/12 | 5/12 | 5/12 | 9/12 | 7/12 |
| | | 0.3 | 0.05 | 10/12 | 9/12 | 8/12 | 6/12 | 7/12 | 7/12 | 9/12 | 4/12 |
| | | | 0.1 | 9/12 | 8/12 | 7/12 | 7/12 | 3/12 | 3/12 | 9/12 | 5/12 |
| 1 | 2 | 0.25 | 0.05 | 8/12 | 7/12 | 5/12 | 3/12 | 7/12 | 7/12 | 8/12 | 6/12 |
| | | | 0.1 | 8/12 | 7/12 | 6/12 | 6/12 | 5/12 | 5/12 | 7/12 | 5/12 |
| | | 0.3 | 0.05 | 11/12 | 10/12 | 9/12 | 7/12 | 9/12 | 8/12 | 10/12 | 8/12 |
| | | | 0.1 | 12/12 | 9/12 | 8/12 | 8/12 | 5/12 | 4/12 | 9/12 | 6/12 |
| 2 | 4 | 1 | 0.05 | 12/12 | 12/12 | 10/12 | 8/12 | 11/12 | 9/12 | 10/12 | 8/12 |
| | | | 0.1 | 12/12 | 12/12 | 9/12 | 7/12 | 11/12 | 11/12 | 9/12 | 9/12 |
| average | | | | 10.1/12 | 8.1/12 | 6.9/12 | 5.7/12 | 6.6/12 | 6.0/12 | 9.3/12 | 6.2/12 |

[1] $\eta$ is the amplitude of white noise we add to our attack examples.
[2] "digital-SR" refers to the success rates of ALIF-OTA attacks towards the APIs in the digital domain. "record-SR" means the success rates of ALIF-OTA attacks in the physical world, where we play and record our attack examples and feed them to the APIs.

TABLE 4. SUCCESS RATES OF ALIF-OTA ATTACKS ON VOICE ASSISTANTS.

| parameters | | | | Echo | | Cortana | |
|---|---|---|---|---|---|---|---|
| $\gamma$ | $\beta$ | $\alpha$ | $\eta$ | digital-SR | physical-SR | digital-SR | physical-SR |
| 0 | 0 | 0.25 | 0.05 | 7/10 | 4/10 | 6/12 | 5/12 |
| | | | 0.1 | 9/10 | 4/10 | 6/12 | 6/12 |
| | | 0.3 | 0.05 | 10/10 | 8/10 | 7/12 | 7/12 |
| | | | 0.1 | 10/10 | 6/10 | 5/12 | 4/12 |
| 1 | 1 | 0.25 | 0.05 | 6/10 | 4/10 | 5/12 | 4/12 |
| | | | 0.1 | 6/10 | 3/10 | 5/12 | 4/12 |
| | | 0.3 | 0.05 | 8/10 | 4/10 | 8/12 | 7/12 |
| | | | 0.1 | 7/10 | 6/10 | 7/12 | 7/12 |
| 1 | 2 | 0.25 | 0.05 | 6/10 | 4/10 | 5/12 | 4/12 |
| | | | 0.1 | 7/10 | 4/10 | 6/12 | 5/12 |
| | | 0.3 | 0.05 | 9/10 | 5/10 | 9/12 | 4/12 |
| | | | 0.1 | 10/10 | 5/10 | 8/12 | 7/12 |
| 2 | 4 | 1 | 0.05 | 10/10 | 7/10 | 10/12 | 4/12 |
| | | | 0.1 | 10/10 | 8/10 | 9/12 | 7/12 |

"digital-SR" refers to the success rates of ALIF-OTA attacks towards the APIs in the digital domain. "physical-SR" means the success rates of attacking the VAs in the physical world.

Pulse5 in the meeting room, which results in a success rate of less than 50% (3/7) for Echo and Cortana.

**Evaluation in different rooms.** Different room types affect the propagation of acoustic signals differently. We pick two typical room types with different reverberation[5] effects to evaluate their impact on our attack. The results show that our samples can get similar success rates within 50 centimeters in both rooms. When the distance increases to 1m and beyond, the performance of our commands becomes less effective in the meeting room than in the bedroom. We suspect this is due to the stronger reverberation in the meeting room since our samples were not explicitly processed for reverberation effect.

**Evaluation of different attack distances.** To evaluate the attack robustness with respect to propagation distance, we conduct the OTA attack across varying distances. As shown in Table 5, when the distance is 15cm in the bedroom, the Marshall speaker achieves the best success rate, which is close to 100% (6/7) for Echo and 100% (7/7) for Cortana. The value drops to 42.9% (3/7) and 85.7% (6/7) at 50cm. When the distance reaches 2m, the success rates are 14.3% (1/7) and 54.1% (4/7). Despite the performance degradation, ALIF-OTA still achieves a success rate of higher than 50% in the challenging 2m distance. To the best of our knowledge, this is the longest attack distance in studies on black-box audio attacks (same as Devil's Whisper). NI-OCCAM [9], which has a similar scenario to ours, achieved a success rate of 60% on Cortana at a distance of 15cm and was not evaluated for performance at a longer distance.

**Evaluation of ambient noise.** We evaluate the robustness of our attack against ambient noise by launching our examples across various levels of white noise. As Table 6 illustrates, our examples demonstrate strong performance at a signal-to-noise ratio (SNR) of roughly 20dB and above, similar to soft speech, but performance diminishes with an increase in ambient noise. This aligns with the SNR level reported in Devil's Whisper [8] and When Evil Calls [11]. Notably, even with a decrease in SNR to 3 dB, we can still achieve a success rate of more than 40% when attacking Echo. This emphasizes the robustness of our attack.

## 7.5. User Study

We carried out a user study to gain deeper insight into human perception of the attack examples we generated. The study comprises two sections: audio incomprehensibility and ablation of different ALIF components. We recruited a varied group of 20 volunteers, aged 22 to 32, all with normal hearing abilities. Notably, five participants are native English speakers.

**Audio comprehensibility.** We selected 32 distinct command examples[6] produced by ALIF-OTL under the parameter settings of $\alpha = 0.3$, $\beta = 1$, and $\gamma = 1$. We recruited volunteers

5. A $5m * 3.8m$ bedroom and a $5.5m * 5.5m$ meeting room.

6. The audio samples include all commands and sentences in Footnote 3, 4 and Appendix A.

TABLE 5. ATTACK SUCCESS RATES OF ALIF-OTA UNDER DIFFERENT ENVIRONMENTS AND DEVICE SETTINGS.

| | speaker | EDIFIER | | | | marshall | | | | JBL | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | voice assistant | Echo | | Cortana | | Echo | | Cortana | | Echo | | Cortana | |
| | distance | SR | dB | SR | dB | SR | dB | SR | dB | SR | dB | SR | dB |
| bedroom | 15cm | 4/7 | 76.04 | 5/7 | 76.54 | 6/7 | 76.1 | 7/7 | 74.59 | 6/7 | 74.37 | 6/7 | 73.93 |
| | 30cm | 2/7 | 70.19 | 6/7 | 69.13 | 4/7 | 69.56 | 6/7 | 67.56 | 2/7 | 69.34 | 6/7 | 68.3 |
| | 50cm | 3/7 | 64.56 | 7/7 | 64.8 | 3/7 | 64.13 | 6/7 | 62.94 | 2/7 | 64.49 | 3/7 | 64.23 |
| | 100cm | 3/7 | 60.57 | 7/7 | 60.9 | 2/7 | 59.34 | 5/7 | 59.09 | 3/7 | 60.19 | 5/7 | 60.76 |
| | 200cm | 3/7 | 55.43 | 3/7 | 56.59 | 1/7 | 56.6 | 4/7 | 56.86 | 4/7 | 57.17 | 3/7 | 57.54 |
| meeting room | 15cm | 5/7 | 74.84 | 5/7 | 75.91 | 5/7 | 74.54 | 6/7 | 74.44 | 3/7 | 72.03 | 3/7 | 71.06 |
| | 30cm | 6/7 | 70.59 | 5/7 | 69.67 | 6/7 | 68.07 | 6/7 | 68.87 | 1/7 | 67.71 | 2/7 | 66.53 |
| | 50cm | 5/7 | 65.44 | 6/7 | 65.64 | 3/7 | 63.47 | 6/7 | 63.91 | 1/7 | 64.17 | 3/7 | 63.54 |
| | 100cm | 2/7 | 61.36 | 2/7 | 59.91 | 2/7 | 57.69 | 2/7 | 59.46 | 1/7 | 60.43 | 2/7 | 59.76 |
| | 200cm | 2/7 | 56.39 | 3/7 | 56.95 | 0 | 56.69 | 0 | 54.96 | 1/7 | 55.77 | 1/7 | 55.07 |

TABLE 6. IMPACT OF AMBIENT NOISE.

| Voice assistant | $SNR \approx 37$ | $SNR \approx 25$ | $SNR \approx 13$ | $SNR \approx 3$ |
|---|---|---|---|---|
| Echo | 6/7 | 6/7 | 3/7 | 3/7 |
| Cortana | 7/7 | 5/7 | 3/7 | 2/7 |

TABLE 7. USER STUDY ON AUDIO COMPREHENSIBILITY.

| | Non-native Speakers | Native Speakers | All volunteers |
|---|---|---|---|
| Score | 0.77 | 1.70 | 1.0 |
| CER | 0.79 | 0.45 | 0.71 |
| WER | 0.88 | 0.55 | 0.80 |

"Score" is the average score calculated based on feedback provided by all volunteers. In addition to score the intelligibility, participants are asked to transcribe audio samples. "CER" and "WER" are character error rate and word error rate obtained by calculating the difference between their transcription and the ground truth text.

to assess each sample using a 0-4 intelligibility scale[7] and subsequently transcribe them. The results are presented in Table 7. On average, participants assigned an intelligibility score of 1.0. Their transcriptions achieved a CER of 71% and a WER of 80%, indicating a significant deviation from the original commands. These statistics confirm the poor intelligibility of the audio samples.

**Distortion effect of ALIF's each component.** We aim to determine various components' effects on audio comprehensibility: perturbations on linguistic features and three Mel spectrogram blurring components. To do this, we selected five examples and ablated each component, resulting in 25 audio samples. We then asked volunteers to rate the comprehensibility of these audios and collected their feedback. As indicated in Table 8, the perturbation on linguistic features, which received the lowest score of 2.09, was identified as the most critical factor impacting human perception.

We conducted the ablation study to show the effects of the different components on user comprehension. Our ALIF algorithm ensures all generated attack samples can be correctly transcribed to target commands. Therefore, this experiment also shows that ablating various parts of the attack pipeline does not have a significantly negative effect on attack success.

7. Scores range from 0 to 4; 0: completely incomprehensible, 1: few parts are understandable, 2: some parts are understandable, 3: most parts are understandable, and 4: completely understandable.

TABLE 8. USER STUDY ON BLURRING ABLATION.

| | L.P. Only | Beta Only | Gamma Only | Alpha Only | All Mel Blurring |
|---|---|---|---|---|---|
| Non-native Speaker | 1.84 | 3.68 | 3.68 | 2.93 | 2.96 |
| Native Speaker | 2.84 | 3.96 | 3.76 | 3.04 | 3.24 |
| All volunteers | 2.09 | 3.75 | 3.7 | 2.96 | 3.03 |

Each column depicts the mean intelligibility score of audio samples produced by removing specific components from the ALIF pipeline. This illustrates the influence of our method's various components on user comprehension. "L.P. Only" indicates audio samples with solely linguistic feature perturbation. "Beta Only" means we only eliminate extremely low frequencies during the generation of attack audio samples. "Gamma Only" and "Alpha Only" follow a similar pattern. "All Mel Blurring" signifies the application of all Mel spectrogram blurring techniques. It should be noted that linguistic feature perturbation is exclusively implemented in the "L.P. Only" category.

## 7.6. Long-Term Effectiveness of ALIF

We validate the robustness of ALIF attacks against model updates by evaluating the long-term effectiveness of our attack audio samples.

We generate 50 to 100 effective attack examples targeting each of the four commercial APIs on October $1^{st}$, 2022, and test them after three months. The success rates of our attack audios on different dates are presented in Table 9. Although the success rate on iFLYTEK decreases by about 25%, the success rates on Amazon and Azure drop by less than 10%. Notably, the attack performance on Amazon remains unchanged after three months. Furthermore, after extending the testing period by an additional half month, the performance across all ASRs remained steady, with multiple examples of each command successfully executing the attack.

ALIF was employed to optimize the added perturbation to the linguistic feature to decrease audio comprehensibility. This is in contrast to existing black-box methods that add perturbation to shift the command signal towards another unsuspicious audio sample, bringing the signal closer to the ASR's decision boundary. Although ASR updates may still affect ALIF's performance, it has shown relative resilience.

## 7.7. Comparisons with related adversarial attacks

The key distinction between our work and existing AE studies is the way of constructing adversarial audio samples. Our work functions uniquely compared to existing works to

TABLE 9. CHANGE OF THE SUCCESS RATES OF ALIF-OTL OVER TIME.

| API | Amazon | Azure | iFLYTEK | Tencent |
|---|---|---|---|---|
| 2022.10.01 | 100% | 100% | 100% | 100% |
| 2023.01.01 | 100% | 91.53% | 74.03% | 94.12% |
| 2023.01.06 | 100% | 91.53% | 74.03% | 94.12% |
| 2023.01.11 | 100% | 91.53% | 74.03% | 94.12% |
| 2023.01.16 | 100% | 91.53% | 74.03% | 94.12% |

TABLE 10. COMPARISONS WITH RELATED ADVERSARIAL AUDIO ATTACKS.

| Attacks | Gradient | Conf | Targeted | Over the air | Distance | Query |
|---|---|---|---|---|---|---|
| CommanderSong [36] | ✓ | | ✓ | | | 1000 |
| Hidden voice [18] | | | ✓ | ✓ | 300 cm | 5000 |
| Devil's whisper [8] | | ✓ | ✓ | ✓ | 5-200 cm | 1500 |
| OCCAM [9] | | | ✓ | | | 30000 |
| TAINT [11] | | | ✓ | ✓ | 50 cm (VoIP) | 1500 |
| ALIF-OTL | | | ✓ | | | **50** |
| ALIF-OTA | | | ✓ | ✓ | 15-200 cm | **400** |

"Gradient" / "Conf" means whether the method needs the gradient/confidence score of the target model. "Targeted" denotes whether the attack is targeted or non-targeted. "Over the air" means whether the attack can work in the physical world. "Distance" is the distance between speaker and microphone in experiments ([11] launches the attack through VoIP). "Query" means the total number of queries for one attack example.

improve the query efficiency and robustness against model updates. Existing works add noise to the raw waveform to mislead the ASR. Their training objective is to ensure the ASR can be misled while minimizing the waveform-level perturbation, as formulated in Equation 2. In contrast, our approach generates audio signals that are incomprehensible to humans but recognizable by ASR. The training objective is to ensure that the ASR can recognize the command while maximizing the perturbations.

We compare our work and existing black-box AE attacks, as demonstrated in Table 10. Because the objectives of adding perturbation differ, we do not compare aspects such as the perturbation level. Our approach shows an evident advantage in query efficiency and attack distance.

# 8. Related Work

In this section, we survey studies related to speech command injection attacks. Note that we mainly cover research focusing on signal generation and do not include the work utilizing hardware property [5], [37], [38].

## 8.1. White-Box Adversarial Example

In recent years, extensive studies apply AE techniques to the speech domain to achieve command injection. Carlini et al. [6] first successfully generated the attack examples towards Deepspeech, an open-sourced end-to-end ASR platform. CommanderSong [36] tried to embed the malicious voice command into songs so that it won't attract human attention but could mislead Kaldi successfully. However, these works are fragile in the physical world. Specifically, the adversarial examples will fail when being played over the air. To address this problem, Yakura et al. and Imperio [39], [40] simulated the transformations caused by the physical world and introduced these transformations into the process of adversarial example generation. Metamorph [17] revealed that the signal distortion in the physical world is mainly caused by the device and channel frequency selectivity. They proposed a "generate-and-clean" two-phase design. To mitigate the human perception of adversarial perturbation, Schönherr et al. [41] generated adversarial examples based on psychoacoustic hiding. Inspired by the universal adversarial perturbations in the image domain [42], Neekhara et al. [43] generated the universal perturbations in the speech domain and successfully attacked Mozilla DeepSpeech. AdvPulse [44] also proposed universal, synchronization-free adversarial perturbations which make the attack scenario more practical.

## 8.2. Black-Box Adversarial Example

Despite the success of adversarial examples in speech recognition attacks, the white-box premise of many tools poses a significant practicality concern; knowledge of the architecture and parameters of the model is unrealistic for a real-world attack scenario. Therefore, recently, black-box attacks have become an active research area. Taori et al. [15] combined the approaches of genetic algorithms with gradient estimation to attack Deepspeech. This approach has a low success rate and is ineffective on commercial models. Devil's Whisper [8] utilized the confidence scores of the commercial ASR APIs. They built a substitute model to approximate the target model and launch the attack. However, most commercial APIs only return the final results without any confidence scores, which limits the practicality of this method. OCCAM [9] is the first approach that attacks commercial ASR APIs successfully in the completely black-box scenario, i.e., no confidence scores are required. However, the cost of this method is high. A recent work, TAINT [11], considers the impact of the VoIP channel and uses gradient estimation to generate adversarial examples, resulting in a more robust and efficient attack.

## 8.3. Hidden Voice Attacks

Besides adversarial examples, "hidden voice command" is another line of attack method targeting speech recognition systems. Cocaine Noodles [45] first used the inverse MFCC technique to create the attack sample which will be recognized by the devices but won't be understood by humans. Carlini et al. [18] extended it to the white-box scenario and proposed a "hidden voice command" attack that can't be understood at all. Abdullah et al. [7] then exploited signal processing algorithms to make hidden voice commands more practical.

# 9. Discussion

## 9.1. Defense against ALIF.

**Downsampling.** Downsampling is a commonly-used method to neutralize AE attacks. According to Nyquist's

TABLE 11. SUCCESS RATES AFTER DOWNSAMPLING THE EXAMPLES.

| API | Amazon | Azure | iFLYTEK | Tencent | total |
|---|---|---|---|---|---|
| 8kHz | 3/12 | 3/10 | 1/11 | 2/11 | 20.45% |
| 12kHz | 1/12 | 4/10 | 7/11 | 5/11 | 38.64% |

We use the attack audio samples generated to evaluate ALIF-OTL performance in Section 7.2. With the parameters set at $\alpha = 0.3$, $\beta = 1$, and $\gamma = 1$, 12, 10, 11, and 11 audio samples were successfully generated to attack Amazon, Azure, iFLYTEK, and Tencent ASR, respectively.

TABLE 12. SUCCESS RATES AFTER FILTERING THE EXAMPLES.

| API | Amazon | Azure | iFLYTEK | Tencent | total |
|---|---|---|---|---|---|
| High-Pass 500Hz | 1/12 | 3/10 | 0/11 | 2/11 | 13.64% |
| Low-Pass 4000Hz | 1/12 | 5/10 | 3/11 | 3/11 | 27.27% |
| Low-Pass 6000Hz | 2/12 | 5/10 | 5/11 | 4/11 | 36.36% |

theorem, the high-frequency component of the original audio will be removed after downsampling. Since AE adds perturbation at the high-frequency area, downsampling can effectively defend against existing black-box attacks. Liu et al. [11] downsampled attack audio to 16 kHz, then upsampled to 48 kHz, causing none of their attacks to succeed. When OCCAM [9] downsampled signals to 12 kHz and then upsampled back to 16 kHz, their attack fails, and the success rate of NI-OCCAM decreases to 30%. To evaluate the robustness of ALIF audio against downsampling, we pick about 10 ALIF audio samples (one sample for each speech command) and perform the same 12 kHz downsampling action, resulting in a success rate of 38.64%. To further push the limit, we perform downsampling to 8 kHz and still achieve a success rate of 20.45%. The experimental results are shown in Table 11, which demonstrate that ALIF has better robustness against downsampling, which makes sense because ALIF mainly affects the frequency range related to speech, which is usually below 8 kHz and has less dependence on higher frequencies. Overall, downsampling is an effective defense. ALIF could be more robust if the attacker knows the downsampling strategy and adapts the audio generation, but this is a strong assumption.

**Frequency filtering.** Apart from downsampling, we evaluate the impact of filtering on mitigating our attacks. We test three potential frequency filters: a 4000Hz low-pass filter, a 6000Hz low-pass filter, and a 500 Hz high-pass filter. Table 12 displays that the three methods successfully defended against our attack. The 500Hz high-pass filter performed best, with only 1, 3, 0, and 2 attack commands accurately recognized by the respective ASRs.

**Adversarial training.** In addition to downsampling, the defender could utilize adversarial training to enhance model robustness [46], [47]. Because ALIF is based on a different working mechanism than AE attacks, hypothetically, the effect of typical adversarial training on the input waveform domain is limited. However, embedding-level adversarial training is potentially effective. The models are not accessible to perform adversarial training because we primarily attack commercial ASRs in this paper. We leave studies on this new type of adversarial training for future work.

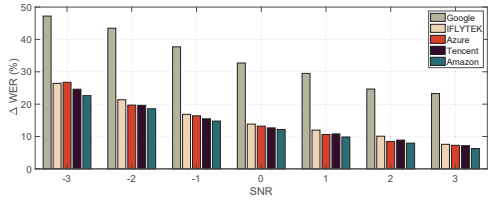


Figure 8. Robustness of commercial ASRs against white noise

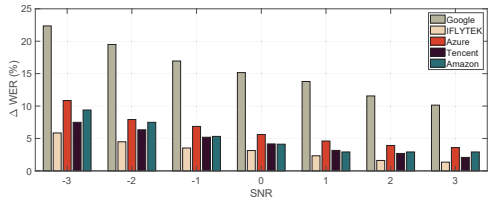


Figure 9. Robustness of commercial ASRs against event noise

## 9.2. Attack Success and ASR Robustness

Black-box attack training follows a specific paradigm in which a target phase is perturbed to both make the audio sound different as well as to limit a victim's awareness of the attack. The limit on the perturbation energy is no longer primarily designed for imperceptibility as in the white-box attack. Instead, it aims to prevent the transcription from being different than the target phrase. In this context, it is possible that the success of a black-box attack also depends on the ASR's robustness to noise.

To investigate if a correlation exists, we test the robustness of five commercial ASRs and study the connection between the attack performance of ALIF-OTL and the noise robustness. We randomly choose 200 sentences from the LibriSpeech test-clean subset and mix them with Gaussian white noise and event noise under different SNRs, respectively. Then, we evaluate them using ASRs and calculate the increase of word error rate ($\Delta$WER) caused by white noise, where smaller $\Delta$WER indicates better robustness, and vice versa. Figure 8 and Figure 9 demonstrate that Amazon is fairly robust while Google is the most vulnerable one, which is consistent with some of our observations during the attack sample generation. That is, we can generate attack samples against Amazon API relatively easily but cannot generate attack examples of high quality against Google API. The experimental results suggest the possibility of *the more robust the ASR is, the more vulnerable it is to our attack.* We will conduct a more comprehensive study to verify the assumption in the future.

## 9.3. Limitations

In this paper, we utilize PSO as the primary optimization method for ALIF-OTA and have not explored other techniques. Different optimization techniques can result in

various search routes in the problem space, and it is worth more investigation.

In both algorithms for ALIF-OTL and ALIF-OTA, we use a fixed threshold to control the amplitude of perturbation added to the linguistic feature, which is not precise enough, since an ASR service usually recognizes different commands with various accuracy, indicating ASRs have different sensitivity to command contents. Setting the threshold dynamically depending on the attack command is more practical. We leave it for future work.

## 10. Conclusion

In this paper, we propose the first black-box adversarial audio attack methods that are highly efficient and robust against model updates: ALIF-OTL and ALIF-OTA. The two attack schemes are based on ALIF, a novel adversarial linguistic feature-based attack pipeline. We directly add perturbation to the low dimensional manifold where the decision boundary lies to generate adversarial audio samples, which overcomes the inherent shortcomings of conventional black-box AEs: query inefficiency and the vulnerability to model updates. ALIF-OTL only requires an average of 35 queries to generate attack audios against cloud ASR APIs. Experiments show ALIF-OTL is effective against four well-known commercial cloud ASRs, achieving an attack success rate of 95.8%. ALIF-OTA uses the PSO method to incorporate environmental interference into the training and only requires 400 queries for attack sample generation. Experiments show the efficacy of our attack audio samples in attacking four commercial ASRs and two VAs in the physical playback environment, achieving average success rates of 81.3% and 69.2%, respectively. Critically, the test-of-time experiment verifies the long-term effectiveness of the ALIF attack, indicating its robustness to model changes.

## Acknowledgment

This work is partially supported by the National Key R&D Program of China (Grant No. 2020AAA0107700), the National Natural Science Foundation of China (Grant No. 62172359, 61972348, and 62102354), the National Key R&D Program of China (Grant No. 2021ZD0112803), the Funding for Postdoctoral Scientific Research Projects in Zhejiang Province (Grant No. ZJ2021139), the Fundamental Research Funds for the Central Universities (Grant No. 2021FZZX001-27), and the Hangzhou Leading Innovation and Entrepreneurship Team (TD2020003).

## References

[1] B. Thormundsson, "Number of voice assistant users in the United States from 2017 to 2022," https://www.statista.com/statistics/1029573/us-voice-assistant-users/, 2022, online; accessed 04-Jul-2022.

[2] F. Laricchia, "Number of digital voice assistants in use worldwide from 2019 to 2024 (in billions)*," https://www.statista.com/statistics/973815/worldwide-digital-voice-assistant-in-use/, 2022, online; accessed 04-Jul-2022.

[3] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 163–168.

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[5] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "DolphinAttack: Inaudible voice commands," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*. New York, NY, USA: ACM, 2017, pp. 103–117. [Online]. Available: http://doi.acm.org/10.1145/3133956.3134052

[6] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*, 2018, pp. 1–7.

[7] H. Abdullah, W. Garcia, C. Peeters, P. Traynor, K. R. B. Butler, and J. Wilson, "Practical hidden voice attacks against speech and speaker recognition systems," in *Proceedings of the 2019 Network and Distributed System Security Symposium (NDSS '19)*, 2019.

[8] Y. Chen, X. Yuan, J. Zhang, Y. Zhao, S. Zhang, K. Chen, and X. Wang, "Devil's whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices," in *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Aug. 2020, pp. 2667–2684. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/chen-yuxuan

[9] B. Zheng, P. Jiang, Q. Wang, Q. Li, C. Shen, C. Wang, Y. Ge, Q. Teng, and S. Zhang, "Black-box adversarial attacks on commercial speech platforms with minimal information," *arXiv preprint arXiv:2110.09714*, 2021.

[10] S. Khare, R. Aralikatte, and S. Mani, "Adversarial black-box attacks on automatic speech recognition systems using multi-objective evolutionary optimization," *Proc. Interspeech 2019*, pp. 3208–3212, 2019.

[11] H. Liu, Z. Yu, M. Zha, X. Wang, W. Yeoh, Y. Vorobeychik, and N. Zhang, "When evil calls: Targeted adversarial voice over ip network," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 2009–2023. [Online]. Available: https://doi.org/10.1145/3548606.3560671

[12] J. Chen, M. I. Jordan, and M. J. Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack," in *2020 IEEE Symposium on Security and Privacy (SP)*, 2020, pp. 1277–1294.

[13] F. Suya, J. Chi, D. Evans, and Y. Tian, "Hybrid Batch Attacks: Finding black-box adversarial examples with limited queries," in *USENIX Security Symposium*, 2020.

[14] X. Tan, T. Qin, F. Soong, and T.-Y. Liu, "A survey on neural speech synthesis," *arXiv preprint arXiv:2106.15561*, 2021.

[15] R. Taori, A. Kamsetty, B. Chu, and N. Vemuri, "Targeted adversarial examples for black box audio systems," *CoRR*, vol. abs/1805.07820, 2018. [Online]. Available: http://arxiv.org/abs/1805.07820

[16] Amazon, "Amazon transcribe - automatically convert speech to text," https://aws.amazon.com/transcribe/, 2022.

[17] T. Chen, L. Shangguan, Z. Li, and K. Jamieson, "Metamorph: Injecting inaudible commands into over-the-air voice controlled systems," in *Proc. NDSS'20*, 2020.

[18] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou, "Hidden voice commands," in *Proceedings of the 25th USENIX Security Symposium (USENIX Security'16)*. Austin, TX: USENIX Association, Aug 2016, pp. 513–530. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/carlini

[19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[20] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Proceedings of the 2013th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part III*, ser. ECMLPKDD'13. Berlin, Heidelberg: Springer-Verlag, 2013, p. 387–402. [Online]. Available: https://doi.org/10.1007/978-3-642-40994-3_25

[21] A. Shamir, O. Melamed, and O. BenShmuel, "The dimpled manifold model of adversarial examples in machine learning," *arXiv preprint arXiv:2106.10151*, 2021.

[22] D. L. Ruderman, "The statistics of natural images," *Network: computation in neural systems*, vol. 5, no. 4, p. 517, 1994.

[23] P. Pope, C. Zhu, A. Abdelkader, M. Goldblum, and T. Goldstein, "The intrinsic dimension of images and its impact on learning," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=XJk19XzGq2J

[24] Google, "Embeddings," https://developers.google.com/machine-learning/crash-course/embeddings/video-lecture, 2022.

[25] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[26] Q. Tahmina, F. Chen, and Y. Hu, "Perceptual contribution of vowels and consonants to sentence intelligibility by cochlear implant users," in *2014 International Symposium on Integrated Circuits (ISIC)*, 2014, pp. 200–203.

[27] D. Fogerty and L. E. Humes, "The role of vowel and consonant fundamental frequency, envelope, and temporal fine structure cues to the intelligibility of words and sentences," *The Journal of the Acoustical Society of America*, vol. 131, no. 2, pp. 1490–1501, 2012.

[28] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4779–4783.

[29] S. Xie, H. Wang, Y. Kong, and Y. Hong, "Universal 3-dimensional perturbations for black-box attacks on video recognition systems," in *2022 IEEE Symposium on Security and Privacy (SP)*, 2022, pp. 1390–1407.

[30] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the sixth international symposium on micro machine and human science*. Ieee, 1995, pp. 39–43.

[31] Azure, "Azure - speech to text," https://azure.microsoft.com/en-us/products/cognitive-services/speech-to-text/\#features, 2022.

[32] iFLYTEK, "iflytek open platform - speech transcribe," https://www.xfyun.cn/services/lfasr, 2022.

[33] Tencent, "Tencent cloud automatic speech recognition," https://www.tencentcloud.com/products/asr, 2022.

[34] iFLYTEK, "iflytek subtitling," https://zimu.iflyrec.com/machine/index.html, 2022.

[35] Clipchamp, "Clipchamp," https://clipchamp.com/en/, 2022.

[36] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "CommanderSong: A systematic approach for practical adversarial voice recognition," in *Proceedings of the 27th USENIX Security Symposium (USENIX Security '18)*. Baltimore, MD: USENIX Association, Aug 2018, pp. 49–64. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/yuan-xuejing

[37] N. Roy, H. Hassanieh, and R. Roy Choudhury, "BackDoor: Making microphones hear inaudible sounds," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '17)*. New York, NY, USA: ACM, 2017, pp. 2–14. [Online]. Available: http://doi.acm.org/10.1145/3081333.3081366

[38] N. Roy, S. Shen, H. Hassanieh, and R. R. Choudhury, "Inaudible voice commands: The long-range attack and defense," in *Proceedings of the 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI '18)*. Renton, WA: USENIX Association, Apr 2018, pp. 547–560. [Online]. Available: https://www.usenix.org/conference/nsdi18/presentation/roy

[39] H. Yakura and J. Sakuma, "Robust audio adversarial example for a physical attack," *CoRR*, vol. abs/1810.11793, 2018. [Online]. Available: http://arxiv.org/abs/1810.11793

[40] L. Schönherr, S. Zeiler, T. Holz, and D. Kolossa, "Imperio: Robust over-the-air adversarial examples for automatic speech recognition systems," 2019.

[41] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," in *Proceedings of the 2019 Network and Distributed System Security Symposium (NDSS '19)*, 2019.

[42] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," 2016.

[43] P. Neekhara, S. Hussain, P. Pandey, S. Dubnov, J. McAuley, and F. Koushanfar, "Universal Adversarial Perturbations for Speech Recognition Systems," in *Proc. Interspeech 2019*, 2019, pp. 481–485. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1353

[44] Z. Li, Y. Wu, J. Liu, Y. Chen, and B. Yuan, "Advpulse: Universal, synchronization-free, and targeted audio adversarial attacks via subsecond perturbations," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1121–1134. [Online]. Available: https://doi.org/10.1145/3372297.3423348

[45] T. Vaidya, Y. Zhang, M. Sherr, and C. Shields, "Cocaine noodles: exploiting the gap between human and machine speech recognition," in *9th USENIX Workshop on Offensive Technologies (WOOT 15)*, 2015.

[46] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016, pp. 582–597.

[47] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 2017, pp. 506–519.

TABLE 13. EVALUATION OF ALIF ON A LARGER DATASET.

| | Echo | Cortana | Amazon-online | Amazon-offline | Azure-online | Azure-offline | iFLYTEK-online | iFLYTEK-offline | Tencent-online | Tencent-offline |
|---|---|---|---|---|---|---|---|---|---|---|
| Clear notification | ✓ | - | ✓ | - | ✓ | - | ✓ | - | ✓ | - |
| Good morning | ✓ | ✓ | ✓ | - | ✓ | - | ✓ | - | ✓ | - |
| How old are you | ✓ | ✓ | ✓ | - | ✓ | - | ✓ | - | ✓ | ✓ |
| Play music | ✓ | ✓ | ✓ | - | ✓ | - | ✓ | - | ✓ | - |
| Sing me happy birthday | ✓ | - | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | ✓ |
| Take a picture | ✓ | - | ✓ | - | ✓ | ✓ | ✓ | ✓ | ✓ | - |
| Tell me a story | ✓ | - | ✓ | ✓ | ✓ | - | ✓ | - | ✓ | ✓ |
| What's the weather | ✓ | ✓ | ✓ | - | ✓ | - | ✓ | ✓ | ✓ | - |
| Where is my car | ✓ | - | ✓ | - | ✓ | - | ✓ | - | ✓ | ✓ |
| Where is my home | ✓ | ✓ | ✓ | - | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| Ask me a question | ✓ | ✓ | ✓ | - | ✓ | - | ✓ | ✓ | ✓ | ✓ |
| Clean my room | ✓ | - | ✓ | - | ✓ | - | ✓ | - | ✓ | - |
| Find a hotel | - | ✓ | ✓ | ✓ | ✓ | - | ✓ | - | ✓ | - |
| Make it warmer | ✓ | - | ✓ | - | ✓ | - | - | - | ✓ | - |
| Open the box | ✓ | - | ✓ | - | ✓ | - | ✓ | - | ✓ | ✓ |
| Open the website | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | - | ✓ | - |
| Reading a book | ✓ | - | ✓ | - | ✓ | - | ✓ | - | ✓ | - |
| Show me the money | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | - |
| Turn off the computer | ✓ | ✓ | ✓ | ✓ | ✓ | - | ✓ | - | ✓ | - |
| Turn on bluetooth | ✓ | - | ✓ | - | ✓ | - | ✓ | - | ✓ | - |
| Total | 19/20 | 10/20 | 20/20 | 6/20 | 20/20 | 3/20 | 19/20 | 4/20 | 20/20 | 7/20 |

The terms "Echo" and "Cortana" refer to OTA attacks on Echo and Cortana, respectively, which is the same as Table 4. The terms "xxx-online" and "xxx-offline" represent the two variants of OTL attacks executed against various APIs, aligning with the information in Table 1. The checkmark symbol "✓" indicates the successful generation of examples and execution of attacks on the target API or VA. All examples adhere to a parameter setting of $\alpha = 0.3, \beta = 1, and \gamma = 1$. In cases of OTA attacks, $\eta = 0.1$ is also applied.

# Appendix A. Additional Evaluations

We expand the command dataset size to demonstrate our method's efficacy. Besides the ten commands and two sentences discussed in the main paper, we also produce adversarial audio samples employing an additional 20 commands. The results are shown in Table 13.

# Appendix B. Meta-Review

## B.1. Summary

The authors propose two attack schemes: ALIF-OTL and ALIF-OTA to generate adversarial black-box attack pipelines based on the linguistic feature space of TTS. The authors propose over-the-line and over-the-air approaches and implement a comprehensive analysis against commercial ASRs, reaching more than 95% query efficiency improvement compared with the state-of-the-art.

## B.2. Scientific Contributions

- Provides a Valuable Step Forward in an Established Field

## B.3. Reasons for Acceptance

1) The paper identifies and uses linguistic structures present in Text-to-Speech algorithms as a means to improve the generation of adversarial audio for their attacks. This allows for the model during the training process to target audio features that human listeners are sensitive to, thus allowing for the model to be trained faster and with less resources.
2) The work highlights an interesting possible external source of a priori knowledge for future work to explore, namely the use of linguistic structures found in TTS.

## B.4. Noteworthy Concerns

1) The authors only provide a small, informal user study consisting of 20 volunteers who evaluated 32 audio samples. Given the attacker's goal in this work is to create audio samples that are incomprehensible to human listeners, but still comprehensive to machines,

this user study is insufficient. Additionally, several of the reviewers of this work found the audio samples easily intelligible. Thus, the overall incomprehensibility of the created audio samples cannot be determined at this time.

2) One of the main contributions of this work is the improved training efficiency of both techniques. Despite this, the paper uses a small number of audio samples, two sentences and ten commands[8], to evaluate the work. While this number is inline with previous work, we would expect that with efficiency improvements we would see larger datasets used for evaluation.
3) The real-world attack scenario of this work is lacking. In Section 7 the authors evaluate the performance of the OTA attack and find that the attack success drops significantly past 15 cm (less than 50% at 15 cm for certain ASRs). Additionally, on their website and in their rebuttal the authors detail the attack "performs well at a SNR of $\sim$20[dB]." In the context of acoustics, this is the difference between a bedroom and a restaurant[9]. This detail gives a possible explanation to the performance loss seen as the attack distance increased since acoustic energy reduces with the inverse square law. These aspects of the paper bring into question the deployability of the proposed attacks.
4) Additional concerns were raised over the selection and effects of the different parameters used in the attacks. An ablation study was provided, however, it focused on evaluated human comprehension, not attack success.

## Appendix C. Response to the Meta-Review

We are grateful to our reviewers and the S&P 2024 program committee for accepting our paper, and we appreciate the effort put into summarizing and identifying the limitations of our work.

We would like to address some points raised in the meta-review:

**Comment 2** notes that the reviewers would like us to use larger command datasets for generating attack audio samples.

We have taken this into consideration and increased our dataset size to include an additional 20 commands, making a total of 32 commands and sentences. These commands were carefully selected from common user interactions, thereby representing significant security threats. The successful generation of attack audio samples from this corpus affirms the effectiveness of our method, and we believe that further increase would not yield additional technical challenges. Thus, we believe that the current size is adequate for the purpose of our study.

**Comment 3** points out the lack of real-world attack scenarios.

We would like to note that the Signal-to-Noise Ratio (SNR) and attack distance requirements of our study align with related work. We plan to enhance the practicality of our method in our future research.

**Comment 4** notes that the ablation study focuses on evaluating human comprehension rather than attack success.

Our attack algorithm can ensure successful attacks under various ablation conditions, making this aspect less crucial for evaluation in our study. Therefore, we did not ablate components of the attack pipeline to study the impact on attack success.

8. The authors added an additional 20 commands in the Appendix.

9. Britannica - https://www.britannica.com/science/sound-physics/The-decibel-scale