Query-Efficient Adversarial Attack With Low Perturbation Against End-to-End Speech Recognition Systems

Shen Wang, Zhaoyang Zhang, Guopu Zhu, *Senior Member, IEEE*, Xinpeng Zhang, *Member, IEEE*, Yicong Zhou, *Senior Member, IEEE*, and Jiwu Huang, *Fellow, IEEE*

***Abstract*—With the widespread use of automated speech recognition (ASR) systems in modern consumer devices, attack against ASR systems have become an attractive topic in recent years. Although related white-box attack methods have achieved remarkable success in fooling neural networks, they rely heavily on obtaining full access to the details of the target models. Due to the lack of prior knowledge of the victim model and the inefficiency in utilizing query results, most of the existing black-box attack methods for ASR systems are query-intensive. In this paper, we propose a new black-box attack called the Monte Carlo gradient sign attack (MGSA) to generate adversarial audio samples with substantially fewer queries. It updates an original sample based on the elements obtained by a Monte Carlo tree search. We attribute its high query efficiency to the effective utilization of the dominant gradient phenomenon, which refers to the fact that only a few elements of each origin sample have significant effect on the output of ASR systems. Extensive experiments are performed to evaluate the efficiency of MGSA and the stealthiness of the generated adversarial examples on the DeepSpeech system. The experimental results show that MGSA achieves 98% and 99% attack success rates on the LibriSpeech and Mozilla Common Voice datasets, respectively. Compared with the state-of-the-art methods, the average number of queries is reduced by 27% and the signal-to-noise ratio is increased by 31%.**

***Index Terms*—Adversarial example, automatic speech recognition, black-box attack, Monte Carlo tree search.**

I. Introduction

NEURAL networks have incredible representative abilities, which makes them suitable for various signal processing and analysis tasks. Prior works [1], [2], [3] have verified the success of adversarial attacks on neural networks. These attacks bring great potential threats to artificial intelligence systems [4]. Most existing works are devoted to the image domain [5], [6], [7], [8], [9], [10] and text domain [11], [12]. With the wide application of audio speech recognition (ASR) systems, the issue of adversarial examples against ASR has not received enough attention [13]. As shown in Fig. 1, the audio adversarial attack deceives the victim model by adding adversarial noise to the original signal. Due to some non-trivial challenges [14], [15], [16], studies on adversarial attacks in the audio domain is still limited. Existing works on audio adversarial attacks have mainly focused on white-box scenarios, [3], [5], [17] which assume that attackers have access to all the parameters of the target systems. However, this assumption does not hold in practice [18]. In most cases, attackers can only obtain the output of ASR systems [19], [20], [21], [22]. With a sufficient number of queries, black-box attack methods utilize the feedback information to generate adversarial examples; thus, they have become major threats to artificial intelligence models for signal processing.

Existing black-box attack methods are generally query-intensive [18], [23], [24]. The speech signal usually contains numerous sampling points, and we call these sampling points the elements of the speech signal. As the number of elements of the original sample increases, the search space increases rapidly, which results in an excessive number of queries per attack. For most black-box attacks, however, an excessive number of queries is unacceptable [25], [26], [27], [28]. Existing black-box attack methods [18], [19], [20] add perturbations to the adversarial example in each iteration. These perturbations cover almost each element of the speech signal when the targeted attack is completed. This leads to the fact that adversarial examples are easily detected by detectors. We attribute the excessive queries required by the existing methods to the blind selection of elements during the generation of adversarial examples.

In this paper, we propose a new black-box attack method called the Monte Carlo gradient sign attack (MGSA). Compared with prior works, the proposed method improves the

Manuscript received 16 February 2022; revised 30 July 2022 and 21 October 2022; accepted 4 November 2022. Date of publication 17 November 2022; date of current version 7 December 2022. This work was supported in part by the National Defense Basic Scientific Research Program of China under Grant JCKY2018603B006; in part by the National Natural Science Foundation of China under Grant 62172402, Grant U19B2022, Grant U22B2047, Grant U1936214, and Grant 61872350; in part by the Science and Technology Development Fund, Macau SAR, under Grant 0049/2022/A1; in part by the University of Macau under Grant MYRG2022-00072-FST; and in part by the Fundamental Research Funds for the Central Universities under Grant FRFCU5710011322. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Tiziano Bianchi. *(Corresponding author: Guopu Zhu.)*

Shen Wang, Zhaoyang Zhang, and Guopu Zhu are with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: shen.wang@hit.edu.cn; 20B903055@stu.hit.edu.cn; guopu.zhu@hit.edu.cn).

Xinpeng Zhang is with the School of Computer Science, Fudan University, Shanghai 200433, China (e-mail: zhangxinpeng@fudan.edu.cn).

Yicong Zhou is with the Department of Computer and Information Science, University of Macau, Macau (e-mail: yicongzhou@um.edu.mo).

Jiwu Huang is with the Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen Key Laboratory of Media Security, Shenzhen University, Shenzhen 518060, China, and also with the Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen 518060, China (e-mail: jwhuang@szu.edu.cn).

Digital Object Identifier 10.1109/TIFS.2022.3222963

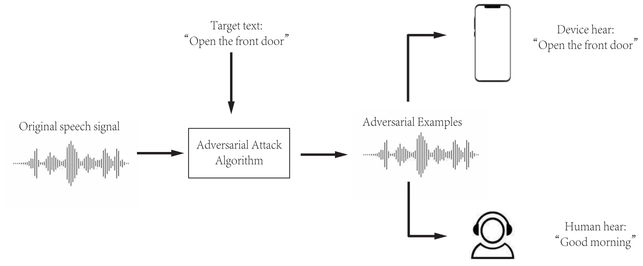


Fig. 1. Illustration of the adversarial attack against ASR systems.

number of queries and the SNR of the adversarial examples. The main contributions of this paper are summarized as follows:

- We first investigate the phenomenon that only a few elements with large gradients have a sufficient impact on the output of ASR, which is called the dominant gradient phenomenon. The experimental results verify that the utilization of elements with dominant gradients can significantly reduce the number of queries of black-box attacks.
- We propose a new black-box attack, called MGSA for ASR systems; this attack, builds a Monte Carlo tree to search elements with the dominant gradients. Furthermore, we introduce a sampling gradient sign strategy and a momentum iterative strategy to accelerate the convergence speed of the algorithm.
- We conduct extensive experiments on the Mozilla Common Voice and LibriSpeech datasets using DeepSpeech as the target model. The results verify that the proposed method achieves a 98% attack success rate. The average number of queries is reduced by 27% and the signal-to-noise is increased by 31% compared with the state-of-the-art methods.

The rest of the paper is organized as follows. The background of ASR systems and the related work of adversarial attack methods are provided in Section II. The principle and details of the proposed algorithm are provided in Section III. In Section IV, we carry out experimental verification and comparative analysis of the experimental results. We conclude this paper in Section V.

II. Background and Related Work

In this section, we first briefly introduce the background knowledge of ASR systems and then review the related research on adversarial attacks and Monte Carlo tree search methods.

A. Deep Learning-Based ASR Systems

Audio signals are usually sampled as N-dimensional vectors in the time domain. Mel-frequency cepstrum (MFC) is often used as the feature extraction method in the ASR systems to obtain an audio representation by simulating the human auditory system [15], [29]. MFC divides each sound wave into 50 frames per second and projects each frame into the frequency domain. In ASR systems, as shown in Fig. 2, recurrent neural networks (RNN) are usually used to map a preprocessed audio signal to the probability distribution on the output tag [30]. The RNN in ASR systems can be expressed as $(s_{i+1}, y_i) = f(s_i, v_i)$, where v_i is the i_{th} frame of the input vector, s_i is the state vector of v_i, and y_i is the probability distribution of the output of v_i.

In this paper, the DeepSpeech system [31] is employed as the target ASR system. Since the inputs of ASR are unaligned, connectionist temporal classification (CTC) [32] is introduced to DeepSpeech to find the precise mapping relationship between a pair of inputs and outputs. The objective of CTC is to maximize the sum of the probabilities of all possible alignment paths between the input and the target sequence:

$$\Pr(t|y) = \sum_{\pi \in D(t)} \Pr(\pi|y) = \sum_{\pi \in D(t)} \prod_{i} y^{i}_{\pi_i}, \tag{1}$$

where π denotes the possible alignments of target sequence t through the mapping operation $D(\cdot)$, and π_i is the characteristic of π at element i [33]. The loss function used in the DeepSpeech system for training the RNN is the following negative logarithm maximum likelihood functions:

$$L_{CTC}(y, t) = -\log \Pr(t|y). \tag{2}$$

Since the search space is large, dynamic programming is usually employed to improve the computational efficiency.

B. Adversarial Attacks on ASR Systems

The success of adversarial attacks on images has inspired the study of adversarial audio attack on ASR systems [14]. Due to the high dimensionality of the inputs and the almost infinite number of possible target texts, targeted attacks on ASR systems are far more difficult than those on image classification models [34]. Most existing attack methods for ASR systems are only applicable in white-box settings [3], [20], [21], [22], [35]. Iter et al. [36] added perturbations to MFCC features and then reconstructed the speech signal from the perturbed MFCC features. However, the perturbations introduced by the inverse-MFCC process are proven to be too noisy for human ears [14]. Carlini et al. [22] addressed this problem by rebuilding the feature extraction function and proposed an iterative optimization method to generate adversarial examples. Nevertheless, their method takes more than one hour to generate a 3-second adversarial audio sample and thus is very inefficient.

Since an attacker may not be able to obtain enough system information in real scenarios, the black-box attacks have also attracted the attention of researchers. Cisse et al. [21] proposed a transferable adversarial attack, which is only effective for target text that is similar to the content of the original audio. Yuan et al. [37] transferred adversarial examples from Kaldi [38] to DeepSpeech [31] and revealed the poor performance of the transferability of adversarial examples. Zheng et al. [39] claimed that minimal information could be used to improve the transferability of adversarial examples. Other recent works [10], [40], [41] have also implemented black-box attacks based on the transferability of adversarial examples. Despite such attacks are successful, they have major

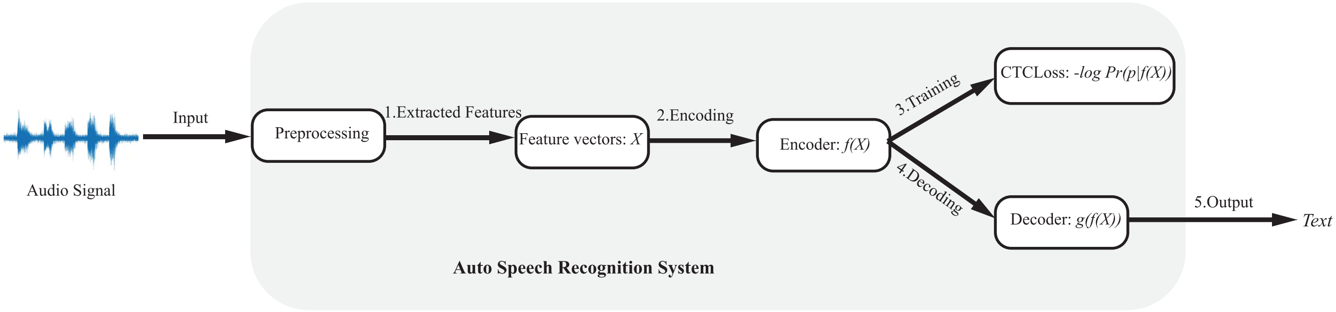


Fig. 2. Typical structure of end-to-end ASR system.

limitations because attackers usually cannot obtain enough training data with the same distribution as that used in the target model. Even if the attacks are implemented, it is too expensive to retrain one or more alternative models for adversarial examples. Another class of black-box attacks is based on queries and these models generate adversarial examples using the feedback information from the target model. Mun et al. [42] proposed an adversarial attack method based on particle swarm optimization and applied it to the speech command model. Zong et al. [43] implemented a variational autoencoder (VAE) based black-box attack. Although the algorithm has the characteristic of low queries, it is only suitable for untargeted attacks. Alzantot et al. [44] first searched for targeted attacks based on the output probabilities of the victim model. They applied a genetic algorithm to the speech command model and achieved an attack success rate of 87%. Taori et al. [45] combined a genetic algorithm and gradient estimation to attack ASR systems, and they successfully realized the targeted attack of two-word phrases. However, this method is ineffective for long speech and complex target texts, and it takes an extremely long time and requires a great number of queries. Wang et al. [18] proposed a selective gradient estimation attack (SGEA), which selects a small batch of elements to perturb in each iteration based on the results of the previous iterations. However, the effective elements change during the generation of adversarial examples. At the later stage of algorithm execution, the efficiency of this method is similar to that of selecting elements randomly. Although SGEA requires fewer queries than other methods, the noise introduced by this method can still be noticed by the human ear. Khare et al. [20] proposed a multiobjective genetic algorithm to improve the success rate of black-box attacks, but the algorithm requires more queries during the attack than other methods.

In summary, the existing black-box attack methods still have many problems to solve, such as alternative model dependencies, excessive queries, long time- consumption, heavy additional noise, and low success rates for long speech or complex target texts.

C. Monte Carlo Tree Search

The Monte Carlo tree search (MCTS) is the best-first search method under the guidance of Monte Carlo simulation [46]; it was proposed for finite zero-sum games. The goal of MCTS is to give the best next action for a given game state. Compared with other tree search algorithms, MCTS does not need a state evaluate function, so it is widely used in fields, such as robotics planning [47], [48], [49], [50] and optimization [51], [52], [53], [54], where the value of each state is difficult to evaluate.

The main idea of MCTS is to expand an incomplete search tree and predict the best next action using the simulation results. After the Monte Carlo tree (MCT) is initialized, MCTS mainly iterates through four steps: selection, expansion, simulation and backtracking. (1) In the selection step, the algorithm selects a path from the root node of MCT to the end node through some strategies. (2) In the expansion step, new nodes are added to MCT as the children of the nodes selected in the selection step. (3) In the simulation step, the algorithm simulates from the new nodes generated in the expansion step to the final state several times to obtain feedback information. (4) In backtracking step, the algorithm uses simulated feedback information to update the statistical information of all nodes on the backpropagation path.

In MCTS, the statistical information including simulation reward $Q(v)$ and total number of visits $N(v)$, is important attribute of a node and are used to determine the probability that the node will be selected in the selection step. A simple form of $Q(v)$ is the sum of the simulation feedback of the node and its children, and $N(v)$ represents the times the node appears on the backpropagation path.

III. Monte Carlo Gradient Sign Attack

In this section, we first explain the motivation of the MGSA algorithm and then introduce the details of the proposed algorithm.

A. Problem Formulation

To attack a black-box target model F with high-dimensional input, an attacker needs to inject an invisible perturbation into the original example, so that the model outputs the target text. Given an original speech example X and a target text t, the problem of searching adversarial examples can be formulated as

$$\begin{aligned} &\min ||\delta||_2, \\ &\text{s.t.} \quad F(X+\delta)=t,\ X+\delta \in [-m,m], \end{aligned} \tag{3}$$

where $||\cdot||_2$ is the L_2 norm, δ denotes adversarial perturbation, and m denotes the boundary parameter of legal examples. However, it is difficult to guarantee that the constraint $F(X+\delta) = t$ during the optimization process. Therefore, we optimize the L_2 norm of adversarial perturbation $||\delta||_2$ and the distance $d(F(X), t)$ between the model output $F(X)$ and the target text t at the same time. In the field of image recognition, the distance can be measured easily based on the output $F(X)$ and the target label, because the dimension of $F(X)$ is usually set to be the same as the number of labels. However, in the field of speech recognition, many sequences can be normalized to the target text. We take the sequence that has the lowest CTC loss with the original example as the optimization target:

$$\pi_t = \arg\min_{\pi \in D(t)} (L_{CTC}(F(X), \pi)). \tag{4}$$

Therefore, the distance between $F(X)$ and target text t can be caculated by

$$d(F(X), t) = L_{CTC}(F(X), \pi_t). \tag{5}$$

By taking the L_2 norm of the adversarial perturbation as the restriction, the optimization problem in Eq. 3 can be transformed to

$$\begin{aligned} \delta &= \arg\min_{\delta}(L(X, \delta, t)), \\ L(X, \delta, t) &= c \cdot ||\delta||_2 + d(F(X+\delta), T), \end{aligned} \tag{6}$$

where c is a balance parameter.

In a black-box setting, the gradient of the target model is not available, so a zero-order optimization method is needed to solve the optimization problem described in Eq. (6). In this paper, we estimate the gradient of the loss function by

$$\nabla L_{\delta}(X, \delta, t) \approx \begin{bmatrix} (L(X, \delta_1, t) - d(F(X), t))/|\delta_1| \\ (L(X, \delta_2, t) - d(F(X), t))/|\delta_2| \\ \vdots \\ (L(X, \delta_n, t) - d(F(X), t))/|\delta_n| \end{bmatrix}, \tag{7}$$

where $(\delta_1 \cdots \delta_n)$ is a group of sparse orthogonal vectors, and δ_i is set to one at element i and zero at the other elements. However, the number of queries required to estimate the gradient for each element is equal to the dimension of the speech signal. For example, when the sampling rate is 16 kHZ, a 3-second audio sample has 48,000 elements, which results in at least 48,000 queries for a round of gradient estimation. Moreover, even if the gradient is estimated for each element, more queries are needed to obtain the appropriate step size.

B. Dominant Gradient Phenomenon

Since the input of an ASR system has the characteristic of high dimensionality, attackers usually adopt iterative algorithms to draw the adversarial examples gradually to the target text. Perturbations accumulate gradually in the iterative process. As a result, it is easy for human ear to detect the final adversarial example.

As shown in Fig. 3, a perturbation in an invalid direction increases the additional noise. Therefore, it is critical to select an effective direction to update the adversarial examples during each iteration. Therefore, we investigate the CTC-Loss gradient distribution of an end-to-end ASR system. As shown in Fig. 4, the output of the ASR system is not strongly correlated with each element of the input example. The gradients of most elements in the input example are almost zero. Elements whose gradient reaches 80% of the maximum gradient account for only 5% of the total number of elements, which is called the dominant gradient phenomenon. This phenomenon means that the benefits of gradient estimation for most elements are not evident. The direction represented by each element can be viewed as mutually orthogonal in the search space, and only a few elements can draw the adversarial example to the target text. Therefore, it is instructive that searching the effective elements of the input example with a few queries will improve the attack efficiency.

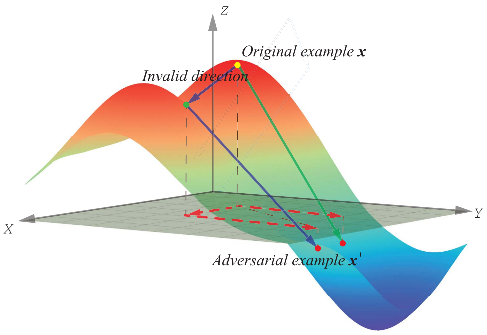
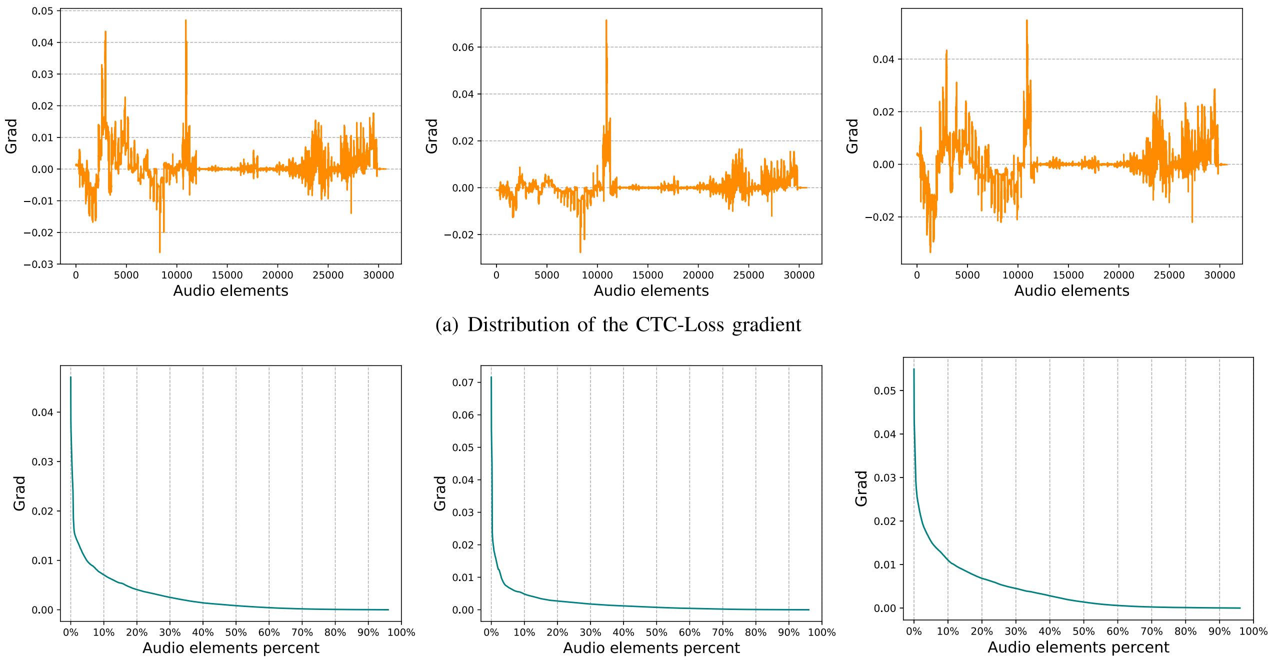


Fig. 3. **Illustration of the invalid direction.** The surface represents the loss function, and the gradient update in the invalid dimension x will increase the noise of the adversarial example.

C. Efficient Element Search

According to the dominant gradient phenomenon, the key to reducing the number of queries is to select the elements with dominant gradients, as mentioned in Section III-B. In this section, an efficient element search algorithm based on MCTS is proposed.

Different from the common application domains of MCTS such as multistep decision games, we study how to quickly find the most effective elements for adversarial examples in this paper. There are two main challenges: (1) In a multistep decision game, the algorithm needs to provide the best next action which corresponds to the first layer under the root node of MCT. However, in our problem, since the algorithm needs to provide the best leaf node (the most effective elements), the middle nodes from the end nodes of MCT to the final states are not important in the simulation step. Therefore, we directly sample elements from the end nodes to evaluate the path. (2) In a multistep decision game, the difficulty comes from the uncertain action of the opponent. MCT can be subsequently deployed by simulating the opponent's actions in a multistep decision game. Since it is easy to determine the reward based on the endgame state, obtaining the final value of each path is inexpensive. In our problem, the main difficulty is obtaining the value of each path, which comes from the output of the model and therefore relies on expensive queries. To reduce



(a) Distribution of the CTC-Loss gradient

(b) Statistics of the gradient value

Fig. 4. **CTC-Loss gradient distribution of the end-to-end ASR system:** For a 2-second speech signal with a sampling rate of 16000, it is shown in (a) that the gradient of CTC-Loss for each element in three different target phrases (i.e. OK Google, Hello world, send a message). In (b), we show the gradient statistics corresponding to (a), where the abscissa is the percentage of elements sorted by gradient.

the number of queries, we should set a reasonable number of branches of MCT to reduce the number of nodes in MCT. We analyze the influence of branches on the search efficiency in section IV-B.

The search process of efficient elements is shown in Algorithm 1. We first take all elements of the original example as the root node, called $node_0$. The visit number of $node_0$ is initialized to zero. MCTS in the proposed attack can be divided into four steps: selection, splitting, sampling, and backtracking.

Selection is implemented layer by layer from the root to the end of MCT base on the value of each node. To ensure that each node has a chance to be selected, we add a penalty term based on the visit number of the nodes. As the visit number of a node increases, the probability of the node being selected will decrease. Therefore, the value of each node in the search process is defined as

$$V_{node_k} = \frac{S_{node_k}}{\sum_{node_i \in D} S_{node_i}} + \alpha \cdot \frac{\sqrt{2 \log N_p}}{N_{node_k}}, \tag{8}$$

where S_{node_k} is the score of $node_k$ and the $node_k$ is obtained from sampling setp; D is the layer where $node_k$ resides; N_{node_k} represents the visit number of $node_k$; N_p represents the visit number of $node_k$'s parent; and α is a coefficient to control the trade-off between the node score and the penalty term. We select the node with the highest score in the end layer of MCT and mark it as $Selected\ node$.

If the size of $Selected\ node$ is less than the set threshold, it is added to the selected leaf set, called $Leaves$, and marked as unelectable for this round of search; otherwise, we split $Selected\ node$ randomly into n new child nodes $\mathbf{N} = \{node_1, node_2, \ldots node_n\}$, and add $\mathbf{N}$ into MCT.

After splitting, random sampling is performed on the elements to obtain m leaves $\mathbf{L} = \{l_1, l_2 \ldots l_m\}$ for each $node_i$ in $\mathbf{N}$ respectively. Note that $\mathbf{L}$ is not added to MCT. The score of $node_i$ is set as the average of the gradients of its leaves:

$$S_{node_i} = \frac{1}{m} \cdot \sum_{j=1}^{m} \frac{L(x_1^{l_j}, \ldots, x_p^{l_j}) - L(x_1^{l_j} + \delta, \ldots, x_p^{l_j} + \delta)}{p \cdot \delta}, \tag{9}$$

where p is the element number of the $node_i$ and $x_p^{l_j}$ is the p_{th} element of l_j.

In backtracking step, one is added to the visit number of each node on the path of $Selected\ node$. Moreover, the score of each node on the path of $Selected\ node$ is updated in a bottom-up fashion using the average of the original score and the scores of its children.

As shown by the workflow in Fig. 5, we repeat selection, splitting, sampling, and backtracking until there are enough nodes in the leaf set.

D. Design of the Iterative Method

After selecting effective elements, we need to obtain the accurate update direction of the adversarial example. In previous black-box attack methods [18], [22], [24], [25], [44], [45], gradient estimation is usually employed to caculate the gradient sign. The vector composed of the gradient sign is regarded as the update direction, and the adversarial example is updated by

$$X_{t+1} = X_t + \beta \cdot \text{sign}(G_t), \tag{10}$$

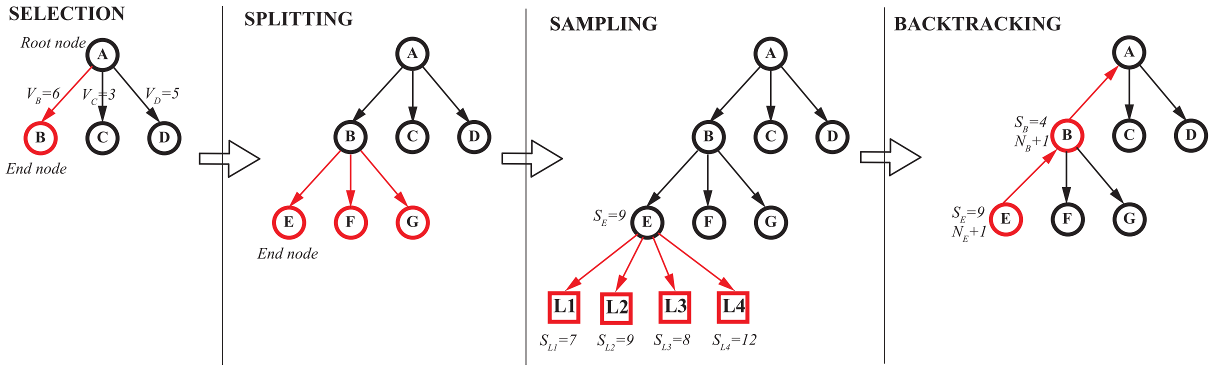


Fig. 5. **Workflow of Monte Carlo tree search for effective elements:** In an iteration, MCTS starts with building the tree via splitting the root node, then it expands the tree with selection, splitting, sampling, and backtracking. It performs the search iteratively, and select a leaf in each round of search until there are enough leaves selected for update.

Algorithm 1 Efficient Gradient Search Algorithm (EGSA)

```
Input:
  Original benign input $X$; Target text $y$; Maximum iteration $T$; Victim model $F$; Monte Carlo Tree $MCT$
Output:
  Selected elements $best\ elements$;
Initialize the following parameters:
  • Set of leaves $Leaves$; Set of node selection times $H$
  • Branches of nodes $m$; Size of leaves $b$
$Selected\ node \leftarrow$ root node of MCT
For $t$ from 1 to $T$ do
  While the length of $Selected\ node > b$ do
    If $Selected\ node$ is end node of MCT do
      Select node $n_p$ according to Eq. (8)
    Else do
      Divid $Selected\ node$ into $\mathbf{N} = \{n_1, n_2, \ldots, n_m\}$
      For $n_i \in \mathbf{N}$ do
        Random sampling to obtain $m$ leaves $\mathbf{L} = \{l^1_{n_i}, l^2_{n_i} \ldots l^m_{n_i}\}$
        Calculate the score $S_{n_i}$ of $n_i$ using Eq. (9)
      End For
      Add $\mathbf{N}$ into $MCT$
      Select node $n_p$ according to Eq. (8)
    End If
    $Selected\ node \leftarrow n_p$
    For $n_i$ on the path of $Selected\ node$
      $H_{n_i} \leftarrow H_{n_i} + 1$
      $S_{n_i} \leftarrow \frac{1}{m+1} \cdot (S_{n_i} + \sum_{j=1}^{m} S_{n_i^j})$
      // $n_i^j$ is the $j_{th}$ child of $n_i$
    End For
  End While
  Add $Selected\ node$ into $Leaves$
End For
Select nodes with highest score in $Leaves$ as $best\ elements$
return $best\ elements$
```

where β is the step size for updating. It is obvious that estimating the gradient of each element is a query intensive strategy. To reduce the number of queries per iteration, the authors in [45] updated the adversarial example with the batch gradient. For selected elements $\mathbf{B}_t = [x_t^1, x_t^2 \ldots, x_t^n]$ in the t_{th} iteration, the batch gradient is given by

$$G(\mathbf{B}_t) = \frac{L(X + \delta \cdot \sum_{x_t^i \in \mathbf{B}_t} e_i) - L(X)}{\delta}, \tag{11}$$

where e_i is a unit vector with 1 at element x_i and 0 at the other elements.

However, elements with positive and negative gradients are often included in the same batch, resulting in $G(\mathbf{B}_t)$ not being able to correctly reflect the gradient sign of the effective element. The problem is alleviated after the introduction of MCTS. Since the elements with the largest gradient are concentrated in leaves, the gradient signs of most selected elements are the same. Based on the characteristics mentioned above, we propose a sampling gradient sign strategy to further increase the utilization of the queries during the update process, and introduce a momentum iteration strategy to accelerate the algorithm.

1) Sampling Gradient Sign Strategy: In black-box attacks, it is proven to be necessary to obtain the gradient sign of the selected elements. As MCTS progresses, elements with positive gradients gradually become concentrated in the leaf set. Therefore, we use a few queries to select elements with negative gradients among the selected elements. We first divide the selected element set $\mathbf{B}_t$ into m subsets $\mathbf{D}_t = \{\mathbf{D}_t^1, \mathbf{D}_t^2 \cdots \mathbf{D}_t^m\}$, and perform batch gradient estimation on each subset to obtain $\{G(\mathbf{D}_t^1), G(\mathbf{D}_t^2) \cdots G(\mathbf{D}_t^m)\}$. Then the expectation of the gradient sign of $x_p \in \mathbf{B}_t$ is evaluated as

$$E_t(x_p) = \frac{\sum_{i=1}^{m} \mathbb{I}(x_p) \cdot G(\mathbf{D}_t^i)}{\sum_{i=1}^{m} \mathbb{I}(x_p)}, \tag{12}$$

where $E_t(x_p)$ is the expectation of the gradient of element x_p and $\mathbb{I}(x_p)$ is an indicator function that is defined as

$$\mathbb{I}(x_p) = \begin{cases} 1 & \text{if} \quad x_p \in \mathbf{D}_t^i \\ 0 & \text{if} \quad x_p \notin \mathbf{D}_t^i \end{cases} \tag{13}$$

Finally, the update direction GS_t is set as the vector composed of $\text{sign}(E_t(x_p) - \tau)$ for $x_p \in \mathbf{B}_t$, where τ is a threshold to control the ratio of the negative gradient.

2) Momentum Iteration Strategy: In addition to the elements selected by MCTS, the k elements with the largest momentum gradient are used to update the gradient. The momentum iteration strategy proposed in this paper can be expressed as

$$\begin{aligned} \mathbf{A}_t &= \mathrm{top}_k M_t, \\ M_t &= \mu \cdot M_{t-1} + (X_{t-1} - X_{t-2}), \quad M_0 = \mathbf{0}_{1\times N}, \end{aligned} \tag{14}$$

where M_t is the momentum of the t_{th} iteration, μ is the decay factor, and $\mathbf{A}_t$ is the set of k selected elements. The update direction GM_t is set as the vector $\mathrm{sign}(G(\mathbf{A}_t))$, where $G(\mathbf{A}_t)$ is the gradient of $\mathbf{A}_t$ calculated by Eq. (11). The momentum iteration method can accumulate gradients and accelerate the convergence speed of gradient descent.

Algorithm 2 Monte Carlo Gradient Sign Attack (MGSA)

```
Input:
  Original benign input $X$; Target phrase $y$; Victim model $L$
Output:
  Adversarial audio sample $X'$;
Initialize the following parameters:
  • Maximum iteration $T$;
  • Expectation threshold $\tau$;
  • Decay factor $\mu$; Learning rate $\beta$.
  • Root node $N_0 = \{x_1, x_2, \ldots, x_p\}$
$M_0 \leftarrow \mathbf{0}_{1\times N}$
$t \leftarrow 0,\ X_0 \leftarrow X$
$MCT \leftarrow N_0$
While $t < T$ and $Decode(x_t)! = y$ do
  $\mathbf{B}_t \leftarrow EGSA(x_t, y, T, L, MCT)$ // Call Algorithm 1
  Divide $\mathbf{B}_t$ into $\mathbf{D}_t = \{\mathbf{D}_t^1 \cdots \mathbf{D}_t^m\}$
  Perform batch gradient estimation on $\mathbf{D}_t$ to obtain
  $\{G(\mathbf{D}_t^1), G(\mathbf{D}_t^2) \cdots G(\mathbf{D}_t^m)\}$
  Calculate the expectation of gradient sign by Eq. (12)
  $GS_t \leftarrow \sum_{x_p \in B_t} e_p \cdot \mathrm{sign}\ (E_t(x_p) - \tau)$
  $M_{t+1} \leftarrow \mu \cdot M_t + (X_t - X_{t-1})$
  $\mathbf{A}_t \leftarrow \mathrm{top}_k(M_t^1, M_t^2, \ldots, M_t^n)$
  For $x_i \in \mathbf{A}_t$ do
    $G(x_i) \leftarrow \frac{L(X_t + e_i \cdot \delta, y) - L(X_t, y)}{\delta}$
  End For
  $GM_t \leftarrow \sum_{x_i \in \mathbf{A}_t} e_i \cdot \mathrm{sign}\ G(x_i)$
  $X_{t+1} \leftarrow X_t + \beta \cdot (GS_t + GM_t)$
  $t \leftarrow t + 1$
End While
$X' \leftarrow X_t$
return $X'$
```

The details of MGSA are provided in Algorithm 2. In each iteration, we update the adversarial example with the above two strategies as follows:

$$X_{t+1} = X_t + \beta \cdot (GS_t + GM_t), \tag{15}$$

where GS_t and GM_t are the gradient signs from the sampling gradient sign strategy and momentum iteration strategy, respectively.

E. Robustness Training

In real scenarios, adversarial examples often need to be played outside the target ASR system and re-recorded, which introduces new noise and sequential offset. To improve the robustness of adversarial examples, we introduce robustness training to the proposed method.

The frequency range of human speech is typically 20 to 200,00 HZ, and adversarial examples outside the legal frequency range will be truncated when they are re-recorded. Therefore, we first perform bandpass filtering on the adversarial examples. The range of the filter is set to 1000 to 4000 HZ in our experiments.

ASR systems conform to the acoustic time window to retain speech features, so that the speech features satisfy the short-term stability of the speech even if there is time sequence offset after framing. Different from natural speech, adversarial examples are randomly searched in the gradient direction and have no continuity or smoothness. Therefore, high-frequency enhancement and framing operations after the time sequence offset of speech will destroy adversarial examples. To improve the offset robustness, random offset are introduced to adversarial examples in MGSA.

The perturbation introduced during the rerecording varies and includes not only the noise in the air, but also the room reverberation and electronic noise related to equipment. Therefore, it is difficult to directly capture the distribution of noise in the real world. According to relevant studies in the field of speech enhancement, a random noise fragment in a natural environment can be generated by a linear combination of basic noise. We sample white Gaussian noise to simulate natural noise, so the transfer on the adversarial examples can be expressed as

$$t(x) = pad(r, x) + \epsilon, \quad \epsilon \sim \mathcal{N}, r \sim \mathcal{R}, \tag{16}$$

where, $pad(r, x)$ is filled with r bit zeros before x, $\mathcal{N}$ is a Gaussian distribution and $\mathcal{R}$ is the distribution of the sequential offset that is setted as a uniform distribution in our experiments. Therefore, the objective function with robustness training can be formulated as

$$L_{RT} = E_{\epsilon \sim \mathcal{N}, r \sim \mathcal{R}}[L_{CTC}(t(x + B(\delta)), T)], \tag{17}$$

where T is the target text of the adversarial attack and B is the bandpass filter. We optimize L_{RT} with MGSA, and the experiments are described in detail in section IV

IV. Experiment

In this section, experiments are conducted to test the effectiveness of the proposed attack. The experimental settings, including the dataset, the target model and evaluation metrics, are described in Section IV-A. The influence of the hyperparameters of MCT is analyzed in Section IV-B. The effects of the sampling gradient sign strategy and momentum iteration strategy are presented in Section IV-C. Finally, we compare the proposed method with the state-of-the-art methods in Section IV-D.

A. Experimental Settings

1) Datasets: Mozilla common voice dataset (MCVD) and LibriSpeech dataset are chosen in our experiments. MCVD is the largest publicly available voice dataset. It contains nearly 1,400 hours of voice data from more than 42,000 contributors. The LibriSpeech corpus [55] is a collection of approximately 1,000 hours of audiobooks, and it is suitable for training and evaluating ASR systems. We randomly choose 100 audio samples from the LibriSpeech and MCVD corpus, respectively, and segment each sample into two-second clips. Unless otherwise specified, all our experimental results are averaged over these 200 instances.

2) Target Model: DeepSpeech [31] is adopted as the target model in our experiment. As a current state-of-the-art ASR model, this model has become a common target for counter-attacks [14], [18], [56], [57]. We treat it as a black-box model and can only access the output logits of the model.

3) Evaluation Metrics: We use the following indicators to measure the performance of the attack methods.

- **The success Rate (SR)** refers to the proportion of adversarial examples recognized as target texts by the target model.
- **The number of queries (NoQ)** refers to the visit number of the target model during the generation of adversarial examples.
- **The pearson correlation coefficient (PCC)** is an indicator used to measure the similarity between the original audio and adversarial examples.
- **The word error rate (WER)** is an indicator used to measure the difference between the target text and the decoded text, and is calculated by

$$WER = 100 \cdot \frac{D_{word} + S_{word} + I_{word}}{N_{word}}\%, \tag{18}$$

where D_{word}, S_{word} and I_{word} are the numbers of deletions, substitutions, and insertions of words, respectively, and N_{word} is the total number of words in the target text.

- **The signal-to-noise ratio (SNR)** is used to measure the level of noise added by the attack algorithm, and can be calculated by

$$SNR = 10\log_{10}(\frac{\mathrm{dB}_{audio}}{\mathrm{dB}_{noise}}), \tag{19}$$

where dB_{audio} and dB_{noise} are the energies of the original audio and the noise, respectively.

4) Target Text: Untargeted attacks against the ASR model are usually considered meaningless, while the existing black-box attacks introduce excessive noise or require intensive queries for complex target texts. To better evaluate the effectiveness of the proposed method, we first adopt short texts commonly used in related works as the target texts, such as "OK Google", "Thank You", "Hello World" and "Open the Door". Then, we choose more complex texts, such as "Ask capital one to make a credit card payment" and "Call the police for help quickly".

TABLE I
ACCURACY RATE OF SAMPLING GRADIENT ESTIMATION IN THE WHOLE ITERATION PROCESS OF MGSA

Target text	Call the police for help quickly	Send a message to Derek	Please restart the phone
LibriSpeech	82.2%	85.1%	81.8%
Common Voice	81.0%	83.4%	81.7%

B. Hyperparameter Analysis of the Monte Carlo Tree Search

In this section, we analyze the influence of the hyperparameters in MCTS and determine the appropriate hyperparameter values.

The hyperparameters that play a key role in the search efficiency include the number m of branches of each layer, the size b of the leaf, and the number s of search rounds. Since the target of the search step is to obtain effective elements with the least number of queries, we define the search effect E as the ratio of the average gradients of the selected elements to the number of queries:

$$E = \frac{\sum_{i\in s} grad_i}{n \cdot \sum_{i\in s} Q_i}, \tag{20}$$

where n is the number of selected elements, $grad_i$ is the gradient of the selected leaf in the i_{th} search, and Q_i is the number of queries.

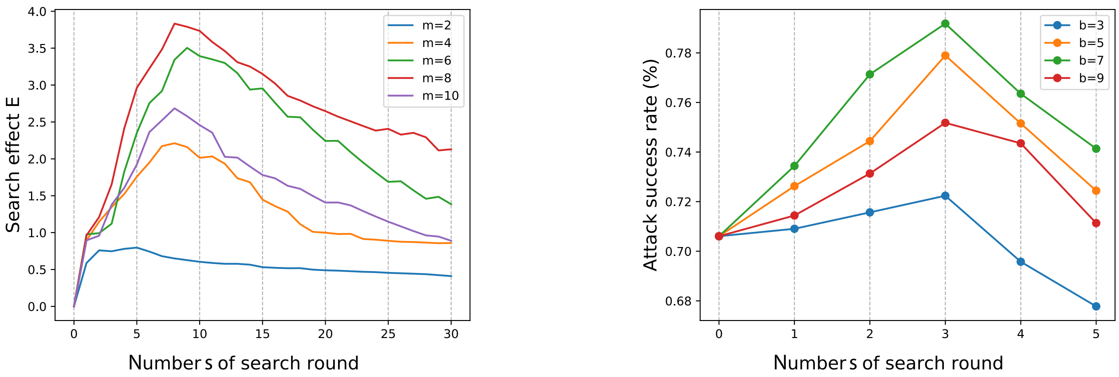
The number m of branches affects the number of queries in each round of the search, and the number s of search rounds affects the gradients of the selected elements. A larger m leads to more queries, while a smaller m means that there is more effective element loss at the high layer of the search tree. With an increase in s, an increasing number of effective elements are selected, but the gain of the new queries decreases. The influence of m and s on the search effect E is shown in Fig. 6(a). The experimental results are obtained after 100 iterations. It can be seen that the search performance reaches the highest level when $s = 7$ and $m = 8$.

In addition, we test the influence of leaf size b on the performance of MCTS under the conditions of $Q = 50{,}000$ and $m = 8$. Fig. 6(b) shows that random selection (the number of search rounds $b = 0$) can only achieve a 71% attack success rate after 50,000 queries, while the attack success rate increases to 79% when $b = 7$. The algorithm performs best when the leaf size is set to 7, and invalid elements will be mixed into the selected leaves when the leaf size continues to increase, thereby reducing the success rate of the attack.

C. Experimental Analysis of the Iterative Method

In Section III-D, we propose a sampling gradient sign strategy and momentum iteration strategy to improve the performance of the algorithm. In this section, we test the accuracy of the gradient sign estimation strategy and the effect of the momentum iteration strategy.

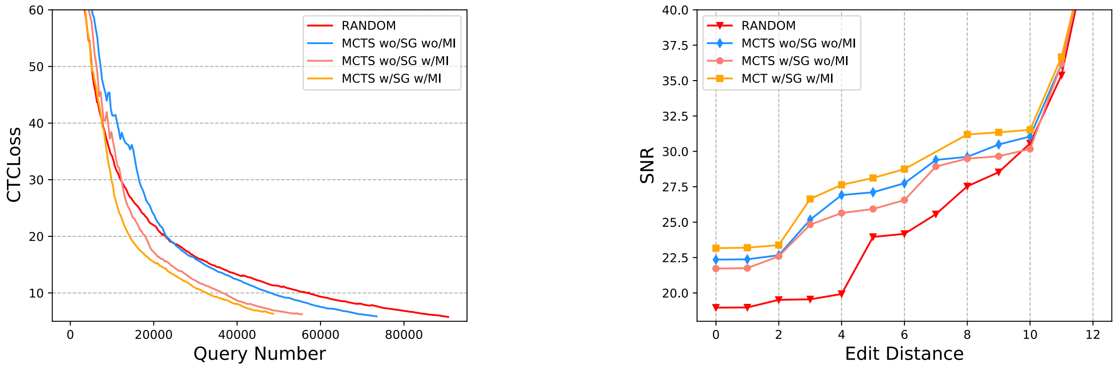
Table I reports the accuracy of sampling gradient sign estimation in the whole iteration process, where the sampling



(a) Influence of m and s on the search effect E

(b) Influence of b on the attack success rate

Fig. 6. **Hyperparametric analysis:** search efficiency and success rate of MGSA on the branch m of each layer, the size b of leaf, and the number s of search rounds.



(a) Average search efficiency

(b) Success rate

Fig. 7. Comparison of convergence speed and SNR between with and without Monte Carlo tree, sampling gradient sign strategy, momentum iterative strategy.

TABLE II
IMPACT OF MOMENTUM ITERATION STRATEGY (MI) ON ATTACK EFFECT AFTER 200,000 QUERIES

Evaluation Metrics	WER	Success Rate	SNR(dB)	PCC
Without MI	13%	76.5%	21.7	0.9981
With MI	4%	92.8%	20.9	0.9973

TABLE III
ABLATION STUDY OF MGSA ON LIBRISPEECH

Settings	SR	NoQ	SNR
Random select	86.8%	165400	13.7
MCTS	98.7%	84200	18.9
+Momentum iteration	99.1%	80600	19.2
+Sampling gradient estimation	96.5%	72300	18.3

batch is set to 5. It can be seen that the accuracy of sampling gradient sign estimation is above 80%, which means that the sign of the gradient can be estimated correctly for most selected elements. The experimental results of the momentum algorithm are shown in Table II. In the experiment, we set the number of elements updated in each iteration to 100, and the decay factor μ is set to 0.9. It can be seen from Table II that WER is reduced by 9% when the momentum iteration strategy is used.

A comparison of the algorithm convergence with and without the proposed search strategy is presented in Fig. 7. Since there are fewer and fewer effective elements in the later stage of the process, random sampling cannot accurately capture the effective elements. Therefore, the efficiency of the proposed algorithm can be greatly improved by using MCTS. As shown in Fig. 7(a), although the convergence rate of the algorithm is slow in the early stage of the process after the introduction of MCTS, the total number of queries is less than the number of queries required for random sampling. After using the sampling gradient sign strategy, the number of queries for gradient updating decreases, which significantly reduces the total number of queries. As a result, the convergence speed of the algorithm is accelerated. A comparison of the SNRs is shown in Fig. 7(b). The SNR of the adversarial examples generated by MGSA is 30% higher than that of the adversial examples generated by the baseline method. The ablation study of the MGSA on LibriSpeech is shown in Table III. It can be seen that the number of queries decreases by 22% with MCTS compared with the baseline method. The experimental results show that the use of the sampling gradient sign strategy further reduces the number of queries by 17%.

TABLE IV

COMPARISON RESULTS OF GAA [45], GEA [27], PSO [42], NES [26], NGD [58], SGEA [18], AND MGSA (THE PROPOSED ATTACK) ON LIBRISPEECH AND COMMON VOICE DATASETS

Target text		**Send a message**					**Call the police**					**Restart the phone**				
Dataset	**Method**	**SR**	**NoQ**	**PCC**	**WER**	**SNR**	**SR**	**NoQ**	**PCC**	**WER**	**SNR**	**SR**	**NoQ**	**PCC**	**WER**	**SNR**
LibriSpeech	GAA	43%	279900	0.933	42%	13.7	40%	285800	0.935	42%	13.8	46%	273100	0.936	36%	13.6
	GEA	81%	172800	0.952	16%	14.2	80%	183100	0.955	15%	13.9	84%	178400	0.953	17%	14.3
	PSO	52%	259600	0.954	38%	14.5	41%	281600	0.951	59%	14.5	54%	246400	0.949	48%	15.9
	NES	89%	114200	0.981	6%	17.1	85%	135600	0.978	11%	16.9	88%	118400	0.977	7%	17.7
	NGD	88%	126600	0.984	5%	17.2	88%	124800	0.982	8%	17.0	93%	120100	0.981	4%	18.1
	SGEA	97%	84300	0.961	2%	14.7	95%	84500	0.961	5%	15.1	98%	78300	0.966	2%	15.4
	MGSA	**99%**	**63200**	**0.997**	**1%**	**19.5**	**99%**	**75200**	**0.997**	**1%**	**19.7**	**100%**	**63800**	**0.997**	**0%**	**20.5**
Common Voice	GAA	41%	285200	0.937	58%	13.6	41%	272400	0.938	33%	13.7	45%	268800	0.939	30%	13.9
	GEA	80%	168400	0.956	22%	14.1	82%	175200	0.954	14%	14.5	88%	169200	0.958	9%	14.2
	PSO	49%	265400	0.95	42%	15.4	55%	267800	0.964	32%	16.1	50%	272500	0.96	37%	15.9
	NES	82%	104200	0.981	6%	16.7	76%	133500	0.971	13%	15.8	86%	152900	0.976	5%	17.9
	NGD	85%	96300	0.988	4%	17.9	79%	129400	0.980	10%	16.2	90%	138900	0.982	3%	18.5
	SGEA	98%	80400	0.97	1%	15.2	97%	86300	0.964	2%	15.1	98%	82600	0.97	1%	15.7
	MGSA	**98%**	**72800**	**0.998**	**1%**	**19.7**	**98%**	**68200**	**0.997**	**1%**	**19.1**	**99%**	**71200**	**0.998**	**1%**	**18.9**

TABLE V

COMPARISION RESULTS OF GEA, PSO, NES, NGD, SGEA, AND MGSA ON MORE COMPLEX TARGET TEXTS ON LIBRISPEECH DATASETS

Adversarial Audio	**NGD**					**SGEA**					**MGSA**				
(Recognized result of DeepSpeech)	**SNR**	**NoQ**	**PCC**	**SR**	**WER**	**SNR**	**NoQ**	**PCC**	**SR**	**WER**	**SNR**	**NoQ**	**PCC**	**SR**	**WER**
Turn on flashlight	16.9	149500	0.977	87%	12%	15.4	133000	0.971	90%	9%	19.2	94600	0.996	97%	3%
Read last sms from boss	16.2	179800	0.975	84%	20%	14.7	197800	0.967	83%	21%	18.7	89500	0.994	91%	12%
Please restart the phone	15.8	204200	0.972	82%	24%	14.7	197200	0.968	83%	22%	18.6	108200	0.992	92%	10%
Send a message to Derek	15.7	213700	0.972	77%	29%	14.3	204800	0.965	79%	26%	18.4	121090	0.988	88%	15%
Show me my last messages	15.5	229800	0.968	67%	43%	14.2	207800	0.962	72%	36%	18.1	126100	0.989	82%	23%
Call the police for help quickly	15.3	236600	0.966	65%	61%	13.9	204700	0.954	70%	52%	17.6	162900	0.985	79%	36%
Remove all photos in my phone	15.3	244500	0.965	59%	64%	13.9	214500	0.956	61%	62%	17.7	157600	0.983	71%	46%
Clear SMS history from my phone	15.1	249200	0.963	52%	81%	13.8	229000	0.953	55%	76%	16.9	164200	0.978	65%	59%
What is my schedule for tomorrow	15.0	256300	0.963	44%	98%	13.7	236400	0.951	48%	91%	16.8	168200	0.976	59%	72%

D. Comparison Results

In this section, the proposed algorithm is compared with a genetic algorithm based attack (GAA) [45], gradient estimation attack (GEA) [27], selective gradient estimation attack (SGEA) [18], particle swarm optimization (PSO) attack [42], natural evolution strategy (NES) attack [26] and natural gradient descent (NGD) attack [58]. In our experiments, we set the same learning rate and decay rate for five algorithms (GEA, SGEA, NES, NGD and MGSA), as in [44]. It is worth noting that for NGD, we need to calculate the Fisher information matrix $F = \sum_{t=1}^{T} p(t|x,\delta)\nabla \log p(t|x,\delta)\nabla \log p(t|x,\delta)^T$, where T denotes all possible classes. Since NGD is used to attack the image classification model in [58], the classes are limited. However, the ASR model is sequence to sequence, and the possible outputs are almost infinite. To enable NGD to work on the ASR model, we reconstruct the Fisher information matrix as $F = p(t'|x,\delta)\nabla \log p(t'|x,\delta)\nabla \log p(t'|x,\delta)^T$, where t' is the target text. We adopt the same experimental setup for PSO as in [42]. In the comparison experiments, we set the upper limit of the number of queries to 300,000. Therefore, attacks are considered to have failed when the number of queries is greater than 300,000. Table IV reports the average results in terms of SR, PCC, NoQ, WER and SNR for the 200 samples sampled in section IV-A. It can be seen from Table IV that GAA and PSO usually cannot find effective adversarial examples even after 300,000 queries. Although NES, NGD, GEA and SGEA are able to realize successful targeted attacks, their SNRs are less than 18 dB. The success

TABLE VI

COMPARISON RESULTS ON ROBUSTNESS OF ADVERSARIAL EXAMPLE AMONG ATTACKS: GAA, GEA, PSO, NES, NGD, SGEA, MGSA (WITHOUT ROBUSTNESS TRAINING) AND MGSA (WITH ROBUSTNESS TRAINING). THE VALUE OF N IS THE SNR OF ADDED WHITE NOISE, AND THE VALUE OF O IS THE AMOUNT OF SEQUENTIAL OFFSET

Target text:	**WER(%)**								**SNR**	**NoQ**
Ok goolge	N(25)	N(20)	N(15)	N(10)	O(10)	O(50)	O(100)	N(20)+O(50)		
GAA	34.3	48.3	66.1	72.8	50.2	62.5	87.5	81.0	13.9	246900
GEA	31.5	48.1	64.1	76.5	52.3	70.5	75.3	90.0	14.4	158200
PSO	46.8	71.7	81.9	90.9	63.2	75.0	97.5	85.4	14.7	210500
NES	41.8	52.8	81.0	86.4	75.6	87.5	92.0	92.3	17.3	136000
NGD	44.4	48.7	63.8	77.0	48.5	63.3	70.5	97.4	17.4	143500
SGEA	24.5	40.3	47.5	70.3	62.5	74.6	85.2	80.3	14.8	72400
MGSA(wo/RT)	21.4	37.1	52.7	68.8	37.5	65.0	75.3	75.5	**20.8**	**49600**
MGSA(w/RT)	**2.4**	**10.7**	**26.3**	**45.8**	**4.3**	**12.5**	**25.1**	**44.7**	16.2	95300

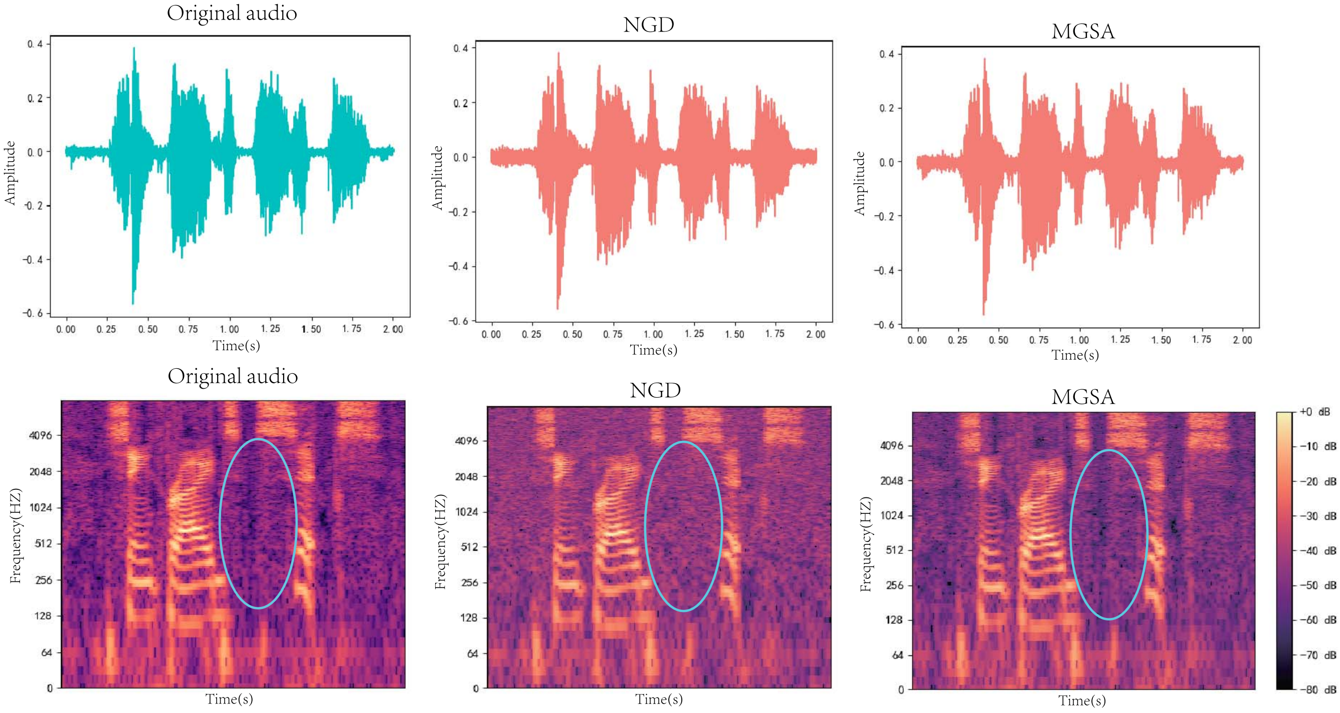


Fig. 8. Comparison of the waveforms and spectrograms among the original audio and the adversarial audios generated by NGD and MGSA.

rate of the MGSA algorithm reaches 98%, and the similarity between the adversarial examples and the original example reaches 0.99. The number of queries required by MGSA is also much lower than those required by the compared methods.

In Table V, we test the algorithm on more complex target texts. As the length of the target text increases, the number of queries required by MGSA grows at a significantly lower rate than that of the compared algorithms. This can be attributed to the selection of effective elements with MCTS.

We further test the robustness of the algorithm against noise and sequential offset for adversarial examples. In the robustness training of MGSA, Monte Carlo approximation is used to replace the expectation in L_{RT}:

$$\hat{L}_{RT} = \frac{1}{n_{MC}} \sum_{i=1}^{n_{MC}} L_{CTC}(t(x + BPF(\delta_i)), T), \tag{21}$$

where n_{MC} is set to 3 and δ_i is sampled from distributions $\mathcal{N}$ and $\mathcal{R}$ every 10 iterations. As shown in Table VI, MGSA has a lower WER score than the compared methods even without robustness training. Because less disturbance is introduced by MGSA, it is harder for the adversial samples to be destroyed by noise and sequential offset. After robustness training, the robustness of the adversarial examples is significantly improved. However, since the task is more difficult, the robustness training requires more queries. Although the algorithm is effective, the WER value is still high when both noise and offset are introduced, which is worth further study in the future.

We show the time consumption of the proposed method and the compared methods in Table VII. Due to the introduction of MCTS, each iteration of MGSA for longer than GEA and SGEA. However, since MGSA requires fewer queries,

TABLE VII
TIME CONSUMPTION OF GAA, GEA, PSO, NES, NGD, SGEA AND MGSA

Time(s)	**Target text**		
	Send a message	**Call the police**	**Restart the phone**
GAA	1536.2	1873.8	2032.9
GEA	373.1	557.2	716.8
PSO	1251.4	1439.6	1787.9
NES	841.7	1074.6	1218.3
NGD	860.1	1097.5	1302.7
SGEA	417.1	595.3	785.2
MGSA	618.7	893.5	932.3

fewer iterations are needed. Especially under long target texts, MGSA achieves a higher SNR value and is not significantly slower than the compared methods.

In Fig. 8, we compare the waveforms and spectra of adversarial examples generated by NGD and MGSA. From the waveforms, it can be seen that the MGSA adversial examples essentially maintain the waveform of the original audio sample, while the NGD adversial examples have obvious noise. In the spectra, the adversarial examples generated bu MGSA are almost indistinguishable from the original audio samples in the areas with no human voice (noted by blue circled), and NGD introduces more noise. This means that the noise introduced by MGSA is mixed with human voice samples and is thus less detectable.

V. CONCLUSION AND DISCUSSION

In this paper, we propose a novel adversarial attack method called MGSA, which takes advantage of the phenomenon of the dominant gradient in ASR systems. First, we introduce MCTS to improve the query efficiency. Then, we propose a sampling gradient sign strategy and momentum iterative strategy for updating adversarial examples. Finally, we introduce robustness training to improve the robustness of the adversarial examples. The extensive experimental results show that the proposed method requires fewer queries and introduces less noise than the existing black-box attacks. After robustness training, the robustness of adversarial examples generated by MGSA is improved significantly. The superiority of MGSA compared with the related methods provides direction for future works and encourages the investigation of queries to search for dominant gradient elements to improve the efficiency of black-box attacks.

Although MGSA achieves superior performance on black-box attack, there are still two issues that should be considered in future research. First, MGSA is a score-based black-box attack that relies on querying the CTC-loss of the target model, which makes it difficult to conduct attacks on the commercial ASR that only provides real-time decoding. An efficient decision-based black-box attack method that only requires the output texts from the target model, is expected in the future. Second, we find in our experiments that the robustness of the adversarial examples generated by the existing methods and the proposed MGSA is insufficient, resulting in a low success rate in "over-the-air" scenarios. Due to the lack of short-term stability, adversarial examples are easily destroyed by transforms such as noise and sequential offset transforms. Therefore, it is worth studying how to improve the stability of adversarial examples in the future.

REFERENCES

[1] B. Biggio et al., "Evasion attacks against machine learning at test time," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2013, pp. 387–402.

[2] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroSP)*, Mar. 2016, pp. 372–387.

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015, *arXiv:1412.6572*.

[4] S. Sun, P. Guo, L. Xie, and M.-Y. Hwang, "Adversarial regularization for attention based end-to-end robust speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 11, pp. 1826–1838, Nov. 2019.

[5] C. Szegedy et al., "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.

[6] J. Kos, I. Fischer, and D. Song, "Adversarial examples for generative models," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2018, pp. 36–42.

[7] A. Arnab, O. Miksik, and P. H. S. Torr, "On the robustness of semantic segmentation models to adversarial attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 888–897.

[8] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proc. ACM Sigsac Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 1528–1540.

[9] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2574–2582.

[10] Y. Zhang, Z. Jiang, J. Villalba, and N. Dehak, "Black-box attacks on spoofing countermeasures using transferability of adversarial examples," in *Proc. INTERSPEECH*, Oct. 2020, pp. 4238–4242.

[11] J. Li et al., "Textshield: Robust text classification based on multimodal embedding and neural machine translation," in *Proc. 29th USENIX Secur. Symp.*, 2020, pp. 1381–1398.

[12] J. Li, S. Ji, T. Du, B. Li, and T. Wang, "TextBugger: Generating adversarial text against real-world applications," 2018, *arXiv:1812.05271*.

[13] S. Hu, X. Shang, Z. Qin, M. Li, Q. Wang, and C. Wang, "Adversarial examples for automatic speech recognition: Attacks and countermeasures," *IEEE Commun. Mag.*, vol. 57, no. 10, pp. 120–126, Oct. 2019.

[14] T. Du, S. Ji, J. Li, Q. Gu, T. Wang, and R. Beyah, "SirenAttack: Generating adversarial audio for end-to-end acoustic systems," in *Proc. 15th ACM Asia Conf. Comput. Commun. Secur.*, Oct. 2020, pp. 357–369.

[15] S. Karpagavalli, R. Deepika, P. Kokila, K. U. Rani, and E. Chandra, "Automatic speech recognition: Architecture, methodologies and challenges—A review," *Int. J. Adv. Res. Comput. Sci.*, vol. 2, no. 6, pp. 326–331, 2011.

[16] Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5231–5240.

[17] Y.-Y. Ding, H.-J. Lin, L.-J. Liu, Z.-H. Ling, and Y. Hu, "Robustness of speech spoofing detectors against adversarial post-processing of voice conversion," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 3415–3426, 2021.

[18] Q. Wang, B. Zheng, Q. Li, C. Shen, and Z. Ba, "Towards query-efficient adversarial attacks against automatic speech recognition systems," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 896–908, 2021.

[19] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

[20] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial Intelligence Safety and Security*. Boca Raton, FL, USA: CRC Press, 2018.

[21] M. M. Cisse, Y. Adi, N. Neverova, and J. Keshet, “Houdini: Fooling deep structured visual and speech recognition models with adversarial examples,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[22] N. Carlini and D. Wagner, “Audio adversarial examples: Targeted attacks on speech-to-text,” in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2018, pp. 1–7.

[23] Y. Liu, X. Chen, C. Liu, and D. Song, “Delving into transferable adversarial examples and black-box attacks,” 2016, *arXiv:1611.02770*.

[24] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. Berkay Celik, and A. Swami, “Practical black-box attacks against machine learning,” 2016, *arXiv:1602.02697*.

[25] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, “Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models,” in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, 2017, pp. 15–26.

[26] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, “Black-box adversarial attacks with limited queries and information,” in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2137–2146.

[27] A. N. Bhagoji, W. He, B. Li, and D. Song, “Practical black-box attacks on deep neural networks using efficient query mechanisms,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 154–169.

[28] M. Zhou, Z. Qin, X. Lin, S. Hu, Q. Wang, and K. Ren, “Hidden voice commands: Attacks and defenses on the VCS of autonomous driving cars,” *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 128–133, Oct. 2019.

[29] V. A. Trinh and M. Mandel, “Directly comparing the listening strategies of humans and machines,” *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 29, pp. 312–323, 2021.

[30] C.-C. Chiu et al., “State-of-the-art speech recognition with sequence-to-sequence models,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4774–4778.

[31] A. Hannun et al., “Deep speech: Scaling up end-to-end speech recognition,” 2014, *arXiv:1412.5567*.

[32] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks,” in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 369–376.

[33] H. Zhou, W. Zhou, and H. Li, “Dynamic pseudo label decoding for continuous sign language recognition,” in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2019, pp. 1282–1287.

[34] X. Liu, K. Wan, Y. Ding, X. Zhang, and Q. Zhu, “Weighted-sampling audio adversarial example attack,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 4908–4915.

[35] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, “Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding,” 2018, *arXiv:1808.05665*.

[36] D. Iter, J. Huang, and M. Jermann, “Generating adversarial examples for speech recognition,” Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, Tech. Rep., 2017. [Online]. Available: https://web.stanford.edu/class/cs224s/project/reports_2017/Dan_Iter.pdf

[37] X. Yuan et al., “Commandersong: A systematic approach for practical adversarial voice recognition,” in *Proc. 27th USENIX Conf. Secur. Symp.*, 2018, pp. 49–64.

[38] M. Ravanelli, T. Parcollet, and Y. Bengio, “The pytorch-kaldi speech recognition toolkit,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6465–6469.

[39] B. Zheng et al., “Black-box adversarial attacks on commercial speech platforms with minimal information,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2021, pp. 86–107.

[40] Y. Zhang, H. Li, G. Xu, X. Luo, and G. Dong, “Generating audio adversarial examples with ensemble substituted models,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2021, pp. 1–6.

[41] Y. Chen et al., “Devil’s whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices,” in *Proc. 29th USENIX Secur. Symp. (USENIX Security)*, 2020, pp. 2667–2684.

[42] H. Mun, S. Seo, B. Son, and J. Yun, “Black-box audio adversarial attack using particle swarm optimization,” *IEEE Access*, vol. 10, pp. 23532–23544, 2022.

[43] W. Zong, Y.-W. Chow, and W. Susilo, “Black-box audio adversarial example generation using variational autoencoder,” in *Proc. Int. Conf. Inf. Commun. Secur.* Cham, Switzerland: Springer, 2021, pp. 142–160.

[44] M. Alzantot, B. Balaji, and M. Srivastava, “Did you hear that? Adversarial examples against automatic speech recognition,” 2018, *arXiv:1801.00554*.

[45] R. Taori, A. Kamsetty, B. Chu, and N. Vemuri, “Targeted adversarial examples for black box audio systems,” in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2019, pp. 15–20.

[46] G. Chaslot, *Monte-Carlo Tree Search*, vol. 24. Maastricht, The Netherlands: Maastricht University, 2010.

[47] D. Silver et al., “Mastering the game of Go with deep neural networks and tree search,” *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[48] J. Schrittwieser et al., “Mastering Atari, Go, chess and shogi by planning with a learned model,” *Nature*, vol. 588, no. 7839, pp. 604–609, 2020.

[49] D. Silver et al., “Mastering the game of go without human knowledge,” *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[50] C. B. Browne et al., “A survey of Monte Carlo tree search methods,” *IEEE Trans. Comput. Intell. AI Games*, vol. 4, no. 1, pp. 1–43, Mar. 2012.

[51] R. Munos, “From bandits to Monte-Carlo tree search: The optimistic principle applied to optimization and planning,” *Found. Trends Mach. Learn.*, vol. 7, no. 1, pp. 1–129, 2014.

[52] A. Weinstein and M. L. Littman, “Bandit-based planning and learning in continuous-action Markov decision processes,” in *Proc. 22nd Int. Conf. Automated Planning Scheduling*, 2012, pp. 1–9.

[53] C. Mansley, A. Weinstein, and M. Littman, “Sample-based planning for continuous action Markov decision processes,” in *Proc. 21st Int. Conf. Automated Planning Scheduling*, 2011, pp. 1–4.

[54] L. Busoniu, A. Daniels, R. Munos, and R. Babuska, “Optimistic planning for continuous-action deterministic systems,” in *Proc. IEEE Symp. Adapt. Dyn. Program. Reinforcement Learn. (ADPRL)*, Apr. 2013, pp. 69–76.

[55] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 5206–5210.

[56] E. Battenberg et al., “Exploring neural transducers for end-to-end speech recognition,” in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2017, pp. 206–213.

[57] A. Sriram, H. Jun, Y. Gaur, and S. Satheesh, “Robust speech recognition using generative adversarial networks,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5639–5643.

[58] P. Zhao, P.-Y. Chen, S. Wang, and X. Lin, “Towards query-efficient black-box adversary with zeroth-order natural gradient descent,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 6909–6916.

Shen Wang received the B.S. and M.E. degrees in electrical engineering and information technology from the TU-Dresden University of Technology, Dresden, Germany, in 2003 and 2007, respectively, and the Ph.D. degree in computer science and technology from the Harbin Institute of Technology, Harbin, China, in 2012. He is currently an Associate Professor with the Department of Computer Science, Harbin Institute of Technology. His current research interests include the adversarial attack and defense based on machine learning, image disguise, digital forensics, and quantum information processing.

Zhaoyang Zhang received the B.S. degree in materials engineering and the M.S. degree in computer technology from the Harbin Institute of Technology, Harbin, China, in 2014 and 2020, respectively, where he is currently pursuing the Ph.D. degree in cyberspace security. His current research interests include the adversarial attack and defense based on machine learning.

Guopu Zhu (Senior Member, IEEE) received the B.S. degree in transportation from Jilin University, China, in 2002, and the M.S. and Ph.D. degrees in control science and engineering from the Harbin Institute of Technology, China, in 2004 and 2007, respectively. He is currently a Professor with the Harbin Institute of Technology. He has authored or coauthored more than 50 articles in peer-reviewed international journals. His main research interests include multimedia security, image processing, and control theory. He serves as an Associate Editor for several journals, including IEEE TRANSACTIONS ON CYBERNETICS, IEEE SYSTEMS JOURNAL, *Journal of Information Security and Applications*, and *Electronics Letters*.

Xinpeng Zhang (Member, IEEE) received the B.S. degree in computational mathematics from Jilin University, China, in 1995, and the M.E. and Ph.D. degrees in communication and information system from Shanghai University, China, in 2001 and 2004, respectively. Since 2004, he was with the Faculty of the School of Communication and Information Engineering, Shanghai University, where he is currently a Professor. He was a Visiting Scholar with The State University of New York at Binghamton from 2010 to 2011, and also with Konstanz University as an Experienced Researcher, sponsored by the Alexander von Humboldt Foundation from 2011 to 2012. He is also with the Faculty of the School of Computer Science, Fudan University. His research interests include multimedia security, AI security, and image processing. He has published over 300 papers in these areas. He was an Associate Editor of the IEEE Transactions on Information Forensics and Security from 2014 to 2017.

Yicong Zhou (Senior Member, IEEE) received the B.S. degree in electrical engineering from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA, USA.

He is currently a Professor with the Department of Computer and Information Science, University of Macau, Macau. His research interests include image processing, computer vision, machine learning, and multimedia security.

Dr. Zhou is a fellow of the Society of Photo-Optical Instrumentation Engineers (SPIE) and was recognized as one of "Highly Cited Researchers" in 2020 and 2021. He received the Third Price of Macao Natural Science Award as a Sole Winner in 2020 and a co-recipient in 2014. He serves as an Associate Editor for the IEEE Transactions on Cybernetics, IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Circuits and Systems for Video Technology, and IEEE Transactions on Geoscience and Remote Sensing.

Jiwu Huang (Fellow, IEEE) received the B.S. degree from Xidian University, Xi'an, China, in 1982, the M.S. degree from Tsinghua University, Beijing, China, in 1987, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 1998. He is currently a Professor with the College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China. He has coauthored more than 300 publications in journals and conferences, with 71 of H-index. His current research interests include multimedia forensics and security.