Enabling Fast and Universal Audio Adversarial Attack Using Generative Model

Yi Xie[1], **Zhuohang Li**[2], **Cong Shi**[1], **Jian Liu**[2], **Yingying Chen**[1], **Bo Yuan**[1]
[1]Rutgers University
[2]The University of Tennessee, Knoxville
yi.xie@rutgers.edu, zli96@vols.utk.edu, cs1421@winlab.rutgers.edu,
jliu@utk.edu, yingche@scarletmail.rutgers.edu, bo.yuan@soe.rutgers.edu

Abstract

Recently, the vulnerability of deep neural network (DNN)-based audio systems to adversarial attacks has obtained increasing attention. However, the existing audio adversarial attacks allow the adversary to possess the entire user's audio input as well as granting sufficient time budget to generate the adversarial perturbations. These idealized assumptions, however, make the existing audio adversarial attacks mostly impossible to be launched in a timely fashion in practice (e.g., playing unnoticeable adversarial perturbations along with user's streaming input). To overcome these limitations, in this paper we propose fast audio adversarial perturbation generator (FAPG), which uses generative model to generate adversarial perturbations for the audio input in a single forward pass, thereby drastically improving the perturbation generation speed. Built on the top of FAPG, we further propose universal audio adversarial perturbation generator (UAPG), a scheme to craft universal adversarial perturbation that can be imposed on arbitrary benign audio input to cause misclassification. Extensive experiments on DNN-based audio systems show that our proposed FAPG can achieve high success rate with up to 214× speedup over the existing audio adversarial attack methods. Also our proposed UAPG generates universal adversarial perturbations that can achieve much better attack performance than the state-of-the-art solutions.

Introduction

As the current most powerful artificial intelligence (AI) technique, deep neural networks (DNNs) have been widely adopted in many practical applications. Despite their current success and popularity, DNNs suffer from several severe limitations, especially the inherent high vulnerability to adversarial attack (Goodfellow, Shlens, and Szegedy 2014; Carlini and Wagner 2017), a very harmful attack approach that imposes well-crafted adversarial perturbation on the benign input of DNNs to cause misclassification. Being originally discovered in the image classification applications, to date the vulnerability of DNNs, especially various types of adversarial perturbation generation methods (Kurakin, Goodfellow, and Bengio 2016; Poursaeed et al. 2018; Moosavi-Dezfooli et al. 2017), has been extensively investigated in many image-domain applications.

Considering the rapidly increasing use of DNNs in modern audio-domain applications and systems, such as smart speaker (e.g., Apple Homepod, Amazon Echo) and voice assistant (e.g., Siri, Google Assistant, Alexa), recently both machine learning and cybersecurity communities have begun to study the possibility of adversarial attack in the audio domain. Some pioneering efforts (Carlini and Wagner 2018; Neekhara et al. 2019) in this topic have demonstrated that the idea of injecting inconspicuous perturbations into benign voice inputs to mislead the DNN-based audio systems is not just conceptually attractive but also practically feasible. To date, several works have reported the successful adversarial attacks in different audio-domain applications, including but not limited to speaker verification (Kreuk et al. 2018; Chen et al. 2019), speech command recognition (Alzantot, Balaji, and Srivastava 2018; Gong et al. 2019), speech-to-text transcription (Carlini and Wagner 2018; Yuan et al. 2018), and environmental sound classification (Abdoli et al. 2019).

Limitations of Prior Work. Although the existing work have already demonstrated the feasibility of audio adversarial attack, they are still facing several challenges. More specifically, the state-of-the-art audio adversarial attack approaches make several idealized assumptions on the attacking setting: 1) *Having large time budget for generating adversarial perturbation.* In practical audio applications, the benign inputs are typically quickly-streaming voice input. Therefore, due to such temporal constrain, the existing audio adversarial attacks, which rely on time-consuming iterative optimization approaches such as C&W (Carlini and Wagner 2018) or genetic algorithms (Alzantot, Balaji, and Srivastava 2018), are too slow to launch the attack against these real-time audio processing systems; 2) *Owning authorization to observe the context of the benign input.* Since the existing perturbation generation methods require to preknow the full content of the ongoing voice input, the inherent sequential nature of audio signals makes it impossible for the adversary to generate adversarial perturbation during input-streaming phase. Consequently, the current audio adversarial attack can only be performed against the recorded or playback voice instead of real-time audio signals, thereby making them impractical for various real-world audio-domain attacking scenarios.

Technical Preview and Contributions. To overcome these limitations, in this paper we propose to use generative model to produce adversarial perturbations in the audio domain. This generative model learns the distribution of adversarial perturbations from training data in an offline way. Once being well-trained, the generative model can generate audio adversarial perturbations very quickly, thereby unlocking the possibility of realizing audio adversarial attack in the real-time setting. Our main contributions of this paper are summarized as follows:

- We, for the first time, propose a generative model-based fast audio adversarial perturbation generator (FAPG). Unlike existing methods requiring considerable adversarial perturbation generation time, our proposed FAPG generates the desired audio adversarial perturbation through a well-trained generative model Wave-U-Net (Stoller, Ewert, and Dixon 2018) in a single forward pass, thereby greatly accelerating the perturbation generation speed.
- We propose to integrate a set of trainable class-wise embedding feature maps into FAPG to encode all the label information in the audio data to a unified model. Unlike conventional generative model-based image-domain adversarial attacks, which require different generative models for different targeted classes, the proposed audio-domain FAPG can generate adversarial perturbation targeting at any adversary-desired class using a single generator model. Such reduction significantly saves the memory cost and model training time if the adversary expects to launch attacks with multiple target classes.
- Built on top of the input-dependent FAPG, we further propose an input-independent universal audio adversarial perturbation generator (UAPG). UAPG is able to generate a single *universal audio adversarial perturbation* (UAP), which can be applied and re-used on different benign audio inputs without the need of input-dependent perturbation re-generation. In addition, since the universality of UAP exists across different benign inputs, such important characteristic removes the prior constraint on needing to observe the entire input for perturbation generation, thereby enabling the realization of real-time audio adversarial attack.
- We evaluate the attack performance using FAPG and UAPG against three DNN-based audio systems: speech command recognition model on the Google Speech Commands dataset (Warden 2018), speaker recognition model on VCTK dataset (Christophe, Junichi, and Kirsten 2016) and environmental sound classification model on UrbanSound8k dataset (Salamon, Jacoby, and Bello 2014). Compared with the state-of-the-art input-dependent attack, our FAPG-based attack achieves $214\times$ speedup with the comparable success rate. Compared with the existing input-independent (universal) attack, our UAPG-based attack achieves 37.22% and 29.98% fooling rate increase in white-box setting and black-box setting, respectively.

Fast Audio Adversarial Perturbation Generator (FAPG)

Motivation

Dilemma Between Speed and Performance. Despite the current progress of the existing audio adversarial attacks, as analyzed in the Introduction, one of the most challenging limitations is their inherent slow generation process for adversarial perturbations. This is because: 1) the current commonly adopted underlying adversarial perturbation-generating approaches, such as PGD (Madry et al. 2017), C&W (Carlini and Wagner 2018) and genetic algorithms (Alzantot, Balaji, and Srivastava 2018), are built on numbers of iterations to optimize or search the perturbations. Although this iterative mechanism can bring high attack performance, the corresponding required generation time is prohibitively long, such as seconds or even hours for producing one well-crafted perturbation. 2) Reducing the number of iterations to make generation time satisfy the real-time requirement is an alternative solution; however, as shown in our experiments that will be reported later, when the iteration-based attack method is performed in a restricted time budget, the corresponding attack performance is severely degraded. 3) On the other hand, the existing one-step perturbation generation methods, e.g. FGSM (Goodfellow, Shlens, and Szegedy 2014), though enjoy the advantage on fast generation, suffer from the poor attack performance limitation – they typically have much lower attack success rates than their iteration-based counterparts.

Why Fast Perturbation Generation Matters? Some readers may have questions about the necessity and motivation of the fast generation of adversarial perturbations. Why should the perturbations be generated in a real-time manner? Cannot the attacker just record the benign voice input, generate the perturbation offline under a sufficient time budget, and then play the generated adversarial audio? Indeed, the above hypothesized attacking strategy may fit some time-budget-relaxed scenarios; however, in practical attacking scenarios, it is more likely that the attacker does not have many opportunities to approach the victim for either recording speech or altering the victim's speech on the fly. If there is a chance, the attacker might want to record the speech, then instantly generate the adversarial perturbation (preferably using their mobile devices) and inject it onto the victim's interactive speech on the spot. This would leave a very limited time budget and computational resource for the process of perturbation generation and injection. Thus, an efficient way to craft robust adversarial perturbations in a very timely and low computational complexity manner is highly desirable.

Generative Model-based Solution in Image Domain. The above demand for fast adversarial perturbation generation is not an audio-specific problem, but also widely exists in the image domain. To satisfy this timing requirement, recent image-domain studies (Poursaeed et al. 2018; Song et al. 2018; Phan et al. 2020) have proposed to utilize generative models, such as Generative adversarial network (GAN) (Goodfellow et al. 2014) and autoencoder (Vincent et al. 2008), to accelerate the generation of image adversarial perturbations. Different from the multi-step optimization-based approaches (e.g. C&W and PGD), the generative model-based solutions aim to learn the distribution of adversarial perturbations from the training images. After being well-trained, the generative model performs one-step generation from input image to adversarial perturbation, where such

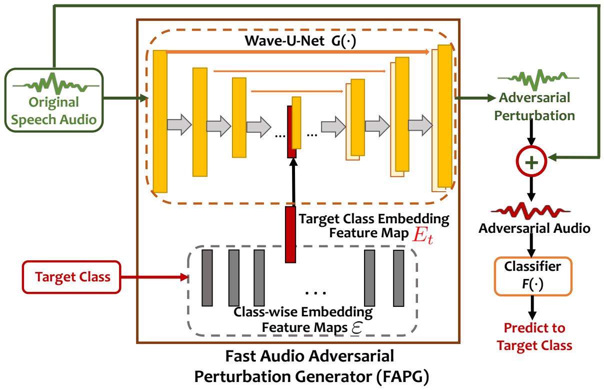
Wave-U-Net $G(\cdot)$
Original Speech Audio
Adversarial Perturbation
Adversarial Audio
Classifier $F(\cdot)$
Predict to Target Class
Target Class Embedding Feature Map E_t
Target Class
Class-wise Embedding Feature Maps ε
Fast Audio Adversarial Perturbation Generator (FAPG)

Figure 1: Overall architecture of the proposed FAPG.

process is essentially a fast one-pass forward propagation over the generative model, thereby significantly improving the generation speed for image adversarial perturbations.

Challenges in Audio Domain. Such progress on image domain naturally encourages the exploration of using generative model to accelerate audio adversarial perturbation generation. However, audio signals have a huge difference from images. A speaker's voice is essentially a 1-D time-serial signal that contains very important sequential order information. Also, unlike well-defined fixed-size image data, voice data typically have very different signal lengths even from the same user and in the same dataset. Besides these new audio-specific challenges, generative model-based audio adversarial perturbation also suffers the same class-specific model preparation problem of image-based counterparts. To be specific, when utilizing generative model to perform targeted attack, for each target class, an individual generative model has to be trained for specific use. Considering the number of classes can be very high, e.g., hundreds or even thousands, the required memory cost for launching the attack is very high.

Proposed FAPG: Construction & Training

Overall Architecture. To address these challenges, we propose FAPG, a fast audio adversarial perturbation generator, to launch the audio-domain adversarial attack in a rapid, high-performance and low-memory-cost way. Figure 1 illustrates the overall architecture of FAPG, which contains a generative model ($G(\cdot)$), e.g., Wave-U-Net (Stoller, Ewert, and Dixon 2018), and multiple class-wise embedding feature maps. During the training phase, both the generative model and embedding feature maps are jointly trained on the training dataset. After proper training, given a benign audio input and a target class label y_t that the adversary plans to mislead the DNN classifier ($F(\cdot)$) to, the corresponding audio adversarial perturbation can be quickly generated via performing inference of the benign input over the well-trained generative model, where the embedding feature map for the target class y_t is concatenated to one intermediate feature map of $G(\cdot)$. Next, we describe the details of the used generative model and the set of embedding feature maps as follows.

Audio-specific Generative Model. Generative model is the core component of FAPG. Although various types of generative models have been widely used in image-domain applications, they are not well-suited for the use in FAPG due to the inherent difference between image and audio signals (e.g., sequence order and varying length). To address these challenges, we adopt Wave-U-Net (Stoller, Ewert, and Dixon 2018), which was originally used for audio source separation, as the underlying generative model of FAPG. Wave-U-Net is a special type of CNN containing 1-D convolutional, decimal down-sampling blocks and linear interpolation up-sampling blocks. Such inherent encoder-decoder structure makes Wave-U-Net exhibit strong distribution modeling capability. Meanwhile, its unique design of first-layer 1-D convolution and up/down sampling blocks also enables Wave-U-Net can naturally capture the temporal information from 1-D varying-length data.

Class-wise Embedding Feature Maps. The purpose of using k-class embedding feature maps is to ensure that a single generative model can be re-used for attacking against different target classes instead of class-specific design. To this end, those class-aware embedding feature maps, denoted as $\varepsilon = \{E_1, E_2, ...E_k\}$, are designed to be trainable, and each of them corresponds to one target class. After joint training of generative model $G(\cdot)$ and these embedding feature maps ε, the label information for class y_t is encoded in the corresponding feature map E_t. Then during the generation phase E_t is concatenated with one intermediate feature map of $G(\cdot)$ to craft the adversarial perturbation for the target class y_t. In our design, E_t has the exact same shape of the intermediate feature map to which it will be concatenated. To be specific, E_t is typically aligned with the intermediate feature map at the intersection between the encoder and decoder parts of Wave-U-Net. This is because the feature map has the smallest size at this position, and thereby minimizing the storage cost of the corresponding E_t.

Training Procedure of FAPG. Next we describe the training procedure of FAPG, or more specifically, the joint training for $G(\cdot)$ and ε. In the forward propagation phase of the entire training procedure, for each batch of input voice data $\boldsymbol{X}$, we first randomly select one target class y_t, and fetch the corresponding embedding feature map E_t. This selected feature map is concatenated into the generative model $G(\cdot)$ to form an overall model $G_t(\cdot)$. A forward pass on $G_t(\cdot)$ will be performed with input $\boldsymbol{X}$. The result, denoted as $\boldsymbol{\delta_t}$, is clipped to the range of $\{-\tau, +\tau\}$ to constrain the generated perturbation $\boldsymbol{\delta_t}$ to be imperceptible, where τ is a threshold parameter. Notice that according to our experiments, τ should be set as a relatively large value initially, and gradually decreased during the training procedure. Empirically such adjusting scheme can bring better training convergence.

After perturbation $\boldsymbol{\delta_t}$ is calculated from the generative model, it is imposed on the benign data to form the adversarial input, which can cause the misclassification of DNN classifier $F(\cdot)$. Then, the loss function, which is the key of the entire training procedure, is formulated as follows:

Algorithm 1: Training Procedure of FAPG

```
Require: Training dataset $\chi = \{x^{(1)}, ..., , x^{(n)}\}$, class
  label $\{y_1, ..., , y_k\}$, DNN classifier $F(\cdot)$, noise constraint
  constant $\tau$
Result: Trained FAPG: generative model $G(\cdot)$, class-wise
  embedding feature maps $\varepsilon = \{E_1, ..., E_k\}$
Initialize $G(\cdot)$, $\varepsilon$ and $\tau$
for number of training iterations do
    for number of steps do
        $\boldsymbol{X} \leftarrow$ minibatch of $m$ samples from $\chi$;
        $y_t \leftarrow get\_random\_target \in \{y_1, ..., y_k\}$;
        $G_t(\cdot) \leftarrow G(\cdot)$ embeds with $E_t \in \varepsilon$;
        $\boldsymbol{\delta_t} \leftarrow Clip(G_t(\boldsymbol{X}), \{-\tau, +\tau\})$;
        $\boldsymbol{X'} \leftarrow \boldsymbol{X} + \boldsymbol{\delta_t}$;
        $\boldsymbol{y_{pred}} \leftarrow F(\boldsymbol{X'})$;
        Loss $\leftarrow$
          $\frac{1}{m}\sum_i^m(\text{CrossEntropy}(\boldsymbol{y_{pred}^{(i)}}, y_t) + \beta \cdot \left\|\boldsymbol{\delta_t^{(i)}}\right\|_2)$;
        minimize Loss to update $G(\cdot)$ and $E_t$;
        decrease $\tau$
    end
end
```

$$Loss(\boldsymbol{X}, y_t) = -y_t \cdot log(F(\boldsymbol{X} + G_t(\boldsymbol{X}))) + \beta \cdot \|G_t(\boldsymbol{X})\|_2, \tag{1}$$

where the first and second terms are the cross-entropy loss and L_2 loss, respectively, and β is a pre-set coefficient. The existence of L_2 loss in the entire loss function is to control the attack strength and make the generated adversarial perturbation imperceptible.

Consequently, in the backward propagation phase both the generative model $G(\cdot)$ and the current selected embedding feature map E_t are updated simultaneously by minimizing the loss function. Notice that for each batch of data, E_t is randomly selected. Therefore after rounds of iterations the generative model $G(\cdot)$ itself learns the general distribution of adversarial perturbations, and different E_t learns the encoded information for each specific target class. The details of the entire FAPG training procedure are summarized in Algorithm 1.

Universal Audio Adversarial Perturbation Generator (UAPG)

Motivation

Reducing the Observation of Full Content – Why It Matters? As presented in the previous section, FAPG provides a fast solution to generate audio adversarial perturbations. However, it is essentially an input-dependent generating approach. In fact, most of the state-of-the-art adversarial attack methods, in both audio and image domains, belong to the input-dependent attack category. In other words, the underlying perturbation generation mechanism is based on the observation of the entire benign input. Although such assumption may hold for most image processing applications, it is very challenging to satisfy such requirement in the practical real-time audio applications. This is because audio signals have inherent temporal sequence, it is unrealistic to pre-know the full content of the ongoing voice input during the input-streaming phase. In other words, the attacks can only be performed against the recorded or playback voice, thereby severely limiting the attacking feasibility and scenarios. Consequently, besides significantly reducing the perturbation generation time, a practical audio adversarial attack should also reduce the demand of observing the content of benign input as low as possible.

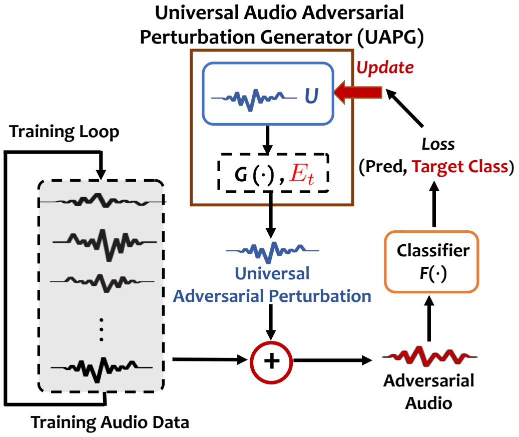

Figure 2: Overall architecture and training scheme of the proposed UAPG.

Universal Audio Adversarial Perturbation Generator (UAPG). To achieve this, we further develop universal audio adversarial perturbation generator (UAPG) to craft audio-domain universal adversarial perturbation (UAP). As revealed by its name, a single universal adversarial perturbation can be applied and re-used on different benign inputs to cause mis-classification without the need of input-dependent perturbation re-generation. Such unique universality completely removes the prior constraint on observation of the entire input, and makes UAPG very suitable to launch real-time audio adversarial attack with zero time cost.

Challenges of UAPG Design. The attractive benefits of UAP have already led to some efforts on studying image-specific UAP (Moosavi-Dezfooli et al. 2017; Poursaeed et al. 2018). Lending the methodology used in those research progress in image domain, recent works (Vadillo and Santana 2019; Neekhara et al. 2019) report audio-domain UAP generating methods for speech command recognition and speech-to-text systems, respectively. Besides, (Gong et al. 2019) also proposes a technique to realize real-time audio adversarial attack without using the entire voice input, which has the similar effect as using UAP.

Despite these existing efforts, designing a robust and powerful UAPG is still non-trivial but facing two main challenges: 1) the experimental results show that the current audio-domain UAPs typically have much lower attack performance than the input-dependent perturbations; and 2) the attack enabled by some audio-domain UAPs are only untargeted attack, where the adversary cannot precisely obtain the desired target results.

Proposed UAPG: Construction & Training

Overall Scheme. Different from the existing studies, we aim to devise a UAPG which can achieve high targeted attack performance. Figure 2 shows the key idea: we produce an input-dependent UAP based on a signal vector U, which is to be trained to exhibit a certain degree of universality. After initialization, U is used to produce UAPs, and it is updated in an iterative way by gradually improving the universality of the derived UAPs across different training data samples. Finally, an effective UAPG is able to be constructed via evolving well-trained U.

From FAPG to UAPG. The underlying method used for generating UAPs is our proposed FAPG. Intuitively, FAPG learns to estimate the distribution of adversarial perturbations instead of iteratively optimizing the perturbation for a specific audio input. Therefore the FAPG-generated perturbation naturally exhibits better universality than that the one comes from the non-generative method. Moreover, our FAPG is designed to integrate various target classes information into a single generative model, thereby enabling the capability of producing targeted universal perturbations.

Training Procedure of UAPG. We then introduce the training details to facilitate an effective UAPG. In general, to formulate an input-agnostic universal attack, our goal is to find a universal perturbation υ_t to satisfy:

$$argmax\ F(\boldsymbol{x^{(i)}} + \upsilon_t) = y_t \text{ for most } x \ \sim \ \chi. \qquad (2)$$

The training procedure of UAPG is shown in Algorithm 2. We aim to generate a single universal perturbation υ_t via the well-trained $G(\cdot)$ and the corresponding $E_t \in \varepsilon$, which can be obtained from the well-trained input-dependent FAPG. Different from input-dependent scenario, the audio input signal is now replaced by a single trainable vector U. Then the universal perturbation is returned and imposed on the benign data to craft the adversarial audio example. Through feeding such an adversarial audio into the DNN classifier F, we can update U by minimizing the following loss function:

$$Loss = -y_t \cdot F(\boldsymbol{X} + G_t(U)) + \beta \cdot \|G_t(U)\|_2, \qquad (3)$$

where the first and second terms represent the cross-entropy loss and L_2 loss, respectively. With the guidance of the above loss function, we optimize U by iteratively applying the derived υ_t across the entire training data. In particular, in order to construct a UAPG that can be universally applied to any target class, at each training step, a random target class is selected to help U to learn inter-class representations. After constructing the unified U, the universal perturbations computed by our UAPG can be effectively applied on any input data to fool the DNN model in an audio-agnostic way, without requiring re-generating adversarial perturbation for each individual audio input.

Attack Evaluation

Experimental Methodology

Target Model and Dataset. We evaluate the proposed FAPG and UAPG on three types of the DNN-based audio systems, namely, *speech command recognition*, *speaker recognition*, and *environmental sound classification*.

Algorithm 2: Training Procedure of UAPG

```
Require: Training dataset $\chi = \{x^{(1)}, ..., , x^{(n)}\}$, class
  label $\{y_1, ..., , y_k\}$, DNN classifier $F(\cdot)$, generative
  model $G(\cdot)$, class-wise embedding feature maps $\varepsilon$, noise
  constraint constant $\tau$
Result: Trained UAPG
Random initialize $U$
for number of training iterations do
    for number of steps do
        $y_t \leftarrow get\_random\_target \in \{y_1, ..., y_k\}$;
        $G_t(\cdot) \leftarrow G(\cdot)$ embeds with $E_t \in \varepsilon$ ;
        UAP $\upsilon_t \leftarrow Clip(G_t(U), \{-\tau, +\tau\})$ ;
        $\boldsymbol{X} \leftarrow$ minibatch of $m$ samples from $\chi$;
        $\boldsymbol{X'} \leftarrow \boldsymbol{X} + \upsilon_t$ ;
        $\boldsymbol{y_{pred}} \leftarrow F(\boldsymbol{X'})$ ;
        Loss $\leftarrow$
          $\frac{1}{m}\sum_i^m(\text{CrossEntropy}(\boldsymbol{y^{(i)}_{pred}}, y_t) + \beta \cdot \|\boldsymbol{\upsilon_t}\|_2)$;
        minimize Loss to update $U$;
    end
end
```

- **Speech Command recognition.** We use a convolutional neural network (CNN)-based speech command recognition model (CNN-trad-fpool3) as proposed in (Sainath and Parada 2015), which has served as the target model in many previous studies (Alzantot, Balaji, and Srivastava 2018; Abdoli et al. 2019; Yu et al. 2018). The network is trained on a crowd-sourced speech command dataset (Warden 2018), which contains $46,278$ utterances from 10 representative speech commands sampled at $16kHz$, with each recording being cropped to 1s. 40-dimensional MFCC features are extracted as the input of the model. We randomly separate the dataset into training and testing set with a ratio of 4 to 1, and the recognition accuracy of this baseline model on the testing dataset is 89.2%.
- **Speaker Recognition.** A pre-trained X-vector model[1] (Snyder et al. 2018) with DNN-based embedding model and probabilistic linear discriminant analysis (PLDA) backend is used as the target speaker recognition model. The features are 30-dimensional MFCC features with a frame length of $25ms$. The dataset we use is an English multi-speaker corpus provided in CSTR voice cloning toolkit (VCTK) (Christophe, Junichi, and Kirsten 2016), which contains 44217 utterances spoken by 109 speakers, with each recording being cropped to 1.75s. The speakers are enrolled utilizing 80% of the data, while the rest is reserved for testing. This results in a baseline accuracy of 92.8% on 8896 testing utterances from 109 speakers.
- **Environmental Sound Classification.** A 1-dimensional CNN model (referred as *CNNrand* in (Abdoli, Cardinal, and Koerich 2019)) is used as the target model. The model is trained on the UrbanSound8k dataset (Salamon, Jacoby, and Bello 2014), which contains a total number of 8732 audio clips from 10 different environmental scenes. Each

[1] Avaible at https://kaldi-asr.org/models/m3

	FGSM	PGD	C&W	FAPG
Command Recognition	11.89%	11.96%	13.26%	**97.77%**
Speaker Recognition	1.65%	0.96%	11.09%	**98.35%**
Sound Classification	14.46%	10.08%	11.42%	**92.93%**

Table 1: Success rate (SR) of audio-dependent targeted attacks under constrained time budget (0.065s).

	Metric	FGSM	PGD	C&W	FAPG
Command Recognition	SR(%)	11.89	96.03	**97.92**	97.77
	Time	0.05s	1.36s	9.16s	**0.05s**
Speaker Recognition	SR(%)	1.65	97.47	98.08	**98.35**
	Time	0.05s	4.33s	10.74s	**0.05s**
Sound Classification	SR(%)	14.46	91.74	92.55	**92.93**
	Time	0.05s	1.85s	4.69s	**0.05s**

Table 2: Success rate (SR) and the corresponding attack generation time of audio-dependent targeted attacks under sufficient time budget.

recording is cropped to 50999 samples which corresponds to roughly 3 seconds at 16 kHz. The dataset is split into training, validation and testing set with a ratio of $8:1:1$. After training, the classification accuracy on 10 classes is 83.4%.

Evaluation Metrics. (1) *Fooling Rate (FR)* is used for evaluating both targeted and untargeted attacks, which shows the ratio of the number of adversarial examples that lead to a false classification and the total number of adversarial examples; (2) *Success Rate (SR)* is only used for evaluating targeted attacks, which is the ratio of the number of attacks resulting in the adversarial example being classified as the target class and total attack attempts; (3) *Distortion Metric:* We quantify the relative noise level of δ_t with respect to the original audio x_i in decibels (dB): $D(x_i, \delta_t) = 20log_{10}\frac{max(\delta_t)}{max(x_i)}$.

Audio-dependent Targeted Attack via FAPG

FAPG Generator Implementation. We use model $M1$ of Wave-U-Net (Stoller, Ewert, and Dixon 2018) to construct our FAPG. Specifically, our model contains 5 down-sampling blocks and 5 up-sampling blocks. The feature map size of the last encoding layer is also the size of each additional class-wise embedding feature map. For FAPG, a total of 10,000 training steps are conducted using Adam optimizer with the batch size of 100. The initial learning rate is set to $1e^{-4}$ and gradually decayed to $1e^{-6}$. β is set as 0.1 for all dataset. τ is initially set as 0.1 and reduces to 0.05 and 0.03 at step of 3,000 and 7,000 for command recognition and speaker recognition, and it stops reducing as 0.05 for sound classification model, which leads to an approximate noise level of -30 dB and -18 dB respectively.

Attack Speedup and Performance. To demonstrate the ability of the proposed FAPG in terms of achieving high success rate while maintaining a short attack generation time, we conduct experiments on the three aforementioned target models under different time conditions. Table 1 compares the attack performance of the proposed FAPG with commonly-used attacks, i.e., FGSM (Goodfellow, Shlens, and Szegedy 2014), PGD (Madry et al. 2017) and C&W (Carlini and Wagner 2017) under constrained time budget scenario, which requires to generate adversarial example with no more than 0.065s (the approximate execution time for one iteration in PGD and C&W attack. For fair comparison, we constrain the perturbations generated by these attacks with an infinity norm of 0.03 for speech command classification and speaker recognition, and 0.05 for environmental sound classification, which are the same as used in FAPG implementation. As shown in Table 1, the proposed FAPG can achieve high attack success rate (SR) (over 90%) under the short time budget for all the three target models, while FGSM, PGD and C&W attack can only achieve less than 15% SR with limited attack time budget.

Additionally, we also conduct experiments when sufficient time budget is granted. As shown in Table 2, though PGD and C&W achieve the very similar SRs to our proposed FAPG, they require much longer adversarial perturbation generation time. For instance, for speaker recognition task PGD needs 4.33s and C&W even requires more than 10s to launch the attack, while the time period for each data is only 1.75s. Such huge gap makes the PGD and C&W-based attacks infeasible in the practical real-time attack scenarios. On the other hand, with achieving the very similar high SR, our proposed FAPG only needs 0.05s to generate adversarial perturbation, thereby bringing very high speedup (up to 86× and 214× as compared with PGD and C&W, respectively). Also, compared with another fast generation approach FGSM, FAPG achieves much higher SR.

Memory Cost Reduction. Our proposed trainable class-wise feature maps can reduce the memory cost significantly. Without the class-wise feature embedding maps, launching targeted attack requires to train one generative model for each target class, which results in a memory consumption of 23.8 MB, 259 MB, and 23.8 MB for the speech command recognition, speaker recognition, and sound classification model, respectively. In contrast, by utilizing the class-wise embedding feature maps, our proposed FAPG only requires to train a single generative model and a set of embedding maps, regardless of the number of target classes, and therefore only takes up 2.4 MB, 3.53 MB, and 2.44 MB for these three target models respectively. This leads to a memory cost reduction of 9.9×, 73.5×, and 9.8×, respecitively.

Audio-agnostic Universal Attack via UAPG

UAPG Implementation. The proposed UAPG is built on a pre-trained FAPG model and a trainable universal adversarial input vector U with the same size of the original audio input. The vector U is then trained on the same training set as used for the target model training. A total number of 8000 training steps are conducted using Adam optimizer with a learning rate of $1e^{-4}$ and a batch size of 100. We set τ to 0.03 which corresponds to an average distor-

	Command Recognition		Speaker Recognition		Sound Classification	
Attack Method	UAP-HC	UAPG	RURA	UAPG	UAAP	UAPG
FR	52.78%	**90.03%**	N/A	**90.05%**	N/A	**91.01%**
SR	N/A	**89.90%**	86.17%	**89.59%**	85.40%	**86.05%**

Table 3: Success rate (SR) of audio-agnostic targeted attacks under white-box setting.

tion of $-30.21dB$ of the generated adversarial perturbations for speech command recognition model and speaker recognition, and $\tau = 0.05$ for environmental sound classification.

Analysis of Learned Representation. To investigate the effectiveness of UAPG, we plot the audio-dependent perturbations generated by FAPG as well as the audio-agnostic perturbations generated by UAPG on the speech command recognition model using principal component analysis (PCA) (Wold, Esbensen, and Geladi 1987). We show the adversarial perturbations targeting at five commands in Figure 3. Although the universal perturbations are created without accessing the distribution of real speech commands, all universal perturbations locate within the manifold of corresponding audio-dependent perturbations generated for the same target class. This demonstrates that our UAPG can efficiently learn the inherent adversarial representations with respect to each target command.

White-box Attack Performance. We compare the performance of our proposed UAPG with several state-of-the-art audio universal attacks, including UAP-HC (Vadillo and Santana 2019) which is based on DeepFool algorithm (Moosavi-Dezfooli, Fawzi, and Frossard 2016), RURA (Xie et al. 2020), and UAAP (Abdoli et al. 2019). To evaluate UAPG attack, for each target model, we generate one universal perturbation for each target class. Table 3 presents the results of UAP-HC on the speech command model, RURA on the speaker recognition model, UAAP on the sound classification model, and the proposed UAPG on all three of these models. Specifically, since UAP-HC is designed to be an untargeted universal attack, we only report its average fooling rate (FR). We observe that our proposed UAPG outperforms existing methods on all three tasks, with an average SR of 89.90%, 89.59, and 86.05% when evaluated on the three models respectively.

Black-box Attack Performance. We also evaluate the performance of the proposed UAPG under black-box settings, where the architecture and parameters of the target victim model is unknown. For each task, we first train UAPG on a substitute model (CNN-3 model (Zhang et al. 2017), d-Vector (Variani et al. 2014), EnvNetV2 (Tokozume and Harada 2017)) as is shown in Table 4, and then evaluate the generated adversarial examples on the target model to study its transferbility. For the speech command recognition model, we compare the performance of the proposed UAPG with the recent untargeted real-time adversarial attack (RAA) (Gong et al. 2019) on the same target model in a black-box manner. As shown in Table 4, our proposed UAPG achieves high FR even when tested in a black-box setting on different tasks. Compared with the state-of-the-art untargeted real-time attack RAA, our UAPG achieves 29.98% FR increase.

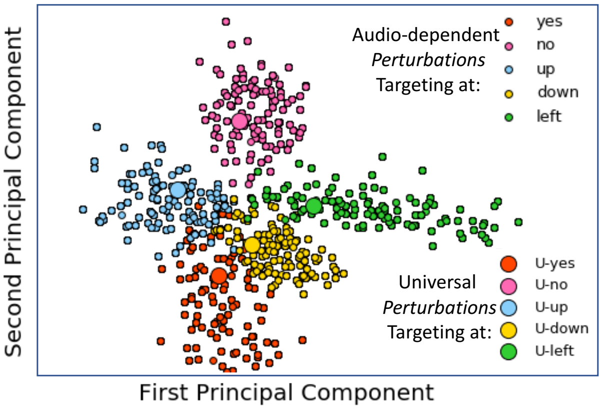

Figure 3: Visualization of audio-dependent perturbations and universal perturbations targeting at different speech commands.

	Speech Recognition		Speaker Recognition	Sound Classification
Substitute Model	CNN-3		d-Vector	EnvNetV2
Target Model	CNN-trad-fpool3		X-Vector	CNNrand
Method	RAA	UAPG	UAPG	UAPG
FR	43.5%	**73.48%**	80.50%	69.26%

Table 4: Fooling rate (FR) of audio-agnostic targeted attacks under black-box setting. RAA (Gong et al. 2019) only reports result on speech command recognition task.

Conclusion

In this work, we propose a fast and universal adversarial attack on three audio processing systems: speech command recognition, speaker recognition and environmental sound classification. By exploiting Wave-U-Net and the class-wise feature embedding maps, our proposed FAPG can launch fast audio adversarial attack targeting at any speech command using a unified generative model within a single pass of feed-forward propagation, which results in an adversarial perturbation generation speedup up to 214× comparing to the state-of-the-art solutions. Moreover, built on the top of FAPG, our proposed UAPG is able to generate universal adversarial perturbation that can be applied on arbitrary benign audio input. Extensive experiments demonstrate the effectiveness of the proposed FAPG and UAPG.

Acknowledgments

This work is partially supported by Air Force Research Lab (AFRL) under Grant No. FA87501820058, the Army Research Office (ARO) grant W911NF1910405 and National Science Foundation (NSF) award CCF-1937403, CCF-1909963, CCF-2028876 and CNS-1820624.

References

Abdoli, S.; Cardinal, P.; and Koerich, A. L. 2019. End-to-end environmental sound classification using a 1D convolutional neural network. *Expert Systems with Applications* 136: 252–263.

Abdoli, S.; Hafemann, L. G.; Rony, J.; Ayed, I. B.; Cardinal, P.; and Koerich, A. L. 2019. Universal adversarial audio perturbations. *arXiv preprint arXiv:1908.03173* .

Alzantot, M.; Balaji, B.; and Srivastava, M. 2018. Did you hear that? adversarial examples against automatic speech recognition. *arXiv preprint arXiv:1801.00554* .

Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, 39–57. IEEE.

Carlini, N.; and Wagner, D. 2018. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, 1–7. IEEE.

Chen, G.; Chen, S.; Fan, L.; Du, X.; Zhao, Z.; Song, F.; and Liu, Y. 2019. Who is Real Bob? Adversarial Attacks on Speaker Recognition Systems. *arXiv preprint arXiv:1911.01840* .

Christophe, V.; Junichi, Y.; and Kirsten, M. 2016. CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit. *The Centre for Speech Technology Research (CSTR)* .

Gong, Y.; Li, B.; Poellabauer, C.; and Shi, Y. 2019. Real-Time Adversarial Attacks. *CoRR* abs/1905.13399. URL http://arxiv.org/abs/1905.13399.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* .

Kreuk, F.; Adi, Y.; Cisse, M.; and Keshet, J. 2018. Fooling end-to-end speaker verification with adversarial examples. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1962–1966.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533* .

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* .

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Fawzi, O.; and Frossard, P. 2017. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1765–1773.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2574–2582.

Neekhara, P.; Hussain, S.; Pandey, P.; Dubnov, S.; McAuley, J.; and Koushanfar, F. 2019. Universal adversarial perturbations for speech recognition systems. *arXiv preprint arXiv:1905.03828* .

Phan, H.; Xie, Y.; Liao, S.; Chen, J.; and Yuan, B. 2020. CAG: A Real-Time Low-Cost Enhanced-Robustness High-Transferability Content-Aware Adversarial Attack Generator. In *AAAI*, 5412–5419.

Poursaeed, O.; Katsman, I.; Gao, B.; and Belongie, S. 2018. Generative adversarial perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4422–4431.

Sainath, T. N.; and Parada, C. 2015. Convolutional Neural Networks for Small-Footprint Keyword Spotting. In *Sixteenth Annual Conference of the International Speech Communication Association.*

Salamon, J.; Jacoby, C.; and Bello, J. P. 2014. A Dataset and Taxonomy for Urban Sound Research. In *22nd ACM International Conference on Multimedia (ACM-MM'14)*, 1041–1044. Orlando, FL, USA.

Snyder, D.; Garcia-Romero, D.; Sell, G.; Povey, D.; and Khudanpur, S. 2018. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5329–5333. IEEE.

Song, Y.; Shu, R.; Kushman, N.; and Ermon, S. 2018. Constructing unrestricted adversarial examples with generative models. In *Advances in Neural Information Processing Systems*, 8312–8323.

Stoller, D.; Ewert, S.; and Dixon, S. 2018. Wave-u-net: A multi-scale neural network for end-to-end audio source separation. *arXiv preprint arXiv:1806.03185* .

Tokozume, Y.; and Harada, T. 2017. Learning environmental sounds with end-to-end convolutional neural network. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2721–2725. IEEE.

Vadillo, J.; and Santana, R. 2019. Universal adversarial examples in speech command classification. *arXiv preprint arXiv:1911.10182* .

Variani, E.; Lei, X.; McDermott, E.; Moreno, I. L.; and Gonzalez-Dominguez, J. 2014. Deep neural networks for small footprint text-dependent speaker verification. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4052–4056. IEEE.

Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P.-A. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, 1096–1103.

Warden, P. 2018. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209* .

Wold, S.; Esbensen, K.; and Geladi, P. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems* 2(1-3): 37–52.

Xie, Y.; Shi, C.; Li, Z.; Liu, J.; Chen, Y.; and Yuan, B. 2020. Real-time, universal, and robust adversarial attacks against speaker recognition systems. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1738–1742. IEEE.

Yu, F.; Xu, Z.; Wang, Y.; Liu, C.; and Chen, X. 2018. Towards robust training of neural networks by regularizing adversarial gradients. *arXiv preprint arXiv:1805.09370* .

Yuan, X.; Chen, Y.; Zhao, Y.; Long, Y.; Liu, X.; Chen, K.; Zhang, S.; Huang, H.; Wang, X.; and Gunter, C. A. 2018. Commandersong: A systematic approach for practical adversarial voice recognition. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 49–64.

Zhang, Y.; Suda, N.; Lai, L.; and Chandra, V. 2017. Hello edge: Keyword spotting on microcontrollers. *arXiv preprint arXiv:1711.07128* .