UniAP: Protecting Speech Privacy with Non-targeted Universal Adversarial Perturbations

Peng Cheng, Yuexin Wu, Yuan Hong, *Senior Member, IEEE,* Zhongjie Ba, *Member, IEEE,* Feng Lin, *Senior Member, IEEE,* Li Lu, *Member, IEEE,* and Kui Ren, *Fellow, IEEE*

Abstract—Ubiquitous microphones on smart devices considerably raise users' concerns about speech privacy. Since the microphones are primarily controlled by hardware/software developers, profit-driven organizations can easily collect and analyze individuals' daily conversations on a large scale with deep learning models, and users have no means to stop such privacy-violating behavior. In this paper, we propose UniAP to empower users with the capability of protecting their speech privacy from the large-scale analysis without affecting their routine voice activities. Based on our observation of the recognition model, we utilize adversarial learning to generate quasi-imperceptible perturbations to disturb speech signals captured by nearby microphones, thus obfuscating the recognition results of recordings into meaningless contents. As validated in experiments, our perturbations can protect user privacy regardless of what users speak and when they speak. The jamming performance stability is further improved by training optimization. Additionally, the perturbations are robust against noise removal techniques. Extensive evaluations show that our perturbations achieve successful jamming rates of more than 87% in the digital domain and at least 90% and 70% for common and challenging settings, respectively, in the real-life chatting scenario. Moreover, our perturbations, solely trained on DeepSpeech, exhibit good transferability over other models based on similar architecture.

Index Terms—Adversarial Examples, Speech Recognition, Privacy, Voice Assistants.

1 Introduction

Microphone deployment is surging with the popularity of smart devices. Meanwhile, the privacy implication of pervasive microphones has been the center of substantial debate [1], [2], [3]. Microphones are increasingly equipped to support hands-free experience (e.g., voice control) and voice related functions. However, many of them are out of users' control, even worse, out of users' awareness. For example, Google integrates a dormant microphone secretly in its home security device – Google Nest. No one realized its existence until Google announced that its voice assistant service would roll out to Nest [4]. These ubiquitous sound receivers can easily record users' daily conversations in many situations (e.g., secretly recording or voice call). Such large volumes of recordings, no longer controlled by users, contain private user information attractive to commercial companies. They are motivated to analyze the speech contents from the recordings for various purposes, such as algorithm enhancement, data trading and targeted advertising as shown in Figure 1. In 2019 Google and Amazon both admitted their analyses on the semantic contents of consumers' recordings for service improvement [5], [6]. A large amount of these audio clips are unintended records [7].

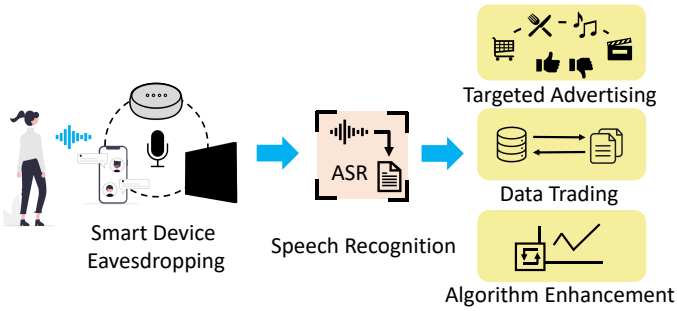

Fig. 1: Organizations collect and illegitimately analyze speech contents for different purposes.

Users cannot fully control the recording behaviors of nearby microphones and cannot stop the illegitimate collection and analysis of their own speech signals. Such a considerable privacy threat raises concerns from government, industry and academia. Security and privacy compliance obligations such as EU's General Data Protection Regulation (GDPR) [8] and California Consumer Privacy Act (CCPA) [9] have been enacted to empower users with more rights to autonomy over their personal data including voice recordings. Companies are upgrading the privacy control on their product. Google introduces a switch to disable all apps' access to the microphones on Android 12 [10]. Lenovo builds microphone mute switch on its laptops. In academia, related studies focus on anti-eavesdropping techniques [7], [11]. In short, communities have realized the privacy threat of pervasive recording devices and illegitimate usage of recordings, and they start to define measures from legal,

- *Peng Cheng, Yuexin Wu, Zhongjie Ba, Feng Lin, Li Lu, and Kui Ren are with the School of Cyber Science and Technology, Zhejiang University, Hangzhou, Zhejiang, China, 310027 and ZJU-Hangzhou Global Scientific and Technological Innovation Center, No.733 Jianshe San Road, Xiaoshan District, Hangzhou, Zhejiang, China, 311200. E-mail: {peng_cheng,wuyuexin,zhongjieba,flin,li.lu,kuiren}@zju.edu.cn.*
- *Yuan Hong is with University of Connecticut. E-mail: yuan.hong@uconn.edu.*

(Corresponding author: Zhongjie Ba.)

policy, and technical perspectives.

The above approaches attempt to give users more control over their private data, but they have limitations. The software can still access the microphone data even though mute switches are activated [12]. Even worse, the manufacturers/service providers may peep into the private information for their interest (e.g., the case of Amazon and Google). Users can only rely on their honesty and hope the claimed software/hardware measures would work. The anti-eavesdropping techniques [7], [11] are deployed at the user side, but they require special hardware to facilitate ultrasonic jamming, which is similar to [13], [14]. However, they make devices deaf thus stopping normal voice activities (e.g., listeners cannot understand the jammed recordings in a video conference.). Furthermore, their constant jamming induces health issues [15].

In this paper, we seek to answer the question: *can users proactively protect their speech content privacy while their voice activities are not interfered?* To this end, we propose a privacy-preserving technique that supports normal voice recording but stops the automatic analysis of the content of vast quantities of speech. Since organizations generally use an Automatic Speech Recognition (ASR) for such analysis otherwise entailing too much human labor [16], we propose UniAP, which fools the ASR recognition results into meaningless texts without affecting human perception. Note that UniAP needs not a microphone that induces extra privacy concerns.

UniAP generates specially-designed perturbations to be played by a commercial off-the-shelf (COTS) speaker (i.e., a jammer) completely controlled by a user. When a user decides to protect speech privacy, he/she first activates the jamming and then freely speaks or starts a video conference with an app (i.e., Zoom). The jamming noises along with speech signals will be captured by malicious recording devices close to users or the same device hosting the conference. Once the recordings are analyzed by an ASR, resulting transcriptions would be meaningless texts. Meanwhile, humans would not be bothered seriously by the jamming noises, and the speech content is still intelligible for the call receiver in the virtual conference. UniAP can be a standalone device without sophisticated hardware or an app on normal smart appliances.

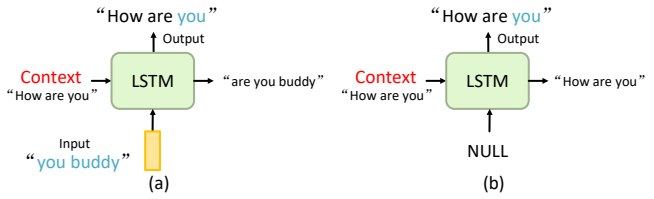

Fig. 2: Context dominates the transcribing process of LSTM-based ASR. "you" is correctly output even if the input of the current step is none, given the necessary information is provided in the context component.

UniAP is a novel jamming technique utilizing non-targeted adversarial examples (AEs) to obfuscate ASR systems. Our ultimate goal is to generate perturbations effective on commercial ASRs (black box), but as the first step towards this direction, we first work on an open-source ASR (white box) in this paper. The white-box setting is widely applied in speech AE works [17], [18], [19], [20], [21]. Considering small business entities are likely to adopt a well-known and open-source ASR rather than developing one from scratch, we use one of the high-performing ASRs, Deep Speech [22], as our jamming target. Deep Speech is widely used in both academia [17], [20], [23] and in industrial products such as voice assistants and online speech-to-text (STT) platforms [20]. Its most popular open-source implementation is named as DeepSpeech. We use "DeepSpeech" to refer to both the model and the implementation in this paper.

Generic AE methods cannot be directly applied to preserve privacy due to several technical challenges. First, the jamming signal should be input-agnostic to be effective regardless of what the user speaks (i.e., universality). Second, the jamming signal should be synchronization-free to be effective regardless of when the user starts talking. Third, the jamming performance of different perturbations should be stable in practical usage, otherwise will harm users' trust in the approach. Fourth, the robustness of jamming noises against denoising should be considered in practical use. Most speech AE techniques [17], [18], [24], [25], [26], [27] focus on targeted attack and cannot satisfy the above requirements. Neekhara et.al [28] achieves universal AE but fails on other requirements. Their AE generation algorithm, similar to [29], is based on aggregating atomic perturbation vectors that disturb specific data points to form a universal AE. In comparison, we redesign the perturbation structure, optimize the training process and induce randomness to satisfy all four requirements. AdvPulse [21] and Vadillo et.al. [30] focus on synchronization-free and universality respectively on command classification models of dozens of words, but the feasibility differs from fooling speech transcription models (our case) [30]. Besides, we have not seen works considering the impact of AE robustness against denoising techniques. To our best knowledge, this paper takes the first cut to apply AE to preserve speech privacy while addressing these new challenges.

Targeting DeepSpeech as an example, the recognition mechanism of Long Short Term Memory (LSTM)-based ASR is studied. We experimentally find the dominating role of context in the transcription process. Figure 2 shows the core idea: even if the input is empty, "you" can be decoded if the context part contains the relevant information and the role of input is more about updating the context. Based on the observation, we implement a practical jamming noise generation method to address the aforementioned challenges. We utilize batch training and create a unique structure to make our perturbation content-agnostic and synchronization-free. We then optimize the training method to increase the jamming stability and the perturbation robustness against noise removal methods. Lastly, we consider over-the-air jamming scenarios in real life and evaluate the transferability of our noises across models with similar network structures. We encourage readers to listen to our jammed audio samples at the demo website[1]. Highlights of our original contributions are summarized as follows:

1) To our best knowledge, we propose the first work utilizing universal adversarial perturbations (UAPs)

1. https://github.com/UniAP2022/UniAP

for protecting user speech
2) We conduct comprehens DeepSpeech, unveiling t based ASRs that dominat
3) To defend against LSTM UniAP, which generates r ming signals satisfying pr study validates the low in model experimental resul ity of these UAPs on LST
4) Extensive evaluations or over-the-air show the jar and validate the robustr against noise removal me

2 PRELIMINARY

Because ASR is our fooling tar concepts in this section. Especiall ings of LSTM-based ASRs as DeepSpeech belongs to this category. Then necessary information about AEs and metrics used in this paper are provided.

2.1 ASR Introduction

2.1.1 Function of an ASR

The task of an ASR is to transcribe an audio signal to its corresponding semantic content. The speech recognition process can be formulated as:

$$y = \underset{\widetilde{y}}{\operatorname{argmax}}\, p(\widetilde{y}|x) \tag{1}$$

x is the audio signal, and $\widetilde{y}$ are all possible transcription candidates. ASR finds the most likely transcription given the audio input. We simplify Equation 1 as $y = f(x)$, and formulate the process of an individual perceives the x as $f_H(x)$, where $f_H(\cdot)$ represents the human capability of understanding speech. A good ASR should result in $f(x) \simeq f_H(x)$. With the advancement of deep learning in recent years, ASRs based on neural networks achieve cutting-edge performance and good usability. LSTM-based ASRs are one of the mainstream learning-based ASRs. **DeepSpeech**, one of the state-of-the-art models [20], is an important representative of this category, primarily studied in this paper. The findings on DeepSpeech are applicable to other ASRs of the LSTM category.

2.1.2 LSTM-based ASR

LSTMs are a special type of recurrent neural networks(RNNs). It introduces the cell state C_t (the top line of an LSTM module shown in Figure 3) which runs through the whole working chain of LSTMs to address the short-term memory limitation of RNNs [31]. The input pair (x_t, h_{t-1}) is fed into the forget gate, input gate and output gate respectively. The outputs of these three gates decide what information is cast away from C_{t-1}, how to update the C_{t-1}, and the value h_t to be output at current step.

In case of an LSTM-based ASR, the cell state C_t keeps the context information essential to a transcription task, and the hidden state h_t decides the result of current step. As Figure 4 illustrates, the input of DeepSpeech is a Mel-Frequency

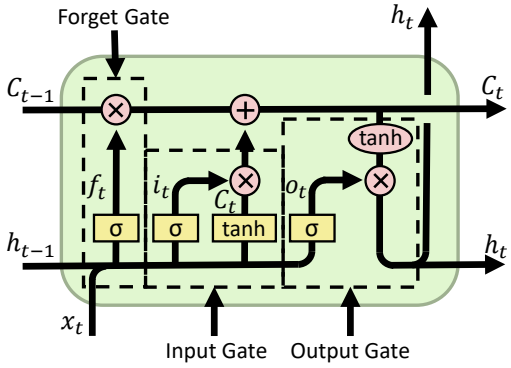

Fig. 3: The internals of the LSTM module.

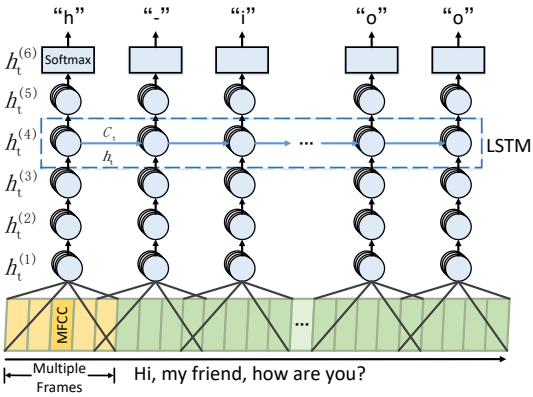

Fig. 4: The structure of DeepSpeech.

Cepstral Coefficient (MFCC) matrix which is the feature of multiple speech frames. The layer highlighted with blue-dashed box is a standard LSTM network. The remaining networks are all feed-forward neural networks with output $h_t^{(i)}$. A softmax layer outputs the probability distribution of every token. Lastly, Connectionist Temporal Classification (CTC) module removes repetitious and redundant symbols and generates the final transcription.

2.2 Adversarial Perturbations/Examples

In the adversarial attack, an adversary adds a small derived perturbation δ on x to generate $x' = x + \delta$. Humans still perceives x' as the original transcription y, while the ASR recognizes it as a different text y'. x' is usually called an adversarial example (AE). There are two types of adversarial attacks: targeted one and non-targeted one. Non-targeted AE is not interested in what results would be decoded by the ASR. The goal is achieved if a user perceives x' different from the ASR does ($f(x) \neq f_H(x)$). δ is derived by an optimization with the goal to increase the difference (loss) between y' and original text y. Considering an ASR as a white-box, gradient descent is used for calculating the perturbations. In this paper, we seek to generate robust acoustic perturbation δ as the jamming noise, where x' is a non-targeted AE and also the perturbed signal.

2.3 Metrics

CER and JSR. For untargeted adversarial attack, Character Error Rate (CER) is a widely used metric to measure the

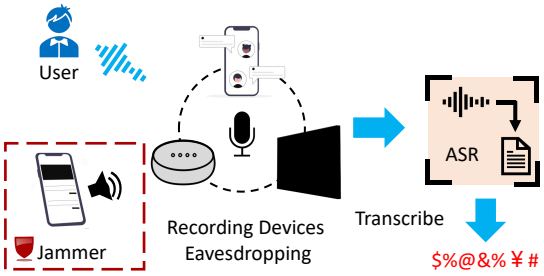

Fig. 5: The system model.

difference between the false transcription y' and the original transcription y: $CER(y, y') = \frac{EditDistance(y,y')}{length(y)}$. The edit distance between two strings y and y' is formulated as $EditDistance(y, y') = (N_{sub} + N_{ins} + N_{del})$, where N_{sub} is the number of characters unmatched between the reference y and y', N_{ins} is the number of characters present in y' but absent in y, and N_{del} is the number of characters which appear in y but absent in y'. Lastly, $length(y)$ equals to N_{ref} that is the number of characters in y. An attack is successful when the CER of an AE is big enough, such that the original semantic content cannot be inferred. In a real privacy-preserving scenario, the jamming is usually treated as a success if a CER is equal to or greater than 50% [28]. Besides CER, the percentage of successful one ($CER > 0.5$) in all jamming trials, named as Jamming Success Rate (JSR), is applied to evaluate jamming performance. Word Error Rate (WER) is also used occasionally Jamming Success Rate (JSR) and it is similar to CER but on the word level.

SNR. The perturbation δ should be small enough to make an AE quasi-imperceptible. We compare the speech signal (jamming target) and our perturbation with signal-to-noise Ratio (SNR) following relevant studies [18], [21]. It is defined as $10 \log_{10} \frac{P_x}{P_\delta}$. The larger the value of SNR, the less likely the perturbation would be noticed by users. To provide some context, the SNR of perturbed signal in CommanderSong [18] ranges from 14 to 18.6 dB on the digital domain, and all SNRs are below 2 dB in the over-the-air case. In AdvPulse [21], the SNRs are 13.7 dB and 6 dB, respectively.

3 Problem Formulation

In this section, we first introduce our system and threat model, then present the design goals of our privacy-preserving method. DeepSpeech is very likely chosen by small companies as their STT engine integrated into products [20], [32], [33], thus it is the main study target in this paper. And our privacy-preserving method shows transferability over other LSTM models. Notice that, although the method allows voice activities like video conferencing, the functioning of voice assistants is not included. We consider it poses privacy risks because of continuous sound monitoring. Such behaviour should not be allowed when a user would like to protect his/her privacy.

3.1 The System and Threat Model

Users cannot prevent organizations from automatically extracting their personal information with an ASR from voice recordings, which raises severe privacy risks [34]. To address this issue, we propose to utilize a jammer that enables users to protect their own speech privacy without relying on the honesty of other parties. Figure 5 shows the system model, including *Users*, *Recording Devices* and *UniAP Jammer*.

1) *Users* are individuals resting in their room. They would have a conversation, a video conference or a voice call at any moment, and they don't want their conversation content to be analyzed without their permission.
2) *Recording Devices* are common smart gadgets equipped with built-in microphones (e.g., a smart TV, a laptop or a smartphone). They are usually deployed around *Users*, can autonomously become active, and record conversations in the environment without *Users*' authorization. The recordings would be further analyzed with an ASR model.
3) *UniAP Jammer* is a device equipped with a built-in speaker. It transmits UniAP perturbations (i.e., non-intrusive jamming noises) over the air after manual activation or at a preset time before *Users* speak. The perturbed speech recordings would be transcribed by an ASR to texts much different from what *Users* actually said, while remaining intelligible to human beings.

We consider the *adversary* as an entity collecting the conversation recordings and thereafter extracting *Users*' private information. The adversary may: (1) record *Users*' conversations without their consent using *recording devices* (e.g., a smart TV) near them; (2) implement a video conferencing app but secretly keep the conversations recorded during a virtual conference for own purposes; (3) lure users to download an eavesdropping app disguised as either a social or communication app. The app can easily obtain the permission to access microphone as Zhou et al. [35] unveiled, and it would record *Users*' conversations when a VoIP call is detected. Once obtaining recordings via one of the three ways, the adversary would extract sensitive information from them. Inferring semantic contents of massive recordings is usually done with an ASR otherwise entailing too much human labor.

The speech privacy threat considered in this paper is the unauthorized semantics interpretation on speech recordings. Small businesses are more likely to be the adversary because they tend to take risks and conduct such behaviour for the sake of profit[2]. A small company would probably choose a well-known and open-source ASR because developing one from ground up takes too much effort and many good open-source ASRs are available.

We consider the opponent would use additional noise reduction techniques to improve the intelligibility of the recordings. A moderate adversary would probably apply spectral-subtractive algorithms [36] to perform a general speech enhancement. A strong adversary would try to obtain our jamming noise (i.e., template) then subtracts the template from the recording before feeding it to an ASR.

2. This is a conservative claim because these giant companies also use speech recordings improperly [5], [6].

3.2 Design Goals

To protect users' speech content privacy under the above model, we aim to design a systematic approach to generate effective and practical jamming noises, achieving the following goals: (1) Low interference: to ensure high SNR of the perturbed speech signal, keeping the interference to users to the minimal. (2) Be content-agnostic and synchronization-free: to ensure our UniAP perturbations can protect the private information contained in most of speech signals regardless of contents. Additionally, the perturbations must stay effective no matter when users speak. (3) Stability: to ensure stability regarding the jamming performance of generated noises, thus maintaining the effectiveness of different trained perturbations. (4) Robustness: to ensure the effectiveness of our noises in a physical playback scenario. We also aim to generate perturbations robust against strong adversaries that may apply various denoising methods.

4 Transcribe Speech via LSTM-based ASRs

Before presenting our perturbation generation approach, we first provide an empirical study on DeepSpeech, then explain our fundamental observations on the working mechanism of LSTM-based ASRs based on the experiment results. These observations construct the basis of UniAP, namely our perturbation generation approach.

Similar to [17], [18], [20], [37], [38], [39], we use Mozilla's Project DeepSpeech [40] as the ASR to study. Section 2.1.2 has shown that DeepSpeech incorporates a standard LSTM as the core component (Figure 6). The LSTM takes X_t, h_{t-1} and C_{t-1} as input and then outputs h_t.

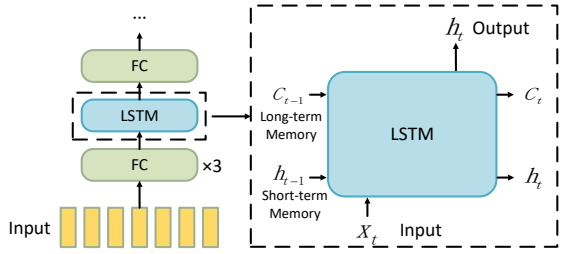

Fig. 6: The LSTM module in DeepSpeech.

We would like to validate the assumption: *The context information plays the key role in inferring the transcription result of an LSTM ASR.* To confirm this assumption, we verify the impact of inputs of the LSTM module on the prediction results. 3300 sentences are randomly chosen from the train-clean-100 subset of Librispeech and fed into DeepSpeech to obtain transcriptions. Then, we look at the interpretation result of each step, collect 50 speech segments which are recognized as the same English characters, and record their corresponding LSTM inputs (e.g., h_t, C_{t-1} and x_t). This gives us 1300 groups of data, each of which is the mapping between the prediction characters and three inputs, noted as $O_{lstm} = f(x_t, h_{t-1}, C_{t-1})$. Then, we set the each input of this mapping to zero vector and calculate what percentage of the pairs in each class is still recognized correctly.

The experiment results are illustrated in Table 1. When x_t are reset to zero vectors, most of the prediction results are correct. When C_{t-1} are settled to zero values, the recognition accuracy fluctuates between 24% and 90%. However, when h_{t-1} values become all zeros, most of the recognition results are wrong. This result shows that x_t is less critical in influencing transcription results. h_{t-1} and C_{t-1} both affect the prediction, and h_{t-1} is more important than C_{t-1}.

The results demonstrate context information especially the short-term memory h_{t-1} dominates the transcription process of LSTM, and information of the current frame only has minor effect. Therefore, we should focus on perturbing the context information to achieve effective jamming. Because h_t and C_t run down the entire ASR processing chain from the utterance beginning, an optimistic solution is to disturb them from the start of a speech signal.

5 UniAP-based Privacy Mechanism Design

Given the importance of context information, we present the key innovations of our perturbation generation approach, namely UniAP in this section. We utilize non-targeted AE training to achieve jamming that fools ASR transcription while keeping low interference to human being. Besides being implicitly quasi-perceptible, perturbations trained with speech AE should be content-agnostic, synchronization-free, stable and robust to be practical.

5.1 Initial Formulation for Content-Agnostic Perturbation

We construct perturbations by solving the non-targeted AE crafting problem. Let x denote a waveform from an audio sample distribution X, we seek to obtain the perturbation δ causing an ASR mis-transcribes the majority of audio data sampled X. Formally, we define the goal of our non-targeted perturbation as:

$$CER(t, f(x+\delta)) > \epsilon, \text{ for majority } x \in X \tag{2}$$

where t is the ground truth transcription, ϵ is empirically set as 0.5, and the "majority" requirement is necessary for jamming arbitrary speech content. Inspired by [29], we use the term Universal Adversarial Perturbation (UAP) to name the problem solution. SNR is applied to quantify the distortion induced by the perturbation. Now we formulate the UAP construction as an optimization problem as below:

$$\begin{aligned} &\max SNR_\delta(x) \\ &\text{subject to } \mathop{\mathbb{E}}_{x \in X}(CER(t, f(x+\delta)) > \epsilon) > TH \end{aligned} \tag{3}$$

where TH refers to a threshold of the jamming success rate to ensure the capability of jamming arbitrary content. However, solving this formula is not trivial. We instead minimize the following objective function which is a relaxation of Equation 3:

$$\begin{aligned} &\min J(\delta, x, t) \\ &\text{where } J(\delta, x, t) = c||\delta||^2 - l(t, f(x+\delta)) \end{aligned} \tag{4}$$

Here l is the CTC loss which measures the distance between the ground-truth transcript and the model transcript. The larger l results in larger CER. The L2 norm aims to restrict the perturbation energy. Constant c is the weight of L2 norm.

To solve Equation 4, we utilize the batch loss to iteratively generate our UAP. Let θ refers to the allowed distortion level, $X_i = x_1, x_2, ..., x_i, ...x_m$ be a batch of speech

TABLE 1: The successful recognition rate of CTC labels after resetting x_t, C_{t-1} and h_{t-1} to 0, respectively. The number of each CTC label is 50. After resetting x_t, C_{t-1} and h_{t-1} to 0, the prediction results achieve average success rates of 96.1%, 68.5% and 2.8%, respectively. Thus, context information (esp. short-term memory h_{t-1}) dominates the transcribing process.

CTC label	$x_t \to 0$	$C_{t-1} \to 0$	$h_{t-1} \to 0$	CTC label	$x_t \to 0$	$C_{t-1} \to 0$	$h_{t-1} \to 0$	CTC label	$x_t \to 0$	$C_{t-1} \to 0$	$h_{t-1} \to 0$
a	98.0%	86.0%	4.0%	j	90.0%	66.0%	2.0%	s	88.0%	62.0%	4.0%
b	100.0%	90.0%	2.0%	k	96.0%	62.0%	2.0%	t	98.0%	78.0%	2.0%
c	96.0%	64.0%	2.0%	l	94.0%	54.0%	4.0%	u	94.0%	60.0%	8.0%
d	98.0%	62.0%	2.0%	m	94.0%	90.0%	0.0%	v	100.0%	90.0%	0.0%
e	90.0%	70.0%	0.0%	n	96.0%	80.0%	0.0%	w	94.0%	68.0%	0.0%
f	100.0%	82.0%	2.0%	o	98.0%	86.0%	2.0%	x	94.0%	32.0%	18.0%
g	98.0%	66.0%	0.0%	p	94.0%	70.0%	2.0%	y	98.0%	64.0%	8.0%
h	98.0%	74.0%	4.0%	q	94.0%	64.0%	0.0%	z	100.0%	24.0%	2.0%
i	98.0%	76.0%	2.0%	r	100.0%	60.0%	2.0%	avg	96.1%	68.5%	2.8%

signals sampled from a distribution X, and m refers to the batch size in training. Algorithm 1 goes through batches of training samples randomly chosen from the training set and builds the perturbation δ iteratively.

Algorithm 1 Training

Input: Training Audios and Texts (X, T), allowed distortion θ, learning rate α, batch size m, L2 penalty constant c
Output: Universal Perturbation δ
  Initialize δ
  while $mean(|\delta|) < \theta$ **do**
    $(X, T) = \{(X_1, T_1), (X_2, T_2), ..., (X_n, T_n)\})$
    for $(X_i, T_i) \in X$ **do**
      $(X_i, T_i) = \{(x_1, t_1), (x_2, t_2), ..., (x_m, t_m)\}$
      $\delta \leftarrow \delta - \alpha \nabla_\delta \frac{1}{m} \sum_{j=1}^{m} J(\delta, x_j, t_j)$
    end for
  end while

5.2 Synchronization-Free with Chunk-based Perturbation

We have constructed a basic non-targeted UAP, next we achieve the synchronization-free goal. In a streaming scenario, the timing of users starting speaking is unpredictable, which is depicted as the **random time delay** in Figure 7. That is to say, we need to significantly disturb the context information under such unpredicted delay condition (i.e., without synchronization with users' speech). If the length of the perturbation is comparable to that of speech signals, it is hard to balance the effect of each portion of the perturbation. In order to disrupt context information along the whole speech signal, it is better to make every part of the perturbation count. We use short-length perturbation to concentrate the disturb effect. The length of a basic perturbation module is small enough such that it is most likely shorter than a signal speech command, then small-sized perturbation chunks are concatenated to form the perturbation, which ensures at least one complete perturbation chunk for influencing users' conversation signals. Specifically, we empirically set each chunk to last for half a second (i.e., 8000 data points when the sampling rate is 16 kHz).

Based on the chunk structure, we further induce random time shifting into the UAP training to address various unsynchronizaed conditions (i.e., unknown speaking timing). We induce random time shifting as shown in Algorithm 2. Rather than starting the adversarial perturbation at a specific point, usually the beginning moment of x, we aim to minimize the loss if the speech signal is delayed randomly by i, where i obeys the uniform distribution of the time interval between 0 and $l-1$. l denotes the length of the basic noise chunk. We copy δ_s to form the repeated chunks illustrated in Figure 7 according to the length of x. The δ of Algorithm 1 is replaced by the repeated-chunk perturbation.

Algorithm 2 Random Shift

Input: Perturbation δ
Output: Shifted Perturbation δ_s
  $l \leftarrow length(\delta)$
  $i \sim Uniform(0, l-1)$
  $\delta_s \leftarrow concatenate(\delta[i : l-1], \delta[0 : i-1])$

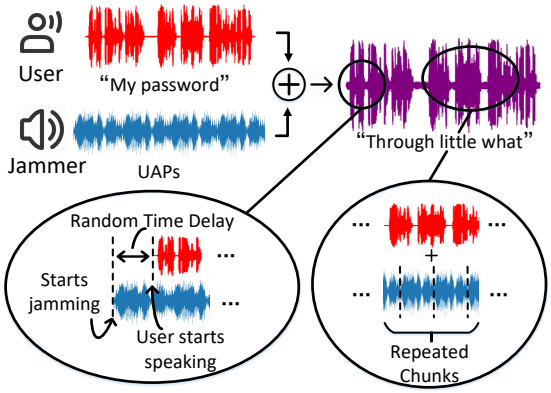

Fig. 7: Random time delay and chunk-based perturbation.

We evaluate the impact of chunk structure on the jamming performance. A dataset containing 5000 instances randomly chosen from the Librispeech-clean-100 is built. The dataset is split into a training set containing 4000 samples and a test set containing 1000 instances. We train ten perturbations utilizing our algorithms, then generate ten perturbations with a comparable length to the target audio clips to jam as the baseline. We choose 10^{-5} as the L2 penalty constant for our algorithm (no penalty for the audio-length scenario considering the training feasibility), 100 as the allowed distortion, 1 as the learning rate and 20 as the batch size for both training process. Besides, we randomly delay speech signal during the evaluation to simulate a user's random talk timing. CER, JSR and SNR introduced in Section 2.3 are applied to evaluate the jamming performance. Larger values of CER and JSR show better jamming effect, and larger SNR indicates less interference to human being. Generating a UAP requires ∼1 hour on an NVIDIA RTX 3090 GPU.

Table 2 shows the average performance of the two types of perturbations on the test set. The repeated-chunk noise

TABLE 2: Performance of perturbations in different forms. The repeated-chunk noise achieves the JSR of 80.3%.

Perturbation form	CER	JSR	SNR (dB)
Audio Length	60.2%	64.5%	22.89
Repeated Chunks	75.5%	80.3%	23

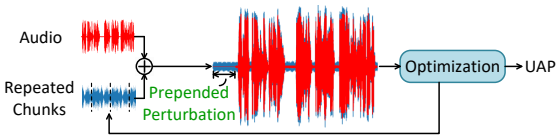

Fig. 8: Illustration of the prepended perturbation.

TABLE 3: Perturbation performance with/without prepending training. Prepending training increases JSR from 85.6% to 89.1%, and significantly reduces the standard deviation of JSR between different UAPs trained with the same process.

Prepending Training	CER	JSR	std-JSR	SNR (dB)
✗	79.3%	85.6%	0.074	22.9
✓	85.7%	89.1%	0.048	22.8

shows a JSR of 80.3%, which is 15.8% higher than the audio-length ones, and the average SNR is 23 dB that is about 28% higher than the best case of the state-of-the-art [18], [21]. The results validate the jamming effectiveness under unsynchronization conditions [28].

5.3 Improve Stability with Perturbation-Prepending Training

Based on the synchronization-free UAP, we further improve its performance stability, which is critical because users would only choose reliable jamming noise. That is to say, when maintaining high jamming effectiveness, we aim to reduce the variance of performance caused by the inherent randomness of the training process and avoid generating invalid perturbations. Considering the actual jamming situation where the user would turn on the jamming before he/she speaks, we introduce the constraint to the training to match the real deployment for stability improvement.

With our chunk-based perturbation structure, we prepend one chunk (half second) before the actual audio signal starts to perform the AE training (see Figure 8). In this way, the perturbation is more compatible with the way it will be utilized, thus reducing the probability of invalid and unstable perturbations.

We evaluate the effect of doing so with and without prepending training. In either condition, we generate perturbations 50 times and record the variation of JSR to calculate its standard deviation value (see Table XII in the demo website for full results). The training and evaluation setting are the same with the chunk perturbation training in Section 5.2 and the length of the prepend perturbation is half a second. Table 3 shows the mean and standard deviation of JSR of prepending training perturbation are 89.1% and 0.048, better than those of non-prepending training perturbation, which are 85.6% and 0.074, respectively. Therefore, we incorporate perturbation-prepend training into UniAP.

5.4 Robustness vs Targeted Noise Reduction

In an actual attack scenario, a strong adversary may perform targeted noise reduction before feeding a speech recording to an ASR to increase the recognition accuracy. In this section, we consider how to improve the robustness of our perturbation in case the opponent performs template cancellation and filtering on the recordings.

5.4.1 Initialization Statuses for UAP Generation

When there is only one UAP as the jamming noise, the adversary can easily obtain the noise as a template and subtract it from the recordings. Therefore, we discuss how to generate multiple non-targeted UAPs thereafter the adversary does not know what jamming noise to expect. In order to improve the number of UAP choices, we generate multiple UAPs from a variety of initialization status to expand the UAP candidate pool.

The frequency of different words/phrases appearing in English is different, which is represented in the English corpora used to train an ASR, we assume certain content can be recognized more easily. Based on this assumption, we train a good many UAPs with abundant (combinations of) words as different starting points, expecting our perturbation gets recognized prior to speech. These starting points belong to one of nine categories including sentence starter words, words containing multiple vowel/phoneme phonemes and their combination, etc. The jamming performance of trained UAPs in five categories are illustrated in Table 4. The performance of words combination is better than that of single word, and we think the initialization statuses of words combination provide more syllables for the training process to manipulate. Please check the demo website for full results.

TABLE 4: Perturbation performance under different initialization statuses. UAPs trained from different starting points achieve a CER of at least 77.1% with a JSR of at least 82.6%.

Initialization Status	CER	JSR	SNR (dB)
starter-combine	85.5%	88.9%	22.6
vowel-single	77.1%	82.6%	22.7
vowel-combine	79.0%	86.0%	22.8
consonant-single	77.9%	84.4%	22.7
consonant-combine	80.3%	85.8%	22.7

It is shown that all UAPs achieve a CER of at least 77.1%, and at least 82.6% of audio clips are transcribed to meaningless texts. This optimistic perturbation performance prove we are able to generate multiple UAPs with small distortion, thus relaxing the limit on the amount of noises that can be chosen. The flexibility in UAP choice makes it harder for the adversary to compromise the jamming through template (i.e., the copy of one of our UAPs) subtraction.

5.4.2 Frequency Matching via Mix Training

The adversary could filter recordings on the frequency domain and only keeps information within the speech range, thus compromising the jamming effect. To defend against such filtering, we empirically study DeepSpeech's frequency dependency. Through matching the frequency dependency of UAPs and DeepSpeech, we force the adversary into a zero-sum game to improve the robustness of UniAP.

Not all frequency range is equally important for an ASR to perform correct recognition. The adversary could filter out less important frequency part of the recording to reduce the jamming effect of noises, while the filtered signal still contains enough information for recognition. We heuristically validate the importance of various frequency bands between 0 and 8kHz (the Nyquist frequency when sampling frequency is 16 kHz). We randomly sample 500 speech signals from the *train-clean-100* subset of Librispeech dataset, filter out certain frequency part of them, and feed them to DeepSpeech for transcription. The CER of the transcriptions is calculated and the results classified according to the passed frequency range are shown in Table 5. A larger value of CER indicates the passed frequency range is of less importance to transcribing speech signals.

TABLE 5: Frequency dependency of DeepSpeech. The most sensitive frequency range is from 0 Hz to 4000 Hz.

Passed Band (Hz)	CER	Passed Band (Hz)	CER
$0 \sim 2000$	55.5%	$500 \sim 4000$	51.3%
$0 \sim 3000$	31.7%	$1000 \sim 4000$	95.7%
$0 \sim 3500$	18.8%	$2000 \sim 4000$	99.8%
$0 \sim 4000$	16.3%	$100 \sim 3500$	18.6%
$0 \sim 5000$	15.3%	$200 \sim 3500$	21.3%
$0 \sim 6000$	11.9%	$4000 \sim 8000$	100.0%

It is obvious that DeepSpeech is sensitive to frequency ranging from 0 Hz to 4000 Hz. By only keeping information from this range and filtering out other part, the adversary may reduce the effect of the perturbation. As shown in Table 6, the JSR of normal UAP reduces from 93.1% to 81.9% after filtering the perturbed signal. To address this issue, we first dedicately mask other frequency to generate perturbation within 0 to 4 kHz. In Table 6, the constraint training strategy results in JSRs of 78.4% and 92.3% in the normal and filtering scenario respectively. Although the robustness against filtering is improved, the performance in the normal condition declines. We further propose a strategy called mix training that randomly decides whether it filters the perturbed signals with a probability in the training. The mix training strategy achieves a more balanced performance, with JSRs of 86.3% and 91.2% in the normal and filtering condition, respectively.

TABLE 6: Jamming performance of normal UAP and UAP processed with frequency matching. CER-f and JSR-f mean CER and JSR of the filtered signals respectively. Filtering range is from 0 Hz to 4000 Hz. Mix training improves the JSR-f from 81.9% to 91.2%.

UAP	CER	JSR	CER-f	JSR-f	SNR (dB)
Normal Training	95.2%	93.1%	77.2%	81.9%	22.8
Constrained Training	70.0%	78.4%	83.2%	92.3%	23.0
Mix Training	76.9%	86.3%	80.8%	91.2%	22.9

5.5 Comparisons with Normal Noises

We compare UAPs with common noises in real life regarding jamming performance and the distortion level. Common noises are roughly categorized as white noise (e.g., engine noise and rainfall sound), pink noise (e.g., wind rustling sound and waves on a beach) and sudden noise (e.g., phone ring and car honking). We generate white noise and pink noise based on their mathematical models, and collect 105 sudden noises from the RWCP sound scene database [41]. We overlap different types of noises with speech samples from the test dataset in the time domain to simulate jamming. The mixed signals are directly fed into DeepSpeech for recognition. The average CER and JSR of the jammed signals are calculated.

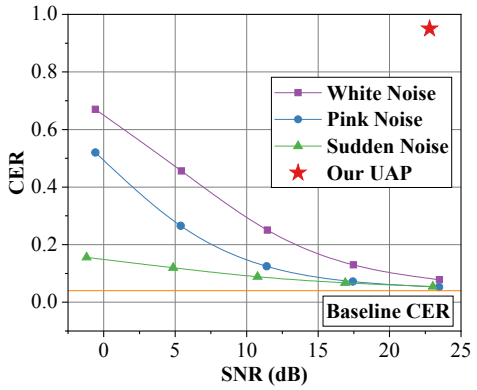

Fig. 9: The jamming performance comparison between common noises and our UAP.

The comparison results are categorized in noise type and shown in Figure 9. The horizontal orange line illustrate the CER of clear audios which is a baseline. We can see that as the power of noise increases, white noise has the best jamming effect among normal noise types, which achieves nearly 70% CER with -0.5 SNR. Our synchronization-free UAP has the best jamming effect, outperfoming white noise with a SNR of 22.8dB while achieving CER of 95%.

5.6 Random-Chunk UAPs

Our UAP so far is formed by repeated chunks, which could cause a problem: a dedicated attacker may discover the repeated pattern and accurately locate the chunk to recover the noise template and filter out the perturbation fully.

To address such an issue, instead of concatenating one chunk repeatedly, we utilize Algorithm 4 to generate UAPs formed with selected chunks, and each chunk is chosen randomly from our UAP pool. The valid perturbation chunks in the pool are generated from different initialization statuses (see 5.4.1) following Algorithm 3. As such, sufficient randomness is induced, and the jamming noise played each time is different, which greatly decreases the noise recovery possibility. Moreover, the UAP pool can be updated constantly with fresh perturbation chunks.

We assess the jamming performance of the random-chunk UAP with the same setting in Section 5.2. 1000 different random-chunk UAPs are generated to perturb the 1000 instances in the test set, which achieves the JSR of 84.9% and the CER of 88.4% with an SNR of 22.7dB. To validate the claimed randomness, we pick 10 chunks generated from different starting points and calculate the similarity between either two of them using the Pearson correlation coefficient (CC). The average CC is 0.003, which shows these chunks show little similarity. As a result, the random combinations of different chunks further enrich the randomness.

Algorithm 3 Generation of UAP Pool

Input: UAP Pool Size N
Output: UAP Pool P
  $i \leftarrow 0$
  $P \leftarrow \{\}$
  while $i < N$ **do**
    Training perturbation δ with different initialization statuses
    $P \leftarrow P \cup \{\delta\}$
    $i \leftarrow i + 1$
  end while

Algorithm 4 Usage of Random-Chunk UAPs

Input: UAP Pool P, Target Length L
Output: Random-Chunk UAP δ_r
  $\delta_r \leftarrow 0$
  while $length(\delta_r) < L$ **do**
    Randomly select δ from UAP Pool P
    $\delta_r \leftarrow concatenate(\delta_r, \delta)$
  end while
  $\delta_r \leftarrow \delta_r[0 : L - 1]$

6 Evaluation

In this section, we comprehensively evaluate the performance of our UAPs (we call it UniAP perturbations/noises alternatively). Note that UAPs used for evaluation in this section are trained without frequency matching because 0-4 kHz is different from the frequency range usually considered in the over-the-air enhancement.

6.1 Experimental Setup

Computing Environment: UniAP is implemented using Tensorflow 1.15.4 and trained by Adam optimizer, on a high-end server equipped with an NVIDIA RTX 3090 GPU and 252 GB RAM. **Metrics:** We rely on mentioned and two new metrics (see below) for evaluation. Besides, we also test ASR recognition performance on clean audios and use it as the baseline for benchmarking. **Parameter Configuration:** We use the same parameters as before (see section 5.2).

6.2 Additional Metrics

Sentence Similarity Scores. Using CER alone is insufficient in some occasions. For example, "how are you" and "how r u" represent the same meaning but the CER between them reaches 36%. Therefore, we introduce sentence similarity scores (SSS) as the supplementary metric to measure the jamming effectiveness. SSS is the cosine similarity between two sentence embeddings extracted with the model proposed by Google [42]. The smaller value indicates the more effective jamming[3].

Distortion. SNR measurement in physical playback is unstable and tedious (verified by experiments), thus we apply Decibels (dB) to quantify the distortion introduced by noise following other works [17]. The distortion $dB_x(\delta)$, calculated by $dB_x(\delta) = dB(\delta) - dB(x)$, shows the relative loudness of the UAP (δ) with respect to the speech (x), and the smaller value indicates less interference. As a reference, -31 dB is approximately the difference between the background noise and a person talking in a silent room [43].

3. The SSS between "how are you" and "how r u" reaches 0.82.

6.3 Digital Domain

Table 7 shows that our repeated-chunk UAPs (10 UAPs) achieve an average JSR of 90.4% on the Librispeech Test-clean set (more than 2000 clips), with an average SNR of 22.4dB (92.5% and 22.3 dB for the best UAP). Additionally, our UAPs also achieve an average JSR of 72.3% on the Gigaspeech Test set (nearly 20000 clips lasting for 40 hours) that is larger and quite different from our training set. The SNRs show that our perturbations are quieter than the AEs of CommanderSong [18] (18.6 dB) and AdvPulse [21] (13.7 dB). The performance of random-chunk UAPs is also analyzed, resulting in average JSRs of 87% and 69.6% on the Test-clean and Gigaspeech-test datasets. CERs of signals jammed by the two perturbations are also shown in Table 7. As a baseline, the average CER for clean audios are 5.7% and 20.2% on the two datasets. Overall, random-chunk UAPs maintain similar performance but show better robustness.

TABLE 7: Perturbation performance on large datasets. UAPs composed of repeated chunks achieve the JSR of 90.4% and 72.3% on the Librispeech Test-clean set and the Gigaspeech test set, while the UAPs composed of random chunks achieve 87.0% and 69.6%, respectively.

Dataset	Reapted Chunks			
	CER	JSR	SSS	SNR (dB)
Test-clean	87.7%	90.4%	0.23	22.4
Gigaspeech-test	69.2%	72.3%	0.27	25.5
Dataset	**Random Chunks**			
	CER	JSR	SSS	SNR (dB)
Test-clean	83.1%	87.0%	0.31	22.4
Gigaspeech-test	66.2%	69.6%	0.25	25.5

6.4 Over the Air

We assess the jamming effect of our perturbations in two physical playback scenarios where users' daily conversations could be covertly recorded.

- **Chatting.** 1) A user is chatting with her partner at home. The malicious microphone is located in the same room (e.g., on a smart TV) with them; 2) A user is having a video conference. The conference app developer illegally keeps the conversation. In both cases, a standalone jammer plays the UniAP noise by the users' side. Note that our chatting experiment setup covers both cases because there is no fundamental distinction between them regarding involved devices and distance settings.
- **Voice Call.** A user is making a voice call using a social media app on his/her smartphone, and a malicious app on the same phone activates itself to secretly record the conversation upon detecting the call activity. The user utilizes the speaker of the same phone to play our UniAP noise.

For the video conferencing scenario, we cannot use the built-in speaker of the computer to play back the jamming noise. The audio fed into the conference software by the OS does not contain the noise. We assume this is because the

TABLE 8: Perturbation performance in the chatting scenario. "UAP-N" means perturbations trained from different initialization states but without over-the-air enhancement, while "UAP-E" means perturbations with the enhancement. CER(V), JSR(V) and SSS(V) show the jamming performance on the vanilla DeepSpeech model. CER(R), JSR(R) and SSS(R) are tested on the robust DeepSpeech model. Our UAPs achieve high JSRs with the microphone deployed at all locations. They also perform well when the speaker is walking.

Perturbations	Location 1 (UMD = JMD = 1m)						Location 2 (UMD = JMD = 2m)						Location 3 (UMD = JMD = 3m)					
	CER(V)	JSR(V)	SSS(V)	CER(R)	JSR(R)	SSS(R)	CER(V)	JSR(V)	SSS(V)	CER(R)	JSR(R)	SSS(R)	CER(V)	JSR(V)	SSS(V)	CER(R)	JSR(R)	SSS(R)
UAP-N	62.8%	86.0%	0.19	61.9%	83.0%	0.24	71.9%	98.0%	0.15	73.1%	92.0%	0.18	62.9%	88.0%	0.19	60.3%	71.0%	0.23
UAP-E	65.8%	90.0%	0.19	61.9%	75.0%	0.23	75.1%	98.0%	0.16	72.4%	97.0%	0.18	65.8%	91.0%	0.18	62.8%	85.0%	0.21
Clean Audio	20.4%	N/A	0.59	17.1%	N/A	0.57	21.5%	N/A	0.51	17.7%	N/A	0.57	24.0%	N/A	0.49	17.0%	N/A	0.60
Perturbations	**Location 4 (UMD = JMD = 4m)**						**Location 5 (UMD = 0, JMD = 1m)**						**Location 6 (UMD = 0, JMD = 2m)**					
	CER(V)	JSR(V)	SSS(V)	CER(R)	JSR(R)	SSS(R)	CER(V)	JSR(V)	SSS(V)	CER(R)	JSR(R)	SSS(R)	CER(V)	JSR(V)	SSS(V)	CER(R)	JSR(R)	SSS(R)
UAP-N	68.3%	93.0%	0.17	68.6%	91.0%	0.20	67.4%	92.0%	0.17	63.2%	78.0%	0.21	63.7%	84.0%	0.18	58.4%	68.0%	0.23
UAP-E	70.5%	100.0%	0.15	67.2%	91.0%	0.20	70.1%	96.0%	0.17	63.5%	81.0%	0.20	70.2%	93.0%	0.16	63.1%	80.0%	0.22
Clean Audio	38.6%	N/A	0.28	26.7%	N/A	0.44	24.8%	N/A	0.45	17.8%	N/A	0.54	24.8%	N/A	0.45	17.8%	N/A	0.54
Perturbations	**Location 7 (UMD = 0, JMD = 3m)**						**Location 8 (UMD = 0, JMD = 4m)**						**Walking**					
	CER(V)	JSR(V)	SSS(V)	CER(R)	JSR(R)	SSS(R)	CER(V)	JSR(V)	SSS(V)	CER(R)	JSR(R)	SSS(R)	CER(V)	JSR(V)	SSS(V)	CER(R)	JSR(R)	SSS(R)
UAP-N	66.8%	89.0%	0.19	64.0%	83.0%	0.23	55.4%	64.0%	0.20	50.4%	50.0%	0.29	71.3%	100.0%	0.15	78.3%	96.0%	0.19
UAP-E	68.0%	92.0%	0.16	62.9%	78.0%	0.21	57.1%	70.0%	0.20	50.8%	53.0%	0.27	75.6%	99.0%	0.15	77.2%	99.0%	0.19
Clean Audio	24.8%	N/A	0.45	17.8%	N/A	0.54	24.8%	N/A	0.45	17.8%	N/A	0.54	34.3%	N/A	0.37	28.9%	N/A	0.46

audio fed into the app is from the phone call data channel, where the echo cancellation mechanism gets activated and eliminates the signal emitted by its speaker from the recorded sound. In contrast, for the voice call case, the audio data acquired by the malicious app does not undergo the echo cancellation process, and the noise is therefore kept. In both the chatting and voice call scenarios, the jamming noises shall experience distortions when a speaker plays them, propagate through the air and get recorded by a microphone. Recent studies [20], [21], [25], [38], [44] address the playback distortion by incorporating device limitations, channel effect and ambient noise into the perturbation generation stage. We follow a similar convention to enhance the robustness of our UAPs in the over-the-air condition, and the Aachen impulse response database [45] is used to include room impulse response (RIR) to handle the channel effect. Users could use the exact RIR of their room environment to enhance robustness further. However, reverberation distortion could be minor in our tasks. For the voice call scenario, the jamming distance is negligible because the jamming noises are both played and recorded by the smartphone's own speaker and microphone. Reverberation does not pose a strong effect when the jamming distance is short (e.g., shorter than 6 m) [20]. In addition, we assume that the unique structure and the perturbation-prepending training method of UniAP improve the robustness of our UAPs. Therefore, we evaluate our UniAP perturbations with and without over-the-air enhancement in both scenarios. To the best of our knowledge, we are the **first** to generate non-targeted UAPs that can attack a complex ASR over the air to protect users' speech privacy.

For the ease of explanation, we use JMD, UMD and UJD to refer to the distance between the jammer and a microphone, a user and a microphone, and a user and the jammer, respectively. All UAPs used for over-the-air evaluation are random-chunk perturbations. UAPs with (without) over-the-air enhancement during training are referred with the notation UAP-E (UAP-N). Since audios recorded over the air endure attenuation and reverberation, which downgrades the recognition accuracy of the DeepSpeech pre-trained model (i.e., vanilla DeepSpeech), we fine-tune the model with the reverberated audio data to obtain a more robust DeepSpeech, which performs better in a practical use case. We conduct experiments in the chatting and voice call scenarios in a quiet room with the sound of the air conditioner running and traffic outside the window being the background noises (37.5 dB).

6.4.1 Chatting Scenario

For the chatting scenario, we conduct experiments in two different settings (i.e., a static scenario and a walking scenario) in a bedroom, and we assume a vigilant user uses our UniAP jammer to protect his/her speech privacy.

Static Scenario. The vigilant user may put the jammer around him/her while the location of an eavesdropping microphone is unknown or while having a video conference using a laptop in front of him/her. We assume the UJD is 1m (i.e., putting the jammer too far away will reduce user's psychological sense of security). Because normal speech signals are not attenuated much over the air as the propagation distance increases [46], the jamming will be ineffective if the UAP is attenuated rapidly as the JMD increases. Therefore, we only vary JMD and maintain UMD = JMD to evaluate whether the UAP will experience serious attenuation over the air. Specifically, the JMD is set to be 1m, 2m, 3m and 4m (i.e., red number 1 to number 4 highlighted in Figure 10(a)).

Next we evaluate the disadvantageous settings when JMD is longer than UMD. Specifically, we conduct jamming experiments with the JMD varying from 1 m to 4 m (as shown in orange number 5 to number 8 in Figure 10a), while the UMD is almost zero. This represents an unlikely case where the eavesdropping microphone is besides the user's mouth and it poses a difficult challenge for jamming.

Walking Scenario. As shown by the green circle in Figure 10(a), the victim is speaking while walking back and forth along a straight line, and the one-way walking distance is about 2m. The carried jammer keeps playing UAPs. The recording device is set at the center of the table.

We use a common USB microphone as the recording device controlled by the adversary. Similar to AdvPulse [21], we use an Edifier m230 Bluetooth speaker to play speech signals (i.e., simulating the victim user) for better control and repeatability, and use a JBL-clip3 Bluetooth speaker to play our UniAP noises to jam the microphone. A sound

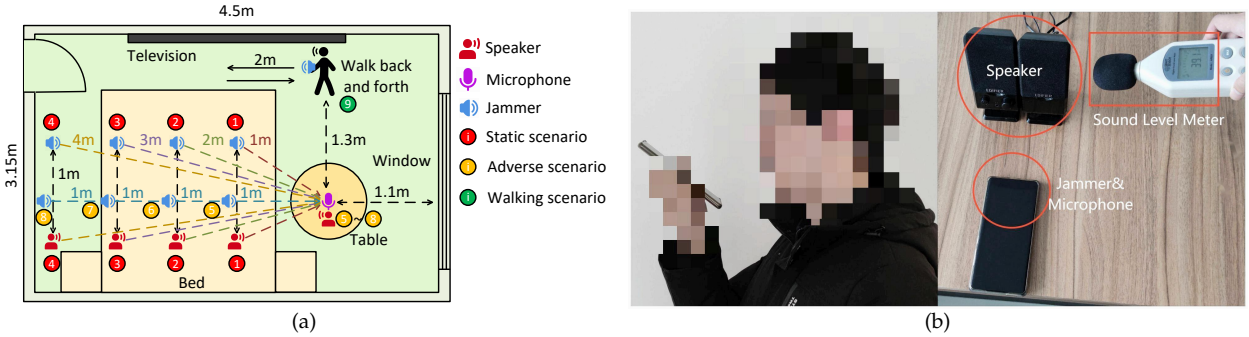

Fig. 10: Evaluation environment of two over-the-air scenarios: (a) chatting and (b) voice call.

level meter measures the loudness of clean audios and perturbations for calculating distortions. Our experiment simulates the scenario: the user would activate the jammer first to ensure her speech privacy is protected, then she could start talking at any moment. The speaker plays 10 speech clips (5 times for each one) without activating the jammer, and the recognition results of these clean audios serve as the baseline. Then the jammer plays a random-chunk UAP signal first, and the speaker starts playing one speech clip at a random point (controlled by an experimenter). Recording and playing speech clip start at the same time. 10 best-performing chunks on the digital domain and 7 different chunks with over-the-air enhancement are shuffled and played randomly by a software player to form the random-chunk perturbations. The experiment is repeated 10 times for each perturbation, which gives us 200 recordings. Metric values are calculated accordingly.

Results. When the UMD and the JMD are set to 1m, 2m, 3m and 4m, UAP-E can achieve JSRs of 90%, 98%, 91% and 100% on the vanilla DeepSpeech and 75%, 97%, 85% and 91% on the robust DeepSpeech as shown in Table 8, while the mean distortion of the UAP-N and the UAP-E are -41.2 dB and -40.9 dB, respectively (perceived by the speaker). For the adverse scenarios (i.e, UMD = 0 and JMD = 1m, 2m, 3m and 4m), UAP-E still achieve a high JSR, even the lowest one is 70% and 53% on vanilla and robust DeepSpeech respectively, and the distortion is 1-2 dB less than the former scenario. In addition, the SSSs of UAP-E in all locations are less than 0.27, which shows the jamming effectiveness.

For the walking scenario, Table 8 shows UAP-E achieves a JSR of 99% on the robust DeepSpeech, while the CER and SSS are of 77.2% and 0.19 respectively. As a baseline, the CER of clean audio recognition is 28.9%.

Remarks. UAP-E achieve equal or better jamming performance than the UAP-N in most cases. UAP-E is enhanced with reverberation simulation dataset. Its better performance indicates the sound absorption and reflection that happened during the multi-path acoustic propagation moderately weaken the jamming effect of UAPs. Still, UAP-N also shows close jamming effectiveness. Our perturbations achieve an impressive average JSR of 90.2% and 80.6% on the vanilla and robust DeepSpeech, respectively with a mean distortion of -39.7 dB. It is worth noting that the reported results are close to the worst case in practical use since the timing of playing speech and recording are synchronized. In practice, an attacker does not know when the user will speak, so he/she would probably record the noise for a while before the victim start talking. The noise segment ahead of the speech will greatly improve the CER.

TABLE 9: Perturbation performance in the voice call scenario. The L label means the loud perturbations while the Q label means the quiet ones. Both UAP-N(L) and UAP-E(L) perturbations achieve at least 49.6% CER and 53% JSR on both vanilla and robust DeepSpeech models.

Perturbations	CER(V)	JSR(V)	SSS(V)	CER(R)	JSR(R)	SSS(R)
UAP-N(L)	60.9%	70.0%	0.32	53.1%	59.0%	0.34
UAP-E(L)	56.2%	61.0%	0.30	49.6%	53.0%	0.34
UAP-N(Q)	54.3%	62.0%	0.42	50.9%	58.0%	0.42
UAP-E(Q)	57.7%	62.0%	0.33	50.7%	55.0%	0.38
Clean Audio	6.8%	N/A	0.85	9.4%	N/A	0.78

6.4.2 *Voice Call Scenario*

As shown in Figure 10b, a mobile phone plays the UAPs (acting as the jammer) while records audios at the same time, and the same 10 speech in the chatting scenario is used. The distance between the smartphone and the speaker is about 6 cm, similar to the distance between human mouth and a smartphone shown in the left half of Figure 10b. A sound level meter is also used for measuring the audio loudness. Distortion level of the perturbation perceived by the speaking person and at the receiver end are measured. The distortion on the speaker end relates to the speaking person's perception on the sound, while the receiver one presents the noise level in the recording. UAP-N and UAP-E perturbations are played with two volume levels of the smartphone, resulting the loud version of them, namely UAP-N(L) and UAP-E(L), and the quiet version, namely UAP-N(Q) and UAP-E(Q).

Table 9 shows that UAP-N(L) achieve JSRs of 70% and 59% on the vanilla and robust DeepSpeech, while UAP-E(L) show slightly worse performances. In these two cases of loud perturbation, the distortion level of the audio heard by the speaker is -45.0dB (i.e., almost imperceptible), while the distortion level of the recorded audio is -22.9dB. It is surprising that the quiet UAPs show similar jamming performance compared with the loud version, while achieving the distortion level of -31.3dB and -46.07dB in the recorded signals and on the speaker end, which ensure the low interference of noises. In particular, quiet UAP-N and UAP-E show similar jamming performance.

Remarks. In the voice call scenario simulating an adversarial app covertly recording users' VoIP call conversation, UAP-E show similar or worse performance than UAP-N,

which verifies our assumption that the reverberation effect may get compromised greatly due to the short distance between the speaker and the microphone. Besides, we conjecture the main propagation medium is the smartphone motherboard in the voice call scenario, which results in almost no obvious gaps regarding performance between the loud and quiet perturbations. We leave the verification in future work. Regarding the distortion level, the speaker-side distortion is less than -45 dB. Under this circumstance, the perturbation is even quieter than the working noise of the air conditioner in the experiment room (confirmed by measurement). Initially, we controlled the distortion level of recordings to be less than -23 dB to ensure the listener's comfort. However, we found the call receiver cannot hear the perturbations because of the echo cancellation mechanism. The perturbation noise played by the smartphone's speaker is subtracted from the recording, and only the speaking person's voice is kept and gets transmitted over the VoIP channel, while the malicious app can get the original recording, including both the speech and the perturbation, which is a great advantage since the comfort of the speaking person is the only limitation regarding the perturbation energy level.

6.5 User Study

We mainly evaluated the interference level of UAPs based on the SNR [18], [21] and the distortion level [17]. The average SNR of our digital-domain signal is about 22.4 dB, and it maintains lower distortion in the over-the-air case. Such numbers show great improvement over existing works. However, we would like to know users' subjective perception on our noise, therefore we conduct a user study to evaluate if users are bothered by the UAPs[4].

A dataset containing 20 noisy speech signals are first created. We choose 5 different speech signals from Librispeech test-clean and overlap each signal with 4 types of noises (white noise, pink noise, random event noise and our UAPs), resulting 20 noisy signals. The power of each type of noise is adjusted to achieve the similar CER of about 80% for a fair comparison. 10 individuals (5 males and 5 females) are recruited to participate in the survey. Each person is asked to listen to the 20 signals and give three scores to each audio regarding intelligibility of the speech content, willingness to tolerate the noise and the noise intensity. Higher score in each category indicates better intelligiblity, higher acceptability, and lower intensity. Table 10 shows our perturbations get 0.96, 0.74 and 0.89 in intelligibility, acceptability and intensity relatively, which is much higher than others in all aspects.

TABLE 10: User study results. UAPs get the highest scores in all categories: intelligibility, acceptability and intensity.

Perturbations	Intelligibility	Acceptability	Intensity
Ours	0.96	0.74	0.89
Event Noise	0.27	0.45	0.28
Pink Noise	0.34	0.36	0.32
White Noise	0.26	0.19	0.16

4. The study is approved by the IRB.

6.6 Robustness against Denoising Methods

General Denoising. We utilize spectral subtraction, a classic and powerful denoising solution to evaluate the robustness of our UAPs against general denoising technique. We choose the best UAP from the candidate pool, use the VOICEBOX on MATLAB as the denoising implementation, and use the same setting as the unsynchronization evaluation 5.2. After denoising, perturbed signals are fed into DeepSpeech. The results show a JSR of 60.0% , a CER of 60.5% , a WER of 101.8% and a SSS of 0.47 are achieved even with denoising. For comparison, these four values are respectively 88.4%, 84.9%, 137.4% and 0.36 without denoising. The optimisitic results show the robustness of our UniAP perturbations.

Targeted Denoising. A strong adversary may perform filtering or even template cancellation to get rid the effect of the UAPs. We have illustrated how UniAP is robust against these attacks in Section 5.4 and 5.6.

Dompteur. Dompteur, a recent study, proposes to use an augmentation module to extend any ASR system [47], which is a denoising method that restricts ASR to human voice frequencies and applies psychoacoustic modeling to remove the inaudible part. We evaluate UAP noises against the DeepSpeech augmented with Dompteur, and a good jamming results regarding CER of 95.8% and JSR of 92.7% are obtained (see Table 11). If a greater scaling factor is utilized to boost the denoising effect, higher CER of 100.1% and JSR of 98% are gained. The results are within expectation because we don't rely on frequency to develop the noises.

Beamforming. Multi-microphone beamforming is a sound source localization technique to improve speech quality by reducing reverberation and ambient noises. We assume our perturbations can still work even in the presence of beamforming because we mainly attack the vulnerability of an ASR rather than relying on energy masking to achieve jamming (e.g., white noise). We use SpeechBrain [48], a state-of-the-art speech toolkit, to perform delay-and-sum beamforming on voice call recordings. After beamforming, compared with the previous experiment shown in Table 9, UAP-N(Q) and UAP-E(Q) get JSRs of 46% and 48% on vanilla model with average CERs of 44% and 47% , while UAP-N(L) and UAP-E(L) get JSRs of 51% and 47% with average CERs of 46% and 43%. The results show that despite of the JSR decrease, the average jamming CERs are still close to 50%, indicating the perturbation robustness.

We conclude some robutness evaluation results in Table 11. Most results are tested with repeated-chunk UAPs, and the results in parentheses are tested with random-chunk ones. Our UAPs show inherently robustness against different kinds of denoising methods never seen in the training.

6.7 Generalization across LSTM-based Models

Considering LSTM-based ASRs work based on the context, our UAPs may have transferability over other LSTM ASRs. We consider two impact factors on the generalization test: different training dataset and different model archietecuture. First, we use dataset Gigaspeech [49] (the M subset) to train a DeepSpeech-architecture model, and validate the transferability of UAPs on this model that has different parameters than the DeepSpeech pre-trained one. For the ease of reference, we call it DeepSpeech_2. Next, we use

TABLE 11: Results with denoising methods and performance cross LSTM-based Models. CER, WER and JSR are shown in each scenario. Numbers in DeepSpeech column are the original jamming results. +Denoiser illustrates the jamming results after VOICEBOX denoising. +Dompteur shows the results with the latest Dompteur denoising. The last two columns present the transferability of UAPs on other LSTM models.

Perturbations	DeepSpeech			+Denoiser			+Dompteur			DeepSpeech_2			ESPnet		
	CER	WER	JSR	CER	WER	JSR	CER	WER	JSR	CER	WER	JSR	CER	WER	JSR
Clean Audio	4.0%	14.0%	N/A	N/A	N/A	N/A	N/A	N/A	N/A	8.0%	25.4%	N/A	1.5%	1.4%	N/A
Our UAPs	95.2%(84.9%)	149.4%(137.4%)	93.1%(88.4%)	55.3%(60.5%)	94.2%(101.8%)	52.8%(60.0%)	95.8%	151.2%	92.7%	78.7%	130.0%	83.0%	46.9%	49.8%	N/A

another LSTM ASR implemented in ESPnet toolkit [50] to validate the transferability of our UAPs on unseen LSTM-based model. We trained the ESPnet model with the entire training set of Librispeech. As a reference, the official RNN-based ESPnet achieves a WER of 4.0% on Librispeech test-clean while our ESPnet model achieved a WER of 4.2% on the same dataset, which shows our trained model is qualified for speech recognition tasks.

We first test the recognition performance of these two models on clean speech. The test-1000 dataset mentioned in Section 5.2 are used. As shown in Table 11, DeepSpeech_2 achieves a WER of 25.4% on the test-1000 dataset, while a WER of 1.4% is achieved by ESPnet. We pick one UAP trained from white noise as the initialization and follow the same test setting as Section 5.2 to evaluate the jamming effectiveness. Table 11 shows the CER and WER go up to 78.7% and 130.0% for DeepSpeech_2. As for ESPnet, they rise to 46.9% and 49.8%. These results validate the transferability of our UAPs. The SNR in both scenarios is 22.8 dB, which is acceptable.

To further improve the transferability, we generate the UAP on DeepSpeech and Espnet jointly by training each epoch alternatively on the two models. The joint-training perturbation achieves a better performance on ESPnet (a CER of 90.5% and a WER of 96.7%) and maintains a similar performance on DeepSpeech.

We also evaluate the effectiveness of the chunk-based UAP in the single-word recognition scenario. A command recognition DeepSpeech is trained with the Google command dataset [51]; we then validate the effectiveness of our UAP (the same one in digital domain evaluation) on the model. The UAP greatly increases the CER from 20.8% to 454.1%, achieving a JSR of 100% when the SNR is 29.23 dB.

7 Related Work

We mainly discuss the studies related to speech privacy preservation and speech AEs as they are the two main components of this work. AE studies targeting other modals like image, video [52] or traffic analysis [53] are not included.

Anti-eavesdropping Approaches. One potential method to protect speech content privacy is jamming microphones surrounding a user [7], [11]. However, these methods make the recorded speech incomprehensible to both ASR and humans, which stops normal voice activities. When they are applied in the video conference case of the chatting set up (see Section 6.4), the listener would seriously be bothered by the noisy speech. Additionally, they require ultrasonic speakers which are not integrated in COTS devices. Cohen-Hadr et al. [54] and Abdullah et al. [16] bring a third party that intercepts and modifies the signal to protect, but these work introduce a new trust issue and intercepting the audio signal is not applicable when defending against secret recordings. Chiquier [23] proposes a method that monitors a two-second audio clip and forecasts noise of 0.5 second to jam future speech. This work needs to record speech, which brings a potential privacy threat. Also, they did not prove the robustness of their method against unseen denoising techniques and user perceptions of their noise is unknown.

Other Speech Privacy-Preserving Methods. Intel's Software Guard Extensions (SGX) can limit the access to data stored within the secure hardware, but it cannot load a large speech recognition model due to memory size limitation [55], and it is vulnerable to side-channel attacks [56]; a recent work integrates cryptographic approaches (e.g., homomorphic encryption) with deep learning, but the conflict between accuracy and efficiency remains unresolved [57].

White-Box Speech Adversarial Examples. Studies on white-box speech AEs have been emerging. Most of these studies focus on targeted AE generation [17], [18], [19], [20], [21], [44]. They evolve from pure digital domain to over-the-air scenario. Li et al. [21] addresses the unsynchronization challenge, but they focus on targeted attack on speaker recognition and command classification tasks.

Universal Adversarial Perturbation (UAP). UAP was first proposed by Moosavi-Dezfooli et al. [29] on images. Li et al. [58] and Xie et al. [59] demonstrated the existence of the UAPs for the speaker recognition models. Lu et al. [60] and Zong et al. [61] explored the **targeted** audio-agnostic adversarial attack. However, the former had a low success rate towards models with CTC loss, and the latter only tried to perturb sentences lasting 2 to 4 seconds.

8 Discussion and Future Work

8.1 Transferability on Black Box ASRs

We tested the transferability of UniAP perturbations on other LSTM-based models. Now we further test the transferability on commercial speech-to-text (STT) APIs that are black boxes to us. With the same average SNR (23.01 dB), we directly feed speech signals (200 clips randomly chosen from Librispeech Train-clean-100) perturbed with our UAPs (4 perturbations including two different repeated-chunk UAPs, one random-chunk UAP and the joint-training UAP) and white noise to three ASR APIs, namely Amazon Transcribe [62], Google Speech-to-Text [63] and Xunfei ASR [64]. The WER of recognizing clean audios is applied as a comparison baseline. The jamming results in Figure 11 show our UAPs have limited jamming performance on commercial models. It is a huge challenge to train a white box UAP with a good transferability on proprietary STT engines. Nevertheless, we observe the joint training UAP

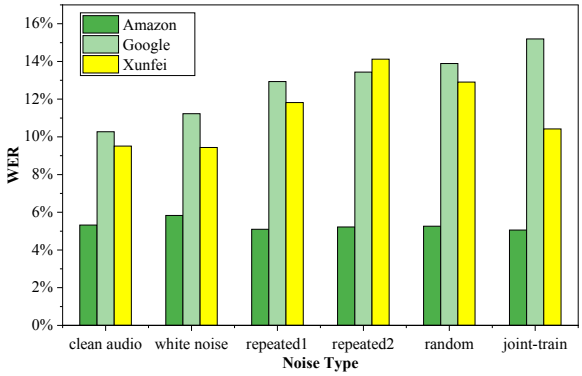

Fig. 11: Jamming performance on commercial ASRs.

and repeated-chunk UAP 2 achieve certain jamming effect on Google and Xunfei ASRs respectively. The joint training UAP achieves WER of 15.2% (10.3% for clean audio) on Google TTS, while the repeated-chunk UAP 2 achieves 14.1% (9.5% for clean audio) on Xunfei. Besides, all four types of UAPs have better jamming performance than white noise on Google and Xunfei ASRs.

Despite the limited effect on commercial STT, the optimistic transferability of UAPs cross LSTM-based models provides us a new way of thinking. We would explore the dominating factors of ASRs with other architectures possibly applied by commercial STTs, and construct UAPs disrupting the dominating factors of each ASR type. If an intersection exists between these factors, a non-targeted UAP effective on multiple black boxes may be possible.

8.2 Miscellaneous

We have implemented UniAP as an Android app that supports various activation modes, including event reservation, all-time jamming, and dedicated app/external jamming. The app enables updating the UAP pool and playing random-chunk UAP. It does not require permission to access the sensitive microphone data and only needs the ones related to its function, such as installing packages, foreground services, and reading phone states[5].

We generate UAP and UAP-E using the iterative optimization training paradigm (i.e., SGD) as shown in Algorithm 1 and the Iterative Fast Gradient Sign Method (I-FGSM), respectively. Besides, we do not consider algorithms only suitable for misleading classification models such as DeepFool [65]. We do not study which training paradigm is optimal and leave it for future work.

UniAP may be countered in the future by adversaries as attacks evolve. However, most privacy preserving systems may not be future-proof (e.g., RSA vs quantum computing). We believe it is important to provide a timely solution to protect user privacy from large-scale speech analysis, and this paper takes the first step in utilizing AEs to thwart automatic and large-scale content snoop by small companies.

9 Conclusion

In this paper, we present UniAP, to protect users from a privacy threat with the big data contexts - companies secretly record and transcribe users' daily conversations for commercial purposes via ubiquitous microphones. We utilize COTS speakers to emit quasi-perceptible noises to jam the speech spoken by users, such that users' voice activity is not affected while an ASR fails to recognize the speech content correctly. We first conduct empirical study to understand the key mechanisms of LSTM-based ASRs, then design AE training process to generate jamming noises which focus on disrupting the context information in ASR recogniton. Extensive experiments show the effectiveness of the UniAP perturbations on both the digital domain test and over-the-air evaluations. Moreover, our UniAP approach enhances the stability of jamming performance, improves the robustness against noise removal techniques, and shows good transferability over models based on LSTM.

5. Please check out the app specifics at the demo website.

References

[1] BBC, "Amazon hands over echo 'murder' data." http://www.bbc.com/news/technology-39191056, 2017.
[2] ——, "Amazon transcribe," http://www.bbc.com/news/technology-43725708, 2018.
[3] A. C. Estes, "Don't buy anyone an echo," https://gizmodo.com/dont-buy-anyone-an-echo-1820981732, 2017.
[4] voicebot.ai, "Nest secure's control hub has a microphone – users only found out when it became google assistant enabled this week," https://voicebot.ai/2019/02/07/nest-secures-control-hub-has-a-microphone-users-only-found-out-when-it-became-google-assistant-enabled-this-week/, 2019, online; accessed 16-Sep-2019.
[5] Bloomberg, "Is anyone listening to you on Alexa? a global team reviews audio," https://www.bloomberg.com/news/articles/2019-04-10/is-anyone-listening-to-you-on-alexa-a-global-team-reviews-audio, 10-Apr-2019, online; accessed 12-Aug-2021.
[6] T. Verge, "Yep, human workers are listening to recordings from google assistant, too," https://www.theverge.com/2019/7/11/20690020/google-assistant-home-human-contractors-listening-recordings-vrt-nws, 11-Jul-2019, online; accessed 12-Aug-2021.
[7] K. Sun, C. Chen, and X. Zhang, "" alexa, stop spying on me!" speech privacy protection against voice assistants," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems (SenSys '20)*, 2020, pp. 298–311.
[8] E. Commission, "2018 reform of eu data protection rules," European Commission, 2018. [Online]. Available: https://ec.europa.eu/commission/sites/beta-political/files/data-protection-factsheet-changes_en.pdf
[9] Office of the Attorney General, "California Consumer Privacy Act (CCPA)," https://oag.ca.gov/privacy/ccpa, 2018, online; accessed 23-April-2021.
[10] A. D. Blog, "What's new in android privacy," https://android-developers.googleblog.com/2021/05/android-security-and-privacy-recap.html, 2021.
[11] Y. Chen, H. Li, S.-Y. Teng, S. Nagels, Z. Li, P. Lopes, B. Y. Zhao, and H. Zheng, "Wearable microphone jamming," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1–12. [Online]. Available: https://doi.org/10.1145/3313831.3376304
[12] Y. Yang, J. West, G. K. Thiruvathukal, N. Klingensmith, and K. Fawaz, "Are you really muted?: A privacy analysis of mute buttons in video conferencing apps," *arXiv preprint arXiv:2204.06128*, 2022.
[13] N. Roy, H. Hassanieh, and R. Roy Choudhury, "BackDoor: Making microphones hear inaudible sounds," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '17)*. New York, NY, USA: ACM, 2017, pp. 2–14. [Online]. Available: http://doi.acm.org/10.1145/3081333.3081366
[14] N. Roy, S. Shen, H. Hassanieh, and R. R. Choudhury, "Inaudible voice commands: The long-range attack and defense," in *Proceedings of the 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI '18)*. Renton, WA: USENIX Association, Apr 2018, pp. 547–560. [Online]. Available: https://www.usenix.org/conference/nsdi18/presentation/roy

[15] G. Zhang, X. Ji, X. Li, G. Qu, and W. Xu, "Eararray: Defending against dolphinattack via acoustic attenuation." in *NDSS*, 2021.

[16] H. Abdullah, M. S. Rahman, W. Garcia, K. Warren, A. S. Yadav, T. Shrimpton, and P. 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)*, 2021, pp. 712–729.

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

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

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

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

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

[22] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," *CoRR*, vol. abs/1412.5567, 2014. [Online]. Available: http://arxiv.org/abs/1412.5567

[23] M. Chiquier, C. Mao, and C. Vondrick, "Real-time neural voice camouflage," in *International Conference on Learning Representations*, 2022. [Online]. Available: https://openreview.net/forum?id=qj1IZ-6TInc

[24] M. Alzantot, B. Balaji, and M. B. Srivastava, "Did you hear that? adversarial examples against automatic speech recognition," *CoRR*, vol. abs/1801.00554, 2018. [Online]. Available: http://arxiv.org/abs/1801.00554

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

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

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

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

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

[30] J. Vadillo and R. Santana, "Universal adversarial examples in speech command classification," *CoRR*, vol. abs/1911.10182, 2019. [Online]. Available: http://arxiv.org/abs/1911.10182

[31] C. Olah, "Understanding LSTM Networks," https://colah.github.io/posts/2015-08-Understanding-LSTMs, 2015, online; accessed 22-Dec-2021.

[32] M. AI, "Mycroft," https://mycroft.ai/, 2016.

[33] B. USA, "Swiftscribe," https://swiftscribe.ai/, 2017.

[34] J. Lau, B. Zimmerman, and F. Schaub, "Alexa, are you listening?: Privacy perceptions, concerns and privacy-seeking behaviors with smart speakers," *Proc. ACM Hum.-Comput. Interact.*, vol. 2, no. CSCW, pp. 102:1–102:31, Nov 2018. [Online]. Available: http://doi.acm.org/10.1145/3274371

[35] M. Zhou, Q. Wang, J. Yang, Q. Li, F. Xiao, Z. Wang, and X. Chen, "Patternlistener: Cracking android pattern lock using acoustic signals," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS'18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1775–1787. [Online]. Available: https://doi.org/10.1145/3243734.3243777

[36] P. C. Loizou, *Speech enhancement: theory and practice*. CRC press, 2007.

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

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

[39] J. Szurley and J. Z. Kolter, "Perceptual based adversarial audio attacks," 2019.

[40] Mozilla, "Project DeepSpeech," https://github.com/mozilla/DeepSpeech, 2017.

[41] S. Nakamura, K. Hiyane, F. Asano, T. Nishiura, and T. Yamada, "Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition," in *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*. Athens, Greece: European Language Resources Association (ELRA), May 2000. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2000/pdf/356.pdf

[42] D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y. Sung, B. Strope, and R. Kurzweil, "Universal sentence encoder," *CoRR*, vol. abs/1803.11175, 2018. [Online]. Available: http://arxiv.org/abs/1803.11175

[43] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*. USA: California Technical Publishing, 1997.

[44] Y. Qin, N. Carlini, I. Goodfellow, G. Cottrell, and C. Raffel, "Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition," *arXiv e-prints*, p. arXiv:1903.10346, Mar 2019.

[45] M. Jeub, M. Schafer, and P. Vary., "A binaural room impulse response database for the evaluation of dereverberation algorithms," in *International Conference on Digital Signal Processing*, 2009, pp. 1–5.

[46] ISO, "Calculation of the absorption of sound by the atmosphere," 1996.

[47] T. Eisenhofer, L. Schönherr, J. Frank, L. Speckemeier, D. Kolossa, and T. Holz, "Dompteur: Taming audio adversarial examples," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 2309–2326. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/eisenhofer

[48] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," 2021, arXiv:2106.04624.

[49] G. Chen, S. Chai, G.-B. Wang, J. Du, W.-Q. Zhang, C. Weng, D. Su, D. Povey, J. Trmal, J. Zhang, M. Jin, S. Khudanpur, S. Watanabe, S. Zhao, W. Zou, X. Li, X. Yao, Y. Wang, Z. You, and Z. Yan, "GigaSpeech: An Evolving, Multi-Domain ASR Corpus with 10,000 Hours of Transcribed Audio," in *Proc. Interspeech 2021*, 2021, pp. 3670–3674.

[50] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-End Speech Processing Toolkit," in *Proc. Interspeech 2018*, 2018, pp. 2207–2211.

[51] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *CoRR*, vol. abs/1804.03209, 2018. [Online]. Available: http://arxiv.org/abs/1804.03209

[52] J. Bai, B. Chen, D. Wu, C. Zhang, and S.-T. Xia, "Universal adversarial head: Practical protection against video data leakage," in *ICML 2021 Workshop on Adversarial Machine Learning*, 2021.

[53] M. Nasr, A. Bahramali, and A. Houmansadr, "Defeating DNN-Based traffic analysis systems in Real-Time with blind adversarial perturbations," in *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 2705–

2722. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/nasr

[54] A. Cohen-Hadria, M. Cartwright, B. McFee, and J. P. Bello, "Voice anonymization in urban sound recordings," in *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2019, pp. 1–6.

[55] F. Brasser, T. Frassetto, K. Riedhammer, A.-R. Sadeghi, T. Schneider, and C. Weinert, "Voiceguard: Secure and private speech processing," in *Proc. Interspeech 2018*, 2018, pp. 1303–1307. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-2032

[56] A. Nautsch, A. Jiménez, A. Treiber, J. Kolberg, C. Jasserand, E. Kindt, H. Delgado, M. Todisco, M. A. Hmani, A. Mtibaa *et al.*, "Preserving privacy in speaker and speech characterisation," *Computer Speech & Language*, vol. 58, pp. 441–480, 2019.

[57] A. Sanyal, M. Kusner, A. Gascon, and V. Kanade, "Tapas: Tricks to accelerate (encrypted) prediction as a service," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4490–4499.

[58] J. Li, X. Zhang, C. Jia, J. Xu, and W. Gao, "Universal adversarial perturbations generative network for speaker recognition," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*, 2020.

[59] Y. Xie, C. Shi, Z. Li, J. Liu, Y. Chen, and B. Yuan, "Real-time, universal, and robust adversarial attacks against speaker recognition systems," *IEEE*, 2020.

[60] Z. Lu, W. Han, Y. Zhang, and L. Cao, "Exploring targeted universal adversarial perturbations to end-to-end asr models," 2021.

[61] W. Zong, Y.-W. Chow, W. Susilo, S. Rana, and S. Venkatesh, "Targeted universal adversarial perturbations for automatic speech recognition," in *Information Security*, J. K. Liu, S. Katsikas, W. Meng, W. Susilo, and R. Intan, Eds. Cham: Springer International Publishing, 2021, pp. 358–373.

[62] Amazon, "Amazon transcribe," {https://aws.amazon.com/cn/transcribe/?nc2=h_ql_prod_ml_ts}, 2022.

[63] Google, "Speech-to-text conversion powered by machine learning," https://cloud.google.com/speech-to-text, 2021.

[64] iFLYTEK, "iflytek speech transcribe," https://www.xfyun.cn/services/lfasr, 2022.

[65] S. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2016, pp. 2574–2582. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.282