Computers & Security 144 (2024) 103948

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

SUETA: Speaker-specific utterance ensemble based transfer attack on speaker identification system

Chu-Xiao Zuo, Jia-Yi Leng, Wu-Jun Li *

National Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China

ARTICLE INFO

Keywords:
Adversarial example
Black-box attack
Transfer-based attack
Utterance ensemble
Speaker identification

ABSTRACT

With the widespread application of speaker identification (SI) systems in security-related tasks, the robustness of SI systems against adversarial examples has garnered increasing attention. Existing works have demonstrated the vulnerability of SI systems to transfer-based black-box attacks, wherein attackers generate adversarial examples with a surrogate model and then transfer them to attack the target system. However, the attack success rate (ASR) of transfer-based black-box attacks is limited by the transferability of adversarial examples. As far as we know, few works have investigated enhancing the transferability of adversarial examples for speech utterances to attack SI systems. Furthermore, existing works only utilize a single utterance in the attack process, but in practical situations, an attacker can usually collect multiple utterances of a speaker. In this paper, we propose a novel transfer-based black-box attack method, called speaker-specific utterance ensemble based transfer attack (SUETA), to attack SI systems. To the best of our knowledge, SUETA is the first transfer-based black-box attack method that utilizes multiple utterances instead of a single one. Furthermore, we also propose an improved variant of SUETA, called SUETA+, by sharing gradients of utterances at the speaker-embedding level. Empirical results show that both SUETA and SUETA+ improve the ASR compared to the baselines under the classical cross-model situation. SUETA+ further shows additional improvement over SUETA, especially in the case of the untargeted attack. SUETA+ also outperforms the baselines in cross-dataset and cross-preprocessor situations, although the ASR for all transfer-based attacks decreases compared to that in the cross-model situation. Moreover, SUETA+ can significantly improve the ASR against commercial APIs (iFlytek and TalentedSoft) and the voice assistant (Tmall Genie) for the untargeted attack compared to the baselines.

1. Introduction

Speaker identification (SI) systems (Jahangir et al., 2021) are widely used to recognize the identity information of a speaker from input speech utterances. It is one of the most popular speaker recognition technologies and has been adopted in many real-world applications, including biometric authentication,[1,2] online payment and smartphone personalized service (Hammi et al., 2022). Due to the promising performance of deep neural network (DNN), an increasing number of existing SI systems (Snyder et al., 2018; Yu and Li, 2020; Desplanques et al., 2020) utilize the DNN-based models. However, existing works (Goodfellow et al., 2015; Szegedy et al., 2014) have shown that DNNs are vulnerable to adversarial examples, raising concerns about the safety and security of SI systems (Li et al., 2023) and highlighting the significance of research on adversarial robustness.

The adversarial robustness of an SI system is typically assessed by subjecting the system to attacks. Early works have found that the white-box adversarial attacks proposed in the image domain (Goodfellow et al., 2015; Kurakin et al., 2017; Madry et al., 2018; Carlini and Wagner, 2017) can also be effectively adapted for speech data to attack speech recognition systems (Ko et al., 2023), speaker verification systems (Kreuk et al., 2018; Li et al., 2020c; Zhang et al., 2021; Villalba et al., 2020), as well as SI systems (Gong and Poellabauer, 2017; Xie et al., 2021). Later works propose to utilize the unique acoustic characteristics of speech data to attack the SI systems. An auxiliary acoustic model is constructed by Li et al. (2020b) to output the adversarial perturbation directly. A frequency masking strategy is proposed by Wang et al. (2020) to generate imperceptible perturbations based on acoustic features. Shamsabadi et al. (2021) propose a steganography-inspired attack method that operates in the discrete cosine transform (DCT)

* Corresponding author.
E-mail addresses: zuochuxiao@smail.nju.edu.cn (C.-X. Zuo), lengjiayi@smail.nju.edu.cn (J.-Y. Leng), liwujun@nju.edu.cn (W.-J. Li).

[1] https://www.tdbank.com/bank/tdvoiceprint.html
[2] http://en.ccb.com/en/home/indexv3.html

https://doi.org/10.1016/j.cose.2024.103948
Received 12 March 2024; Received in revised form 11 May 2024; Accepted 6 June 2024
Available online 10 June 2024

domain. However, although these white-box adversarial attacks (Gong and Poellabauer, 2017; Xie et al., 2021; Li et al., 2020b; Wang et al., 2020; Shamsabadi et al., 2021) are effective, their reliance on full knowledge about the target SI system renders them impractical in many real-world applications. In contrast, black-box adversarial attacks offer greater practicality.

Black-box adversarial attacks can be divided into query-based attacks and transfer-based attacks, depending on whether the attacker interacts with the target system. In query-based attacks, the attacker continuously queries the target system and modifies the attack strategy according to the feedback. Based on the requirement for different types of feedback, the query-based attacks can be further divided into score-based attacks (Chen et al., 2017; Brendel et al., 2018) and decision-based attacks (Andriushchenko et al., 2020; Chen et al., 2020). Fakebob (Chen et al., 2021) and SMACK (Yu et al., 2023) are two representative score-based attacks on SI systems, which utilize the feedback of similarity scores. However, conducting the scored-based attack is unavailable in some practical cases, e.g., the similarity scores are inaccessible when attacking the commercial system Microsoft Azure. Deng et al. (2022) proposed a decision-based attack on the SI system, relying solely on the feedback of decision results. However, this approach requires thousands of queries that could be detected by abnormal query detection methods (Li et al., 2020a). In contrast, transfer-based attacks do not require feedback from the target system. Attackers generate adversarial examples with a surrogate model and then transfer them to attack the target system. However, due to the lack of interaction, the performance of transfer-based attacks is usually more limited than query-based attacks.

The transferability of the adversarial examples is the most important factor that affects the performance of transfer-based attacks. In the image domain, several methods have been proposed to enhance the transferability. Dong et al. (2018) have found that utilizing the momentum method in the optimization process enhances the transferability of the adversarial examples. He et al. (2023) later propose a feature-momentum method to enhance transferability. Model ensemble methods (Liu et al., 2017; Long et al., 2022b; Huang et al., 2023) can enhance the transferability by alleviating the adversarial examples from overfitting to a single local surrogate model. Data augmentation methods (Xie et al., 2019; Zou et al., 2020; Dong et al., 2019), which increase input diversity by using image transformations, can also enhance transferability. However, these methods are speaker-unrelated for speech data. As far as we know, few works have investigated enhancing the transferability of adversarial examples for speech utterances to attack SI systems. Furthermore, existing works only utilize a single utterance in the attack process, but in practical situations, an attacker can usually collect multiple utterances of a speaker.

In this paper, we propose a novel transfer-based black-box attack method, called speaker-specific utterance ensemble based transfer attack (SUETA), to attack SI systems. The main contributions are listed as follows:

- To the best of our knowledge, SUETA is the first transfer-based black-box attack method that utilizes multiple utterances instead of a single one.
- In SUETA, a speaker-specific utterance ensemble loss function is proposed, which enhances the transferability of adversarial examples compared with speaker-unrelated baselines.
- An improved variant of SUETA, called SUETA+, is further proposed by sharing gradients of utterances at the speaker-embedding level. To aggregate the gradients of utterances with different lengths, a technique called chain-rule-based gradient sharing (CRGS) is proposed in SUETA+.
- Empirical results show that both SUETA and SUETA+ effectively improve the attack success rate (ASR) compared to the projected gradient descent (PGD) attacks under the classical cross-model situation. SUETA+ further shows additional improvement over SUETA, especially in the case of the untargeted attack.

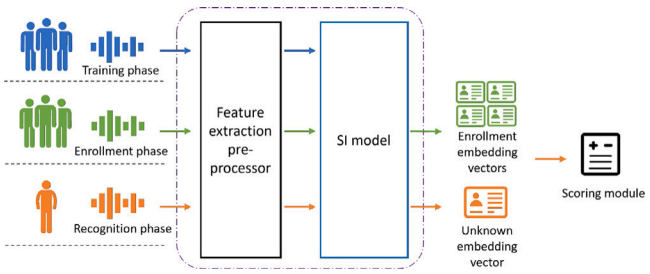


Fig. 1. A general architecture of SI systems.

- SUETA+ also outperforms the baselines in cross-dataset and cross-preprocessor situations, although the ASR for all transfer-based attacks decreases compared to that in the cross-model situation.
- Moreover, SUETA+ can significantly improve the ASR against commercial APIs (iFlytek and TalentedSoft) and the voice assistant (Tmall Genie) for the untargeted attack compared to the baselines.

2. Preliminaries

2.1. Speaker identification system

The SI system is used to identify the speaker of an utterance from a group of enrolled speakers. Fig. 1 shows the general architecture of an SI system, which consists of three phases: training, enrollment, and recognition. In the training phase, an SI model, also known as the background model, is trained by performing a classification task using utterances from the training dataset. This process enables the SI model to extract identity information from each utterance and convert the utterance into a speaker embedding. In the enrollment phase, the SI system aggregates several speaker embeddings for each enrolled speaker to construct distinct enrollment embeddings. In the recognition phase, the SI system first extracts the speaker embedding of an input utterance and then measures the similarity between the speaker embedding and the enroll embeddings. Subsequently, the SI system outputs the identification result according to the similarity scores. The implementation of SI systems can vary significantly according to the configuration of three key modules: feature extraction preprocessor, SI model, and scoring module, as described below.

Feature extraction preprocessor. The feature extraction preprocessor transforms the waveform utterances into acoustic features. Various algorithms have been proposed to extract different types of features, including the mel-spectrogram, mel-frequency cepstral coefficients (MFCC) (Muda et al., 2010), spectral subband centroid (SSC) (Thian et al., 2004), perceptual linear predictive (PLP) (Hermansky, 1990), and the log mel-scale filterbank (Fbank) (Hinton et al., 2012). Among these features, the Fbank feature has become popular when combined with the DNN-based SI model (Pardede et al., 2019). In practical implementation, additional signal processing operations, like data normalization, can be incorporated into the preprocessor. The widely-used open-source toolkit, Kaldi, also includes frequency cutoff, signal preemphasis, remove_dc_offset, edge snipping, and vocal tract length normalization while extracting Fbank features.

SI model. DNN-based SI model has been widely employed in the existing SI system. State-of-the-art (SOTA) SI models include X-Vector (Snyder et al., 2018), D-TDNN (Yu and Li, 2020), and ECAPA-TDNN (Desplanques et al., 2020), which mainly varies in model architecture. The X-vector model utilizes a time-delayed neural network (TDNN) with statistics pooling layers to extract speaker embedding. D-TDNN integrates information across different layers of the model by dense connections. ECAPA-TDNN further combines channel

features across different layers of the model and applies channel attention in both frame and statistics pooling layers, which helps extract the identity information into the speaker embedding.

Scoring module. Probabilistic linear discriminant analysis (PLDA) (Prince and Elder, 2007) and cosine similarity are the two widely used scoring methods. While PLDA is effective, it requires additional training. In contrast, cosine similarity offers effectiveness without additional training.

2.2. Identification task

The identification task of the SI system can be divided into two subtasks: close-set identification (CSI) and open-set identification (OSI). CSI identifies the speaker from a group of enrolled speakers without rejection. OSI deals with situations where an input utterance may belong to an unknown speaker that is not enrolled. The SI system incorporates a similarity threshold to reject an input whose similarity score falls below this threshold. This paper focuses on the CSI task. To formally define the problem, we denote an input utterance as $\boldsymbol{u}$, and the enrolled speaker group as G. The speakers in the enrolled group are numbered by $1, 2, \dots, K$. In the recognition phase, the SI system first extracts the speaker embedding $\boldsymbol{z} = E(\boldsymbol{u})$ using the SI model, then computes a similarity vector between $\boldsymbol{z}$ and the enrollment embeddings, denoted as $S(\boldsymbol{z}) = [[S(\boldsymbol{z})]_1, \dots, [S(\boldsymbol{z})]_K]$. In the CSI task, the SI system identifies the speaker of $\boldsymbol{u}$ according to the highest similarity score, which is defined as follows:

$$D(\boldsymbol{u}) = \arg\max_{i \in G} [S(E(\boldsymbol{u}))]_i.$$

2.3. Adversarial example

The definition of adversarial example (Goodfellow et al., 2015) is initially introduced in the image classification task. More specifically, an adversarial example is defined as an input that an attacker maliciously perturbs to cause misclassifications. The adversarial perturbation, which is usually limited by a perturbation budget ϵ under a certain distance metric $\|\cdot\|$, must be imperceptible to a human observer. Similarly, the adversarial examples for the SI systems can be defined as:

$$\{(\boldsymbol{u} + \boldsymbol{\delta}, k) | D(\boldsymbol{u}) = k, D(\boldsymbol{u} + \boldsymbol{\delta}) \neq k, \|\boldsymbol{\delta}\| \leq \epsilon\},$$

where $\boldsymbol{\delta}$ is the adversarial perturbation added to an input utterance $\boldsymbol{u}$ of speaker k.

Classical white-box attack methods. Existing works have proposed various white-box attack methods to generate adversarial examples on image tasks. Most of the classical white-box attack methods can be directly adapted to attack the SI systems. Fast gradient sign method (FGSM) (Goodfellow et al., 2015) performs a one-step movement from the original input along the gradient direction that maximizes the classification loss $\mathcal{L}$ (e.g., the cross-entropy loss function). Projected gradient descent (PGD) (Madry et al., 2018) is an iterative version of FGSM, exhibiting enhanced effectiveness compared to FGSM. In the attack process, PGD iteratively accumulates the perturbation $\boldsymbol{\delta}$ and projects the perturbation onto the ϵ-ball in each step. To attack the SI system, the attack step of PGD can be defined as:

$$\boldsymbol{\delta}_t = \prod_{B(\mathbf{0},\epsilon)} \left(\boldsymbol{\delta}_{t-1} + \eta \cdot \operatorname{sign}(\nabla_{\boldsymbol{u}} \mathcal{L}(D(\boldsymbol{u}), k))|_{\boldsymbol{u}=\boldsymbol{u}+\boldsymbol{\delta}_{t-1}} \right), \tag{1}$$

where η is the fixed step size, $\prod$ is the projection operator that maps the perturbation onto the ϵ-ball centered at the origin vector $\mathbf{0}$. PGD attack provides a standard pipeline to conduct adversarial attacks in the iterative optimization. By choosing different loss functions or optimization methods, researchers have developed different attack methods (Croce and Hein; Long et al., 2022a). Carlini and Wagner attack (C&W) (Carlini and Wagner, 2017) generates an adversarial example by searching for the minimal perturbation that changes the prediction of the classifier. In the paper of C&W attack, several choices of loss functions are proposed for the targeted attack, among which the most effective loss function is:

$$\mathcal{L}(F(\boldsymbol{x}), y) = -\max\{\max_{i \neq y}([H(\boldsymbol{x})]_i) - [H(\boldsymbol{x})]_y + c,\ 0\}, \tag{2}$$

where H is the pre-softmax layer of an image classifier F, $\boldsymbol{x}$ is an input image, y is a target label, and c is the margin parameter (also called the confidence parameter). Madry et al. (2018) point out that using the loss function in Eq. (2), C&W attack can also be implemented effectively in the form of iterative optimization.

Existing work (Chen et al., 2021) has adapted the C&W loss function to attack the SI system. For the untargeted attack, the loss function is defined as follows:

$$\mathcal{L}^{\text{CW-UT}}(\boldsymbol{u}) = -\max\{[S(E(\boldsymbol{u}))]_k - \max_{j \neq k}([S(E(\boldsymbol{u}))]_j) + c,\ 0\}, \tag{3}$$

where k is the label of the speaker for utterance $\boldsymbol{u}$. For the targeted attack, the loss function is defined as follows:

$$\mathcal{L}^{\text{CW-T}}(\boldsymbol{u}, y) = -\max\{\max_{j \neq y}([S(E(\boldsymbol{u}))]_j) - [S(E(\boldsymbol{u}))]_y + c,\ 0\}, \tag{4}$$

where y is the label of a target speaker.

Black-box transfer-based attack methods. In black-box transfer-based attacks, adversarial examples are usually generated by conducting iterative optimization-based attacks on a local surrogate model. The optimization process greedily moves the adversarial example toward the gradient direction in each step. Consequently, the adversarial example may drop into poor local maxima and overfit the local surrogate model, leading to poor transferability. Existing attacks from the image domain have tried several methods to enhance transferability. For example, the momentum method (Polyak, 1964), proposed to prevent an optimization process from getting trapped in local maxima, can be adopted in the PGD attack (Dong et al., 2018) to enhance the transferability of the adversarial example. He et al. (2023) later propose a feature-momentum method to enhance transferability. The model ensemble method, which leverages multiple local surrogate models (Liu et al., 2017; Zhang et al., 2020) or uses one model to emulate an ensemble of models (Long et al., 2022b; Huang et al., 2023), can alleviate the adversarial examples from overfitting to a single local surrogate model and enhance the transferability. Furthermore, attacks that craft a unified adversarial perturbation on multiple transformation-augmented images (Xie et al., 2019; Zou et al., 2020; Dong et al., 2019) also enhance the transferability. Among these transferability-enhancing methods, the optimization-based and model ensemble methods can be directly adapted to speech data. However, the data augmentation transformations employed for images, such as rotation and translation, cannot be directly adapted to speech data. Furthermore, existing works only utilize a single utterance in the attack process, but in practical situations, an attacker can usually collect multiple utterances of a speaker.

3. Threat model

The threat model specifies the security conditions and defines the situation considered by the attacker when designing the attack method. To define our attack formally, we describe the threat model according to the standards provided by Carlini et al. (2019).

3.1. Adversary goals

An attacker aims to craft a perturbation $\boldsymbol{\delta}$ for an utterance $\boldsymbol{u}$ from speaker k, such that the SI system makes a wrong decision on the perturbed utterance $\boldsymbol{u} + \boldsymbol{\delta}$. Depending on different specific goals, there are two types of attacks: untargeted and targeted. In the case of untargeted attack, the perturbed utterance should be identified as any other wrong speaker. In the case of targeted attack, the perturbed utterance should

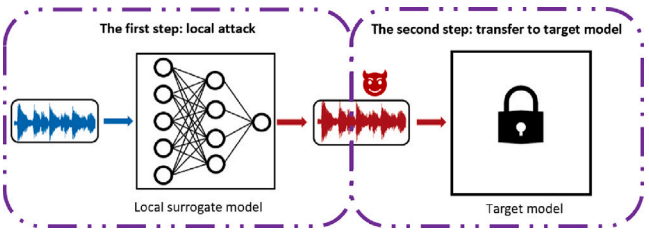


Fig. 2. A general architecture of the transfer-based attack.

be identified as a specified target speaker y. The goals of these attacks are defined as:

$$\begin{cases} D(\boldsymbol{u}+\boldsymbol{\delta}) \neq k & \text{(Untargeted)}, \\ D(\boldsymbol{u}+\boldsymbol{\delta}) = y & \text{(Targeted)}. \end{cases}$$

3.2. Adversary knowledge

In the common settings of transfer-based attacks, the attacker lacks knowledge about the target model and cannot receive feedback from the target model. They can only train surrogate models locally. Differences in implementation details between the surrogate and target models can greatly influence the effectiveness of the transfer-based attack. Existing attacks (Dong et al., 2018, 2019; Zou et al., 2020; Long et al., 2022b) primarily focus on the cross-model attack for image tasks, where only the model architecture differs between the surrogate and target models. However, in addition to the model architecture, the implementation details of an SI system also include the training dataset and feature extraction preprocessor. Furthermore, the attacker may either obtain the exact utterances enrolled in the target SI system or merely collect substitute enrolled utterances. In this paper, we first study the classical cross-model attack and then study the situations beyond the cross-model attack.

3.3. Adversarial capabilities

To conduct meaningful attacks, the attacker needs to limit the perturbation budget. Otherwise, the perturbed utterances will be easily distinguished, which is outside the scope of adversarial examples. In order to measure the transferability reasonably, we conduct attacks under the typical L_∞ constraint.[3]

4. Methodology

Fig. 2 illustrates the standard pipeline of the transfer-based attack, in which an attacker first conducts attacks on the local surrogate model and then transfers the adversarial examples to the target model. The attack method on the local surrogate model will greatly affect the transferability.

Inspired by data augmentation and model ensemble methods, in this section, we propose a novel transfer-based black-box attack method called speaker-specific utterance ensemble based transfer attack (SUETA). SUETA crafts adversarial perturbation on an ensemble batch of utterances by utilizing the unique characteristic of speech data that utterances from a specific speaker may vary in content and length but share a consistent voiceprint. Furthermore, we also propose an improved variant of SUETA, called SUETA+, by sharing gradients of utterances at the speaker-embedding level.

[3] As demonstrated in the conference version of this paper (Zuo et al., 2022), the performance of attack under the L_2 constraint is consistent with that of the attack under the L_∞ constraint. In this journal version, we omit the experiments of the L_2 constraint and leave more space for studying the transfer-based attacks with varying levels of adversary knowledge.

In the following content of this section, we first introduce the three key components of SUETA: speaker-specific utterance ensemble loss function, momentum-based iterative optimization, and buffer-based timely update strategy. Then, we introduce the two key components of SUTEA+: chain-rule-based gradient sharing and buffer-based timely update strategy for SUETA+.

4.1. Speaker-specific utterance ensemble loss function

To craft adversarial perturbation on an utterance from speaker k, SUETA collects N_k utterances to form an ensemble batch, denoted as $\mathbb{U}^k = \{\boldsymbol{u}^{k1}, \dots, \boldsymbol{u}^{kN_k}\}$. For each utterance $\boldsymbol{u}^{ki}$, we aggregate the similarity scores of $\mathbb{U}^k$ into an ensemble similarity vector by:

$$\begin{aligned} \tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k) &= \alpha S(E(\boldsymbol{u}^{ki})) \\ &\quad + (1-\alpha)\frac{1}{N_k - 1} \sum_{\boldsymbol{u}^{kj} \in \mathbb{U}^k, j \neq i} S(E(\boldsymbol{u}^{kj})), \end{aligned} \tag{5}$$

where α is a hyper-parameter. Based on the ensemble similarity vector, we introduce a novel loss function called speaker-specific utterance ensemble loss function ($\mathcal{L}^{\text{SUE}}$). The untargeted version of $\mathcal{L}^{\text{SUE}}$ is defined as:

$$\begin{aligned} &\mathcal{L}^{\text{SUE-UT}}(\boldsymbol{u}^{ki}, \mathbb{U}^k) \\ &= -\max\{[\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_k - \max_{j \neq k}([\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_j) + c,\ 0\} \end{aligned} \tag{6}$$

where c is the margin hyper-parameter. The untargeted version of $\mathcal{L}^{\text{SUE}}$ can be regarded as an utterance-ensemble extension of $\mathcal{L}^{\text{CW-UT}}$ in Eq. (3). But for the targeted attack, we do not directly adopt an utterance-ensemble extension of $\mathcal{L}^{\text{CW-T}}$ in Eq. (4) that can be defined as:

$$\begin{aligned} &\mathcal{L}^{\text{E-CW-T}}(\mathbb{U}^k, y) \\ &= -\max\{\max_{j \neq y}([\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_j) - [\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_y + c,\ 0\}. \end{aligned} \tag{7}$$

Instead, we propose the targeted version of $\mathcal{L}^{\text{SUE}}$ as:

$$\begin{aligned} &\mathcal{L}^{\text{SUE-T}}(\boldsymbol{u}^{ki}, \mathbb{U}^k, y) \\ &= -\max\{[\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_k - [\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_y + c,\ 0\}. \end{aligned} \tag{8}$$

In $\mathcal{L}^{\text{E-CW-T}}$, the maximum operation, $\max_{j \neq y}([\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_j)$, introduces dependence on all the output similarity scores of the local surrogate model, increasing the risk of overfitting. Therefore, in $\mathcal{L}^{\text{SUE-T}}$, we directly compute $[\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_k$ as an alternative. Formally speaking, $\mathcal{L}^{\text{SUE}}$ can be defined as:

$$\mathcal{L}^{\text{SUE}} = \begin{cases} \mathcal{L}^{\text{SUE-UT}} & \text{(Untargeted)}, \\ \mathcal{L}^{\text{SUE-T}} & \text{(Targeted)}. \end{cases} \tag{9}$$

In addition to the utilization of $\mathbb{U}^k$, another key component in the $\mathcal{L}^{\text{SUE}}$ is the margin factor c, which is also known as the confidence parameter. It has been investigated in the existing work (Chen et al., 2021) that increasing the margin factor in $\mathcal{L}^{\text{CW-UT}}$ and $\mathcal{L}^{\text{CW-T}}$ can enhance the transferability of adversarial examples when attacking the ivector-PLDA models (Dehak et al., 2011). Therefore, we fix a non-zero value of the margin factor in $\mathcal{L}^{\text{SUE}}$ to construct a margin-based loss function when attacking the DNN-based models. The margin-based loss function encourages the adversarial example to stay away from the decision boundary by a substantial distance, preventing the adversarial example from overfitting to the local surrogate model.

4.2. Momentum-based iterative optimization

SUETA performs attacks on the ensemble batch $\mathbb{U}^k$ with PGD-based iterative optimization. Since the lengths of the utterances in $\mathbb{U}^k$ are different, combining them into a batch tensor is not feasible for implementation. Therefore, the optimization process is divided into two

Algorithm 1 SUETA

Input:
Local surrogate model S, utterance batch $\mathbb{U}^k$ of speaker k; perturbation budget ϵ, number of iterations T, step size η, momentum factor β, ensemble factor α.

Output:
Adversarial perturbation vectors $\{\boldsymbol{\delta}_T^1, \dots, \boldsymbol{\delta}_T^{N_k}\}$;

Initialize $\mathbf{S}^k$ by Eq. (12), $\{\boldsymbol{\delta}_0^1, \dots, \boldsymbol{\delta}_0^{N_k}\} = \{\mathbf{0}, \dots, \mathbf{0}\}$, $\{\boldsymbol{g}_0^1, \dots, \boldsymbol{g}_0^{N_k}\} = \{\mathbf{0}, \dots, \mathbf{0}\}$;
for $t = 1$ to T **do**
  for $i = 1$ to N_k **do**
    Compute $\tilde{\boldsymbol{s}}^{ki}$ by Eq. (13);
    if Untargeted attack **then**
      $\mathcal{L}^{\text{SUE}} \leftarrow -\max(\tilde{\boldsymbol{s}}_k^{ki} - \max_{j\neq k}\{\tilde{\boldsymbol{s}}_j^{ki}\} + c,\ 0)$;
    end if
    if Targeted attack **then**
      $\mathcal{L}^{\text{SUE}} \leftarrow -\max(\tilde{\boldsymbol{s}}_k^{ki} - \tilde{\boldsymbol{s}}_y^{ki} + c,\ 0)$;
    end if
    $\boldsymbol{g}_t^i \leftarrow \beta \cdot \boldsymbol{g}_{t-1}^i + \nabla_{\boldsymbol{u}}\mathcal{L}^{\text{SUE}}$;
    $\boldsymbol{\delta}_t^i \leftarrow \prod_{B(\mathbf{0},\epsilon)}(\boldsymbol{\delta}_{t-1}^i + \eta \cdot \text{sign}(\boldsymbol{g}_t^i))$;
    $\mathbf{S}_{:,i}^k \leftarrow S(E(\boldsymbol{u}^{kj} + \boldsymbol{\delta}_t^i))$;
  end for
end for
return $\{\boldsymbol{\delta}_T^1, \dots, \boldsymbol{\delta}_T^{N_k}\}$;

layers of loops: the outer loop iterates over total attack steps, and the inner loop iterates over utterances in the ensemble batch. Instead of directly performing the PGD step in Eq. (1), we incorporate momentum into the optimization process following the work of Dong et al. (2018). For the untargeted attack, SUETA crafts perturbation $\boldsymbol{\delta}_t^i$ for the ith utterance $\boldsymbol{u}^{ki}$ at the tth attack step as:

$$\begin{aligned}\boldsymbol{g}_t^i &= \beta \cdot \boldsymbol{g}_{t-1}^i + \nabla_{\boldsymbol{u}}\mathcal{L}^{\text{SUE-UT}}(\boldsymbol{u}, \mathbb{U}^k)|_{\boldsymbol{u}=\boldsymbol{u}^{ki}+\boldsymbol{\delta}_{t-1}^i},\\ \boldsymbol{\delta}_t^i &= \prod_{B(\mathbf{0},\epsilon)}(\boldsymbol{\delta}_{t-1}^i + \eta \cdot \text{sign}(\boldsymbol{g}_t^i)),\end{aligned} \tag{10}$$

where $\boldsymbol{g}_t^i$ is the momentum vector. For the targeted attack, the attack step is similar:

$$\begin{aligned}\boldsymbol{g}_t^i &= \beta \cdot \boldsymbol{g}_{t-1}^i + \nabla_{\boldsymbol{u}}\mathcal{L}^{\text{SUE-T}}(\boldsymbol{u}, \mathbb{U}^k, y)|_{\boldsymbol{u}=\boldsymbol{u}^{ki}+\boldsymbol{\delta}_{t-1}^i},\\ \boldsymbol{\delta}_t^i &= \prod_{B(\mathbf{0},\epsilon)}(\boldsymbol{\delta}_{t-1}^i + \eta \cdot \text{sign}(\boldsymbol{g}_t^i)).\end{aligned} \tag{11}$$

4.3. Buffer-based timely update strategy

SUETA adopts a timely update strategy in the attack process. More specifically, in each iteration of the inner loop, i.e., crafting the perturbation $\boldsymbol{\delta}_t^i$ in Eq. (10) or Eq. (11), we calculate the $\mathcal{L}^{\text{SUE}}$ on the newly perturbed utterances instead of on the original batch $\mathbb{U}^k$. However, the process of recalculating $\mathcal{L}^{\text{SUE}}$ in each iteration requires additional computations, which involve feedforwarding the entire ensemble batch to the SI model. To reduce the computational cost, SUETA employs a memory buffer matrix $\boldsymbol{S}^k \in \mathbb{R}^{K\times N_k}$ to store the similarity vectors for each utterance. The buffer is initialized as[4]:

$$\boldsymbol{S}^k = [S(\boldsymbol{u}^{k1}), S(\boldsymbol{u}^{k2}), \dots, S(\boldsymbol{u}^{kN_k})]. \tag{12}$$

[4] In this paper, the one-dimensional vectors are regarded as column vectors when performing matrix operations. In order to avoid notation conflicts, we use $[]^{tr}$ instead of $[]^T$ to represent the transpose of a matrix. The symbol T will represent the total number of steps of the iterative attack.

In the ith step of the inner loop, we update an ensemble similarity vector $\tilde{\boldsymbol{s}}^{ki}$ for each utterance $\boldsymbol{u}^{ki}$, defined as:

$$\tilde{\boldsymbol{s}}^{ki} = \alpha S(E(\boldsymbol{u}^{kj} + \boldsymbol{\delta}_{t-1}^i)) + (1-\alpha)\frac{1}{N_k - 1}\sum_{j\neq i}\mathbf{S}_{:,j}^k. \tag{13}$$

The loss functions $\mathcal{L}^{\text{SUE}}$ is then computed by substituting $\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)$ with $\tilde{\boldsymbol{s}}^{ki}$ in Eq. (6) or Eq. (8). The memory buffer matrix is updated by $\boldsymbol{S}_{:,i}^k = S(E(\boldsymbol{u}^{kj} + \boldsymbol{\delta}_t^i))$ after obtaining $\boldsymbol{\delta}_t^i$ in each inner loop iteration. The whole optimization process of SUETA is summarized in Algorithm 1.

4.4. Chain-rule-based gradient sharing

While SUETA leverages an ensemble batch of utterances, the gradients of the utterances are not fully utilized. For example, the gradient of $\boldsymbol{u}^{ki}$ with respect to the $\mathcal{L}^{\text{SUE-T}}$ can be factorized as:

$$\begin{aligned}&\frac{d\mathcal{L}^{\text{SUE-T}}(\boldsymbol{u}^{ki}, \mathbb{U}^k, y)}{d\boldsymbol{u}^{ki}}\\ &= \mathbb{I}([\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_k - [\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_y + c > 0)\\ &\qquad\cdot \frac{d([\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_y - [\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_k)}{d\boldsymbol{u}^{ki}}\\ &= \mathbb{I}([\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_k - [\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_y + c > 0)\\ &\qquad\cdot\left(\alpha\frac{d([S(E(\boldsymbol{u}^{ki}))]_y - [S(E(\boldsymbol{u}^{ki}))]_k)}{d\boldsymbol{u}^{ki}}\right.\\ &\qquad\left.+\frac{1-\alpha}{N_k-1}\sum_{j\neq i}\frac{d([S(E(\boldsymbol{u}^{kj}))]_k - [S(E(\boldsymbol{u}^{kj}))]_y)}{d\boldsymbol{u}^{ki}}\right)\\ &= \mathbb{I}([\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_k - [\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_y + c > 0)\\ &\qquad\cdot\alpha\frac{d([S(E(\boldsymbol{u}^{ki}))]_y - [S(E(\boldsymbol{u}^{ki}))]_k)}{d\boldsymbol{u}^{ki}},\end{aligned} \tag{14}$$

where $\mathbb{I}$ is the indicator function defined as:

$$\begin{aligned}&\mathbb{I}([\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_k - [\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_y + c > 0)\\ &= \begin{cases}1 & \text{if } [\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_k - [\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_y + c > 0,\\ 0 & \text{if } [\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_k - [\tilde{S}(\boldsymbol{u}^{ki}, \mathbb{U}^k)]_y + c \leq 0.\end{cases}\end{aligned} \tag{15}$$

In contrast, consider a non-ensemble version of $\mathcal{L}^{\text{SUE-T}}$, which can be defined as:

$$\begin{aligned}&\mathcal{L}^{\text{NE-SUE-T}}(\boldsymbol{u}^{ki}, y)\\ &= -\max\{[S(E(\boldsymbol{u}^{ki}))]_k - [S(E(\boldsymbol{u}^{ki}))]_y + c,\ 0\}.\end{aligned} \tag{16}$$

The gradient of $\boldsymbol{u}^{ki}$ with respect to the $\mathcal{L}^{\text{NE-SUE-T}}$ can be factorized as:

$$\begin{aligned}&\frac{d\mathcal{L}^{\text{NE-SUE-T}}(\boldsymbol{u}^{ki}, y)}{d\boldsymbol{u}^{ki}}\\ &= \mathbb{I}([S(E(\boldsymbol{u}^{ki}))]_k - [S(E(\boldsymbol{u}^{ki}))]_y + c > 0)\\ &\qquad\cdot\frac{d([S(E(\boldsymbol{u}^{ki}))]_y - [S(E(\boldsymbol{u}^{ki}))]_k)}{d\boldsymbol{u}^{ki}}.\end{aligned} \tag{17}$$

The primary difference between Eqs. (14) and (17) lies in the indicator function. When the indicator function is activated, the actual value of the gradient, $\frac{d([S(E(\boldsymbol{u}^{ki}))]_y - [S(E(\boldsymbol{u}^{ki}))]_k)}{d\boldsymbol{u}^{ki}}$, is only correlated to the single utterance $\boldsymbol{u}^{ki}$ for both the ensemble and non-ensemble loss functions.

To leverage the gradient information of the entire ensemble batch, we introduce a technique called chain-rule-based gradient sharing (CRGS) to share the gradients. While the lengths of utterances vary, the speaker embeddings obtained from the SI model share a uniform shape. Consequently, we can aggregate the gradients at the speaker-embedding level and then propagate the aggregated gradient to the utterance using the chain rule. Formally speaking, by denoting the speaker embeddings of the ensemble batch as:

$$\{\boldsymbol{z}^{kj} | \boldsymbol{z}^{kj} = E(\boldsymbol{u}^{kj}), \boldsymbol{u}^{kj} \in \mathbb{U}^k, \boldsymbol{z}^{kj} \in \mathbb{R}^M\}, \tag{18}$$

we decompose the gradient for each utterance $\boldsymbol{u}^{kj}$ by:

$$\frac{d\mathcal{L}^{\text{SUE}}}{d\boldsymbol{u}^{kj}} = \left[\left[\frac{d\mathcal{L}^{\text{SUE}}}{d\boldsymbol{z}^{kj}}\right]^{tr} \cdot \boldsymbol{J}_{\boldsymbol{z}^{kj}}(\boldsymbol{u}^{kj})\right]^{tr}, \tag{19}$$

where $J_{z^{kj}}(u^{kj}) \in \mathbb{R}^{M \times l(u^{kj})}$ is the Jacobian matrix:

$$J_{z^{kj}}(u^{kj}) = \begin{bmatrix} \frac{dz_1^{kj}}{du_1^{kj}} & \frac{dz_1^{kj}}{du_2^{kj}} & \cdots & \frac{dz_1^{kj}}{du_{l(u^{kj})}^{kj}} \\ \frac{dz_2^{kj}}{du_1^{kj}} & \frac{dz_2^{kj}}{du_2^{kj}} & \cdots & \frac{dz_2^{kj}}{du_{l(u^{kj})}^{kj}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{dz_M^{kj}}{du_1^{kj}} & \frac{dz_M^{kj}}{du_2^{kj}} & \cdots & \frac{dz_M^{kj}}{du_{l(u^{kj})}^{kj}} \end{bmatrix}, \tag{20}$$

and $l(u^{kj})$ is the length of utterance. By performing a matrix multiplication of $J_{z^{ki}}(u^{ki})$ and $\frac{d\mathcal{L}^{\text{SUE}}}{dz^{kj}}$, the gradient information of the jth speaker embedding can be conveyed to the ith utterance. Based on this idea, SUETA+ aggregates the gradients of the ensemble batch with CRGS, which is defined as:

$$CRGS(u^{ki}) = \left[[\gamma \frac{d\mathcal{L}^{\text{SUE}}}{dz^{ki}} + \frac{1-\gamma}{N_k - 1} \sum_{j \neq i} \frac{d\mathcal{L}^{\text{SUE}}}{dz^{kj}}]^{tr} \cdot J_{z^{ki}}(u^{ki}) \right]^{tr}, \tag{21}$$

where γ is a hyper-parameter.

4.5. Buffer-based timely update strategy for SUETA+

SUETA+ also adopts the timely update strategy in the optimization process. In addition to the similarity buffer matrix S^k, we employ another memory buffer matrix, $G^k \in \mathbb{R}^{M \times N_k}$, to store the speaker-embedding-level gradients. G^k is initialized as a zero matrix.

In practical implementation, the size[5] of the Jacobian matrix in Eq. (21) is often too large, making it infeasible to compute the CRGS directly. Instead, we rearrange the computation order of the matrix multiplication, i.e., we first aggregate the gradients at the speaker-embedding level as:

$$\tilde{g}^{ki} = \gamma \frac{d\mathcal{L}^{\text{SUE}}}{dz^{ki}} + \frac{1-\gamma}{N_k - 1} \sum_{j \neq i} G_{:,j}^k. \tag{22}$$

Then, we perform a matrix multiplication of the aggregated gradients and the speaker embedding before computing the gradient on utterance, which can be formulated as follows:

$$CRGS(u^{ki}) = \frac{d[[\tilde{g}^{ki}]^{tr} \cdot E(u^{ki})]}{du^{ki}}. \tag{23}$$

SUETA+ updates $G_{:,i}^k = \frac{d\mathcal{L}^{\text{SUE}}}{dz^{ki}}$ in each inner loop iteration. The whole optimization process of SUETA+ is summarized in Algorithm 2.

Algorithm 2 SUETA+

Input:
Local surrogate model S, utterance batch $\mathbb{U}^k$ of speaker k; perturbation size ϵ, number of iterations T, step size η, momentum factor β, ensemble factor α, γ.

Output:
Adversarial perturbation vectors $\{\delta_T^1, \ldots, \delta_T^{N_k}\}$;

Initialize $\mathbf{S}^k$ by Eq. (12), $\{\delta_0^1, \ldots, \delta_0^{N_k}\} = \{\mathbf{0}, \ldots, \mathbf{0}\}$, $\{g_0^1, \ldots, g_0^{N_k}\} = \{\mathbf{0}, \ldots, \mathbf{0}\}$, $G^k = \{\mathbf{0}, \ldots, \mathbf{0}\}$;
for $t = 1$ to T **do**
  for $i = 1$ to N_k **do**
    Compute $\tilde{s}^{ki}$ by Eq. (13);
    if Untargeted attack **then**
      $\mathcal{L}^{\text{SUE}} \leftarrow -\max(\tilde{s}_k^{ki} - \max_{j \neq k}\{\tilde{s}_j^{ki}\} + c,\ 0)$;
    end if
    if Targeted attack **then**
      $\mathcal{L}^{\text{SUE}} \leftarrow -\max(\tilde{s}_k^{ki} - \tilde{s}_y^{ki} + c,\ 0)$;
    end if
    $G_{:,i}^k \leftarrow \frac{d\mathcal{L}^{\text{SUE}}}{dz^{ki}}$;
    Compute $\tilde{g}^{ki}$ by Eq. (22);
    Compute $CRGS(u^{ki} + \delta_{t-1}^i)$ by Eq. (23);
    $g_t^i \leftarrow \beta \cdot g_{t-1}^i + CRGS(u^{ki} + \delta_{t-1}^i)$;
    $\delta_t^i \leftarrow \prod_{B(\mathbf{0},\epsilon)} (\delta_{t-1}^i + \eta \cdot \text{sign}(g_t^i))$;
    $\mathbf{S}_{:,i}^k \leftarrow S(E(u^{kj} + \delta_t^i))$;
  end for
end for
return $\{\delta_T^1, \ldots, \delta_T^{N_k}\}$;

Table 1
Notation of different model architectures.

ID	A	B	C	D	E
Model architecture	X-Vector	X-Vector-CAM	D-TDNN	D-TDNN-CAM	ECAPA-TDNN

Table 2
Notation of combinations of different training datasets and feature extraction preprocessors.

ID	1	2	3	4
Training dataset	Librispeech	Librispeech	Librispeech	VoxCeleb2
Kaldi-style SPO	✓	✓	✗	✓
Normalization scale	32768	1	1	32768

5. Experiments

5.1. Evaluation setup

Feature extraction preprocessor. We evaluate transfer-based attacks between SI systems with different Fbank-based feature extraction preprocessors. Specifically, we employ different preprocessors based on two aspects: the utilization of the Kaldi-style signal processing operations (Kaldi-style SPO) and the scale of data normalization. The Kaldi-style SPO involves frequency cutoff, signal preemphasis, remove_dc_offset, edge snipping, and vocal tract length normalization. We use the *kaldi.fbank*, integrated in the *torchaudio* package, to implement the Fbank-based feature extraction preprocessor with Kaldi-style SPO. For the scale of data normalization, we investigate the impact of normalizing the waveform data value to 1 or 32768. For all of the different preprocessors, the waveform data is transformed to obtain an 80-dimensional Fbank feature using a 25 ms window with a 10 ms overlap.

Model architecture. We employ three SOTA models: X-Vector (Snyder et al., 2018), D-TDNN (Yu and Li, 2020), and ECAPA-TDNN (Desplanques et al., 2020). We also design X-Vector-CAM and Dense-TDNN-CAM models by adding a CAM layer (Yu et al., 2021) to the original models. These models are trained using the AAM-Softmax loss function (Yu et al., 2019).

Scoring module. We employ the cosine similarity to evaluate the similarity between speaker embeddings, aligning with the implementation of the SOTA models (Snyder et al., 2018; Yu and Li, 2020; Desplanques et al., 2020).

Dataset.[6] We employ three widely used datasets: TIMIT (Garofolo, 1993), LibriSpeech (Panayotov et al., 2015), and VoxCeleb (Nagrani et al., 2020). TIMIT comprises 630 speakers and 6300 utterances. We partition the speakers into training, validation, and test sets using a

[5] The length of the waveform obtained by a 5-second sentence at a sampling rate of 16 kHz is 80,000

[6] Please note that in the conference version of this paper (Zuo et al., 2022), the experiments are only conducted on the TIMIT dataset under the cross-model attack. To improve the convincingness of the experimental results, the major experiments are conducted on the larger datasets LibriSpeech and VoxCeleb2 in this journal version. In Section 5.4.3, we also show a summarized version of the original experimental results on the TIMIT dataset.

Table 3
ASR of the cross-model attack.

Surrogate model	Attack	Goal	Target model					Avg
			A1	B1	C1	D1	E1	
A1	Naive-PGD	Untargeted	*	**19.25**	**12.25**	13.88	7.63	13.25
	Margin-PGD		*	95.18	78.33	84.80	78.97	84.32
	SUETA		*	94.84	78.88	84.96	82.09	85.19
	SUETA+		*	**96.13**	**91.34**	**92.42**	**90.08**	**92.49**
C1	Naive-PGD		17.63	11.00	*	12.00	7.25	11.97
	Margin-PGD		74.50	86.01	*	86.68	74.84	80.51
	SUETA		77.75	88.75	*	87.58	78.67	83.19
	SUETA+		**88.88**	**93.71**	*	**93.96**	**90.50**	**91.76**
E1	Naive-PGD		26.50	23.00	21.13	22.38	*	23.25
	Margin-PGD		86.31	91.72	88.48	90.02	*	89.13
	SUETA		89.75	93.67	89.88	91.42	*	91.18
	SUETA+		**92.09**	**94.88**	**93.04**	**93.21**	*	**93.30**
A1	Naive-PGD	Targeted	*	6.00	2.63	3.88	2.38	3.72
	Margin-PGD		*	65.67	38.22	52.19	43.79	49.97
	SUETA		*	70.63	40.46	53.13	**48.42**	53.16
	SUETA+		*	**69.00**	**41.50**	**55.83**	47.71	**53.51**
C1	Naive-PGD		4.63	3.63	*	3.75	2.75	3.69
	Margin-PGD		37.88	50.81	*	53.29	43.78	46.44
	SUETA		37.25	48.71	*	50.96	42.63	44.89
	SUETA+		**41.17**	**52.13**	*	**55.00**	**45.42**	**48.43**
E1	Naive-PGD		8.88	10.50	8.63	9.63	*	9.41
	Margin-PGD		54.47	70.48	55.85	67.77	*	62.14
	SUETA		53.83	**73.59**	55.34	68.83	*	62.90
	SUETA+		**56.75**	72.08	**57.92**	**69.29**	*	**64.01**

ratio of 8:1:1. For the LibriSpeech dataset, we select the "train-clean-100" subset and divide it into Spk_{251}-train, Spk_{251}-test, Spk_{10}-enroll, and Spk_{10}-test, following the procedure outlined by Chen et al. (2021). Spk_{251}-train and Spk_{251}-test are used to train and validate the backbone SI models. These two subsets share a common pool of speakers, consisting of 126 males and 125 females, with utterance counts in a ratio of 9:1. Spk_{10}-test and Spk_{10}-enroll, which include the same speakers (five males and five females) but distinct voices, are used for evaluating the CSI task. The VoxCeleb dataset consists of two subsets: VoxCeleb1 and VoxCeleb2. We use the development set of VoxCeleb2 to train the SI models.

Hyper-parameters of the attack. We set $\epsilon = 0.002$ and step size $\eta = \epsilon/4$ under L_∞ constraint, following the settings in Chen et al. (2021, 2022). The number of attack iterations is set to $T = 10$. When attacking commercial APIs and the voice assistant in Sections 5.8 and 5.9, we set $\epsilon = 0.02$, step size $\eta = \epsilon/4$ under L_∞ constraint, and the number of attack iterations $T = 300$ to conduct stronger attacks. For the targeted attack, we select a random target y for each utterance. The hyper-parameters, α, γ, ensemble number, and β, are determined through a random search conducted on the validation set.

We train a collection of 20 SI models, each characterized by a different combination of the feature extraction preprocessor, model architecture, and training dataset. These models are denoted by a combination of the model architecture ID and the dataset&preprocessor ID, as illustrated in Tables 1 and 2. These models are sequentially labeled from A1 to E4 (A1, A2, A3, A4, … , E1, E2, E3, E4).

5.2. Evaluation metrics and baselines

We adopt ASR as the evaluation metric, which is defined as the proportion of successful attack samples to the total number of attack samples. To calculate the ASR, we iteratively sample all the utterances in the test set and perturb each utterance to generate the adversarial examples.

We compare SUETA and SUETA+ with the baseline method PGD, which uses $\mathcal{L}^{\text{CW-UT}}$ and $\mathcal{L}^{\text{CW-T}}$ as the loss functions. Specifically, we denote the PGD attack with a margin factor of $c = 0$ and a momentum factor of $\beta = 0$ as the Naive-PGD. We denote the PGD attack with a margin factor of $c = 1$ and a momentum factor of $\beta = 1$ as the Margin-PGD.

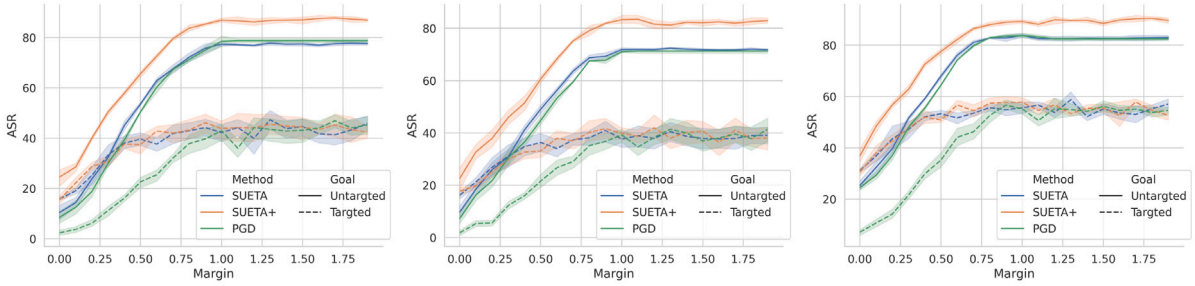
5.3. Experimental design

We first conduct controlled variable experiments on the differences between the surrogate and target models, according to the feature extraction preprocessor, model architecture, and the training dataset. In Section 5.4, we evaluate the classical cross-model situation, which has been studied in numerous existing work (Dong et al., 2018, 2019; Zou et al., 2020; Long et al., 2022b; Kreuk et al., 2018; Li et al., 2020c; Chen et al., 2021). Furthermore, to understand the effect of each component of our method, we conduct ablation study on the loss function (Section 5.4.1), margin factor (Sections 5.4.2 and 5.4.4), momentum factor (Sections 5.4.3 and 5.4.4), ensemble number (Section 5.4.5), and substitute enrolled utterances (Section 5.4.6) in the cross-model situation. Then, we stretch the experiments to the cross-dataset and cross-preprocessor situations in Sections 5.5 and 5.6.

To evaluate the attacks across diverse technology in practical situations, we conduct attacks with speech compression codecs in Section 5.7 under the cross-model situation. Then, in Section 5.8, we attack the SI systems of commercial APIs by feeding adversarial audio files to the exposed APIs. In Section 5.9, we attack the SI system of voice assistant Tmall Genie by playing the adversarial audio over the air.

5.4. Evaluation of the classical cross-model attack

In this part of the experiment, we evaluate the classical cross-model attack, where the training dataset and feature extraction preprocessor are consistent for the local surrogate and the target SI models. In addition, we assume that the attacker has access to the exact enrolled utterances in the target SI system, similar to the situation of cross-model attacks on image tasks (Dong et al., 2018, 2019; Zou et al., 2020; Long et al., 2022b) where only the model architectures are different. We attack three local surrogate models, A1, C1, and E1, and transfer the adversarial examples to the target models, A1, B1, C1, D1, and E1. The experimental results are presented in Table 3. In most cases, SUETA achieves a higher average ASR on each surrogate model compared to the Naive-PGD and Margin-PGD. SUETA+ further improves the performance of SUTEA and achieves the highest average ASR on each surrogate model in all the cases. Particularly in the case of the untargeted attack, SUETA+ demonstrates the highest ASR



(a) Transfer from A1 (b) Transfer from C1 (c) Transfer from E1

Fig. 3. ASR with varying margin. The momentum factor is fixed as $\beta = 0$ and the ASR is averaged on different target models.

Table 4
ASR of targeted attack for different loss functions.

Loss function	Optimization process	ASR
$\mathcal{L}^{\text{CW-T}}$	Naive-PGD	3.45
	Margin-PGD	37.44
$\mathcal{L}^{\text{NE-SUE-T}}$	Naive-PGD	0.10
	Margin-PGD	37.85
$\mathcal{L}^{\text{E-CW-T}}$	SUETA	35.57
	SUETA+	21.55
$\mathcal{L}^{\text{SUE-T}}$	SUETA	47.62
	SUETA+	**48.30**

across all target models. Notably, in the transfer attack from C1 to E1, SUETA+ increases the ASR by 82% compared to Naive-PGD and by 15% compared to Margin-PGD.

5.4.1. The effect of the loss functions for the targeted attack

We evaluate different loss functions for the targeted attack. We compare $\mathcal{L}^{\text{SUE-T}}$ in Eq. (8), the ensemble version of C&W loss $\mathcal{L}^{\text{E-CW-T}}$ in Eq. (7), C&W loss function $\mathcal{L}^{\text{CW-T}}$ in Eq. (4), and the non-ensemble loss function $\mathcal{L}^{\text{NE-SUE-T}}$ in Eq. (16). For the ensemble-based loss functions, $\mathcal{L}^{\text{SUE-T}}$ and $\mathcal{L}^{\text{E-CW-T}}$, the attacks are conducted based on the optimization process of the ensemble-based attack, SUETA and SUETA+. For the non-ensemble loss functions, $\mathcal{L}^{\text{CW-T}}$ and $\mathcal{L}^{\text{NE-SUE-T}}$, the attacks are conducted based on the optimization process of the non-ensemble attacks, Naive-PGD and Margin-PGD. We evaluate attacks across three models: A1, C1, and E1. Table 4 shows the results which are averaged on different surrogate and target models. $\mathcal{L}^{\text{NE-SUE-T}}$ achieves comparable ASR to $\mathcal{L}^{\text{CW-T}}$ for the non-ensemble attacks, $\mathcal{L}^{\text{SUE-T}}$ achieves significantly higher ASR compared to $\mathcal{L}^{\text{E-CW-T}}$ for the ensemble-based attack. Furthermore, the combination of $\mathcal{L}^{\text{SUE-T}}$ and the optimization process of SUETA+ achieves the highest ASR.

5.4.2. The effect of the margin factor

We fix the momentum factor in the loss function of PGD, SUETA, and SUETA+ as $\beta = 0$. Attacks are conducted across three models: A1, C1, and E1. Fig. 3 shows the ASR curve in relation to the margin factor c. When the value of the margin factor is near 0, SUETA and SUETA+ demonstrate notably higher ASR compared to PGD in the case of the targeted attack, and SUETA+ demonstrates notably higher ASR compared to PGD and SUETA in the case of the untargeted attack. As the margin increases, the ASR curves of the three attacks simultaneously rise until they become flat near $c = 1$. With a larger margin, SUETA+ surpasses PGD and SUETA in the case of the untargeted attack while exhibiting comparable performance in the case of the targeted attack. The results indicate that increasing the margin factor significantly enhances the transferability of adversarial examples. Besides, SUETA+ outperforms PGD and SUETA in the case of the untargeted attack regardless of the margin factor.

Table 5
ASR of attacks on TIMIT dataset.

Attack	Momentum	Goal	Target model			Avg
			A1	C1	E1	
PGD	0	Untargeted	21.43	33.02	19.68	24.71
	1		36.83	50.96	42.54	43.44
PGD-ME	0		31.75	48.57	34.29	38.20
	1		45.56	56.19	49.52	50.42
SUETA	0		36.19	46.03	34.29	38.84
	1		43.18	51.59	41.27	45.35
SUETA-ME	0		44.29	52.22	45.4	47.30
	1		**49.68**	**56.67**	**52.22**	**52.86**
PGD	0	Targeted	0.37	3.57	1.48	1.81
	1		7.33	3.4	2.88	4.54
PGD-ME	0		6.03	4.76	3.17	4.65
	1		7.62	4.6	4.44	5.55
SUETA	0		6.8	5.54	3.61	5.32
	1		9.99	5.26	4.66	6.64
SUETA-ME	0		9.21	5.08	5.24	6.51
	1		**10.32**	**5.71**	**5.4**	**7.14**

5.4.3. The effect of the momentum factor

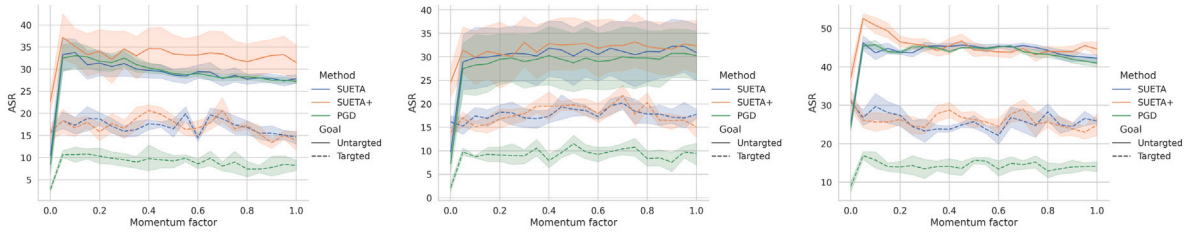
The momentum factor can also enhance the transferability of adversarial examples, which has been shown in the image tasks (Dong et al., 2018). We fix the margin factor as $c = 0$ in the loss functions of PGD, SUETA, and SUETA+. Attacks are conducted across three models: A1, C1, and E1. Fig. 4 presents the ASR curve in relation to the momentum factor β. In the case of the untargeted attack, the ASR curves of the three attacks rise steeply before reaching $\beta = 0.1$. In contrast, in the case of the targeted attack, the growth of ASR is smaller, and the curve exhibits more oscillations. However, when using E1 as the local surrogate model, the highest ASR of SUETA and SUETA+ is obtained at $\beta = 0$. The results indicate that momentum can also help enhance the transferability of adversarial examples in most cases, but a larger momentum factor does not always guarantee a higher ASR. Besides, SUETA and SUETA+ outperform PGD in the case of the targeted attack regardless of the momentum value.

Table 5 presents the summarized results of attacks on the TIMIT dataset, which are conducted in the conference version of this paper (Zuo et al., 2022). These existing experiments are conducted under the setting of $c = 0$ and $\epsilon = 0.001$, and the ASR is averaged on different local surrogate models. Additionally, we combine the attacks with the model ensemble method (Zuo et al., 2022), which are denoted with a suffix of '-ME' in the table.

The results indicate that momentum significantly improves the ASR for all the attacks, and the ASR of SUETA and SUETA-ME is higher than that of the PGD.

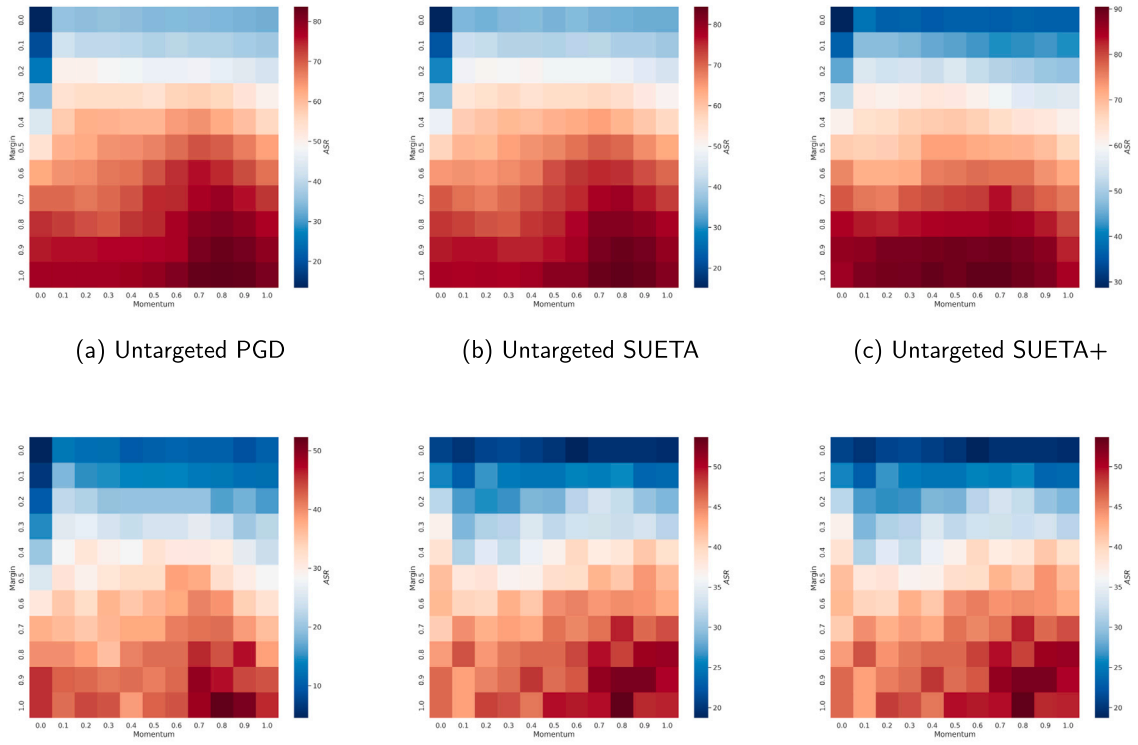
5.4.4. The collaborative effect of margin and momentum factors

Fig. 5 further shows the ASR heatmap of the attacks, where heat intensity corresponds to ASR values, the x-axis corresponds to the



(a) Transfer from A1 (b) Transfer from C1 (c) Transfer from E1

Fig. 4. ASR with varying momentum. The margin factor is fixed as $c = 0$, and the ASR is averaged on different target models.



(a) Untargeted PGD (b) Untargeted SUETA (c) Untargeted SUETA+

(d) Targeted PGD (e) Targeted SUETA (f) Targeted SUETA+

Fig. 5. ASR with varying momentum and margin.

momentum factor, and the y-axis corresponds to the margin factor. Optimal results are achieved when the margin is set to 1, and the momentum factor is set to near 0.8. The findings indicate that the effect of margin and momentum do not entirely overlap. The combination of both momentum and margin can lead to higher ASR.

5.4.5. The effect of ensemble number

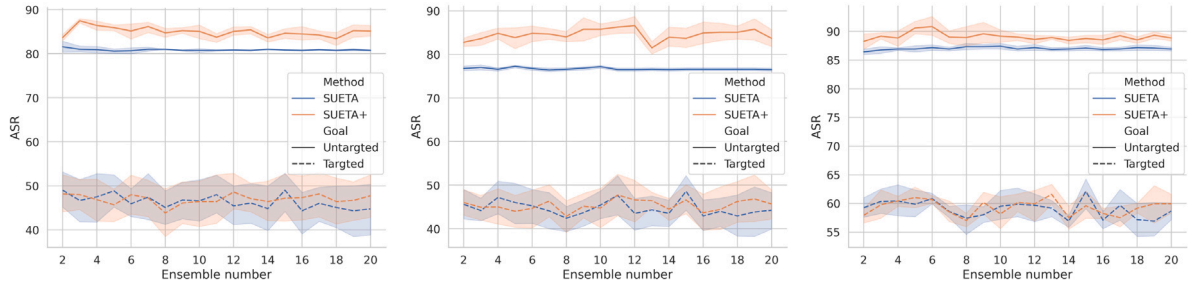
In practical scenarios, the attacker usually cannot ensure collecting a fixed number of utterances for each target speaker. Consequently, it is essential to evaluate the effect of the ensemble number, which refers to the size of the ensemble batch. We conduct attacks across three models, A1, C1, and E1 over three rounds. In each round, we randomly divided the utterances into different ensemble batches. Fig. 6 presents the ASR curve in relation to the ensemble number. The ASR curve oscillates as the ensemble number increases, indicating that the effectiveness of the attack is not strongly associated with the ensemble number with a value larger than one.

5.4.6. Attack with substitute enrolled utterances

In a practical scenario, an attacker might not have access to the exact enrolled utterances the target system retains. To examine transfer-based attacks under this situation, we assume that the attacker can only gather substitute enrolled utterances instead of the precise ones for the target system. We use 10% of the Spk_{10}-test as the substitue enrolled utterances and perform attacks on the remaining data. The experimental results are presented in Table 6. The ASR of all attacks is similar to the results in Table 3, aligning with the conclusions attacking with the exact enrolled utterances. The findings indicate that even if the attacker lacks knowledge about the exact enrolled utterances of the target SI systems, the utilization of substitute utterances remains effective.

5.5. Evaluation of the cross-dataset attack

In this part of the experiment, we maintain consistency in the model architecture and feature extraction preprocessor between the surrogate and target models, with the only variation being in the training dataset. We conduct attacks on three models trained on the LibriSpeech dataset: A1, C1, and E1, and three models trained on the larger VoxCeleb2 dataset: A4, C4, and E4. The results are presented in Table 7, where the notation M1–M2 refers to the transfer from the local surrogate model M1 to the target model M2. Although the ASR of the cross-dataset attacks is significantly lower than that of the cross-model attacks,



(a) Transfer from A1 (b) Transfer from C1 (c) Transfer from E1

Fig. 6. ASR with varying ensemble numbers. The ASR is averaged on different target models.

Table 6
ASR of the cross-model attack with substitute enrolled utterances.

Surrogate model	Attack	Goal	Target model					Avg
			A1	B1	C1	D1	E1	
A1	Naive-PGD	Untargeted	*	12.29	9.57	8.00	5.14	8.75
	Margin-PGD		*	90.43	71.14	77.14	69.57	77.07
	SUETA		*	93.57	74.71	80.43	76.00	81.18
	SUETA+		*	**96.14**	**90.43**	**91.43**	**89.57**	**91.89**
C1	Naive-PGD		13.57	6.43	*	6.71	5.14	7.96
	Margin-PGD		66.57	79.71	*	76.57	62.71	71.39
	SUETA		71.14	84.00	*	78.29	68.57	75.50
	SUETA+		**81.71**	**89.43**	*	**89.57**	**83.14**	**85.96**
E1	Naive-PGD		17.71	14.57	12.71	12.29	*	14.32
	Margin-PGD		81.29	86.43	78.29	83.71	*	82.43
	SUETA		85.57	89.71	81.43	87.57	*	86.07
	SUETA+		**89.43**	**91.43**	**88.29**	**90.57**	*	**89.93**
A1	Naive-PGD	Targeted	*	4.71	1.43	2.29	0.71	2.29
	Margin-PGD		*	57.71	27.14	43.14	32.71	40.18
	SUETA		*	62.14	**32.86**	47.71	**40.14**	**45.71**
	SUETA+		*	**62.57**	**32.86**	**49.14**	38.00	45.64
C1	Naive-PGD		4.14	2.14	*	2.71	2.29	2.82
	Margin-PGD		32.86	43.00	*	44.86	34.00	38.68
	SUETA		33.29	42.00	*	45.71	**35.71**	39.18
	SUETA+		**35.43**	**45.29**	*	**46.86**	**35.71**	**40.82**
E1	Naive-PGD		6.00	5.29	4.57	5.14	*	5.25
	Margin-PGD		51.00	63.57	46.71	59.43	*	55.18
	SUETA		48.86	**68.57**	50.00	**64.43**	*	57.96
	SUETA+		**52.57**	66.43	**52.29**	62.43	*	**58.43**

Table 7
ASR of the cross-dataset attack.

Attack	Goal	A1–A4	A4–A1	C1–C4	C4–C1	E1–E4	E4–E1	Avg
Naive-PGD	Untargeted	0.00	5.41	0.00	3.25	0.00	0.75	1.57
Margin-PGD		0.23	17.62	0.15	**20.79**	0.12	12.23	8.52
SUETA		0.38	16.45	0.30	19.54	0.25	12.08	8.17
SUETA+		**0.59**	**18.46**	**0.34**	19.95	**1.09**	**12.67**	**8.85**
Naive-PGD	Targeted	0.00	0.88	0.00	0.63	0.00	0.13	0.27
Margin-PGD		**0.08**	2.83	**0.08**	2.84	0.08	2.39	1.38
SUETA		0.00	2.17	0.00	2.67	0.09	2.50	1.24
SUETA+		0.00	**3.04**	0.00	**3.25**	**0.17**	**3.17**	**1.61**

SUETA+ achieves the highest ASR in most cases. Higher ASR can be observed when transferring the adversarial examples from surrogate models A4, C4, and E4, which are trained on the larger dataset. The experimental results also highlight the challenges of transfer-based attacks in the cross-dataset situation.

5.6. Evaluation of the cross-preprocessor attack

In this part of the experiment, we maintain consistency in the model architecture and training dataset between the surrogate and target models, with the only variation being in the feature extraction preprocessor. Attacks are conducted across three models: A1, A2, and A3. The experimental results are presented in Table 8. SUETA and SUETA+ outperform Naive-PGD and Margin-PGD on the average ASR for each surrogate model, and SUETA+ achieves the highest average ASR in most cases. Notice when the difference in the preprocessors of surrogate and target models is only in the normalization scale (e.g., A1–A2 and A2–A1), the ASR of the SUETA+ is higher than 95% for untargeted attack and higher than 60% for targeted attack. However, when the utilization of Kaldi-style SPO is different in the preprocessors (e.g., A1–A3, A2–A3, A3–A1, and A3–A2), a significant decline appears in the ASR. The experimental results also highlight the challenges of transfer-based attacks in the cross-preprocessor situation, especially for the difference in the utilization of Kaldi-style SPO.

Table 8
ASR of the cross-preprocessor attack.

Surrogate model	Attack	Goal	Target model			Avg
			A1	A2	A3	
A1	Naive-PGD	Untargeted	*	21.25	2.00	11.63
	Margin-PGD		*	92.54	6.55	49.55
	SUETA		*	93.96	7.04	50.50
	SUETA+		*	**95.71**	**9.46**	**52.58**
A2	Naive-PGD		22.75	*	1.13	11.94
	Margin-PGD		90.22	*	4.52	47.37
	SUETA		92.84	*	4.42	48.63
	SUETA+		**96.54**	*	**7.58**	**52.06**
A3	Naive-PGD		3.75	3.48	*	3.62
	Margin-PGD		23.38	21.53	*	22.45
	SUETA		23.59	22.38	*	22.98
	SUETA+		**29.92**	**30.88**	*	**30.40**
A1	Naive-PGD	Targeted	*	6.88	0.25	3.57
	Margin-PGD		*	56.98	**1.36**	29.17
	SUETA		*	60.21	1.13	30.67
	SUETA+		*	**60.88**	1.21	**31.04**
A2	Naive-PGD		6.13	*	0.38	3.26
	Margin-PGD		59.48	*	0.98	30.23
	SUETA		**61.00**	*	0.92	**30.96**
	SUETA+		60.50	*	**1.25**	30.88
A3	Naive-PGD		0.50	0.63	*	0.57
	Margin-PGD		4.94	4.56	*	4.75
	SUETA		4.79	5.33	*	5.06
	SUETA+		**5.50**	**5.50**	*	**5.50**

Table 9
ASR of the cross-model attack with speech compression codecs.

Attack	Goal	Codec										Avg
		None	G.711a	G.711a-16k	G.711u	G.711u-16k	G.726	G.726-16k	GSM	MP3	AAC	
Naive-PGD	Untargeted	23.25	45.75	20.38	45.97	21.00	45.75	20.38	49.06	46.47	19.81	33.78
Margin-PGD		89.13	71.59	83.09	71.56	82.94	71.59	83.09	67.47	73.53	84.47	77.85
SUETA		91.18	72.91	84.72	73.28	84.34	72.91	84.72	68.34	74.38	87.66	79.44
SUETA+		**93.30**	**77.09**	**90.47**	**77.22**	**90.22**	**77.09**	**90.47**	**69.53**	**79.88**	**92.16**	**83.74**
Naive-PGD	Targeted	9.41	7.91	6.07	8.38	6.31	7.91	6.06	7.66	10.47	5.69	7.59
Margin-PGD		62.14	16.84	50.59	16.41	50.77	16.84	50.59	**13.61**	19.04	53.63	35.05
SUETA		62.90	16.44	47.22	16.31	46.75	16.44	47.22	12.34	18.41	53.13	33.71
SUETA+		**64.01**	**17.56**	**53.75**	**17.63**	**53.75**	**17.56**	**53.75**	12.78	**19.22**	**55.25**	**36.53**

5.7. Attack with speech compression codecs

Speech compression codecs are used to reduce the size of audio files transmitted over the telephone or the Internet. In this part of the experiment, we evaluate the attack performance of adversarially perturbed utterances that are processed through an encode-decode speech compression codec, including G.711a, G.711u, G.726, GSM, MP3, and AAC. The G.711a, G.711u, G.726, and GSM codecs are commonly employed in telephony transformation or Voice over Internet Protocol (VoIP). These codecs compress the utterances with a sample rate of 8 kHz and a bit depth of 8 bits. Besides, we also compress the utterances by employing algorithms of G.711a, G.711u, and G.726 with a sample rate of 16 kHz and a bit depth of 8 bits, designated as G.711a-16k, G.711u-16k, and G.726k-16k respectively. MP3 and AAC codecs are commonly used on mobile devices and streaming services. We employ MP3 and AAC with a sample rate of 16 kHz and a bit depth of 16 bits, which is the same as the uncompressed utterance.

We conduct attacks with substitute enrolled utterances and employ E1 as the surrogate model to craft transferable adversarial examples as it is the most effective surrogate model according to results in Table 6. The adversarial examples are encoded and decoded with speech compression codecs before being sent to the target models, A1, B1, C1, and D1. The experimental results are shown in Table 9, where the ASR for each codec is averaged on different target models. The codecs reduce the ASR for all the attacks, especially for G.711a, G.711u, G.726u, and GSM which compress the sample rate and bit depth into 8 kHz and 8 bits. Nevertheless, SUETA+ still achieves the highest ASR in almost all cases, demonstrating a notable improvement in ASR particularly in untargeted attacks.

5.8. Attack commercial APIs

We attack two commercial SI systems that can be accessed by APIs: iFlytek[7] and TalentedSoft.[8] The feature extraction preprocessors, model architecture, and training datasets of commercial SI systems are completely unknown. We use E1 with substitute enrolled utterances as the local surrogate model and craft transferable adversarial examples. The ASR is shown in Table 10. SUETA and SUETA+ achieve higher ASR in the untargeted attack compared to the baselines. In particular, SUETA+ significantly improves the average ASR by 39.08%, 24.09%, and 18.18% compared to Naive-PGD, Margin-PGD, and SUETA, respectively. For the targeted attack, the attack performance of different attacks becomes closer while Margin-PGD achieves the highest ASR.

5.9. Attack voice assistant

In this part of the experiment, we stretch the attacks from digital lines to the over-the-air situation. We attack the voice assistant Tmall

[7] https://www.iflytek.com/en/index.html
[8] http://www.talentedsoft.com/en/

Table 10
ASR of attacks against commercial APIs.

Attack	Goal	APIs		Avg
		iFlytek	TalentedSoft	
Naive-PGD	Untargeted	10.39	7.72	9.06
Margin-PGD		24.37	23.72	24.05
SUETA		26.05	32.81	29.43
SUETA+		**42.42**	**53.86**	**48.14**
Naive-PGD	Targeted	0.86	0.00	0.43
Margin-PGD		**5.14**	**5.47**	**5.31**
SUETA		4.62	4.52	4.57
SUETA+		5.09	4.43	4.76

Genie,[9] which supports the speaker identification task. The identification mechanism of the voice assistant relies on text-dependent utterances, i.e., we have to ask the Tmall Genie with "TMall Genie, who am I" (in Chinese). Therefore, we engage six volunteers, comprising four males and two females, to utter the specific phrases. We ask volunteers to utter "TMall Genie" three times to register on the Tmall Genie and another three times to obtain the substitute enrolled utterances. We ask volunteers to utter "TMall Genie, who am I?" 10 times per speaker to construct the test set. To construct the ensemble utterance batch for SUETA and SUETA+, we also ask the volunteers to utter 10 different phrases (in Chinese) listed as follows:

- Tmall Genie, play a cheerful song
- Tmall Genie, tell me today's weather forecast
- Tmall Genie, turn on the living room light
- Tmall Genie, lower the air conditioning temperature to 22 degrees
- Tmall Genie, check the high-speed train schedule from Beijing to Shanghai
- Tmall Genie, remind me of the meeting tomorrow morning
- Tmall Genie, play the latest news summary
- Tmall Genie, tell me a bedtime story
- Tmall Genie, check my schedule
- Tmall Genie, help me order a bottle of mineral water

We generate adversarial examples on a Resnet34 model (Wang et al., 2023) pretrained on the Chinese language dataset CNCeleb (Li et al., 2022) since the Tmall Genie only supports the Chinese language. We play the adversarially perturbed utterance by Apple iPhone 15pro smartphone to attack the Tmall Genie Cube Sugar 3 Smart Speaker in a meeting room. The distance between the devices is set to 0.5 m. The experiments are repeated for three rounds. In each round, we regenerate adversarial examples for each utterance in the test set with a different random seed.

For the TMall Genie, the responses to "TMall Genie, who am I" are categorized into three types:

- Identified with high confidence, e.g., "Hello, Sam, happy to serve you".
- Identified with medium confidence, e.g., "I think you are Sam, am I right?"
- Not identified, e.g., "I do not know your name".

A successful untargeted attack occurs when the perturbed utterance is identified as a wrong speaker with high or medium confidence or the perturbed utterance is not identified. Conversely, a successful targeted attack occurs when the perturbed utterance is identified as the target speaker with high or medium confidence.

The results are presented in Table 11, where the ASR-ut refers to the ASR of the untargeted attack, and ASR-t refers to the ASR of the targeted attack. Attacking the voice assistant over the air is the most challenging situation studied in this paper, where the ASR of all attacks further declines compared to attacks on commercial APIs. Nevertheless, SUETA+ still outperforms the baselines. Especially, the improvement in the untargeted attack is more significant compared to the targeted attacks. Besides, we also identified two limitations when conducting the over-the-air attack:

Table 11
ASR of attacks against voice assistant Tmall Genie.

Attack	ASR-ut	ASR-t
Naive-PGD	11.67	1.11
Margin-PGD	23.89	5.56
SUETA	21.11	6.11
SUETA+	**27.78**	**7.22**

- The difficulty of targeting different speakers varies, and successful attacks are concentrated on a few specific target speakers.
- Inter-gender attacks are more difficult compared to intra-gender attacks. Successful attacks only exist in cases where the gender of the target speaker is the same as the original speaker of utterance.

These limitations present challenges for attacking SI systems in the more complicated over-the-air situation compared to the digital line situation.

6. Discussion

The experimental findings also inspire possible directions for future research:

- The controlled variable experiments demonstrate that variations in the training dataset, feature extraction preprocessor, and speech compression codecs exert a more notable influence on the transferability of adversarial attacks than the model architecture. From the attacker's view, reducing the gap induced by training datasets, feature extraction preprocessors, and speech compression codecs could lead to stronger attacks, e.g., ensemble attacks over these different aspects. From the defender's view, increasing the difference between the SI system and potential surrogate models in these aspects could possibly lead to a more robust system.
- The method designed for the digital-line attacks has limited performance in the over-the-air situation. The transition process of audio signals suffers from information degradation and interference with environmental noise. These phenomena have to be considered when designing attacks in the over-the-air situation.

7. Conclusion

In this paper, we propose a novel transfer-based black-box attack method called SUETA. To the best of our knowledge, SUETA is the first transfer-based black-box attack method that utilizes multiple utterances instead of a single one. In SUETA, a speaker-specific utterance ensemble loss function is proposed, which enhances the transferability of adversarial examples compared with speaker-unrelated baselines. An improved variant of SUETA, called SUETA+, is further proposed by sharing gradients of utterances at the speaker-embedding level. Empirical results show that both SUETA and SUETA+ effectively improve the attack success rate (ASR) compared to the PGD attacks under the classical cross-model situation. SUETA+ also outperforms the baselines in cross-dataset and cross-preprocessor situations, although the ASR for all transfer-based attacks decreases compared to that in the cross-model situation. Moreover, SUETA+ can significantly improve the ASR against commercial APIs (iFlytek and TalentedSoft) and the voice assistant (Tmall Genie) for the untargeted attack compared to the baselines.

[9] https://thearf-org-unified-admin.s3.amazonaws.com/MSI/2021/04/MSI_Report_21-114.pdf

CRediT authorship contribution statement

Chu-Xiao Zuo: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jia-Yi Leng:** Validation, Software, Data curation. **Wu-Jun Li:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used ChatGPT3.5 in order to improve readability and language. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Acknowledgments

This work is supported by NSFC, China Project (No. 62192783).

References

Andriushchenko, M., Croce, F., Flammarion, N., Hein, M., 2020. Square attack: A query-efficient black-box adversarial attack via random search. In: ECCV, vol. 12368, pp. 484–501.
Brendel, W., Rauber, J., Bethge, M., 2018. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In: ICLR.
Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I.J., Madry, A., Kurakin, A., 2019. On evaluating adversarial robustness. CoRR arXiv:1902.06705.
Carlini, N., Wagner, D.A., 2017. Towards evaluating the robustness of neural networks. In: S&P. pp. 39–57.
Chen, G., Chen, S., Fan, L., Du, X., Zhao, Z., Song, F., Liu, Y., 2021. Who is real bob? Adversarial attacks on speaker recognition systems. In: S&P. pp. 694–711.
Chen, J., Jordan, M.I., Wainwright, M.J., 2020. HopSkipJumpAttack: A query-efficient decision-based attack. In: S&P. pp. 1277–1294.
Chen, P., Zhang, H., Sharma, Y., Yi, J., Hsieh, C., 2017. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In: AISec@CCS. pp. 15–26.
Chen, G., Zhao, Z., Song, F., Chen, S., Fan, L., Wang, F., Wang, J., 2022. Towards understanding and mitigating audio adversarial examples for speaker recognition. TDSC 1–17.
Croce, F., Hein, M., Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: ICML, vol. 119, pp. 2206–2216.
Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P., 2011. Front-end factor analysis for speaker verification. TASLP 19 (4), 788–798.
Deng, J., Dong, L., Wang, R., Yang, R., Yan, D., 2022. Decision-based attack to speaker recognition system via local low-frequency perturbation. IEEE Signal Process. Lett. 29, 1432–1436.
Desplanques, B., Thienpondt, J., Demuynck, K., 2020. ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In: INTERSPEECH. pp. 3830–3834.
Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J., 2018. Boosting adversarial attacks with momentum. In: CVPR. pp. 9185–9193.
Dong, Y., Pang, T., Su, H., Zhu, J., 2019. Evading defenses to transferable adversarial examples by translation-invariant attacks. In: CVPR. pp. 4312–4321.
Garofolo, J.S., 1993. Timit acoustic phonetic continuous speech corpus. Linguist. Data Consort..
Gong, Y., Poellabauer, C., 2017. Crafting adversarial examples for speech paralinguistics applications. CoRR arXiv:1711.03280.
Goodfellow, I.J., Shlens, J., Szegedy, C., 2015. Explaining and harnessing adversarial examples. In: ICLR.
Hammi, B., Zeadally, S., Khatoun, R., Nebhen, J., 2022. Survey on smart homes: Vulnerabilities, risks, and countermeasures. Comput. Secur. 117, 102677.
He, X., Li, Y., Qu, H., Dong, J., 2023. Improving transferable adversarial attack via feature-momentum. Comput. Secur. 128, 103135.
Hermansky, H., 1990. Perceptual linear predictive (PLP) analysis of speech. J. Acoust. Soc. Am. 87 (4), 1738–1752.
Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al., 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Process. Mag. 29 (6), 82–97.
Huang, H., Chen, Z., Chen, H., Wang, Y., Zhang, K., 2023. T-SEA: Transfer-based self-ensemble attack on object detection. In: CVPR. pp. 20514–20523.
Jahangir, R., Teh, Y.W., Nweke, H.F., Mujtaba, G., Al-Garadi, M.A., Ali, I., 2021. Speaker identification through artificial intelligence techniques: A comprehensive review and research challenges. Expert Syst. Appl. 171, 114591.
Ko, K., Kim, S., Kwon, H., 2023. Multi-targeted audio adversarial example for use against speech recognition systems. Comput. Secur. 128, 103168.
Kreuk, F., Adi, Y., Cissé, M., Keshet, J., 2018. Fooling end-to-end speaker verification with adversarial examples. In: ICASSP. pp. 1962–1966.
Kurakin, A., Goodfellow, I.J., Bengio, S., 2017. Adversarial machine learning at scale. In: ICLR.
Li, J., Chen, C., Azghadi, M.R., Ghodosi, H., Pan, L., Zhang, J., 2023. Security and privacy problems in voice assistant applications: A survey. Comput. Secur. 134, 103448.
Li, L., Liu, R., Kang, J., Fan, Y., Cui, H., Cai, Y., Vipperla, R., Zheng, T.F., Wang, D., 2022. CN-Celeb: Multi-genre speaker recognition. Speech Commun. 137, 77–91.
Li, H., Shan, S., Wenger, E., Zhang, J., Zheng, H., Zhao, B.Y., 2020a. Blacklight: Defending black-box adversarial attacks on deep neural networks. CoRR arXiv:2006.14042.
Li, J., Zhang, X., Xu, J., Zhang, L., Wang, Y., Ma, S., Gao, W., 2020b. Learning to fool the speaker recognition. In: ICASSP. pp. 2937–2941.
Li, X., Zhong, J., Wu, X., Yu, J., Liu, X., Meng, H., 2020c. Adversarial attacks on GMM I-Vector based speaker verification systems. In: ICASSP. pp. 6579–6583.
Liu, Y., Chen, X., Liu, C., Song, D., 2017. Delving into transferable adversarial examples and black-box attacks. In: ICLR.
Long, T., Gao, Q., Xu, L., Zhou, Z., 2022a. A survey on adversarial attacks in computer vision: Taxonomy, visualization and future directions. Comput. Secur. 121, 102847.
Long, Y., Zhang, Q., Zeng, B., Gao, L., Liu, X., Zhang, J., Song, J., 2022b. Frequency domain model augmentation for adversarial attack. In: ECCV, vol. 13664, pp. 549–566.
Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A., 2018. Towards deep learning models resistant to adversarial attacks. In: ICLR.
Muda, L., Begam, M., Elamvazuthi, I., 2010. Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. CoRR arXiv:1003.4083.
Nagrani, A., Chung, J.S., Xie, W., Zisserman, A., 2020. VoxCeleb: Large-scale speaker verification in the wild. Comput. Speech Lang. 60.
Panayotov, V., Chen, G., Povey, D., Khudanpur, S., 2015. LibriSpeech: An ASR corpus based on public domain audio books. In: ICASSP. pp. 5206–5210.
Pardede, H.F., Zilvan, V., Krisnandi, D., Heryana, A., Kusumo, R.B.S., 2019. Generalized filter-bank features for robust speech recognition against reverberation. In: IC3INA. pp. 19–24.
Polyak, B.T., 1964. Some methods of speeding up the convergence of iteration methods. Comput. Math. Math. Phys. 4, 1–17.
Prince, S.J.D., Elder, J.H., 2007. Probabilistic linear discriminant analysis for inferences about identity. In: ICCV. pp. 1–8.
Shamsabadi, A.S., Teixeira, F.S., Abad, A., Raj, B., Cavallaro, A., Trancoso, I., 2021. FoolHD: Fooling speaker identification by highly imperceptible adversarial disturbances. In: ICASSP. IEEE, pp. 6159–6163.
Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S., 2018. X-Vectors: Robust DNN embeddings for speaker recognition. In: ICASSP. pp. 5329–5333.
Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R., 2014. Intriguing properties of neural networks. In: ICLR.
Thian, N.P.H., Sanderson, C., Bengio, S., 2004. Spectral subband centroids as complementary features for speaker authentication. In: ICBA, vol. 3072, pp. 631–639.
Villalba, J., Zhang, Y., Dehak, N., 2020. x-Vectors meet adversarial attacks: Benchmarking adversarial robustness in speaker verification. In: INTERSPEECH. pp. 4233–4237.
Wang, Q., Guo, P., Xie, L., 2020. Inaudible adversarial perturbations for targeted attack in speaker recognition. In: INTERSPEECH. pp. 4228–4232.
Wang, H., Liang, C., Wang, S., Chen, Z., Zhang, B., Xiang, X., Deng, Y., Qian, Y., 2023. Wespeaker: A research and production oriented speaker embedding learning toolkit. In: ICASSP. pp. 1–5.
Xie, Y., Li, Z., Shi, C., Liu, J., Chen, Y., Yuan, B., 2021. Real-time, robust and adaptive universal adversarial attacks against speaker recognition systems. J. Signal Process. Syst. 93, 1187–1200.
Xie, C., Zhang, Z., Zhou, Y., Bai, S., Wang, J., Ren, Z., Yuille, A.L., 2019. Improving transferability of adversarial examples with input diversity. In: CVPR. pp. 2730–2739.
Yu, Z., Chang, Y., Zhang, N., Xiao, C., 2023. SMACK: Semantically meaningful adversarial audio attack. In: USENIX Security.
Yu, Y.-Q., Fan, L., Li, W.-J., 2019. Ensemble additive margin softmax for speaker verification. In: ICASSP. pp. 6046–6050.
Yu, Y.-Q., Li, W.-J., 2020. Densely connected time delay neural network for speaker verification. In: INTERSPEECH. pp. 921–925.

Yu, Y., Zheng, S., Suo, H., Lei, Y., Li, W., 2021. CAM: Context-aware masking for robust speaker verification. In: ICASSP. pp. 6703–6707.
Zhang, Y., Jiang, Z., Villalba, J., Dehak, N., 2020. Black-box attacks on spoofing countermeasures using transferability of adversarial examples. In: INTERSPEECH. pp. 4238–4242.
Zhang, W., Zhao, S., Liu, L., Li, J., Cheng, X., Zheng, T.F., Hu, X., 2021. Attack on practical speaker verification system using universal adversarial perturbations. In: ICASSP. pp. 2575–2579.
Zou, J., Pan, Z., Qiu, J., Liu, X., Rui, T., Li, W., 2020. Improving the transferability of adversarial examples with resized-diverse-inputs, diversity-ensemble and region fitting. In: ECCV, vol. 12367, pp. 563–579.
Zuo, C.-X., Leng, J.-Y., Li, W.-J., 2022. Speaker-specific utterance ensemble based transfer attack on speaker identification. In: INTERSPEECH. pp. 3203–3207.

Chu-Xiao Zuo received the BSc degree in statistics from Nanjing University, China, in 2017. He is currently working toward the PhD degree in the Department of Computer Science and Technology, Nanjing University. His research interests are in the security and privacy of machine learning and multimedia.

Jia-Yi Leng received the BSc degree in computer science from Nanjing University, China, in 2020. He received the MSc degree in computer science from Nanjing University, China, in 2023. His research interests are in machine learning and multimedia.

Wu-Jun Li received the BSc and MEng degrees in computer science from the Nanjing University of China, and the PhD degree in computer science from the Hong Kong University of Science and Technology. He started his academic career as an assistant professor in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. He then joined Nanjing University, where he is currently a professor in the Department of Computer Science and Technology. His research interests are in machine learning, big data, and artificial intelligence. He is a member of IEEE.