SMACK: Semantically Meaningful Adversarial Audio Attack

Zhiyuan Yu, Yuanhaur Chang, Ning Zhang
Washington University in St. Louis

Chaowei Xiao
Arizona State University

Abstract

Voice controllable systems rely on speech recognition and speaker identification as the key enabling technologies. While they bring revolutionary changes to our daily lives, their security has become a growing concern. Existing work has demonstrated the feasibility of using maliciously crafted perturbations to manipulate speech or speaker recognition. Although these attacks vary in targets and techniques, they all require the addition of noise perturbations. While these perturbations are generally restricted to L_p-bounded neighborhood, the added noises inevitably leave unnatural traces recognizable by humans, and can be used for defense. To address this limitation, we introduce a new class of adversarial audio attack, named **S**emantically **M**eaningful Adversarial **A**udio Atta**CK** (SMACK), where the inherent speech attributes (such as prosody) are modified such that they still semantically represent the same speech and preserve the speech quality. The efficacy of SMACK was evaluated against five transcription systems and two speaker recognition systems in a black-box manner. By manipulating semantic attributes, our adversarial audio examples are capable of evading the state-of-the-art defenses, with better speech naturalness compared to traditional L_p-bounded attacks in the human perceptual study.

1 Introduction

The advent of voice recognition is promoting the rapid growth of voice controllable systems (VCS). The application of VCS is ubiquitous with some being security-sensitive, ranging from making phone calls to controlling household security systems. It is reported that over 120 million people in the United States use VCS, and the number is expected to increase to 130.1 million in 2025 [63].

Adversarial Attacks on VCS: On the other hand, the key functionalities of VCS - automatic speech recognition (ASR) and speaker recognition (SR) are driven by deep neural networks (DNNs), which have been shown to be vulnerable to adversarial examples. Existing work in the field of adversarial audio attacks focuses on crafting adversarial examples

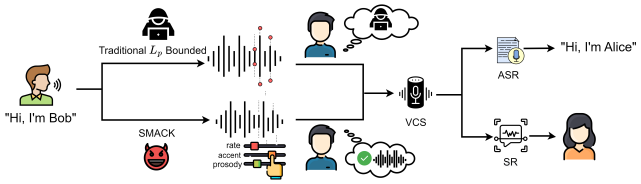


Figure 1: SMACK compared to traditional adversarial audio examples against speech and speaker recognition.

by adding small perturbations to the original audio, such that they will be interpreted differently by humans and recognition algorithms. The perturbed audio can be used to mislead ASR for malicious command injection [8, 16, 20, 60, 79], or confound SR to misidentify attackers as enrolled users [18,41,74]. In these attacks, adversarial example generation is modeled as a constrained optimization problem with restrictions on the magnitude of the perturbations. While existing attacks vary in methodologies and targets, the perturbation is often optimized based on an L_p-bounded norm, and the artificially introduced noises often inevitably leave distinguishable artifacts for both humans and algorithms.

Why Semantic Perturbations: To address this limitation, we propose semantic perturbations to modify the inherent speech attributes, as shown in Figure 1. In contrast to traditional perturbations that can be introduced with the finest granularity (pixels in the image domain and sample points in the audio domain), semantic-preserving perturbations impose additional constraints on the search space to achieve better naturalness. An example of semantic adversarial examples in the image domain against face verification systems is to manipulate inherent attributes (e.g., smile, mustache) to maximize the preservation of the facial features [56]. Similarly, we propose to only perturb the inherent attributes of speech to maximize the preservation of speech quality (naturalness). Particularly, we explore the manipulation of prosody, a representative semantic attribute, to generate adversarial audio examples in this paper. From the security perspective, such naturalness has the potential to significantly improve the

stealthiness of the adversarial examples (see human perceptual study in Section 9). Furthermore, semantic perturbations can better evade existing detection mechanisms that check for artifacts from L_p-bounded adversarial perturbations (see evaluation in Section 8).

Our Attack: To gain a deeper understanding on the feasibility and limitation of adversarial semantic perturbation in audio examples, we present our exploration of a new class of adversarial attack, **S**emantically **M**eaningful Adversarial **A**udio Atta**CK** (SMACK). Similar to how adversarial perturbation in the form of eyeglasses constrains the modification to pixels surrounding the eye in the image domain, semantic preserving adversarial perturbation on speech also has additional constraints. Compared to introducing perturbation to each sampling point with independent optimization, semantically modifying prosody restricts changes to pitch and speech rate that are specified by numerous sample points. These unique characteristics of audio motivate us to investigate temporal perturbations instead of spatial semantic perturbations in the image domain [14, 32, 35, 56]. SMACK leverages an adapted generative model that enables prosody control with a vector in the continuous space. We further develop a novel algorithm incorporating our proposed expanded genetic algorithm and gradient estimation to optimize complex prosody features.

Technical Challenges: The core technical challenges behind SMACK lie in modeling and optimizing semantic features (i.e., prosody) in the context of adversarial examples. First, prosody is a composite attribute that is often described by a combination of several speech characteristics (e.g., speech rate and fundamental frequency), and the effective modeling and natural manipulation of prosody remain an open research problem [31]. To address this challenge, we propose an adapted generative model to enable fine-grained control of prosody. To further preserve the original voice, the generative model also takes the original audio as input on which prosody modification should be performed. Second, semantic attributes in the audio domain are temporal features by nature, and such characteristics can be of varying complexity depending on speech content. This results in a variable-length prosody control vector. We handle this significantly larger search space by introducing a two-stage optimization algorithm consisting of our adapted genetic algorithm and gradient estimation, where a customized genetic operator *insertion and deletion* is introduced to enable variable-length prosody vector optimization. Third, the black-box assumption and additional constraint due to semantic preservation in SMACK present new challenges in navigating the solution space for the adversarial perturbation. Building on the observation that transcription algorithms are more inclined to be confounded with words of similar pronunciation, we propose to incorporate a new adversarial term evaluating phonemic similarity. Furthermore, SMACK also leverages the confidence score in SR to iteratively update estimation on the speaker verification threshold, significantly reducing the number of queries.

Evaluation and Findings: We demonstrate the feasibility and practicality of this new category of adversarial audio examples by evaluating against five ASR systems and two SR systems in a targeted black-box setting, both over-the-line and over-the-air. Our attack was successful in both misleading ASR transcription of commercial products and confounding various speaker identification tasks, achieving a mean success rate of 84.9% and 99.2% respectively. Furthermore, we showcased the physical robustness of semantic adversarial examples by delivering attacks in the air with different noise levels and distances, where we observed that semantic adversarial audio examples gain unique advantages compared to L_p-bounded methods, due to their decoupling from the traditional fine-grained perturbations on each sample points. To understand human perceptions of semantic adversarial audio, we further conducted a user discernability study. Among a total of 168 participants, the majority of them appreciated the fidelity and naturalness of semantic audio examples, as opposed to the two typical L_p-bounded adversarial attacks [17, 18].

Contributions: Our contributions are outlined as follows:

- We introduce *semantic adversarial audio examples*, where speech is perturbed via manipulation of inherent attributes while semantically preserving the original content. Using prosody as the representative semantic attribute, we propose SMACK that generates adversarial audio examples by perturbing prosody features.
- We model temporal semantic attributes by adapting a generative model as a manifold of semantic transformations on speech audio. In addition, we propose a two-stage optimization mechanism, which includes a novel genetic operator and gradient estimation scheme to effectively optimize the variable-length attribute vector that controls the transformation. A new transcription-based loss function and a bound-based threshold estimation method are proposed to attack ASR systems and SR systems.
- We evaluated SMACK against a total of seven state-of-the-art voice systems (five ASR systems and two SR systems) including three real-world commercial products, with successful attacks in both over-the-line and over-the-air attack scenarios. Besides, our attack shows stealthiness to evade state-of-the-art defenses. We further conducted a comprehensive user study on 168 participants. The results indicated that our adversarial examples appear more natural to humans as compared to traditional ones.

2 Existing Work on Adversarial Audio

Existing adversarial audio attacks can be categorized based on the attack targets - ASR systems and SR systems.

Attack ASR: This line of research primarily focuses on crafting adversarial examples that are transcribed differently by machines and humans. Carlini et al. [16] proposed the hidden voice command attack to mislead GMM-based recognition models, where they designed obfuscated audio fragments that can be understood by speech recognition algorithms but remain unintelligible to humans. Following this work, Yuan et al. [79] studied attacks targeting DNN-based speech recognition systems, where they perturb songs to deliver adversarial commands. In the same vein, Carlini et al. [17] optimized adversarial audio examples targeting DeepSpeech [29] with gradient descent based on the CTC loss. However, these attacks rely on white-box knowledge of the targets. To address this limitation, Abdullah et al. [8] developed a model-agnostic attack that exploits signal processing algorithms prior to the DNN-based classification stage. To improve the practicality, Chen et al. [20] studied mechanisms to enhance the survival of the adversarial audio examples in over-the-air transmission. To improve the stealthiness of adversarial audio examples, Schönherr et al. [60] adopted a psychoacoustic model lowering the signal guided by human hearing thresholds to avoid human perception. SMACK explores perturbation mechanisms that preserve semantics yet improve stealthiness.

Attack SR: There are also attempts that leverage adversarial audio to attack speaker recognition systems [18, 40, 41, 74]. Kreuk et al. [40] were the first to develop adversarial attacks on end-to-end speaker verification systems. However, they targeted an end-to-end binary system and the attack was delivered over-the-line. Following this work, Li et al. [41] proposed attacks on the xvector system, and further improved practicality by enabling over-the-air attack delivery with modeled impulse response of the room. Xie et al. [74] also developed over-the-air attacks on xvector systems, but with the additional advantage of efficiency enabled by the universal perturbations that can be directly added to arbitrary utterance. More recently, Chen et al. [18] proposed FakeBob attack, where a threshold estimation scheme was first proposed to improve attack effectiveness. SMACK leverages semantic perturbation which requires a new mechanism for threshold estimation.

3 Semantic Adversarial Examples

Semantic Adversarial Attacks in the Image Domain: The concept of semantic attribute was first introduced in the image domain as a method for data augmentation to mitigate overfitting [58, 70]. The key intuition is to translate a data sample in the linearized feature space along semantic directions, resulting in a feature representation corresponding to another sample with the same class identity but different semantics. Examples of semantic attributes include eyeglasses, beard, as well as changing facial expression and makeup. Semantic adversarial attack on images was later developed to manipulate only higher-level features (e.g., adding eyeglasses to faces) to deceive image recognition algorithms [14, 32, 35, 56].

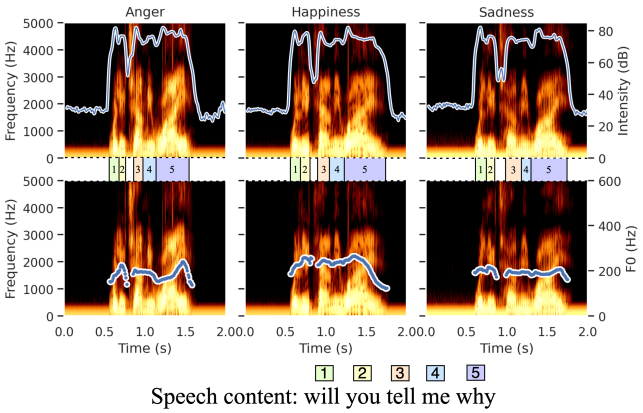


Figure 2: The intensity, word duration, and fundamental frequency of three types of prosody applied to the same speech.

While semantic-preserving visual modifications are known to be effective, semantically meaningful perturbation of audio is less understood. Image and audio exhibit distinct semantic principles: semantic attributes in images (e.g., eyeglasses) are represented by pixels that entail spatial correlations to hierarchical object associations and color descriptions, while semantic properties of audio waveforms (e.g., prosody) take the format of time sequence that possess temporal dependency. As spatial semantic perturbations in images can be leveraged to attack recognition models, it remains unknown if temporal semantic modifications can have a similar effect.

Semantic Attributes in the Audio Domain: Though a few existing works in the audio domain proposed the concept of semantic attributes including prosody [36], accent [25], and word order [66] for data augmentation, the feasibility of using them for stealthy adversarial example generation has not been explored yet. For a semantic feature to be used in stealthy adversarial attacks, it has to meet three key requirements.

Inherent Attributes: In the audio domain, inherent attributes refer to the attributes that should not be dependent on speech content or speaker identity but instead should widely exist in almost all speech. Typical inherent attributes include emotion, speech rate, accent, prosody, etc.

Identity/Content Preserving: Another requirement lies in the preservation of the original label from the human perspective. For ASR systems, the content (i.e., transcripts) of the semantic adversarial audio example should be the same as the original for humans. Note that we do not consider substituting a word with a synonym as satisfying this requirement. For SR systems, the identity information of the semantic adversarial audio example and the original audio should be the same.

Naturalness: Naturalness (or realism) is also an important requirement. Besides preserving the original identity, we also emphasize naturalness in semantic perturbations. Intuitively, although the semantic attributes are the inherent properties of human speech, modifying them to a large degree will introduce artifacts that make the speech sounds abnormal to

human. For instance, speech rate cannot be excessively fast.

Prosody as Semantic Attribute of Speech: In this work, we leverage prosody as the representative semantic attribute. Prosody is often described as intonation and rhythm – the musical qualities and melodic aspect of speech [71]. It encompasses multiple characteristics of a speech, including pitch contour or intonation of an utterance, the length of a syllable, the loudness of a word, etc [75]. Compared to the pixel-level spatial modification in the image domain, perturbations in the audio domain are often temporal modifications. Figure 2 shows the quantitative analysis of prosody on the three speech clips sourced from the SAVEE dataset [33]. These three audio clips contain the same speech content ("will you tell me why") but are spoken in three types of prosody: anger, happiness, and sadness. The three figures at the top depict the spectrum and intensity of the speech, where intensity is measured in decibels (dB) at each moment. The bottom three figures present the fundamental frequency (F0), which is the frequency at which the vocal cords vibrate in voiced sounds. In psychoacoustic models, the F0 frequency is generally perceived as the acoustic pitch of a sound by humans. In addition, the duration of each word is indicated in the middle and represents the rate of speech. Speech with different prosody is observed to have varying time duration and, consequently, varying speech rates. For example, word 3 ("tell") in anger has a duration of 167 milliseconds, which is significantly shorter than word 3 in happiness (186 milliseconds) and sadness (225 milliseconds). Therefore, prosody is a complex attribute described by a variety of frequency and time domain descriptors.

Prosody was selected because it meets the three requirements for semantic adversarial example generation. First, prosody is one of the most important inherent features of human speech that has been widely studied in a variety of domains, including emotion recognition [44], speech synthesis [61], and linguistic research [23, 69]. Second, unlike other language-specific characteristics like accent and word order, prosody is not limited to speech content and is universally applicable. Third, prosody can be represented by fine-grained features in each frame, and semantically restricting the manipulation of prosody can preserve content and naturalness. To achieve coherent manipulation of prosody, we build on top of the existing work in generative model [27].

Relevance to Other Attacks Using Generative Models: Generative models are widely used in different applications including adversarial attacks [21, 73]. One related concept is *Audio DeepFake*, where the attacker aims to impersonate the victim by generating speech recordings in the voice of the target speaker [21]. While DeepFake and SMACK share similarities, such as using generative models and targeting VCS, they differ in attack goals and techniques. First, the synthetic speech from DeepFake is designed to sound like the victim for both humans and computer systems [34, 53, 67]. However, SMACK follows the line of research on adversarial audio generation and aims to create audio examples that are imperceptible, which do not sound like the victim at all to humans and only mislead the recognition algorithms. Second, though both use generative models, DeepFake leverages them to learn and mimic the characteristics of the victim's speech, which often requires non-trivial efforts in collecting the victim's speech for training [50]. In contrast, SMACK uses generative models to create perturbations on the original audio without requiring the victim's voice. Since SMACK aims to mislead the recognition models, it relies on a designed multi-objective function incorporating adversarial loss and human perceptibility, while DeepFake usually does not include adversarial loss. We further evaluated SMACK against a DeepFake defense to raise awareness of the new threats introduced by SMACK and motivate new defenses (Section 8.4).

4 Threat Model

Attack Goal: The adversary aims to conduct *targeted* attacks against ASR or/and SR systems. In the attack against ASR, the adversarial audio shall be transcribed to a different word/sentence compared to what the human would interpret. For the attack on SR systems, the attacker aims to craft and play the adversarial audio, such that it is misrecognized by the SR algorithms as coming from one of the enrolled speakers.

Assumption on Attacker's Knowledge: We assume that the adversary only has *black-box* knowledge (access to neither architecture nor parameters of the targeted models) with limited access to the audio sources [8, 9, 18, 68]. For attacks on ASR, we assume the adversary only has access to the transcription, thus it is considered a hard-label black-box attack [78, 83]. For attacks on SR, we follow the same setting from the most recent attack [18] and assume that the attacker has access to the final result (accept/reject) and confidence score. We also assume the attacker does not have access to the voice samples of the enrolled user in the system, therefore he/she cannot use DeepFake to create audio examples that sound like the victim.

Assumption on Target Systems: Similar to [60], we assume that the ASR and SR systems are configured to give the best possible recognition rate, and the recognition models remain unchanged over time. We also make the same assumption as [22] that the services provided by commercial API and VCS are similar for the same platform.

Adversarial Example Generation and Delivery: Similar to existing work [8, 9, 16, 68, 79], adversarial audio examples are generated ahead of time and can be delivered to the target either *over-the-line* to an API or played *over-the-air* to an ASR/SR device. We assume the attacker can play the entire adversarial audio example rather than part of it.

5 Overview of SMACK

SMACK aims to manipulate the prosody of the original audio to cause misclassification of ASR and SR systems. There are

three key technical challenges.

C1. Prosody is a complex attribute incorporating multiple features, including speech rate, intonation, loudness, lengths of syllables, etc. Such complexity brings the challenge of effectively modeling and controlling it in a given speech. To address it, we designed an adapted generative model for frame-wise fine-grained prosody control (Section 5.1). Prosody involves not only the temporal features of the speech, but also the interactions between the feature vectors. To tackle the lack of concrete mathematical formulation and enable accurate manipulation of speech prosody, we adapt the latest generative model for speech. To further enable preservation of the original voice, we modify the generative model to include the original audio on which prosody modification should occur, allowing modification of voice rather than direct synthesis.

C2. Under the existing approach of prosody manipulation via generative models, the prosody vector is often of a fixed dimension. However, when the speech has different lengths and wording, it requires an appropriate level of manipulation granularity governed by the dimensions of the prosody control vector. To ensure the naturalness of the generated adversarial example, we propose a two-stage optimization framework coupled with a novel *InsDel* operator (Section 5.2) to enable variable-length search of prosody vector.

C3. Unlike previous black-box attacks on ASR, SMACK is based on prosody manipulation, which sets an additional constraint on how perturbations can be added. To facilitate a more accurate calculation of the gradient, we developed a new loss function based on phonemes. To enable a more effective attack against SR systems, we designed an algorithm that leverages the result and confidence score of each query to iteratively estimate the threshold value used in speaker recognition.

5.1 Modeling and Controlling Prosody

Due to the complexity of prosody, its control remains an open research problem [31]. Without concrete analytical solutions, we build on top of existing advances in generative models [19] to enable fine-grained manipulation of prosody, where it is adapted to serve as a transformation function for semantic editing on audio examples. Figure 3 shows the architecture of our generative model design with four major components: the content network, the prosody network, the prosody-content cross-attention module, and the decoder network.

Content Network: It uses the text of the original speech to constrain the content of reconstructed speech (i.e., semantic adversarial audio examples) to be identical to the original.

Prosody Network: The prosody of the generated audio is defined by two components. (1) *Global prosody* represents the tone of voice that is unique to each speaker, which therefore characterizes the speaker identity. It is represented by global prosody embedding extracted from the original audio via Wav2Vec 2.0 [13] feature extractor. To satisfy the *identity preserving* requirement (Section 3), the global prosody embedding is kept unchanged in SMACK. (2) *Local prosody* focuses on frame-level prosody features, which result from matrix multiplication of the variable-length input prosody vector (i.e., the vector that the attacker aims to optimize) and a trainable prosody token codebook. The input prosody vector is of variable length, because its richness (i.e., the dimension of the prosody vector p) is configurable and dependent on the speech content (e.g., the length of the speech content). The prosody representation is obtained by broadcast-adding global and local prosody embedding across the time dimension.

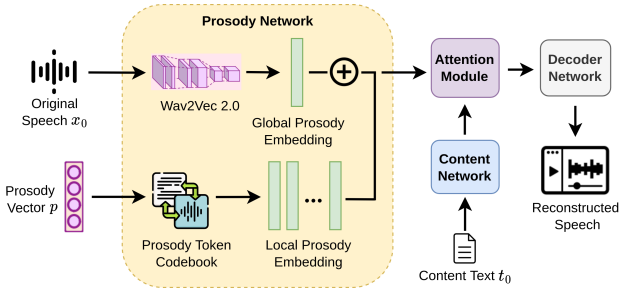


Figure 3: Overall structure of the adapted generative model

Attention Module and Decoder: It aligns the information from *context network* and *prosody network* to get the fused embedding, which will be used to reconstruct the speech audio via the *decoder network* (i.e., WaveGlow vocoder [55]). Incorporating the above elements, we denote the generated semantic example x^* as $x^* = G(t_0, x_0, p)$, where t_0 represents the original content text, x_0 is the original audio, and p is the prosody vector that controls the frame-wise prosody. Within this context, generating semantically meaningful adversarial audio examples is equivalent to optimizing the prosody vector p, which controls the adversarial manipulation. In the next section, we introduce how to optimize p.

5.2 Prosody Optimization Mechanism

As discussed in challenge **C2**, the variable length of the prosody vector p and black-box settings present unique challenges for optimization. To address these challenges, we propose a two-stage optimization mechanism. It consists of an adapted genetic algorithm (AGA) and a gradient estimation scheme (ES), namely AGA-ES algorithm. The first stage (AGA) performs a global search, which is designed to reduce the search space by optimizing both the length and values of the prosody vector. The second stage (ES) serves as a fine-grained local optimization scheme that refines the prosody towards the adversarial goal and speech naturalness. Notably, the proposed AGA-ES framework can be adapted to attack both ASR and SR systems with different loss functions. This section describes the framework, and loss functions adapted to ASR and SR will be defined in Section 6 and 7. For more

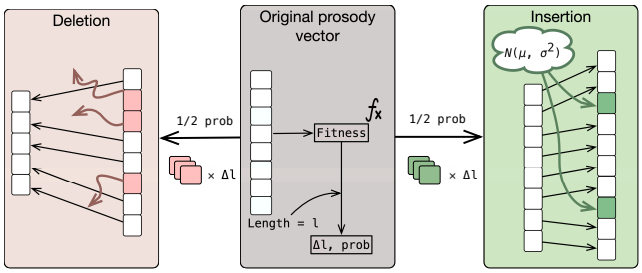


Figure 4: Underlying concept of InsDel operator

implementation details please see the project website[1].

Adapted Genetic Algorithm: Genetic algorithm is a search-based optimization technique based on the principles of genetics and natural selection. It works by applying genetic operators including selection, mutation, and crossover, and has been demonstrated to be competitive with deep reinforcement learning across multiple application domains [65]. In SMACK, two unique advantages make it well-suited for our optimization problem. First, the variable length and unbounded value of the prosody vector result in a large search space. Genetic algorithm has proven effective in searching over a large domain [37]. Second, our attack is developed under a black-box setting. The genetic algorithm works well in this setting since it does not require gradient information.

In SMACK, the potential solutions of p are modeled as chromosomes and thus each gene represents a value in the prosody vector. However, the standard genetic algorithm cannot be directly applied to SMACK because the search space is rigid, meaning that the length of the chromosome is fixed in genetic operators. Building on top of the concept of variable-length genetic algorithm [38], we introduce a new operator named *insertion and deletion* (InsDel), which is prevalent in biological chromosomes [46]. Figure 4 depicts the underlying concept of this procedure. As described in Alg. 1, a portion of genes will be added to or removed from the original chromosome in accordance with an adaptive probability. Down to the design details, there are five factors that characterize the effectiveness of the operator. First, we design the probability of InsDel to be reversely related to the fitness value and score improvements, thus achieving a balance between convergence speed and fitness improvement. Second, the insert and delete are designed to happen with equal probability to introduce unbiased length variations. Third, the inserted/deleted genes were randomly selected to introduce additional value variations. Fourth, the edit length is devised to be dependent on the iteration number and the original prosody control vector to improve naturalness and run time. Lastly, the inserted values are sampled from the distribution of the original prosody vector for naturalness.

(1) Operation probability (line 5 in Alg. 1). We set the probability to be dependent on fitness values, aiming to introduce variations to enhance chromosomes with poor performance. Intuitively, chromosomes with lower scores require more alterations, whereas those with higher scores may have attained optimal sizes. Inspired by momentum update [68], a term that is inversely proportional to changes in fitness is also introduced. As such, the probability increases if the fitness merely changes, thereby facilitating the InsDel operation and assisting in the avoidance of local optimal. As a result, the probability is designed as a weighted term balancing the convergence speed and fitness improvement: $prob_i^k = \alpha \times \frac{fit(i,k)}{\sum_{i \in \mathbf{I}} fit(i,k)} + \frac{\beta}{|fit(i,k) - fit(i,k-1)|}$, where $prob_i^k$ and $fit(i,k)$ are the operation probability and fitness for the i^{th} chromosome in the k^{th} iteration, and α and β are the weights.

(2) Insertion vs. deletion (line 8, 14 in Alg. 1). Due to the uninterpretable nature of the prosody vector, the fitness value alone is insufficient to determine whether insert or delete operations should be performed. To introduce unbiased variations, SMACK ensures that the two actions have equal probability.

(3) Operation location (line 7, 12, and 15 in Alg. 1). Genes are inserted or deleted at multiple positions rather than a fixed location (e.g., at the end of each chromosome). This design introduces additional value variations.

(4) Edit length (line 6 in Alg. 1). The number of inserted and deleted genes characterizes the efficacy of the operation. The design of this parameter needs to consider two aspects. First, the number of altered genes $\triangle l_0$ should be proportional to the original length of the chromosome l_0 to preserve the speech naturalness. Second, it should be adaptive to the optimization procedure in order to strike a balance between the efficiency and effectiveness of searching. Therefore, we develop an adaptive edit length $\triangle l$ as: $\triangle l = \lceil pr \times e^{-iter/c} \times l_0 \rceil$,

Algorithm 1: Insertion and deletion (InsDel) operator

Input: Original population pop
Hyperparameters α, β, pr
Current iteration of optimization iter
Output: New population newPop

1: **function** *Fitness*(individual)
2: **return** fitness value of individual
3: **for** i in *range*(popSize) **do**
4: fitness$[i] \leftarrow$ *Fitness*(pop$[i]$)
5: prob$[i] \leftarrow \alpha \frac{\text{fitness}[i]}{\sum_{i=1}^{\text{popSize}} \text{fitness}[i]} + \beta \times |\frac{1}{fitness[i] - preFitness[i]}|$
6: $\triangle l[i] \leftarrow \lceil pr \cdot e^{-iter/c} \cdot \text{chromoSize}[i] \rceil$
7: locations$[i] \leftarrow$ *Sample*$(0, \text{chromoSize}[i] - 1, \triangle l[i])$
8: **if** $uniform(0,1) \leq 0.5\text{prob}[i]$ **then**
9: **for** gene in range($\triangle l$) **do**
10: calculate μ, σ from $l[i]$
11: draw gene $\sim \mathcal{N}(\mu, \sigma^2)$
12: insert gene to pop$[i]$ at locations$[i]$
13: **end for**
14: **else if** $0.5\text{prob}[i] < uniform(0,1) \leq \text{prob}[i]$ **then**
15: remove genes at location$[i]$ from pop$[i]$
16: **end if**
17: newPop.*append*(pop$[i]$)
18: **end for**
19: **return** newPop

[1] https://semanticaudioattack.github.io/

where pr is the percentage, $iter$ is the current number of iterations, c is a constant that controls the decay.

(5) Insertion values (line 8-12 in Alg. 1). In contrast to *deletion*, which merely eliminates a portion of genes from the original chromosome (line 15 in Alg. 1), *insertion* adds genes whose values have yet to be defined. To preserve the general naturalness of the prosody, we construct the values to be sampled from the original distribution (line 11 in Alg. 1).

Gradient Estimation Scheme: Even though the adapted genetic algorithm works well with large-space exploration, it becomes less effective when approaching the optimal solution. Therefore, we complement the optimization by tuning the prosody guided by gradient directions. In this phase, we do not change the prosody vector size and only refine its values.

As we consider a black-box setting where the gradient of the target model is not accessible to the attacker, we approximate the gradient for i^{th} iteration via a gradient estimation scheme named natural evolution strategy (NES) [72]. It works by adding noises to the prosody vector along various directions and approximating the gradient as the mean direction weighted by loss values. We first create the K Gaussian noises $\mathbf{u}_k \sim \mathcal{N}(0, I)$, with which the gradient is calculated as:

$$\nabla_p \mathcal{L}(p_i) = \frac{1}{\sigma K} \sum_{k=1}^{K} \mathcal{L}(G(t_0, x_0, p_i + \sigma \mathbf{u}_k), t) \times \mathbf{u}_k \quad (1)$$

where t is the attack target and $\mathcal{L}$ is the task-dependent loss function that evaluates the distance between the generated audio $G(t_0, x_0, p_i + \sigma \mathbf{u}_k)$ and target t.

The direction of the estimated gradient will be multiplied with a learning rate η and added to the optimized vector:

$$p_i = p_{i-1} + \eta \cdot sign(\nabla_p \mathcal{L}(p_{i-1})) \quad (2)$$

More details regarding the loss function, fitness heuristics, and the newly proposed threshold estimation technique for SR will be further discussed in Section 6 and 7.

Algorithm 2: SMACK against black-box ASR

Input: Prosody control vector p; Original speech x_0 with content t_0; Target transcription t; Maximum number of iterations N
Output: Adversarial audio example x_{adv}

pop $\leftarrow [p] *$ popSize
function $Fitness$(pop, t)
  popAudio $\leftarrow G(t_0, x_0,$ pop$)$
  fitness $\leftarrow L_{ASR}(ASR($popAudio$), t) + \alpha Q($popAudio$)$
  return fitness
for i in range(N) **do**
  fitness $\leftarrow Fitness($pop, $t)$
  bestPop $\leftarrow$ pop$[Argmax($fitness$)]$
  if $Fitness($bestPop, $t) <$ Threshold **then**
    topPop $\leftarrow Select($pop, fitness$)$
    for j in range(childSize) **do**
      newPop $\leftarrow InsDel($topPop$)$
      newPop $\leftarrow Mutate($newPop$)$
      newPop $\leftarrow Crossover($newPop$)$
    end for
  else
    bestFitness $\leftarrow Fitness($bestPop, $t)$
    probePop $\leftarrow$ [bestPop] $* K + \sigma \boldsymbol{u}$
    gradEst $\leftarrow \frac{1}{\sigma K} \sum Fitness($probePop, $t) \times \mathbf{u}$
    bestPop $\leftarrow$ bestPop $+ \eta \cdot sign($gradEst$)$
  end if
end for
get x_{adv} generated by $G(t_0, x_0,$ bestPop$)$
return x_{adv}

6 Attack Speech Recognition Systems

6.1 Problem Formulation

We denote the target ASR system as $ASR(x)$, which takes an audio clip x as input and returns its transcript. The attacker aims to generate a semantic adversarial example x^* such that:

$$ASR(x^*) = t, \text{s.t. } x^* = G(t_0, x_0, p), t \neq t_0, H(x^*) = H(x_0) \quad (3)$$

where t is the attack target, and $H(\cdot)$ denotes a human that interprets the audio x into transcript as he/she perceives.

To achieve the attack goal shown in Eq. 3, the attacker aims to obtain the adversarial example x_{adv} by minimizing the multi-objective loss function consisting of two components:

$$\mathcal{L} = \text{argmin}_{x^*} \{L_{ASR}(ASR(x^*), t) + wQ(x^*)\}, \quad (4)$$

where L_{ASR} is the adversarial term that pushes the generated audio towards the adversarial transcript, $Q(\cdot)$ is introduced to measure the human interpretability of reconstructed audio, with w serving as the weighting factor to trade-off between the adversarial goal and attack stealthiness.

6.2 Attack Method

The complete algorithm for semantic adversarial example generation based on AGA-ES is presented in Alg. 2. Particularly, we construct a multi-objective loss function $\mathcal{L}$ adapted to SMACK, where we propose a new distance function that incorporates measurement of edit distance and pronunciation similarity, and also introduce the NISQA score [47] as the quality monitoring factor that preserves naturalness.

Adversarial Term with Levenshtein Distance: Ultimately, the adversarial term aims to measure the distance between the current transcript to the target phrase, which allows the algorithm to adjust optimization strategies accordingly. This term is generally derived from the loss values or probability distribution over the labels [17, 79]. However, such information is usually not available in practice as assumed in our threat model. Therefore, our adversarial term can only be built using hard labels of the final transcripts. Levenshtein distance [80], also known as edit distance, has been used to measure the difference between two sentences by computing the minimum cost of transforming one string into another with

a series of single-character edits (e.g., insertions, deletions, or substitutions). However, such measurement is insufficient in our attack, as it only takes character editing into consideration. For example, the edit distance of "*book*"→"*back*" is 2, which is the same as the edit distance of "*book*"→"*ok*". However, "*back*" and "*book*" have a higher degree similarity in their pronunciation, making it relatively easier to confound ASR into misclassification by manipulating speech attributes (i.e., prosody). The underlying reason for such disparity stems from the fact that Levenshtein distance treats each character in a word independently while disregarding the pronunciation factor, which generally depends on the character combinations and correlations.

Adversarial Term with Phonological Similarity: Recognizing the limitations of using Levenshtein distance as the sole metric in our attack, we turn to phonology and linguistics techniques in an effort to quantify distances based on the constituent phonemes, where a phoneme is a unit of sound that can distinguish between words. The phonemes can be obtained directly from transcripts via Grapheme-to-phoneme (G2P) transcriptors. Nonetheless, merely applying edit distance to phonemes has limitations for two reasons. First, vowels and consonants contribute differently to speech. Therefore, substituting a consonant with a vowel should be measured as a larger distance as oppose to replacing it with another vowel. Secondly, some phonemes display more similarities to each other, such that any substitution between these phonemes should incur fewer costs. Take ARPAbet [57] for example, the measured distance of the substitution operation on phonemes "*EM*"→"*EN*" should be at a smaller cost than the measured distance of "*EM*"→"*V*", even though all these phonemes are consonants.

Based on these insights, we propose an improved edit distance, where the edit cost for each phoneme pair is assigned with a customized weight. We follow the assumption in existing phonological research [30], which states that an edit operation is less costly when it occurs frequently between two alternative pronunciations of the same word. As such, we collect phonemes for different pronunciations of words extracted from CMU's Pronouncing Dictionary [15], and adopt Needleman-Wunsch algorithm [4] to align each pair of pronunciations with minimized edit distance. We then develop phonemic similarity $S(a,b)$ via statistical analysis of occurrence frequency: $S(a,b) = \frac{p(a,b)+p(b,a)}{p(a)+p(b)}$, where $p(a,b)$ represents the occurrence of substituting phoneme a with b, $p(a), p(b)$ are the occurrences of phoneme a and b respectively. As such, a larger $S(a,b)$ value indicates a higher similarity between phonemes a and b.

However, such similarity developed from alternative pronunciation focus on extracting similar phonemes, thus incomplete for phonemes pairs with significant disparities (e.g., a vowel and a consonant). We complement this by calculating phonemic differences based on Kondrak's ALINE cognate alignment system [39], where phonemes are described with a set of features **F** weighted on their salience - the features' impact on similarity. The difference is measured as:

$$D(a,b) = \sum_{f \in \mathbf{F}} df(a,b,f) \times salience_f + |V(a) - V(b)|$$
$$V(phoneme) = \begin{cases} v_{cst}, & \text{if } phoneme \text{ is a consonant,} \\ v_{vwl}, & \text{otherwise.} \end{cases} \quad (5)$$

where $df(a,b,f)$ calculates the difference between a and b given the feature f, v_{cst} and v_{vwl} are the heuristic values that determine the relative weights of consonant and vowel.

Lastly, we incorporate the aforementioned Levenshtein distance to avoid local minimum caused by words with very similar pronunciations but different transcripts.

Combining Levenshtein Distance and Phonological Similarity: As a result, the adversarial term can be formulated as:

$$L_{ASR} = w_1 \frac{Leven(t^*,t)}{Len(t^*)+Len(t)} + w_2 D(t_p^*, t_p) - w_3 S(t_p^*, t_p) \quad (6)$$

where t_p^* is the phoneme constitutions of the transcript t^*, $Len(\cdot)$ returns the length of the string input, w_1, w_2, w_3 are the factor weights.

Quality Assessment Term: We also introduce a term to the loss function to evaluate the quality and naturalness of the transformed speech. Traditional L_p norm and signal-to-noise ratio (SNR) are insufficient for our problem because they merely assess the noise level by computing deviations on sample points. Changing prosody could affect a number of values, but the audio example could still sound natural. For instance, a speech uttered in anger could differ significantly from one spoken in neutral when measured in L_p norm, but it remains semantically meaningful and sounds like human. Therefore, we employ and incorporate NISQA [47], a state-of-the-art DNN-based speech assessment system that quantify the overall quality and naturalness of speech on a scale from 1 to 5. This term is weighted and combined with the adversarial term described previously to form a complete loss function.

7 Attack Speaker Recognition Systems

7.1 Problem Formulation

Similar to Section 6.1, we denote the target SR system as $SR(x)$, which takes an audio clip x as input and returns the recognized speaker label. Different from ASR systems, speaker recognition algorithms generally include a threshold θ for decision making. For a SR system that hosts n enrolled speakers $\{s_1, s_2, ...s_n\}$, it can be modeled as:

$$SR(x) = \begin{cases} \underset{s_i \in G}{\text{argmax}}\ S_i(x), & \text{if } \max\limits_{s_i \in G} S_i(x) \geq \theta \\ Reject, & \text{otherwise.} \end{cases} \quad (7)$$

Algorithm 3: SMACK against black-box SR

Input: Initial range of threshold $[inf_0, sup_0]$
Target speaker s_t
Output: Adversarial audio example x_{adv}

pop $\leftarrow [p] *$ popSize
$\theta \leftarrow (inf_0 + sup_0)/2$
function S(pop)
  popAudio $\leftarrow G(t_0, x_0,$ pop$)$
  Get scores $S_i(x)$ for each $s_i \in G$
  return $\max_{s_i \in G} S_i(x)$
for i in range(N) **do**
  fitness $\leftarrow Fitness($pop$, t, \theta)$
  for j in range(popSize) **do**
    if $S($pop$) > \theta$ **and** $SR($pop$[j]) =$ Reject **then**
      inf $\leftarrow S($pop$[j])$
    else if $S($pop$) < \theta$ **and** $SR($pop$[j]) \neq$ Reject **then**
      sup $\leftarrow S($pop$[j])$
    end if
  end for
  $\theta \leftarrow (inf + sup)/2$
  bestPop $\leftarrow$ pop$[Argmax($fitness$)]$
  if $Fitness($bestPop$, t, \theta) <$ Threshold **then**
    Generate children with *InsDel*, *Mutate*, and *Crossover*
  else
    for k in range($\mathcal{K}$) **do**
      $\theta \leftarrow \frac{\mathcal{K}-k}{\mathcal{K}}$inf $+ \frac{k}{\mathcal{K}}$sup
      Estimate gradient and update bestPop
      if $S($bestPop$) > \theta$ **and** attack fail **then break**
    end for
  end if
end for
get x_{adv} generated by $G(t_0, x_0,$ bestPop$)$
return x_{adv}

where $S_i(x)$ is the calculated score for the speaker s_i. As such, the attack objective can be formulated as:

$$SR(x^*) = s_t, \text{s.t. } x^* = G(t_0, x_0, p), s_t \neq s_0, H(x^*) = H(x) \quad (8)$$

where s_0 and s_t are the original and targeted speakers respectively. Similar to Eq. 4, the adversarial example x_{adv} can be obtained by minimizing the loss function consisting of the adversarial term $L_{SR}(SR(x^*), t)$ and quality assessment term $Q(x^*)$: $x_{adv} = \text{argmin}_{x^*} L_{SR}(SR(x^*), t) + wQ(x^*)$, where the speech quality monitoring function $Q(\cdot)$ is the same as the one adopted in Eq. 4 (i.e., NISQA score). However, the adversarial term is significantly different due to the distinct working principles of SR, which will be detailed in the following.

7.2 Attack Method

While attacks against ASR and SR share similar problem formulations, the difference in how these two systems operate necessitates the use of distinct attack approaches. As modeled in Eq. 7, SR recognition tasks incorporate a unique threshold θ, and a speaker s_i will be identified only if its score $S_i(x)$ is the highest amongst all enrolled speakers and also exceeds θ. Therefore, the adversarial term can be written as: $L_{SR}(x) = \max\{\theta, \max_{s_i \in G \setminus \{t\}} S_i(x)\} - S_t(x)$.

However, the threshold θ is not available to the attacker given the black-box setting of our attack. Inspired by the concept of differential in mathematics, we approximate the threshold by iteratively shrinking its range. As illustrated by the complete algorithm outlined in Alg. 3, our proposed threshold estimation scheme runs concurrently with adversarial example optimization. More specifically, we set an initial range $[inf, sup]$ for θ, and the infimum and supremum are iteratively updated using query results and the corresponding confidence scores. Moreover, the exact value for θ that participates in the calculation for each iteration is calculated as the average of the infimum and supremum. (1) If the audio is rejected and the output maximum confidence score is higher than our estimated θ, it indicates that the current estimated θ is under-estimated and therefore the inf should be increased to the query output confidence score. (2) Similarly, if the audio is accepted and the maximum query-output confidence score is lower than θ, then θ is over-estimated and the sup should be decreased to the query output confidence score. The estimation of θ is updated iteratively in a bisection manner until reaching a small range (i.e., $sup - inf < \varepsilon$).

8 Evaluation

Target Systems: The target ASR systems include DeepSpeech 2 [11], CMU Sphinx [1], Google API [5], Microsoft Azure [6], and iFlytek [2]. The Pytorch implementation of DeepSpeech 2 [3] was trained on the LibriSpeech dataset [52], achieving a word error rate (WER) of 10.46 on its clean test set. The two evaluated SR systems include ivector-PLDA [24] and GMM-UBM [59]. The ivector and GMM systems are incorporated in Kaldi [54], both enrolled with 5 speakers (3 female and 2 male) from the LibriSpeech dataset. Although some systems like DeepSpeech 2 have publicly available network structure and model weights, our attack treated them as black-box and did not leverage such information.

Generative Model and Source Audio: Our generative model was trained on LibriSpeech dataset [52] of over 1000 hours of English human speech. For the attack against ASR, we selected the latest English version of Common Voice dataset [12] which contains 81085 sentences of human speech. In addition, we also utilized TIMIT [26] corpus as the audio source for attacking SR systems, which consists of 10 sentences spoken by each of 630 speakers from 8 major dialect regions of the United States. In the attack against SR, we selected four major regions and a male/female speaker for each region, a total of 8 speakers.

Hardware Devices: We conducted experiments on a server with Ubuntu 16.04 and RTX 3080 GPU with 12GB RAM. For over-the-air experiments, we evaluated with four different

Table 1: Results of transcription attacks across ASR platforms

ASR system	WER	MSR	NISQA	Queries
CMU Sphinx	14.0%	79.7%	3.23	1386
DeepSpeech 2	9.8%	88.3%	3.30	1275
Google API	13.6%	81.8%	3.46	1149
iFlyTek	7.3%	90.6%	3.20	908
Azure	12.4%	84.2%	3.34	1067

smartphones: iPhone 8, LG Q6, Nexus 5X, and Redmi 5A.

8.1 Attacking ASR

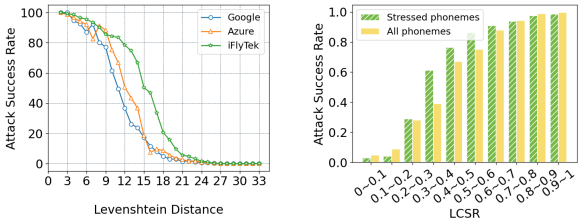
Experimental Design: We evaluated the targeted transcription attack against ASR with a strategy similar to [68]. We randomly selected 500 examples from the Common Voice dataset as source audio, each with 5 targeted phrases. Each target phrase was generated by replacing either part or entire of the sentences with other words of the same part-of-speech (e.g., noun, verb). As a result, the number of characters contained in the original sentence has a range of $[6, 36]$, and that of the generated target phrases is $[2, 49]$. This design enabled us to get source-target pairs at varying Levenshtein distances to understand feasibility limitations. To show the feasible threat in the real world, we also generated semantic adversarial examples that target six malicious commands.

Evaluation Metrics: We evaluated the attack capabilities based on the mean successful rate (MSR) and WER, where same transcript with the target counts as a success. We also measured the computational cost in terms of query number and resource occupation. The audio quality and naturalness of adversarial examples are assessed with the NISQA score.

ASR Attack Results: The results with randomly selected original speech and target phrases are shown in Table 1. It achieves a mean WER of 11.42 and success rate of 84.9% across the five ASR models. A well-trained transcription model generally has a similar WER (e.g., DeepSpeech 2 of 10.46). The crafted semantic adversarial examples exhibit good audio quality with a mean NISQA score of 3.31, which is comparable to normal human speech[2]. The mean number of queries required for a successful attack is 1157. We also randomly selected 100 out of these 2500 crafted examples and generated L_p-based adversarial examples using [68] with the same original audio and target phrases, where the mean number of queries is 9620. The query efficiency of our mechanism is likely due to the better guidance provided by phoneme similarity. Our algorithm occupies 2846 MB GPU memory. Each iteration takes 0.41s, and each example takes 474s on average. Lastly, these adversarial examples are later used to evaluate against defenses (Section 8.4).

The results of malicious commands are summarized in Table 2. SMACK achieves a mean success rate of 87.7%, with

[2]The NISQA of clean speech for another emotional speech dataset RAVDESS [43] is 3.37±0.82.



(a) (b)

Figure 5: Attack success rate with respect to edit distance and LCSR between original and target phrases.

the attack success rate of individual malicious command dependent on the target phrase. In addition, simpler transcription targets such as "open the door" present higher success rates. This is potentially because a longer and more complex target generally requires more perturbations, and therefore such optimal solution may not be easily achieved for some recognition models. We also observed that while the attack succeeds for all six commands, the audio quality degrades as the target phrase becomes longer due to the need for higher dimension of prosody manipulation vector. However, we include these samples in the user discernability study, and the results turned out that humans still perceive them as normal (Section 9).

Table 2: Results on malicious commands

Malicious Command	WER	MSR	NISQA
Airplane mode on	2.2%	93.3%	3.38
Open the door	0	100%	3.46
Turn off the light	5%	86.7%	3.70
Turn on wireless hot spot	2.7%	93.3%	3.31
Transfer evil a thousand dollars	9.3%	73.3%	3.14
Make a credit card payment	8%	80%	2.95

To understand the limitation, we conduct further experiments on attacks against three commercial ASR models with different edit distances between the original and target phrase. As shown in Figure 5(a), we observed that the attack success rate is generally higher when the target phrase spells similar to the original (i.e., edit distance is smaller), and vice versa. However, as we revealed in our design, Levenshtein distance falls short as it fails to incorporate the factor of pronunciation.

To further understand how the attack success rate is affected by phoneme similarities, we borrowed the concept of longest contiguous matching subsequence rate (LCSR) [45] from linguistics. LCS works by finding the longest subsequence that has the same order of elements given two sequences. Such metric in our context of phoneme sequences addresses both pronunciation (i.e., phoneme) and temporal (i.e., phoneme order) information. LCSR is calculated by dividing the length of LCS by the longest length of the two compared phoneme sequences, thereby measuring the similarity. Note that the LCSR for stressed phonemes is calculated as dividing the length of the matched stressed phonemes by the longest length

Table 3: Results of semantic attacks against SR systems

Attack Type	Task	GMM-UBM			ivector-PLDA		
		MSR	NISQA	Queries	MSR	NISQA	Queries
Intra-gender	CSI	100%	3.42	645	100%	3.39	753
	OSI	100%	3.46	574	95%	3.27	966
	SV	100%	3.35	713	100%	3.41	978
Inter-gender	CSI	100%	3.44	879	100%	3.21	868
	OSI	100%	3.46	794	100%	3.32	1034
	SV	100%	3.31	671	95%	3.40	1309

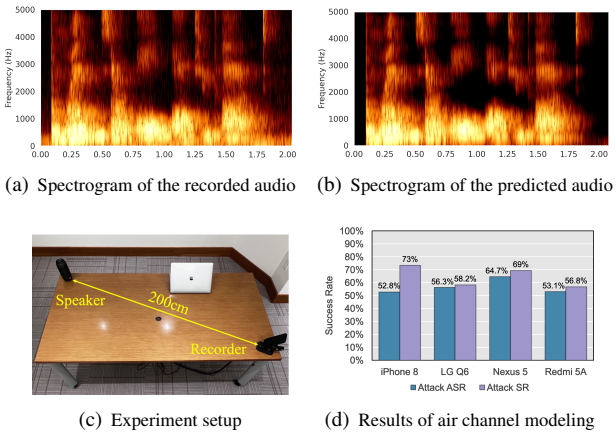
of the two stressed phoneme sequences. The results are shown in Figure 5(b). Overall, the attack success rate is generally higher when more phonemes are overlapped in the original and target phrases. We also observed that under the same LCSR, having overlapped stressed phonemes often leads to a higher success rate compared to general phonemes. This is because stressed phonemes play an important role in transcription for both humans and machines, while overlapping some other types of phonemes (e.g., consonants) are less helpful to execute the attack. Lastly, we observed that it is also possible to mistranscribe two phrases even though they have no similar phonemes (such as "*Like who*"→"*What's wrong*").

8.2 Attacking SR

Experimental Design: For each SR system, we consider three target tasks, open-set identification (OSI), close-set identification (CSI), and speaker verification (SV). The attacks were designed in a targeted manner, where each source audio belonging to a source speaker will be semantically perturbed to be identified as the target speaker enrolled in the system. As the SR system was enrolled with five speakers (3 female and 2 male), we designed two sets of speakers for adversarial attacks: inter-gender and intra-gender, where each source-target speaker pair is of different and same gender respectively. Note that all the source speakers are different from the five enrolled speakers. And four speech files with different utterances were randomly chosen for each source speaker, based on which the adversarial examples were crafted. As a comparison, we also reproduced the latest L_p-based attack [18] against SR systems following the same strategies.

Evaluation Metrics: The attack capability is evaluated based on the success rate, where a successful attack means the adversarial audio is attributed to the target speaker. We also measured queries number, computational costs, and audio quality similar to the experiments in attacking ASR systems.

SR Attack Results: Table 3 summarizes the results of attacks against SR systems. Our attack achieves a mean success rate of 99.2% with two failed attacks in attacking OSI and SV systems, and in both cases the adversarial audio examples were misclassified as another enrolled speaker instead of the targeted one. This was caused by the fact that the optimization exhausted the maximum iteration and the best results turned out to be getting accepted by SR as another speaker. While the attacker still achieved the malicious goal in both "failed" cases,



(a) Spectrogram of the recorded audio (b) Spectrogram of the predicted audio

(c) Experiment setup (d) Results of air channel modeling

Figure 6: Evaluation of our modeling on air channel

we still regard them as failures because the algorithm failed to find the optimal prosody vector. To understand if it was feasible to succeed in the attack, we increased the learning rate and the attack succeeded within 1000 queries. More importantly, while the gender difference between the original speaker and target speaker poses more challenges, we observed that the attack feasibility is not impacted by this factor, with both attack scenarios achieving a mean success rate of 99.2%. However, the mean number of queries indeed increased and indicated more efforts to find such optimal prosody. The mean number of queries made by our attack is 848, which occupies 3463 MB GPU memory. Each iteration involving threshold estimation and adversarial example generation takes 0.62s, and each example takes 525s on average. For comparison, the state-of-the-art black-box attack [18] achieves 99.0% success rate with an average number of queries at 1766. However, compared to white-box-attack in [74] that does not require query and appends a universal perturbation to create adversarial examples, query-based black-box methods like SMACK remains more time-consuming. For a more detailed comparison, please see Appendix A. Lastly, the mean NISQA value for the semantic adversarial examples is 3.37, which is similar to the results in attack ASR systems and indicates relatively good quality and naturalness of the semantic adversarial examples.

8.3 Over-the-air Attack

Over-the-air Attack: In addition to the over-the-line attack in which semantic adversarial examples were fed directly into ASR and SR systems, we also evaluated our attacks in a more realistic scenario where adversarial audio examples were delivered over-the-air. We first re-evaluated the attack performance by directly playing the raw audio in the air, with varying devices, distances, and noise levels. To improve the attack performance, we further employed air channel estimation via room impulse response (RIR) [62, 74] and evaluated the

Table 4: Attack success rate on over-the-air attacks

Attack		Distance (cm)				Noise (dB)		
		50	100	150	200	-10	-5	0
Raw	SMACK	56.8%	44.0%	38.1%	30.4%	41.7%	28.6%	12.3%
	[68]	23.6%	10.4%	4.7%	4.7%	29.5%	18.8%	9.6%
	[18]	33.7%	26.6%	16.3%	12.8%	26.8%	11.9%	4.7%
RIR	SMACK	64.7%	45.2%	39.6%	30.4%	43.1%	24.3%	16.3%
	[68]	30.2%	21.4%	9.6%	7.3%	31.6%	16.3%	8.2%
	[18]	40.3%	29.5%	17.9%	14.6%	28.3%	20.0%	11.9%

attacks. Our semantic adversarial audio examples were compared against the traditional L_p-bounded examples generated from the aforementioned existing work [18, 68].

Over-the-air Attack Results: We conducted experiments in a room with size $4.14 \times 5.38 \times 3.14$ meters. Figure 6(c) shows the experiment setup where we used a JBL Pulse 2 speaker to play audio clips, and each of the four phones was used as the microphone for recording. The phones were placed at different locations, and we simulated varying ambient noise strengths by playing white noises in the background under controlled volumes. The full results are shown in Table 4. We observe that semantic adversarial examples have a relatively higher over-the-air attack success rate compared to traditional adversarial audio examples [18,68]. We believe this is because semantic adversarial examples did not rely on fine-grained L_p-bounded perturbations that were susceptible to environmental distortions. On the other hand, both SMACK and traditional attacks can still be impacted by ambient noise, especially when the noise level is comparable to the adversarial audio.

We also measured the performance of our developed RIR model. Figure 6(b) shows an example of our prediction of the audio transmitted through the air, which is very close to the audio played in the air shown in Figure 6(a), indicating the effectiveness of our air-channel modeling. In order to quantitatively measure the effectiveness of our predicted air channel, we placed the speaker at a distance of 200cm from the microphone, which recorded a total of 117 regular speech clips for transcription and speaker recognition. As shown in Figure 6(d), 56.7% of these speech clips were correctly interpreted on ASR systems and 64.2% on SR systems across all speaker identification tasks. With this model, we observed a significant increase in success rate. We have also conducted experiments in a smaller room with size $1.91 \times 2.57 \times 2.36$ meters, and obtained similar results and insights.

Therefore, similar to other adversarial audio attacks, the performance of SMACK is affected when the distance is large (e.g., 2 meters) or the ambient noise is high (e.g., the devices are deployed in a room next to a street). However, the attacker could play the adversarial examples multiple times to improve the success rate, though at the risk of raising suspicion.

8.4 Evaluation against Defense

SMACK against Adversarial Audio Defense: We use the defense proposed in [77], which aims to detect adversarial audio examples based on temporal dependencies. We follow the official implementation and original setup, where a released DeepSpeech model is used for transcription. We randomly selected 100 semantic adversarial examples, and generated 100 corresponding L_p-based adversarial audio examples with the same original audio and adversarial transcriptions. The corresponding original audios were combined with adversarial examples to form the test set. The defense mechanism achieved 0.92 AUC on the generated L_p-based adversarial audio consistent with the original 0.936 AUC. However, it only achieved 0.534 AUC in detecting our semantic adversarial audio examples.

SMACK against DeepFake Defense: DeepFake detector aims to detect DeepFake-spoofed audio, another type of synthesized audio. Thus, we evaluate our method against Deepfake detector [82] to assess feasibility. We followed the official implementation and used their pre-trained model on the ASVspoof corpus [76], which contains 107 speakers (46 male, 61 female). We randomly selected 3 female and 3 male speakers to be enrolled in the SR systems. To avoid bias, we leveraged the benign audio clips contained in the ASVspoof speech corpus as the original audio. Specifically, we randomly selected 10 benign audio clips for each of the 3 female and 3 male speakers (different from the enrolled speakers), and generated a total of 120 semantic adversarial audio examples against SR following the same strategy. We also conducted post-processing on the sampling rate and frequency to match the expected input format of the defense. As a result, 94.2% (n=113) of our semantic adversarial examples were classified as bonafide by the detection system.

9 Human Perceptual Studies

In this study, we recruited human participants to analyze the naturalness of semantic adversarial audio examples and compared our attack against the two most representative traditional adversarial audio attacks [17, 18]. The study was approved by our University's Institutional Review Board (IRB).

Survey Design: The study procedure consists of two phases: pre-test surveys and listening tests. The participants were not informed of our study goal to minimize bias, and all the responses were collected anonymously to preserve the participants' privacy. Each participant was asked about their age, gender, and familiarity with VCS. Additionally, as the study calls for human empirical judgments, participants' prior experience in evaluating speech quality can affect the evaluation results. Therefore, participants are asked to rate their familiarity in speech quality assessment on a 5-point Likert scale, 1 for very unfamiliar and 5 for very familiar. During the listening test, 28 selected speech samples of 6-8 seconds each are played to the participants followed by questions. A concentration test is used to filter distracted participants with a silent file, a Gaussian noise file, and a human speech. We include the detailed survey questions in Appendix B.

Table 5: Results of Human Perceptual Studies

Target	Audio Groups	WER	Iden. ER	MOS
ASR	Original	2.15%	-	3.77
	Semantic Adversarial	4.68%	-	3.58
	Traditional Adversarial [17]	18.8%	-	1.89
SR	Normal Same	-	9.23%	3.82
	Normal Different	-	4.6%	3.77
	Semantic Adversarial	-	7.82%	3.61
	Traditional Adversarial [18]	-	13.6%	2.17

Iden. ER = Identification Error Rate

Adversarial Examples against ASR: We selected 12 speech clips in the attack against ASR systems: 4 semantic adversarial audio examples, 4 L_p-bounded adversarial examples [17], and 4 original speech of the adversarial audio examples. The participants were asked to transcribe each file, and the WER was calculated to indicate how well the speech is perceived. By definition, a higher WER indicates a noisier and less natural audio file hindering human perception. In addition, mean opinion score (MOS) [64] is used to evaluate speech quality and naturalness on a scale from 1 to 5, with 1 being the worst quality and 5 being the best.

Adversarial Examples against SR: In our evaluation against SR systems, participants were asked to judge whether a pair of speeches come from the same speaker. We prepared 4 groups of samples with 4 pairs of audio clips each, namely (1) *normal-same*, two original speeches of the same speaker, (2) *normal-different*, two original speeches of different speakers, (3) *semantic-adversarial*, one semantic adversarial audio and one original speech of the targeted speaker, and (4) *traditional-adversarial*, one traditional adversarial example [18] and one original speech of the targeted speaker. Each was followed by a multiple-choice question on whether they are of the same speaker ("Yes", "No", "I am not sure"). We calculated the identification error rate as error cases divided by trials, with the wrong answers and "I am not sure" counted as error cases. Participants were then asked about the quality of speech and naturalness similar to the ASR experiments.

Analysis Results: After filtering the responses by the concentration test, a total of 168 individuals participated in the study, including 97 females (57.7%) and 71 males (42.3%). The majority of participants (n=137, 81.5%) rated their prior experience as 2-3, while only a few (n=8, 4.8%) self-reported as more experienced (larger than 4), leading to an average of 2.34 and a standard deviation of 0.81. Such distribution indicates that most of the participants are regularly experienced in speech quality assessment.

The test results are summarized in Table 5. The adversarial examples against ASR show that our method gains similar WER and MOS as original speech, indicating better intelligibility and sound quality compared to the traditional L_p-bounded method [17]. For the adversarial examples against SR systems, the identification error rates for *normal-same* and *normal-different* are 9.23% and 4.6% respectively, as participants deem it high quality. Interestingly, the error rate of *normal-different* is lower than that of *normal-same*, this is potentially because humans are inclined to be more sensitive to perceive differences than similarity [51]. In comparison to the traditional attack [18], the semantic adversarial examples achieved lower identification error rates and higher opinion scores, indicating their superior quality. Lastly, the majority of participants (n=149, 88.7%) claim familiarity with VCS, yet showed poor discernibility to the semantic adversarial examples, further highlighting the significance of our attack.

10 Limitations and Discussions

Security Impact of SMACK: Despite being better in preserving speech quality due to attribute-only manipulation, the real-world attack it enables is categorically identical to the existing line of work on adversarial audio attacks [8, 18, 68, 79], where the adversarial example either triggers a mistranscription in ASR or misidentification in SR. Even though the impacts of the attacks are similar, SMACK does make adversarial audio examples more potent by improving stealthiness through the preservation of speech quality. The user study in Section 9 indicates preliminary evidence of improved naturalness of prosody-based semantic adversarial examples. Furthermore, the demonstration of the feasibility of adversarial semantic attributes manipulation to create adversarial audio examples raise a new threat vector that system defender need to take into consideration. Our preliminary investigation on testing SMACK against both adversarial audio example detection system and DeepFake example detection (Section 8.4) shows that semantic attribute manipulation is quite effective in evading existing detection mechanisms, since it contains significantly fewer artifacts compared to L_p-based adversarial perturbation. Lastly, we hope a deeper understanding of the semantic adversarial example would inform the development of more robust (often safety-critical) voice-controlled systems in the future.

DeepFake Recognition Algorithms and Potential Defense: While adversarial examples obtained from SMACK are different from DeepFake, it is important to understand how potential defenses for the two attacks may share similarities or differences. Existing DeepFake detection methods usually take two steps: (1) conduct feature selection to extract distinguishable features, and (2) train a classifier based on these features [84]. Thus, the key in this domain is to design effective feature extractors. Within this context, researchers have made efforts to extract various features, including short-term power spectrum features (e.g., log-spectrum, cepstrum, MFCC) [21, 49], short-term phase features (e.g., modified group delay function, relative phase shift) [85], spectral features with long-term processing (e.g., modulation spectrum) [48], and first-order Fourier coefficients [42] or second-order power spectrum [7]. However, one common problem with these methods is the lack of generalizability, where they usually work for only certain types of datasets, models, or spoofing techniques. Besides, adversarial examples in SMACK are obtained via prosody

manipulation of real speech, which further challenges the effectiveness of DeepFake detector algorithms.

The effectiveness of existing defenses that aims to detect artifacts from L_p-based adversarial audio examples [77] can be limited since SMACK relies on the malicious manipulation of prosody. One potential direction is to introduce our attack into the training pipeline with the adversarial training strategy to generate a new model [28, 81]. To achieve this, the defender needs to access the model and data, and design an effective white-box attack to enable iterative adversarial training. Another potential defense lies in the so-called liveness detection [10], where the acoustic characteristics induced by the physical aspects of human speech can be leveraged to detect the attack.

Limitations of SMACK: In this work, we assume the adversarial audio examples can be pre-generated ahead of time and the entire adversarial audio examples (not just perturbations) can be delivered to the target either over-the-line to an API or played over-the-air to an ASR/SR device via a speaker. However, we did not consider the real-time online attack scenario: the attacker will generate real-time adversarial audio perturbation and play it along with background sounds from the victim. In this scenario, the attacker needs to know the starting time of the audio and generate the corresponding adversarial perturbation in real-time. This can be challenging for SMACK, which manipulates temporal features. One possible workaround is to generate word-based prosody perturbation. Second, changing certain semantic attributes such as speech rate or accent is typically difficult by simply adding noises which is also the main difference between our adversarial attack and the traditional L_p-norm attacks. Another key limitation as shown in our experiment is that semantic-preservation limits adversarial perturbation space. As a result, for original speeches that have large differences in phoneme with the target transcription, it could be quite difficult to generate adversarial examples. To address this, a future direction is to explore the adversarial manipulations of more semantic attributes.

11 Conclusion

In this work, we introduce a new class of adversarial audio examples - semantically meaningful adversarial audio examples. Using prosody as the representative semantic attribute, we propose SMACK to mislead speech transcription and speaker recognition systems. Our experiments show that SMACK is effective against five ASR systems and two SR systems, as well as evading state-of-the-art defenses. During the human perceptual study, semantic adversarial examples exhibit better audio quality and speech naturalness compared to traditional adversarial audio examples. By showing the feasibility and practicality of semantic audio attacks, we hope our work shed light on a deeper understanding of the security of voice assistant technology and inspire future defense mechanisms.

Acknowledgment

We thank the reviewers for their feedback and Shixuan Zhai for his help in this project. This work is supported in part by US National Science Foundation under grants CNS-1916926, CNS-2038995, CNS-2154930, and CNS-2229427, and by Army Research Office under contract W911NF-20-1-0141.

References

[1] Cmusphinx open source speech recognition. https://cmusphinx.github.io/, May 2022.

[2] iflytek. http://www.iflytek.com/en/index.html, May 2022.

[3] Implementation of deepspeech2 for pytorch using pytorch lightning. https://github.com/SeanNaren/deepspeech.pytorch, May 2022.

[4] Nwalign 0.3.1. https://pypi.org/project/nwalign/, Sept 2022.

[5] Speech-to-text: Automatic speech recognition. https://cloud.google.com/speech-to-text/, May 2022.

[6] Speech to text – audio to text translation. https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/, May 2022.

[7] Ehab Alsayed Albadawy Abdrabuh. *AI-Synthesized Speech: Generation and Detection*. PhD thesis, State University of New York at Albany, 2022.

[8] Hadi Abdullah, Washington Garcia, Christian Peeters, Patrick Traynor, Kevin R. B. Butler, and Joseph Wilson. Practical hidden voice attacks against speech and speaker recognition systems. In *26th Annual Network and Distributed System Security Symposium, NDSS*, 2019.

[9] Hadi Abdullah, Muhammad Sajidur Rahman, Washington Garcia, Kevin Warren, Anurag Swarnim Yadav, Tom Shrimpton, and Patrick Traynor. Hear" no evil", see" kenansville"*: Efficient and transferable black-box attacks on speech recognition and voice identification systems. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 712–729. IEEE, 2021.

[10] Muhammad Ejaz Ahmed, Il-Youp Kwak, Jun Ho Huh, Iljoo Kim, Taekkyung Oh, and Hyoungshick Kim. Void: A fast and light voice liveness detection system. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 2685–2702, 2020.

[11] Dario Amodei et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*. PMLR, 2016.

[12] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*, 2019.

[13] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33, 2020.

[14] Anand Bhattad, Min Jin Chong, Kaizhao Liang, Bo Li, and David A. Forsyth. Unrestricted adversarial examples via semantic manipulation. In *8th International Conference on Learning Representations, ICLR 2020, April 26-30, 2020*. OpenReview.net, 2020.

[15] Alan W Black and Kevin A Lenzo. Building synthetic voices. *Language Technologies Institute, Carnegie Mellon University and Cepstral LLC*, 4(2):62, 2003.

[16] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. Hidden voice commands. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 513–530, 2016.

[17] Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 1–7. IEEE, 2018.

[18] Guangke Chen, Sen Chenb, Lingling Fan, Xiaoning Du, Zhe Zhao, Fu Song, and Yang Liu. Who is real bob? adversarial attacks on speaker recognition systems. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 694–711. IEEE, 2021.

[19] Li-Wei Chen and Alexander Rudnicky. Fine-grained style control in transformer-based text-to-speech synthesis. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7907–7911. IEEE, 2022.

[20] Tao Chen, Longfei Shangguan, Zhenjiang Li, and Kyle Jamieson. Metamorph: Injecting inaudible commands into over-the-air voice controlled systems. In *Network and Distributed Systems Security (NDSS) Symposium*, 2020.

[21] Tianxiang Chen, Avrosh Kumar, Parav Nagarsheth, Ganesh Sivaraman, and Elie Khoury. Generalization of audio deepfake detection. In *Odyssey*, 2020.

[22] Yuxuan Chen, Xuejing Yuan, Jiangshan Zhang, Yue Zhao, Shengzhi Zhang, Kai Chen, and XiaoFeng Wang. Devil's whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices. In *USENIX Security Symposium*, pages 2667–2684, 2020.

[23] Anne Cutler, Delphine Dahan, and Wilma Van Donselaar. Prosody in the comprehension of spoken language: A literature review. *Language and speech*, 40(2), 1997.

[24] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4), 2010.

[25] Takashi Fukuda, Raul Fernandez, Andrew Rosenberg, Samuel Thomas, Bhuvana Ramabhadran, Alexander Sorin, and Gakuto Kurata. Data augmentation improves recognition of foreign accented speech. In *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association*. ISCA, 2018.

[26] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon technical report n*, 93:27403, 1993.

[27] Liang Gonog and Yimin Zhou. A review: generative adversarial networks. In *2019 14th IEEE conference on industrial electronics and applications (ICIEA)*, pages 505–510. IEEE, 2019.

[28] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[29] Awni Hannun et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[30] Ben Hixon, Eric Schneider, and Susan L Epstein. Phonemic similarity metrics to compare pronunciation methods. In *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[31] Zack Hodari, Alexis Moinet, Sri Karlapati, Jaime Lorenzo-Trueba, Thomas Merritt, Arnaud Joly, Ammar Abbas, Penny Karanasou, and Thomas Drugman. Camp: a two-stage approach to modelling prosody in context. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6578–6582. IEEE, 2021.

[32] Hossein Hosseini and Radha Poovendran. Semantic adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1614–1619, 2018.

[33] Philip Jackson and SJUoSG Haq. Surrey audio-visual expressed emotion (savee) database. *University of Surrey: Guildford, UK*, 2014.

[34] Ye Jia et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *Advances in neural information processing systems*, 31, 2018.

[35] Ameya Joshi, Amitangshu Mukherjee, Soumik Sarkar, and Chinmay Hegde. Semantic adversarial attacks: Parametric transformations that fool deep classifiers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4773–4783, 2019.

[36] Hemant Kumar Kathania, Mittul Singh, Tamás Grósz, and Mikko Kurimo. Data augmentation using prosody and false starts to recognize non-native children's speech. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, 25-29 October 2020*, pages 260–264. ISCA, 2020.

[37] Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, 80(5), 2021.

[38] Il Yong Kim and OL De Weck. Variable chromosome length genetic algorithm for progressive refinement in topology optimization. *Structural and Multidisciplinary Optimization*, 29(6):445–456, 2005.

[39] Grzegorz Kondrak. *Algorithms for language reconstruction*. PhD thesis, University of Toronto, Canada, 2002.

[40] Felix Kreuk, Yossi Adi, Moustapha Cisse, and Joseph Keshet. Fooling end-to-end speaker verification with adversarial examples. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 1962–1966. IEEE, 2018.

[41] Zhuohang Li, Cong Shi, Yi Xie, Jian Liu, Bo Yuan, and Yingying Chen. Practical adversarial attacks against speaker recognition systems. In *Proceedings of the 21st international workshop on mobile computing systems and applications*, pages 9–14, 2020.

[42] Suk-Young Lim, Dong-Kyu Chae, and Sang-Chul Lee. Detecting deepfake voice using explainable deep learning techniques. *Applied Sciences*, 12(8):3926, 2022.

[43] Steven R Livingstone and Frank A Russo. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one*, 13(5):e0196391, 2018.

[44] Iker Luengo, Eva Navas, Inmaculada Hernáez, and Jon Sánchez. Automatic emotion recognition using prosodic parameters. In *Ninth European conference on speech communication and technology*. Citeseer, 2005.

[45] I Dan Melamed. Bitext maps and alignment via pattern recognition. *Computational Linguistics*, 25(1), 1999.

[46] Ryan E Mills, Christopher T Luttig, Christine E Larkins, Adam Beauchamp, Circe Tsui, W Stephen Pittard, and Scott E Devine. An initial map of insertion and deletion (indel) variation in the human genome. *Genome research*, 16(9):1182–1190, 2006.

[47] Gabriel Mittag, Babak Naderi, Assmaa Chehadi, and Sebastian Möller. NISQA: A deep cnn-self-attention model for multidimensional speech quality prediction with crowdsourced datasets. In *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association*, pages 2127–2131. ISCA, 2021.

[48] Hannah Muckenhirn, Pavel Korshunov, Mathew Magimai-Doss, and Sebastien Marcel. Long-term spectral statistics for voice presentation attack detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(11):2098–2111, 2017.

[49] Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques. *arXiv preprint arXiv:1003.4083*, 2010.

[50] Aaron Oord et al. Parallel wavenet: Fast high-fidelity speech synthesis. In *International conference on machine learning*, pages 3918–3926. PMLR, 2018.

[51] Andrew J Oxenham. How we hear: The perception and neural coding of sound. *Annual review of psychology*, 69:27–50, 2018.

[52] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[53] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan Ömer Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep voice 3: Scaling text-to-speech with convolutional sequence learning. In *6th International Conference on Learning Representations, ICLR 2018, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[54] Daniel Povey et al. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*, number CONF. IEEE Signal Processing Society, 2011.

[55] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.

[56] Haonan Qiu, Chaowei Xiao, Lei Yang, Xinchen Yan, Honglak Lee, and Bo Li. Semanticadv: Generating adversarial examples via attribute-conditioned image editing. In *Computer Vision - ECCV 2020 - 16th European Conference, Proceedings, Part XIV*, volume 12359 of *Lecture Notes in Computer Science*. Springer, 2020.

[57] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of speech recognition*. Prentice-Hall, Inc., 1993.

[58] Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[59] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn. Speaker verification using adapted gaussian mixture models. *Digital signal processing*, 10(1-3), 2000.

[60] Lea Schönherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding. In *26th Annual Network and Distributed System Security Symposium (NDSS)*, 2019.

[61] RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A Saurous. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. In *international conference on machine learning*, pages 4693–4702. PMLR, 2018.

[62] Guy-Bart Stan, Jean-Jacques Embrechts, and Dominique Archambeau. Comparison of different impulse response measurement techniques. *Journal of the Audio engineering society*, 50(4):249–262, 2002.

[63] Statista. Number of voice assistant users in the united states from 2021 to 2025. https://www.statista.com/statistics/1299985/voice-assistant-users-us/, Apr 2022.

[64] Robert C Streijl, Stefan Winkler, and David S Hands. Mean opinion score (mos) revisited: methods and applications, limitations and alternatives. *Multimedia Systems*, 22(2):213–227, 2016.

[65] Felipe Petroski Such, Vashisht Madhavan, Edoardo Conti, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *CoRR*, abs/1712.06567, 2017.

[66] Jianwei Sun, Zhiyuan Tang, Hengxin Yin, Wei Wang, Xi Zhao, Shuaijiang Zhao, Xiaoning Lei, Wei Zou, and Xiangang Li. Semantic data augmentation for end-to-end mandarin speech recognition. In *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association*. ISCA, 2021.

[67] Yaniv Taigman, Lior Wolf, Adam Polyak, and Eliya Nachmani. Voiceloop: Voice fitting and synthesis via a phonological loop. In *6th International Conference on Learning Representations, ICLR 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[68] Rohan Taori, Amog Kamsetty, Brenton Chu, and Nikita Vemuri. Targeted adversarial examples for black box audio systems. In *2019 IEEE security and privacy workshops (SPW)*, pages 15–20. IEEE, 2019.

[69] Michael Wagner and Duane G Watson. Experimental and theoretical advances in prosody: A review. *Language and cognitive processes*, 25(7-9):905–945, 2010.

[70] Yulin Wang, Xuran Pan, Shiji Song, Hong Zhang, Gao Huang, and Cheng Wu. Implicit semantic data augmentation for deep networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[71] Ann Wennerstrom. *The music of everyday speech: Prosody and discourse analysis*. Oxford University Press, 2001.

[72] Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

[73] Yi Xie, Zhuohang Li, Cong Shi, Jian Liu, Yingying Chen, and Bo Yuan. Enabling fast and universal audio adversarial attack using generative model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14129–14137, 2021.

[74] Yi Xie, Cong Shi, Zhuohang Li, Jian Liu, Yingying Chen, and Bo Yuan. Real-time, universal, and robust adversarial attacks against speaker recognition systems. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 1738–1742. IEEE, 2020.

[75] Yi Xu. Speech prosody: A methodological review. *Journal of Speech Sciences*, 1(1):85–115, 2011.

[76] Junichi Yamagishi, Massimiliano Todisco, Md Sahidullah, Héctor Delgado, Xin Wang, Nicolas Evans, Tomi Kinnunen, Kong Aik Lee, Ville Vestman, and Andreas Nautsch. Asvspoof 2019: The 3rd automatic speaker verification spoofing and countermeasures challenge database. 2019.

[77] Zhuolin Yang, Bo Li, Pin-Yu Chen, and Dawn Song. Characterizing audio adversarial examples using temporal dependency. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[78] Zhiyuan Yu, Zack Kaplan, Qiben Yan, and Ning Zhang. Security and privacy in the emerging cyber-physical world: A survey. *IEEE Communications Surveys & Tutorials*, 23(3):1879–1919, 2021.

[79] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, Xiaofeng Wang, and Carl A Gunter. Commandersong: A systematic approach for practical adversarial voice recognition. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 49–64, 2018.

[80] Li Yujian and Liu Bo. A normalized levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1091–1095, 2007.

[81] Haichao Zhang and Jianyu Wang. Defense against adversarial attacks using feature scattering-based adversarial training. *Advances in Neural Information Processing Systems*, 32, 2019.

[82] You Zhang, Fei Jiang, and Zhiyao Duan. One-class learning towards synthetic voice spoofing detection. *IEEE Signal Processing Letters*, 28:937–941, 2021.

[83] Baolin Zheng, Peipei Jiang, Qian Wang, Qi Li, Chao Shen, Cong Wang, Yunjie Ge, Qingyang Teng, and Shenyi Zhang. Black-box adversarial attacks on commercial speech platforms with minimal information. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021.

[84] Yipin Zhou and Ser-Nam Lim. Joint audio-visual deepfake detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[85] Pedram Abdzadeh Ziabary and Hadi Veisi. A countermeasure based on cqt spectrogram for deepfake speech detection. In *2021 7th International Conference on Signal Processing and Intelligent Systems (ICSPIS)*, pages 1–5. IEEE, 2021.

A Performance Comparison between SMACK and White-box Attack

On the other hand, compared to another attack against speaker recognition [74], SMACK has a weaker threat model assuming black-box knowledge, as compare to the white-box setting in [74]. This difference in attack knowledge entails different approaches in adversarial example generation. Furthermore, in order to preserve speech quality, prosody manipulation in SMACK can not be applied to a small snippet of audio, as compared to [74] where the perturbation is added to different locations of the audio example. The success rate of universal perturbation can also vary based on the input. Lastly, [74] targeted one open source SR system, and SMACK works on multiple open-source/commercial ASR and SR systems. To summarize, while the generation of adversarial audio example in [74] is much faster than SMACK due to its addition of a universal perturbation, SMACK assumes a different attacker model, preserves the speech quality, and targets multiple voice systems.

B Survey Questions of Human Studies

At the beginning, we asked each participant about their past experience on speech quality assessment. The exact question is phrased as "On a scale from 1 to 5, please rate your familiarity in assessing the quality of the given speech audio clips, with 1 indicating very unfamiliar and 5 indicating very familiar." For the adversarial examples against ASR systems, we instructed the participant as "In the following tasks, you will be guided to listen to audio clips and answer questions accordingly. Specifically, please write down your transcript of each of the audio clip as you listen, and rate your perception on the quality of the audio clip." The questions are as follows:

- For the following audio clip, please write down its transcript as you listen.
- For this audio clip, please rate your perception on its quality on a scale from 1 to 5, with 1 indicating the worst quality and 5 indicating the best.

For the adversarial examples against SR systems, we instructed the participants as "In the following, you will be guided to listen to audio clips and answer questions accordingly. Specifically, you will listen to two audio clips in a group, and please judge if they are uttered from the same person." The exact questions we asked are listed below:

- For the two audio clips in the following, do you think they were uttered by the same person? ("Yes", "No", "I am not sure".)
- For the two audio clips, please rate your perception on its quality on a scale from 1 to 5, with 1 indicating the worst quality and 5 indicating the best.

At last, we ask the participants how frequently they use voice assistant in their daily lives. The question is phrased as "On a scale from 1 to 5, please rate how frequently you use voice assistants in your daily life, with 1 being never and 5 being very frequently."