Can I Hear Your Face? Pervasive Attack on Voice Authentication Systems with a Single Face Image

Nan Jiang, Bangjie Sun, and Terence Sim, *National University of Singapore;*
Jun Han, *KAIST*

https://www.usenix.org/conference/usenixsecurity24/presentation/jiang-nan

This paper is included in the Proceedings of the 33rd USENIX Security Symposium.

August 14–16, 2024 • Philadelphia, PA, USA

978-1-939133-44-1

Open access to the Proceedings of the 33rd USENIX Security Symposium is sponsored by USENIX.

Can I Hear Your Face? Pervasive Attack on Voice Authentication Systems with a Single Face Image

Nan Jiang[†], Bangjie Sun[†], Terence Sim[†], and Jun Han[‡]

[†]*National University of Singapore,* [‡]*KAIST*

Abstract

We present *Foice*, a novel deepfake attack against voice authentication systems. *Foice* generates a synthetic voice of the victim from *just a single image* of the victim's face, *without requiring any voice sample*. This synthetic voice is realistic enough to fool commercial authentication systems. Since face images are generally easier to obtain than voice samples, *Foice* effectively makes it easier for an attacker to mount large-scale attacks. The key idea lies in learning the partial correlation between face and voice features, and adding to that a face-independent voice feature sampled from a Gaussian distribution. We demonstrate the effectiveness of *Foice* with a comprehensive set of real-world experiments involving ten offline participants and an online dataset of 1,029 unique individuals. By evaluating eight state-of-the-art systems, including WeChat's Voiceprint and Microsoft Azure, we show that all these systems are vulnerable to *Foice* attack.

1 Introduction

Voice authentication is being adopted more widely as an alternative to traditional password-based security measures for social media platforms [13], telebanking [6], and personal smart devices [2, 9, 10]. However, recent advancements in voice deepfake techniques have enabled the generation of synthetic voices that are sufficiently convincing to bypass even the most advanced voice authentication systems. For example, recent deepfake attacks have reportedly compromised the voice authentication systems of platforms such as WeChat, Microsoft Azure's voice authentication API, and Amazon Alexa [16, 57].

The state-of-the-art voice deepfake attacks, however, typically necessitate the attackers to obtain voice recordings of the victim. This assumption inherently limits and confines the pervasiveness of these attacks. As a result, targets are often limited to celebrities whose voice samples are widely available online. However, there have been minimal efforts to enhance the pervasiveness of the voice deepfake attacks to a wider range of victims.

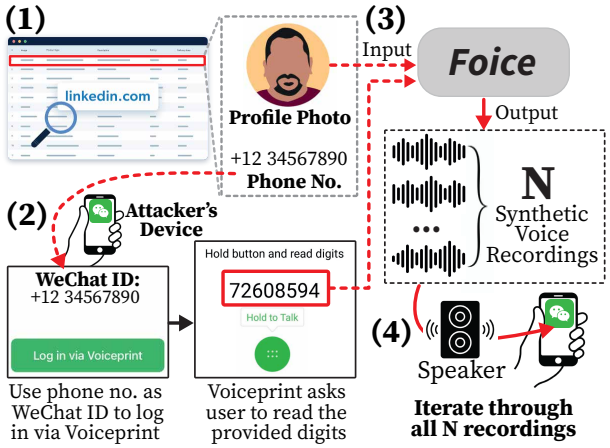


Figure 1: Figure depicts an example attack scenario of *Foice*. The attacker (1) obtains the victim's photo along with the corresponding phone number (often used as WeChat ID) from online social media such as LinkedIn; (2) attempts to log in to victim's WeChat via Voiceprint (voice authentication) system on the attacker's device; (3) inputs the victim's photo and WeChat's provided digits to *Foice* to generate N synthetic voice recordings where the speaker is speaking the provided digits; and (4) manages to successfully authenticate and log in as the victim on WeChat by sequentially playing each of the N synthetic recordings through an external speaker.

In light of this, we pose the following question: *is it possible to design a more pervasive voice deepfake attack that eliminates the need for direct access to the victim's voice recordings, thus increasing the attack scalability by broadening the target to a wider range of victims?* To address this, we propose *Foice*, a novel voice deepfake attack that can synthesize a victim's voice leveraging ***only a single photograph (i.e., face image) without requiring voice recordings*** of the individual. Such images are easily accessible, especially with the proliferation of social media platforms such as Instagram, Facebook, and LinkedIn. The feasibility of *Foice* stems from

the inherent biological correlation between facial attributes (e.g., gender, age, facial bone structure, mouth, and lips) and voice features (e.g., volume, pitch, timbre, pace, and vocal range). This correlation becomes clear when we intuitively gauge an individual's voice from their physical appearance – for instance, predicting a deeper tone from a more robust individual while expecting a higher pitch from someone with a slender build. Figure 1 illustrates how *Foice* launches an attack on WeChat's Voiceprint, namely its voice authentication system, by leveraging the biological correlation. The attacker (1) obtains the victim's photo along with the corresponding phone number (often used as WeChat ID) from online social media such as LinkedIn; (2) attempts to log in to the victim's WeChat via the Voiceprint (voice authentication) system on the attacker's device; (3) uses the victim's photo and WeChat's provided digits to generate N synthetic voice recordings where the victim is speaking the provided digits; and (4) manages to successfully authenticate and log in as the victim on WeChat by sequentially playing each of the N synthetic recordings through a speaker.

Designing *Foice* presents significant challenges, primarily due to the intricate relationships between facial and voice features. Specifically, facial features influence only a portion of voice features (e.g., pitch), while the remaining (e.g., timbre) are influenced by internal physiological structures, like the vocal cords, which are not discernible from facial appearance (see §2.3). To bridge this gap, we design a generative deep-learning model to generate a search space containing all the potential voice features that cannot be derived from the face image. To launch the attack on the victim, *Foice* extracts a portion of voice features from the victim's face image (i.e., *face-dependent* features). The attacker then takes N samples of *face-independent* features from the search space and combines them with the *face-dependent* features to obtain N synthetic voice recordings. By increasing the number N, the attacker has a higher chance of obtaining a synthetic recording that sounds similar to the victim, hence successfully compromising the voice authentication systems.

We evaluate *Foice* to demonstrate its feasibility through comprehensive real-world experiments. We evaluate *Foice*'s attack performance on authentication systems of commodity products, including WeChat's Voiceprint as well as smart assistants such as Siri [9], Google Assistant [10], and Bixby [2], by inviting ten participants. We also evaluate *Foice* on the state-of-the-art cloud services, including Microsoft Azure [8], iFlyteck [7], VGGVox [43], and DeepSpeaker [35], with a public dataset of 1,029 unique individuals. Our results demonstrate that **all** the tested authentication systems and voice assistants are vulnerable to *Foice* attack. In particular, *Foice* successfully bypasses all ten invited participants' WeChat Voiceprint system. On average, about 30% of the synthetic recordings for each participant are successful in the attack. In addition to demonstrating the effectiveness of *Foice* attack, we quantify how a face image contributes to the success of *Foice* attack. Finally, we evaluate the feasibility of augmenting the conventional voice deepfake attack with *Foice*. Our results demonstrate that *Foice* significantly improves the attack's performance by threefold.

These findings highlight the urgent need for heightened awareness to safeguard against novel voice deepfake attacks, such as *Foice*. Through this study, we also hint at a new avenue of information leakage, namely how a person's face potentially reveals how they sound. We hope that this paper would stimulate further exploration within the security community regarding the threats posed by the correlation of biometric information.

2 Background

We introduce the background of voice deepfake, voice authentication systems, and the correlation between a speaker's face and voice.

2.1 Voice Deepfakes

The state-of-the-art voice deepfake utilizes a short voice recording of the victim to synthesize another recording with a similar vocal style but with new content. A typical example of a voice deepfake circulating on social media is to manipulate the content of the U.S. president's speech while keeping a similar vocal style. Figure 2 depicts an overview of how the state-of-the-art voice deepfake operates. Specifically, the voice deepfake produces a new voice recording in two steps.
(1) Voice Feature Extraction. Given a voice recording, the voice deepfake first extracts a *voice feature vector*, which captures and numerically represents the speaker's unique *voice features*, such as pitch, timbre, and pronunciation. Intuitively, when we listen to two different speakers saying the same sentence (e.g., "Hello, how are you?" as depicted in Figure 2(a)), we can easily differentiate between them due to their unique *voice features*. The voice deepfake utilizes a deep-learning model, namely the *Speaker Encoder* to extract the *voice feature vector*, which captures relevant *voice features* of the speaker that can distinguish him/her from other speakers, regardless of the speech content.
(2) Voice Synthesis. Subsequently, the voice deepfake synthesizes a new voice recording with the extracted *voice feature vector* and a text input (e.g., "Hey, Google!" as depicted in Figure 2(b)). The voice deepfake utilizes another deep-learning model, namely the *Synthesizer*, to perform this task. Specifically, the *Synthesizer* first generates audio signals whose waveform is primarily determined by the text input. It then modifies the audio signals based on the *voice features* described in the *voice feature vectors*.

Different from the state-of-the-art voice deepfake, *Foice* only utilizes a face image of the speaker to generate the *voice feature vectors* (see §4.3 and §4.4). We observe that both *Foice*

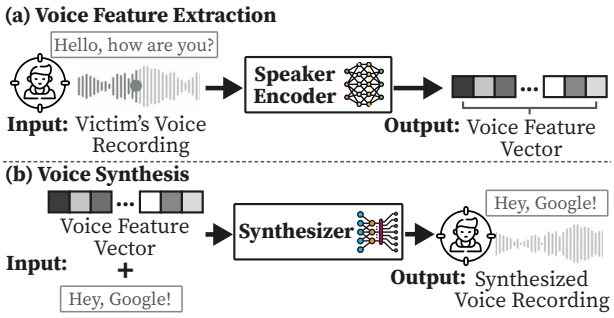


Figure 2: Figure depicts an overview of how voice deepfake operates. (a) depicts the *Voice Feature Extraction* step, where the *Speaker Encoder* extracts the *voice feature vector* from a voice recording. (b) depicts the *Voice Synthesis* step, where the *Synthesizer* uses the *voice feature vector* and a text input to synthesize a new voice recording.

and the voice deepfake yield similar vocal styles when using the face images and voice recordings, respectively (see §5.2).

2.2 Voice Authentication

Voice authentication is often incorporated into many online applications, such as social media platforms (e.g., WeChat), and telebanking, as an authentication method. Modern voice assistants, such as Siri, Google Assistant, and Bixby, also require voice authentication when they are activated by the user (e.g., "Hey, Google" when activating Google Assistant). In general, existing voice authentication solutions consist of two phases, namely the *Enrollment Phase* and the *Authentication Phase*. During the *Enrollment Phase*, the system provides several digits or sentences and enrolls the user's voice. In the *Authentication Phase*, the system records the user's voice again and compares the recorded voice with the enrolled voice to obtain a similarity score. The authentication is successful when the score is above a certain threshold.

However, we notice that the existing systems are vulnerable to voice deepfake attacks due to two reasons.

(1) Low Threshold. For the user's convenience, voice authentication systems usually configure a considerably low threshold of around 0.5 to 0.6 (e.g., Microsoft and iFlytek APIs) [7, 8]. This is to ensure that users can successfully authenticate themselves in noisy environments, such as restaurants and cafes, with background noises (e.g., speech from other people).

(2) Unlimited Number of Attempts. Many voice authentication systems do not restrict the number of authentication attempts for the user's convenience. For example, WeChat does not limit the number of *unsuccessful* login attempts. However, there is a limitation on the number of *successful* daily logins [13].

Foice exploits the aforementioned vulnerability to successfully bypass real-world voice authentication systems even

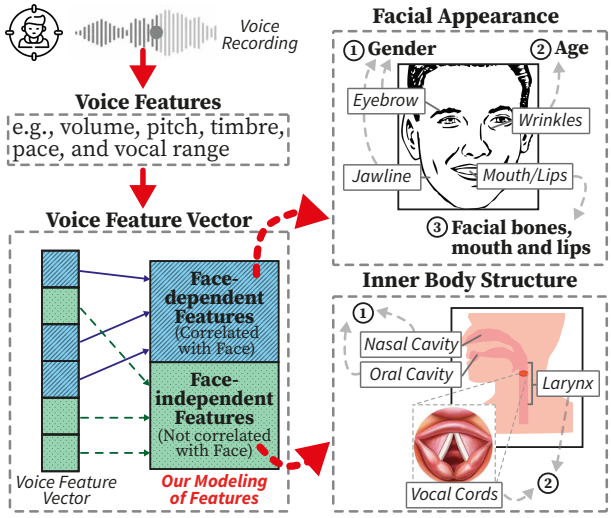


Figure 3: Figure depicts an example of the *voice feature vector* extracted from a voice recording, which captures the speaker's *voice features*. Based on our modeling, the *voice feature vector* consists of *face-dependent* and *face-independent* features, which are affected by the speaker's facial appearance or inner body structure, respectively.

with a single face image from the victim.

2.3 Correlation Between Face and Voice

Recall that existing voice deepfake utilizes the *voice feature vector* to capture and numerically represent *voice features*, including volume, pitch, timbre, pace, and vocal range, all of which constitute the uniqueness of the person's voice (see §2.1). Importantly, a subset of values in the *voice feature vector* is correlated with the person's ***facial appearance*** [47] (i.e., *face-dependent* features). The remaining feature values are largely affected by the person's ***inner body structure*** that produces the sound [49] (i.e., *face-independent* features). Figure 3 illustrates our modeling of *face-dependent* and *face-independent* features and their correlations with the speaker's facial appearance and inner body structure, respectively.

Facial Appearance. Facial appearance, including gender, age, the shape of the mouth, facial bone structure, and thin or full lips, may affect how the person's voice sounds (i.e., the *face-dependent* features) [47]. We present three examples as depicted in Figure 3 (*Facial Appearance*). ① Gender: Males typically have a more prominent eyebrow ridge and broad jawline compared to females while having a lower voice pitch. This is because hormone levels (i.e., the major factor determining gender) during puberty affect both face morphology and voice pitch [55]. ② Age: Age significantly affects a person's face and voice due to physiological and structural changes that occur over time. As people age, we can observe wrinkles and fine lines, especially around the eyes, forehead, and mouth. At

the same time, their *vocal cords* (i.e., a pair of muscular structures located in the voice box within the human throat) may become less flexible and thinner, resulting in a higher pitch in both men and women [42]. ③ Facial bones, mouth, and lips: Facial bone structure, the shape of the mouth, and thin or full lips may interfere with the sound wave propagating from the *vocal cords* to the mouth [23]. For example, people with thin lips may find it more challenging to articulate labial sounds (i.e., sounds like *'p', 'b', 'm', and 'w'* involving movement of lips). It is because they have less lip tissue to create the necessary closure or friction [15].

Inner Body Structure However, a significant portion of the *voice feature vector* is not correlated with the face (i.e., the *face-independent* features). Instead, it is largely affected by the inner body structure, such as the *nasal cavity*, *oral cavity*, and *larynx*. Figure 3 (*Inner Body Structure*) depicts two examples. ① The nasal cavity and oral cavity: The *nasal* and *oral cavities* are hollow spaces or passages in the head that extend from the nose or mouth, respectively, to the back of the throat. The shape and size of these cavities affect how the sound waves resonate and are modified, which, in turn, influences the timbre or quality of a person's voice [27]. ② The larynx: *The larynx*, commonly known as the voice box, plays a vital role in speech production. It is located in the neck. The *larynx* contains the *vocal cords*, which are two pairs of folds made up of muscle and tissue. The shape, size, and tension of the *vocal cords* significantly affect the timbre of the voice [51].

Key Takeaway. The *voice feature vector* describing a person's voice consists of the *face-dependent* (i.e., correlated with face) and *face-independent* (i.e., not correlated with face) features. Leveraging this observation, *Foice* first utilizes information in the face image to extract the *face-dependent* features (see §4.3). However, as the face image provides almost no insight into the inner body structure, it is challenging for *Foice* to obtain the *face-independent* features. To overcome this challenge, *Foice* designs a deep-learning model to generate a *search space* containing all the possible *face-independent* features subject to different inner body structures (see §4.4).

3 Threat Model

We highlight the attacker's goals and capabilities and further present *Foice*'s assumptions. The *goal* of the attacker is to compromise voice authentication in order to gain unauthorized access to the victim's accounts on platforms such as social media or gain access to personal voice assistants like Siri or Bixby. To achieve this goal, the attacker possesses the *capability* to acquire at least one facial image of the victim along with their credentials (such as email addresses, often used as account IDs). This information is extracted from publicly available data on online platforms such as Instagram, Facebook, and LinkedIn. Moreover, we assume that the attacker is capable of crawling through multiple online platforms to gather information from a broad cross-section of potential victims, making *Foice* a scalable attack. To execute the attack successfully, the attacker is also capable of collecting a large training dataset. This dataset includes a large collection of facial images of different speakers paired with their corresponding ground truth voice data (see §4.1).

4 System Design

We now present an overview of *Foice*'s design in §4.1, followed by details of each module from §4.2 to §4.5.

4.1 System Overview

Foice produces a set of synthetic voice recordings of the victim from a single face image leveraging the biological correlation between facial appearance and voice features (see §2.3). The attacker could iterate through all synthetic recordings to authenticate and log in as the victim to voice authentication systems. As depicted in Figure 4, *Foice* is divided into the *Training* and *Attack* phases.

❶Training Phase. The *Training Phase* is a one-time phase where the attacker utilizes online public datasets, containing face images and corresponding voice recordings, to train deep-learning models to learn the correlation. Specifically, *Foice* first processes the online public dataset in the *Data Processing* module (§4.2) to remove noise in face images, such as head orientations and background scenes, and extract ground truth voice feature vectors from voice recordings. Recall from §2.3 that the voice feature vector consists of *face-dependent* (i.e., correlated with face) and *face-independent* (i.e., not correlated with face) features. For clarity, we use F_{dep} and F_{indep} for these two kinds of features. In the *Face-dependent Voice Feature Extractor* module (§4.3), *Foice* utilizes processed face images and the ground truth voice feature vector to learn the relationship between F_{dep} and facial appearances. Subsequently, in the *Face-independent Voice Feature Generator* module (§4.4), *Foice* learns to identify F_{indep} by removing F_{dep} from the ground truth voice feature vector. It then generates a ***search space*** enumerating all possibilities of F_{indep}.

❷Attack Phase. Subsequently, in the *Attack Phase*, the attacker inputs one face image of the victim and takes N random samples from the search space to obtain ***N* synthetic voice recordings**, which are played iteratively to the authentication system until gaining access. Specifically, *Foice* first processes the face image in the *Face Processing* module (§4.2.1). *Foice* then utilizes the trained model in the *Face-dependent Voice Feature Extractor* module (§4.3) to extract F_{dep} of the victim from the processed face image. In the *Face-independent Voice Feature Generator* module (§4.4), *Foice* combines N samples from the search space with F_{dep} to reconstruct a set of N voice feature vectors of the victim. Finally, *Foice* converts N reconstructed vectors and a text input into N synthetic voice recordings, respectively, in the *Voice Synthesizer* module (§4.5).

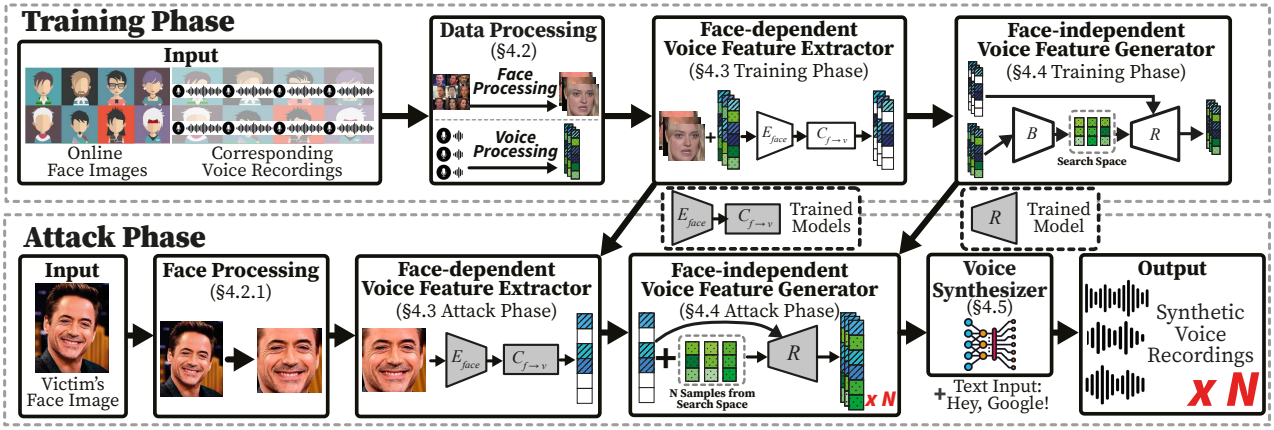


Figure 4: Figure depicts *Foice*'s system design. *Foice* is divided into *Training Phase* and *Attack Phase*. During the *Training Phase*, the attacker utilizes online public face images and corresponding ground truth voice recordings to train deep-learning models in the *Face-dependent Voice Feature Extractor* (§4.3) and *Face-independent Voice Feature Generator* (§4.4). During the *Attack Phase*, the attacker inputs the victim's face image to *Foice* to synthesize *N* number of voice recordings of the text that the attacker chooses (e.g., "Hey, Google!") in an attempt to bypass the victim's voice authentication or voice assistant systems (e.g., Google Assistant). The attacker iterates through the *N* synthetic voice recordings until gaining access.

4.2 Data Processing

Data processing module processes the online public dataset containing face images and corresponding voice recordings. It consists of two sub-modules. First, the *Face Processing* sub-module (§4.2.1) removes noise from the face image, such as the background scene and head orientation. Second, the *Voice Processing* sub-module (§4.2.2) takes as input a voice recording and outputs the voice feature vector that can uniquely identify a speaker.

4.2.1 Face Processing

Figure 5(a) illustrates the processing pipeline of this sub-module. *Foice* processes face images in two steps.

Face Cropping and Normalization. In this step, *Foice* removes background scenes and normalizes head orientations and face sizes in the input face image. Specifically, we detect and crop the face appearing in the image using a face detection model. Then, the cropped face is normalized to ensure that each face has approximately the same size and is rotated so that two eyes lie on a horizontal line.

Face Blurriness Assessment. Subsequently, *Foice* evaluates the image quality of the cropped and normalized face and discards blurry face images from the dataset. Specifically, *Foice* evaluates the clarity and sharpness of the face image by obtaining a *quality score*, which we compute based on the observation that a sharp image generally contains a large number of edges. Here, we apply the *Sobel* edge detection [32] on the face to detect all the edges. Examples of images with different *quality scores* are listed in Figure 14. We only keep images with scores above a threshold (i.e., empirically set to 100). However, we note that *Foice* attack is still effective when taking blurry faces (i.e., those below the threshold) as input (see §5.5.3).

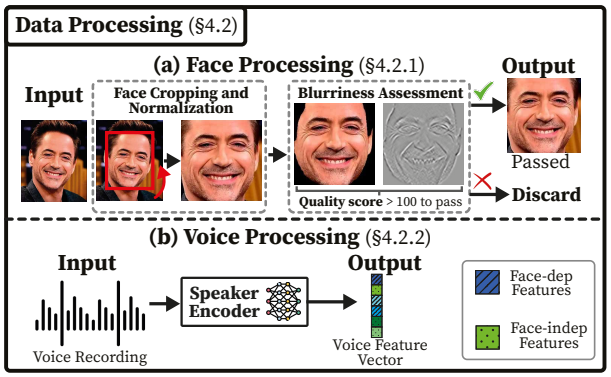


Figure 5: Figure depicts the pipeline of *Foice*'s *Data Processing* module (§4.2). (a) depicts the *Face Processing* (§4.2.1) which removes noise like head orientations and background scenes from the face images and filters out blurry images. (b) depicts the *Voice Processing* module (§4.2.2) which extracts the voice feature vector from the voice recording.

4.2.2 Voice Processing

Foice only utilizes this sub-module in its *Training Phase*. Figure 5(b) depicts the audio processing pipeline. We leverage a deep-learning model, namely the *Speaker Encoder* (see §2.1), to extract a voice feature vector from a voice recording. The extracted voice feature vector serves as the *ground truth* to facilitate the training of deep-learning models in §4.3 and §4.4.

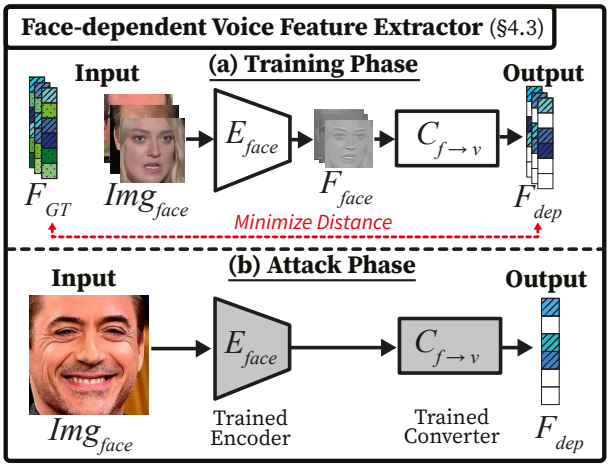


Figure 6: Figure depicts the pipeline of the *Face-dependent Voice Feature Extractor* module (§4.3). Specifically, it extracts relevant facial features from the processed face images and then converts them to *face-dependent* features in the voice feature vector.

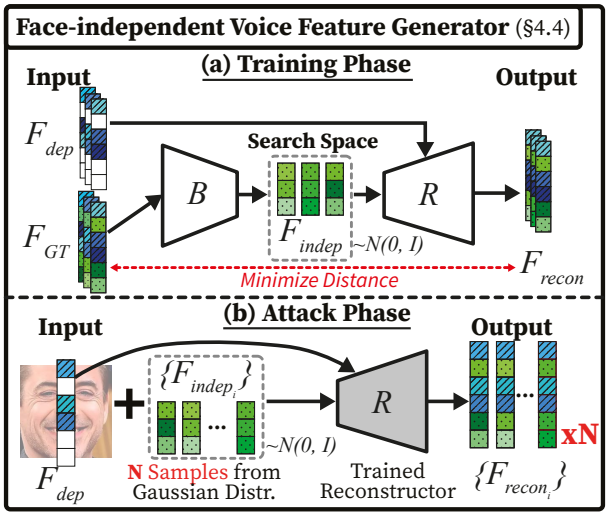


Figure 7: Figure depicts the pipeline of the *Face-independent Voice Feature Generator* module (§4.4). Specifically, it generates a search space of all possible *face-independent* features subject to different inner body structures by analyzing the dataset. It then combines both *face-dependent* and *face-independent* features to reconstruct the voice feature vectors of the victim, ultimately generating N reconstructed voice feature vectors.

4.3 Face-dependent Voice Feature Extractor

In this module, *Foice* aims to extract the *face-dependent* features, F_{dep}, from a processed face image. Figure 6 depicts the module pipeline in the *Training* and *Attack* phases.

Training Phase. In this phase, *Foice* trains a deep-learning model using processed face images, Img_{face}, and corresponding ground truth voice feature vectors, F_{GT}, from online public datasets (i.e., training dataset). The training objective is first to identify the facial features in face images that are correlated to the voice of the speaker, and then convert the relevant facial features to partial voice features, namely F_{dep}, as the model output. Figure 6(a) depicts the deep-learning models, namely an *Encoder*, $E_{face}(\cdot)$, and a *Converter*, $C_{f\to v}(\cdot)$[1]. Specifically, $E_{face}(\cdot)$ learns to identify relevant facial features from Img_{face} leveraging convolutional neural networks (CNN) such as ResNet [29]. $C_{f\to v}(\cdot)$ learns to convert facial features to F_{dep}.

$$F_{face} = E_{face}(Img_{face}), \qquad F_{dep} = C_{f\to v}(F_{face}),$$

where F_{face} denotes the facial features extracted by $E_{face}(\cdot)$.

To ensure that the extracted F_{face} correlates with the speaker's voice features, we train the two models jointly. We design the loss function to maximize the similarity (or minimize the distance) between F_{dep} and F_{GT}, such that

$$\min_{E_{face}(\cdot), C_{f\to v}(\cdot)} Err(F_{dep}, F_{GT}),$$

where $Err(\cdot,\cdot)$ denotes distance-based loss function. We note that our training scheme is effective in extracting *face-dependent* features with the following Observation 1. We present an empirical analysis of this observation in §A.2.1.

[1] See §A.1 for model architecture and implementation.

Observation 1. Following the training scheme in §4.3, *Foice* can extract *face-dependent* voice features, such as voice gender, from a single face image.

Attack Phase. The attacker inputs the victim's face image to the trained *Encoder*, $E_{face}(\cdot)$, and the *Converter*, $C_{f\to v}(\cdot)$, to extract F_{dep} of the victim. Figure 6(b) depicts the pipeline. Note that the attacker can input face images that are ***unseen*** in the *Training Phase*, hence attacking a wider range of victims.

4.4 Face-independent Voice Feature Generator

In this module, *Foice* aims to obtain the victim's *face-independent* features, F_{indep}. However, this is extremely challenging because F_{indep} are largely affected by the victim's inner body structure (see §2.3) which a face image provides almost no insight into. To overcome this challenge, the core idea is to analyze the known ground truth voice feature vectors, F_{GT}, of speakers in the public online dataset (i.e., training dataset) to generate a ***search space*** which enumerates all possible F_{indep}. Figure 7 illustrates this core idea. To launch the attack on the victim, the attacker can take N samples from the search space and combine them with the *face-dependent* features, F_{dep}, extracted from the face image to obtain a set of N reconstructed voice feature vectors, $\{F_{recon_i}\}$, *where* $i \in [1,N]$. By increasing N, the attacker has a higher chance of obtaining a F_{recon_i} sufficiently similar to the victim's F_{GT}. We now present how *Foice* generates the search space in the *Training*

Phase and how the attacker could use *Foice*'s model to attack a victim in the *Attack Phase*.

Training Phase. In this phase, we train a deep-learning model, which learns how to generate a search space from F_{dep} and F_{GT} of speakers in the training dataset. However, there are two main challenges.

First, separating F_{indep} from F_{GT} is non-trivial. To overcome this challenge, we design a *bottleneck* in the model (inspired by AutoVC [46]) to reduce the dimension of F_{GT} until it only contains F_{indep} (i.e., with as little F_{dep} as possible). The *bottleneck* is analogous to a "funnel" that filters out F_{dep} from F_{GT}. This is feasible by selecting a proper size of the "narrow opening of the funnel", or the dimension of the output of the *bottleneck*.

Second, the F_{indep} of speakers in the training dataset form **discrete** data points. It is still difficult to generalize to new speakers who are not in the dataset. For example, a victim's F_{indep} may be different from all speakers in the dataset and *Foice*'s search space should still contain it. To address this issue, we ensure that the search space adheres to a continuous distribution (i.e., standard Gaussian distribution). Intuitively, we may find the victim's F_{indep} by interpolating between discrete data points in the dataset.

Figure 7(a) depicts the details of our model design. Specifically, the *Bottleneck*, $B(\cdot)$, takes F_{GT} as input and outputs F_{indep} that has a smaller dimension and follows the continuous standard Gaussian distribution. Subsequently, the *Reconstructor*, $R(\cdot,\cdot)$, combines F_{indep} and F_{dep} to obtain the reconstructed voice feature vector, F_{recon}.

$$F_{indep} = B(F_{GT}), \qquad F_{recon} = R(F_{indep}, F_{dep})$$

We train the deep-learning model by minimizing a custom loss function to achieve two training objectives. First, to ensure that the search space is continuous, we minimize the KL-divergence loss (i.e., a standard loss often used to force feature vectors to follow the standard Gaussian distribution). Second, to ensure that $R(\cdot,\cdot)$ can accurately reconstruct voice features, we minimize the reconstruction error (i.e., the difference between F_{GT} and F_{recon}). Hence, the custom loss function is:

$$\min_{B(\cdot),R(\cdot,\cdot)} Err(F_{GT}, F_{recon}) + KL[P_{F_{indep}}(\cdot) \parallel \mathcal{N}(0,I)],$$

where $KL(\cdot \parallel \cdot)$ denotes the KL-divergence loss, $Err(\cdot,\cdot)$ represents the reconstruction error (e.g., distance-based loss function), and $P_{F_{indep}}(\cdot)$ denotes the probability density function of F_{indep}.

Note that *Foice* only utilizes $B(\cdot)$ in the *Training Phase* to obtain accurate extraction of F_{indep} from F_{GT} to train $R(\cdot,\cdot)$. We note that our bottleneck structure and training scheme are effective in extracting *face-independent* features with the following Observation 2, and we present an empirical analysis of this observation in §A.2.2.

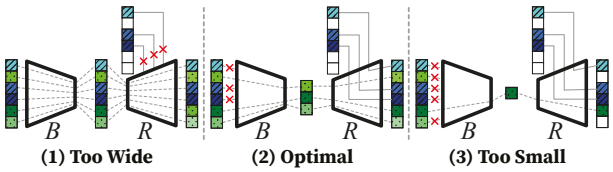


Figure 8: Figure depicts the implications of the bottleneck dimension to *Foice* when it is (1) too wide, (2) optimal, and (3) too small. *Foice* selects the optimal dimension.

Observation 2. By adjusting the bottleneck dimension, *Foice* is able to produce *face-independent* voice feature vectors that contains little *face-dependent* voice features.

The Rationale Behind Observation 2. Figure 8 depicts three scenarios to demonstrate the implications to *Foice* if the bottleneck dimension is (1) too wide, (2) optimal, and (3) too small. ***(1) Too wide:*** When the bottleneck dimension is too wide (i.e., as wide as F_{GT}), the bottleneck tends to copy F_{GT} directly to its output, F_{indep}, and the reconstructor could simply utilize F_{indep} as its output, F_{recon}, to achieve a minimal reconstruction loss (i.e., the difference between F_{GT} and F_{recon}). However, the search space in this case is overly large (i.e., as large as the entire voice feature vector space). ***(2) Optimal:*** When the bottleneck dimension decreases, F_{indep} is forced to discard some portion of it. To minimize the reconstruction loss, F_{indep} will first discard information preserved in F_{dep}. When the bottleneck dimension reaches the optimal, F_{indep} contains just the right amount of information and involves as few *face-dependent* features as possible in order to minimize the reconstruction loss. ***(3) Too small:*** When the bottleneck dimension is too small, F_{indep} starts to lose *face-independent* features in this case, leading to large reconstruction errors. As a result, the attacker may not be able to find the voice feature vector associated with the victim in the search space.

Attack Phase. Figure 7(b) depicts how the attacker can utilize F_{dep} of the victim and *Foice*'s trained model to obtain a set of voice feature vectors of the victim, $\{F_{recon_i}\}$, *where* $i \in [1,N]$. Specifically, the attacker first samples N random vectors as $\{F_{indep_1}, \ldots, F_{indep_N}\}$ from the standard Gaussian distribution, $\mathcal{N}(0,I)$. Then the attacker leverages the trained $R(\cdot,\cdot)$ to combine each F_{indep_i} with F_{dep} previously extracted from the victim's face image (see §4.3).

$$F_{indep_i} \sim \mathcal{N}(0,I), \qquad F_{recon_i} = R(F_{indep_i}, F_{dep}),$$

where $i \in [1,N]$. Finally, the attacker obtains $\{F_{recon_1}, \ldots, F_{recon_N}\}$ and then generates N synthetic voice recordings in the *Voice Synthesizer* module (§4.5).

4.5 Voice Synthesizer

Foice now takes as input a set of N reconstructed voice feature vectors, F_{recon}, and text of a given content (e.g., "Hey, Google!"). *Foice* then outputs ***N* synthetic voice recordings**

of the victim speaking the given content. Specifically, *Foice* adopts the deep-learning model used in the state-of-the-art voice deepfake system (see §2.1). Finally, the attacker enumerates through all synthetic voice recordings to authenticate and log in as the victim to voice authentication systems.

5 Evaluation

Through comprehensive real-world experiments, we seek to answer the following questions: ❶ Is *Foice* attack effective against diverse modern implementations of speaker authentication systems and voice assistants? (see §5.2) ❷ Can *Foice* provide more voice information other than age and gender? (see §5.3) ❸ Can we combine *Foice* and the existing voice deepfake system to improve the attack's effectiveness? (see §5.4) ❹ How do different experimental conditions affect the effectiveness of *Foice*? (see §5.5) ❹ How does *Foice* behave when converting facial features into voice features? (see §5.6)

5.1 Experiment Setup

We now present the experiment setup to evaluate *Foice*.

5.1.1 Voice Authentication System Selection

We evaluate *Foice* on various state-of-the-art voice authentication systems and off-the-shelf voice assistants, listed in Table 1. The list provides a representative collection of systems that can be acquired and implemented for experiments.

Voice Authentication Systems. We select two types of voice authentication systems for our evaluation - *on-device systems* and *cloud services*. First, we choose commercial on-device voice authentication systems installed on commodity smartphones. In addition, we experiment on *commercial* cloud authentication services, such as the voice authentication API from Microsoft and iFlytek, and deep learning models proposed in *academic* papers, such as VGGVox and DeepSpeaker, which could be deployed as cloud services. We select the official GitHub implementation of VGGVox [11] and DeepSpeaker [3], which are pre-trained on VoxCeleb2 [21] and LibriSpeech dataset [45], respectively. Cloud services may exhibit superior authentication capabilities compared to on-device systems. This is because on-device systems are designed to be lightweight due to limited computing resources on the edge device. In contrast, cloud services can utilize large deep-learning models without such constraints.

Voice Assistants. Initially designed for automatic speech recognition (i.e., conversion from speech recordings to written text), voice assistants are increasingly integrating voice authentication for personalized user services. This paper primarily focuses on voice assistants incorporating voice authentication. Consequently, we select and experiment with off-the-shelf systems - e.g., Siri, Google Assistant, and Bixby.

Authentication Mechanism. The aforementioned systems work by **verifying** the user's identity. Each system has exactly one enrolled speaker and checks if an input voice is uttered by the enrolled speaker or not. Specifically, by comparing the enrolled voice with the input voice, the system computes a match score to indicate how closely they match each other. If the match score exceeds a pre-defined threshold, then the system accepts the input as a match; otherwise, it rejects the input voice (see §2.2). The *on-device systems* (i.e., WeChat, Siri, Google Assistant, and Bixby) have fixed thresholds that users cannot modify. Cloud services are initially configured with default thresholds, yet users can adjust the threshold based on their specific usage scenarios. This paper assesses cloud services across various threshold values, with the threshold configuration discussed in §5.2.2. It is worth taking note that while the main purpose of voice assistants like Siri, Google Assistant, and Bixby is to recognize activation words such as "Hey, Google", these systems also include the above-mentioned authentication measures to verify the speaker's identity.

5.1.2 Benchmark Voice Deepfake System

We utilize the state-of-the-art voice deepfake system with public code and pre-trained models as our benchmark. Specifically, we select SV2TTS [31] because it outperforms other state-of-the-art voice deepfake systems due to its outstanding performance in generalizing to *unseen* speakers. SV2TTS takes as input a single voice recording from the victim and outputs one synthetic recording. We use a well-known GitHub implementation [1] pre-trained on the VoxCeleb and LibriSpeech datasets.

5.1.3 Speaker Dataset

We use three different speaker datasets, namely VoxCeleb1 [43], VoxCeleb2 [21], and a custom dataset, that contain face images and associated voice recordings for our experiments. We use each speaker's face image as the input to *Foice* to produce multiple synthetic voice recordings. We also input the corresponding voice recording to the benchmark voice deepfake model (i.e., SV2TTS) to generate a synthetic voice recording as the baseline.

VoxCeleb. VoxCeleb1 and VoxCeleb2 are public datasets containing YouTube videos. In particular, VoxCeleb1 contains over 100,000 videos for 1,251 celebrities, and VoxCeleb2 contains over a million videos from 6,112 celebrities. Each speaker in the dataset has multiple face images and voice recordings. There is no shared speaker identity between these two datasets. Both datasets demonstrate relatively equal representation of genders, with male gender accounting for 55% in VoxCeleb1 and 61% in VoxCeleb2, respectively. However, the videos are collected in many noisy environments – e.g., red carpets, outdoor stadiums, and lecture halls. Con-

Category	System	System Type	Commercial/ Academic	Eval. Param.		Overall Success Rate			Average Individual Success Rate (Foice)
				#Spk.	Threshold	SV2TTS [31]	Foice	Augmentation Attack (Foice + SV2TTS)	
 On-Device System	WeChat	Authentication	Commercial	10	—	50.0%	**100%**	—	29.7%
	Siri	Voice Assistant	Commercial	10	—	50.0%	**70.0%**	—	40.9%
	Google Assistant	Voice Assistant	Commercial	10	—	50.0%	**60.0%**	—	10.3%
	Bixby	Voice Assistant	Commercial	10	—	30.0%	**50.0%**	—	3.6%
 Cloud Service	Microsoft API	Authentication	Commercial	597	0.1 - 0.8	0% - 86.9%	0% - 95.0%	**0% - 99.6%**	0% - 29.5%
	iFlytek API	Authentication	Commercial	1021	0.1 - 0.8	0% - 100%	0% - 100%	**0% - 100%**	0% - 99.5%
	VggVox	Authentication	Academic	1029	0.1 - 0.8	2.9% - 97.8%	8.9% - 99.3%	**26.1% - 99.9%**	2.3% - 84.6%
	DeepSpeaker	Authentication	Academic	1029	0.1 - 0.8	0.2% - 99.5%	0.5% - 100%	**2.4% - 100%**	1.0% - 99.4%

Table 1: Table depicts the four on-device systems and four cloud services tested in our experiment. We present the evaluation parameters for each system, the overall success rate of *SV2TTS* [31], *Foice*, and *Augmentation Attack* (i.e., *Foice* + *SV2TTS*), and the average individual success rate achieved by *Foice*.

sequently, voice recordings in the dataset may contain background conversations, laughter, speech overlap, and diverse ambient noise. Similarly, face images experience variations in pose, lighting conditions, and blurriness due to motion. We use VoxCeleb2, a larger dataset, for training and VoxCeleb1 for evaluation of *Foice* on cloud-based authentication services.

Custom Dataset. We evaluate the real-world implications of *Foice* on on-device authentication systems by creating our own dataset of ten participants with different ages (six in their 20s; two in their 30s; one in their 40s; one in their 50s) and genders (seven male and three female). Each participant utilizes the Voice Memo app on the iPad Air (4th generation) for recording. In a quiet meeting room, they read two sentences from the Rainbow Passage [26]. This voice recording is input to SV2TTS to generate a synthetic voice recording, which serves as our benchmark. We also collect a profile photo from each participant as the input to *Foice*. We conduct the experiments and data collection by adhering to our university's Institutional Review Board (IRB) approval (see §A.4). Table 3 summarises the dataset employed to train and evaluate *Foice*.

5.1.4 Performance Metrics

We define the following three metrics to evaluate *Foice*:

Overall Success Rate: Percentage of *speakers* with at least one synthetic voice recording that passes the authentication (or successfully activate the voice assistant).

Individual Success Rate: Percentage of *synthetic voice recordings* of a particular person that passes the authentication (or successfully activate the voice assistant).

Foice Individual Success Rate: Fraction of voice cloned from a single face image that can pass the verification. A higher individual success rate indicates that the participant is more likely to be attack by *Foice*.

The overall success rate captures the general view of the security vulnerability of the target authentication system. In contrast, the individual success rate indicates the chances of attacking a victim successfully.

5.2 Can *Foice* Attack Real-world Systems?

We present our experiment results on eight modern voice authentication systems and voice assistants as listed in Table 1

5.2.1 Analysis of On-device Systems

Users interact with on-device systems (i.e., WeChat, Siri, Google Assistant, and Bixby) by engaging in direct conversation with the device microphones. To resemble real-world scenarios, we recruit ten participants in our experiments.

Data Preparation. We evaluate the effectiveness of *Foice* and SV2TTS attacks on these systems. Specifically, we use our custom dataset (see §5.1.3) to generate synthetic voice recordings of each participant using both *Foice* and SV2TTS. Ultimately, for each speaker, we generate 100 synthetic voice recordings using *Foice* and one synthetic recording using SV2TTS. This is due to *Foice*'s ability to produce multiple synthetic voice recordings from a single face image (see §4.4), whereas SV2TTS can generate only a single synthetic recording from a single voice input (see §5.1.2).

Experiment Setup. Each participant enrolls their voice with a dummy account on each system. Subsequently, we use a laptop speaker to directly play the synthetic voice recordings to the smartphone hosting the on-device system. The smartphone is positioned next to the laptop, with the smartphone's microphone facing directly towards the speaker. We consider an attack successful if the synthetic voice recording can log in to the WeChat account or activate the voice assistant.

Results of WeChat. As depicted in Table 1, *Foice* yields an overall success rate of 100%. In other words, all the participants have at least one *Foice*-generated synthetic voice recording that can successfully log in to the WeChat account protected by their voice. In addition, *Foice* achieves an average individual success rate of 29.7%. This indicates that, on average, approximately 30 out of 100 synthetic recordings of each participant can successfully log in to WeChat. Given that WeChat allows ***unlimited*** login attempts, this result shows that *Foice* poses a significant threat to WeChat's Voiceprint.

By comparison, SV2TTS only achieves an overall success rate of 50%. This is because the synthetic voice recordings generated by SV2TTS exhibit audible noise. Some are even unintelligible to the human ear, rendering the authentication impossible. It appears that SV2TTS is sensitive to noise in the input audio. Although we recorded the input voice in a *quiet* and *empty* meeting room (see §5.1.3), we still observed echos in the recording, which could be the primary cause of the noisy output. This poses a notable disadvantage for attackers utilizing SV2TTS because noise-free audio is not easily available. Also, removing noise from audio is still a difficult research problem [22].

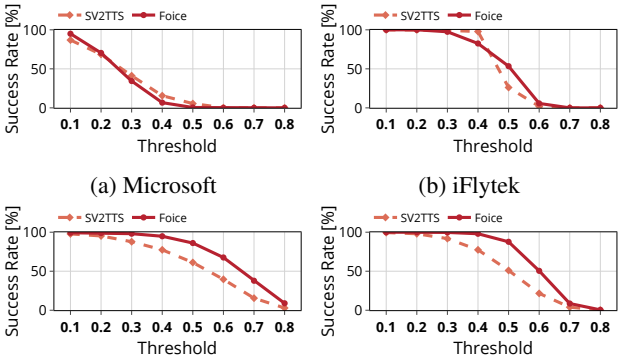
Results of Siri, Google Assistant, and Bixby. Table 1 summarizes the overall success rate and average individual success rate. Overall, *Foice* yields an overall success rate ranging from 50% to 70% across all the three voice assistants while the highest success rate for SV2TTS is 50%. Again, this is due to the presence of noise in SV2TTS' outputs, causing the voice assistant to be unable to recognize the activation words. In addition, we note that the *voice synthesizer* (see §4.5) struggles to accurately pronounce specialized vocabulary. For example, "Siri" tends to be pronounced as "sigh ree". In turn, this impacts the success rate of both *Foice* and SV2TTS. As the technology of voice synthesizers improve, so should the performance of both *Foice* and SV2TTS.

5.2.2 Analysis of Cloud Services.

We also evaluate *Foice* on two state-of-the-art cloud-based commercial voice authentication APIs (Microsoft and iFlytek) and two academic proposals (VGGVox and DeepSpeaker). Unlike on-device systems, cloud services accept inputs through their software interface rather than through a microphone. This allows us to conduct a large-scale experiment using a public dataset.

Data Preparation. We experiment on the VoxCeleb1 dataset. **(1) Enrollment.** We filter out blurry face images (see §4.2.1) and select high-quality voice recordings using NISQA [41] (i.e., a deep learning model evaluating the audio quality). In total, we have a dataset with 1,029 speakers (out of a total of 1,251 speakers; see §5.1.3). Table 1 summarizes the number of enrolled speakers. Not all speakers are successfully enrolled to Microsoft and iFlytek due to poor-quality recordings (i.e., audio signal-to-noise ratio below 2dB). **(2) Authentication.** We generate synthetic voice recordings using both *Foice* and SV2TTS. *Foice* takes as input one face image, whereas SV2TTS uses a *new* voice recording (i.e., not used for enrollment) as input. The generated recordings are then sent to each cloud service for authentication.

Threshold. Cloud services rely on an adjustable threshold (a numerical value between 0 and 1) to decide if two recordings are from the same speaker (see §2.1). Table 1 lists the threshold values used for evaluation. Although Microsoft and iFlytek recommend fixed thresholds for their systems, we vary said thresholds to investigate their overall performance.



(a) Microsoft (b) iFlytek

(c) VGGVox (d) DeepSpeaker

Figure 9: (a) to (d) depict the overall success rate across a range of thresholds from 0.1 to 0.8 for four tested cloud services, respectively.

System	Default/Optimal Threshold	Overall Success Rate		Average Individual Success Rate
		SV2TTS [31]	*Foice*	
Microsoft	0.5	**5.5%**	0.5%	1%
iFlytek	0.6	2.0%	**5.7%**	3.3%
VGGVox	0.6	39.6%	**67.6%**	15.4%
DeepSpeaker	0.5	51%	**87.7%**	32.7%

Table 2: Table depicts the default/optimal thresholds of four tested cloud services and the overall success rate of *SV2TTS* and *Foice*, respectively. In addition, it presents the average individual success rate achieved by *Foice*.

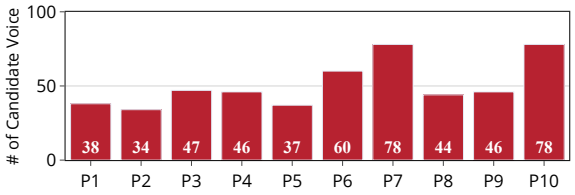
Results. Figure 9 depicts the attack success rate across a range of thresholds from 0.1 to 0.8. *Foice* achieves comparable results as SV2TTS in attacking Microsoft and iFlytek. Surprisingly, *Foice* yields higher success rates than SV2TTS in attacking VGGVox and DeepSpeaker across all the thresholds. ***We note that regardless of differences in system design and implementation, all systems are susceptible to Foice when the threshold is set at their default or optimal value*** as summarized in Table 2. The optimal threshold of VGGVox and DeepSpeaker is set to minimize their equal error rate. It is possible to guard against *Foice* by setting a higher threshold, but this will result in higher false rejects, impairing usability. On the other hand, setting a lower threshold can significantly increase *Foice*'s success, as Figure 9 shows. In addition, we observe that academic models exhibit greater vulnerability than commercial systems. This could be due to insufficient training resources (e.g., training data, computing power) or an inadequate optimization strategy.

5.3 Does *Foice* Leverage Voice Information Beyond Gender and Age?

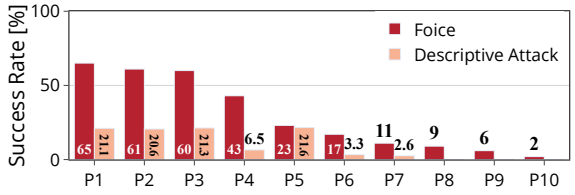
Foice leverages the correlation between face and voice to extract *face-dependent* features, such as gender and age (see

Dataset	#Speaker	#Instances	Details
Train Set	4108	1273017	Part of **VoxCeleb2** [21] used for training *Foice*
Test-1 Set	1029	1029	Part of **VoxCeleb1** [43] used for testing Microsoft, iFlytek, VGGVox, and DeepSpeaker
Test-2 Set	10	10	Data from **recruited participants** used for testing WeChat, Siri, Google Assistant, and Bixby

Table 3: Table summarises the datasets for training and testing. "#Speaker" specifies the total number of speakers. "#Instances" specifies the total number of face images. "Details" highlights the source and purpose of each dataset.



(a) Number of candidate voice recordings



(b) *Foice* vs. *Descriptive Attack*

Figure 10: (a) depicts the number of candidate recordings obtained from the dataset for each participant. (b) depicts the individual success rate of *Foice* and *Descriptive Attack* on WeChat's Voiceprint, respectively.

§4.3). However, apart from the face image, the attacker could gather the victim's gender and age from other sources (e.g., healthcare databases). Hence, we aim to investigate the following: (1) Can a voice deepfake attack be launched using only descriptive features (e.g., gender and age)? (2) Does *Foice* leverage more information other than gender and age?

Descriptive Attack. As such, we define a *Descriptive Attack* that only leverages descriptive information (i.e., age and gender) as a baseline. To launch the attack, the attacker searches from a large-scale voice dataset for speakers of similar age and the same gender as the victim. Subsequently, these speakers' voice recordings are input to voice deepfake systems (e.g., SV2TTS) to generate candidate synthetic voice recordings. Finally, the attacker attempts to compromise the authentication system using all the candidate recordings.

Data Preparation. We utilize VoxCeleb2, on which *Foice* is trained, as the dataset for the *Descriptive Attack*. We ensure that *Foice* and *Descriptive Attack* are implemented with the same dataset. To attack a victim, we search for voice recordings with similar age (i.e., age difference no larger than two years) and the same gender from VoxCeleb2. For example, if the attacker wants to attack a 30-year-old male speaker, all the male speakers between the ages of 28 and 32 are selected. Figure 10(a) provides an overview of the number of candidate recordings for each participant.

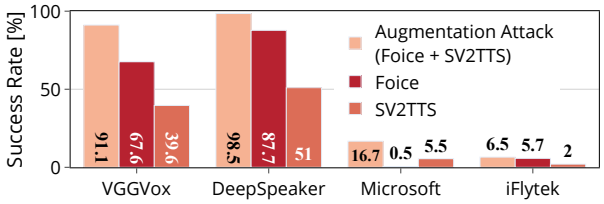


Figure 11: Figure depicts the overall attack success rate of *Augmentation Attack* (i.e, *Foice* + *SV2TTS*), *Foice*, and *SV2TTS*, for four tested cloud services (with the default or optimal threshold), respectively.

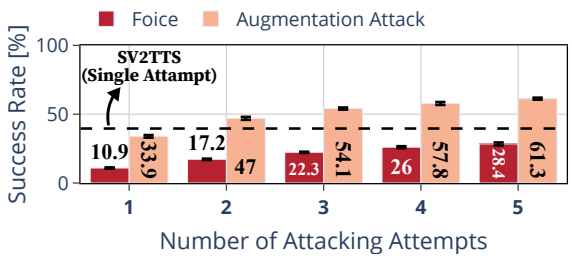
Descriptive Attack* vs. *Foice*.** We evaluate the *Descriptive Attack* on WeChat with the same experiment setup in §5.2.1. Figure 10(b) compares the individual success rate (i.e., the success rate in attacking a specific speaker) of *Foice* and *Descriptive Attack*. *Foice* outperforms the *Descriptive Attack* across all the participants. *Foice* yields an average individual success rate of 29.7%, and the *Descriptive Attack* has an average individual success rate of 9.7%. These results indicate that ***Foice leverages voice information beyond age and gender for voice synthesis. Moreover, we can observe from Figure 10(a) that even if VoxCeleb2 contains recordings from 6,112 speakers, each participant has less than 100 candidate synthetic recordings. On the other hand, *Foice*'s capability in enumerating all the possible voice recordings is not constrained by the training dataset.

5.4 Can *Foice* Augment Existing Voice Deepfake Systems?

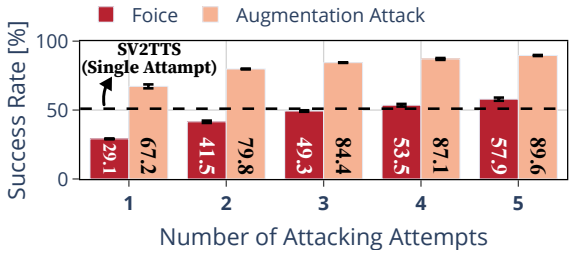
Recall that existing voice deepfake systems (e.g., SV2TTS) can only generate one synthetic recording given one voice input. This section investigates if *Foice* can be used to augment SV2TTS attack to improve the attack performance.

Augmentation Attack. We assume that the attacker can obtain a short voice recording (i.e., less than a minute) and a face image of the victim. To launch the attack, the attacker averages the voice feature vector generated by SV2TTS with each of the N vectors produced by *Foice*. Then, the attacker generates N synthetic voice recordings from the averaged feature vectors. Finally, the attacker enumerates through all synthetic recordings to compromise the authentication system.

Evaluation of *Augmentation Attack*. We conduct a large-scale experiment with commercial APIs and academic models, utilizing the VoxCeleb1 dataset. We follow the same experiment setup in §5.2.2, where each speaker has 100 synthetic



(a) VGGVox



(b) DeepSpeaker

Figure 12: Figure depicts the *Foice*'s overall success rate with varying numbers of attacking attempts on (a) VGGVox and (b) DeepSpeaker, respectively.

voice recordings output by the *Augmentation Attack*. Figure 11 depicts the overall success rate of SV2TTS and *Augmentation Attack* when targeting voice authentication systems using either the default or optimal threshold. *Augmentation Attack* outperforms SV2TT and *Foice* across all the four evaluated systems. ***This finding highlights the effectiveness of Foice in augmenting existing voice deepfake attacks.***

5.5 How Robust is *Foice*?

We evaluate the practicality of *Foice* over several different conditions. In total, our experiments target 1,029 speakers from the VoxCeleb1 dataset (see §5.2.2 Experiment Setup) on VGGVox and DeepSpeaker. We choose the optimal threshold of 0.5 and 0.6 for VGGVox and DeepSpeaker, respectively.

5.5.1 Impact of Limiting Attack Attempts

Some real-world authentication systems, such as mobile banking apps, generally allow three to five login attempts for improved security. We evaluate *Foice* and *Augmentation Attack*'s performance in attacking academic models (e.g., VGGVox and DeepSpeaker) with **a maximum of five attack attempts**. Figure 12 depicts the results. The success rate grows with the number of attempts for both VGGVox and DeepSpeaker. With a single face image, *Foice* is effective with only five attempts, yielding a success rate of 29.4% for VGGVox and 58.6% for DeepSpeaker. With a face image and a short voice recording, the *Augmentation Attack* outperforms the single-attempt voice-only attack (e.g., SV2TTS) with less than five attempts.

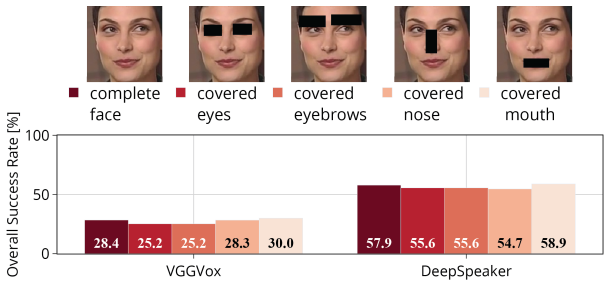


Figure 13: Figure depicts *Foice*'s overall success rate with *varying missing facial features* in the face images on VGGVox and DeepSpeaker, respectively.

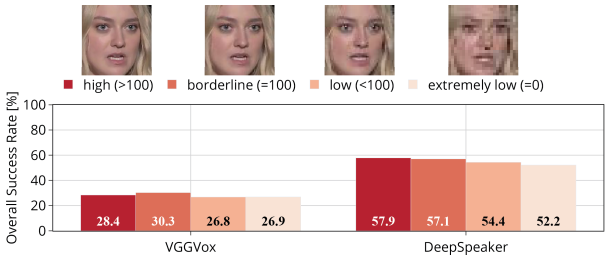


Figure 14: Figure depicts *Foice*'s overall success rate with *varying resolutions of face images* on VGGVox and DeepSpeaker, respectively.

The *Augmentation Attack* achieves a success rate above 50% across both systems with only five attempts. ***These results demonstrate the potential of Foice in realizing a practical attack that limits the number of attack attempts.***

5.5.2 Impact of Image Occlusion

The attack performance may be constrained by image occlusion – e.g., wearing sunglasses or masks. We cover critical facial features, such as eyebrows, eyes, nose, and mouth as depicted in Figure 13. We evaluate *Foice*'s performance with five attack attempts. For both VGGVox and DeepSpeaker, we observe a decrease in the success rate when the eyes, eyebrows, and nose are covered. In contrast, there is a subtle increase in the success rate when the mouth is covered. We attribute this to the varying mouth shapes in our training images due to people speaking. Hence, mouth might be considered by *Foice* as noise since the same speaker could have different mouth shapes. With a cleaner dataset, *Foice*'s performance may improve as it may utilize mouth features as well. ***Overall, Foice is robust to image occlusion.***

5.5.3 Impact of Image Resolution

The attacker might not always obtain high-resolution face images of the victim. To explore the impact of image resolution, we down-sample the face image evaluated in §5.2.2 to different resolutions. By assessing the resolution using

our *Face Blurriness Assessment* algorithm (see §4.2.1), we categorize images into four resolution levels: high resolution (score > 100), borderline resolution (score ≈ 100), low resolution (score < 100), and extremely low resolution (score ≈ 0). Figure 14 shows the success rate of *Foice* with five attack attempts. We observe that the success rate increases with the image resolution. *Foice* achieves a surprisingly high success rate on both systems, even when taking the extremely low-resolution image as input. We conjecture that *Foice* mainly relies on the overall face structure rather than detailed features (e.g., texture) to derive voice features. As a result, *Foice* nearly remains unaffected when we blur the face image. ***The results indicate that Foice is effective even when taking the blurry face as input.***

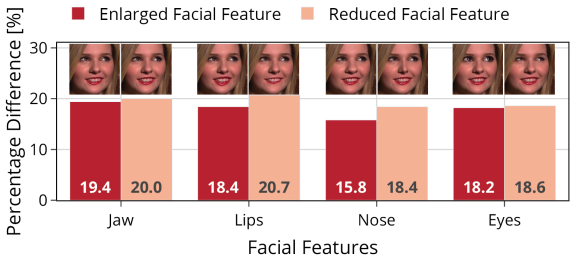
5.6 How Do Facial Features Affect the Output?

We now investigate the behavior of *Foice*'s *Face-dependent Voice Feature Extractor* (see §4.3) by analyzing how facial features affect its output. Our analysis involves altering various facial features (e.g., eyes, noses, jawlines, and lips) through enlargement or reduction in the input face image. Subsequently, we evaluate the resulting audio output to compare voice features with those obtained from the ground truth audio.

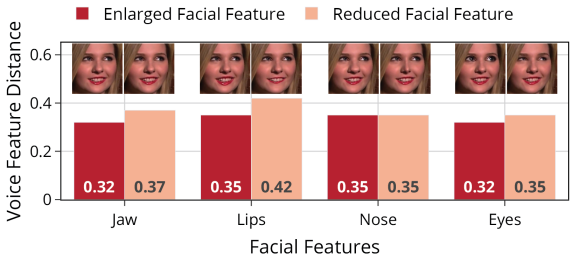
Data Preparation. We select 50 speakers (25 males and 25 females) randomly from the VoxCeleb1 dataset. Using Adobe Photoshop, we then change the size of facial features such as eyes, noses, jawlines, and lips in each image (see Figure 15). Subsequently, we employ *Foice*'s *Face-dependent Voice Feature Extractor* (see §4.3) and *Voice Synthesizer* (see §4.5) to convert these modified images into audio outputs.

Voice Features. We calculate voice features, including pitch (i.e., highness or lowness of a sound), formant frequency (i.e., timbre), spectral centroid (i.e., the brightness of a sound), and spectral bandwidth (i.e., the sharpness of a sound), from each generated audio file. We quantify the pitch difference between the synthetic audio, x, and the speaker's actual voice, y, using percentage difference, $PD = |\frac{x-y}{y} \times 100\%|$. We then compare PD when enlarging or reducing the size of facial features to calculate the delta, $\Delta = |PD_{enlarge} - PD_{reduce}|$, which indicates pitch sensitivity to changes in facial features. We use Euclidean distance for the remaining voice features to quantify the difference as they are high-dimensional.

Results. We observe notable effects on pitch and formant frequency with modifications to facial features. However, changes in facial features have minimal impact on other voice attributes (i.e., spectral centroid and spectral bandwidth). Figure 15a illustrates the effect on pitch, while Figure 15b depicts the effect on formant frequency, resulting from enlarging and reducing facial features in the image. In Figure 15a, it is evident that *Foice*'s deep-learning model associates **lips** and **noses** with pitch. For instance, enlarging and reducing noses result in a percentage difference of 15.8% and 18.4%, respectively, with the largest delta of 2.6%. Similarly, modifying



(a) Pitch



(b) Formant Frequency

Figure 15: Figure depicts the impact on (a) Pitch and (b) Formant Frequency with enlarged and reduced facial features in the face image.

lips yields the second-largest delta of 2.3%. This aligns with the biological association between facial features and voice features (see §2.3). In Figure 15b, we also notice a similar pattern where lips emerge as the most influential facial feature, followed by jawlines. When enlarging and reducing lips, we observe a feature distance of 0.35 and 0.42, respectively, with the largest delta being 0.07. Similarly, modifying jawlines results in the second-largest delta of 0.05. Jawlines exert a more pronounced effect on formant frequency compared to the nose because formant frequencies represent the resonant frequencies of the vocal tract, which can be influenced by the width or narrowness of jawlines. Therefore, it is clear that *Foice* can effectively extract pertinent facial features and convert them into pitch and formant frequencies, which are crucial features often used in speaker identification [48].

6 Discussion

We present relevant discussion points of *Foice* in this section.

Deployment Considerations. We note that there is an increasing trend of incorporating voice authentication into real-world systems [2,9,10,13]. Hence, it is not difficult to imagine voice authentication replacing password-based authentication to improve user experience. For example, many instant messaging services, including Facebook Messenger and WhatsApp, could employ voice authentication in the near future for increased usability [12]. Furthermore, we note that existing authentication systems may adopt multi-factor authentication (MFA) to enhance their security. For example, voice biomet-

rics and one-time passwords (OTP) are often used as MFA factors for telebanking [6] and WeChat also adopts multiple factors to protect users' accounts. Hence, if *Foice* is to be deployed to gain unauthorized access to the victim's account, we envision that *Foice* would compromise the voice factor while leveraging additional attack methods to compromise other factors (e.g., password and OTP) [34].

Improving *Foice*'s Performance. While we demonstrate the effectiveness in launching *Foice* attack on commercial systems, there is still room for improvement. First, we can enhance *Foice*'s processing pipeline such as improving noise reduction methods to normalize facial expressions and lip movements in face images (see §4.2). Second, we can incorporate diverse information from the victim. For example, we can combine face images with the victim's voice samples (see §5.4) or utilize 3D face photos, to capture additional details of the victim's face structures [24, 39]. Furthermore, as a proof-of-concept, *Foice* only adopts a simple architecture for the deep-learning models (see §A.1). Further optimization of the model architecture would contribute to further improving *Foice*'s performance.

Countermeasures. One simple countermeasure is to restrict the number of login attempts to authentication systems. However, we note that the tested systems are still vulnerable to *Foice* with a maximum of five login attempts (see Figure 12 in §5.5.1). Hence, we envision two potential countermeasures. First, integrating *deepfake voice detection* methods [20, 38, 54] into authentication systems could mitigate problems that may arise from attacks like *Foice*. Second, authentication systems could also incorporate *liveness detection* methods [14, 40, 53] to determine the source of the input voice signal, such as whether it originates from a human voice or a pre-recorded voice playback. However, it is important to note that these two directions have not been incorporated into real-world systems. Hence, additional measures need to be implemented and deployed to adequately safeguard against voice deepfake attacks such as *Foice*.

7 Related Work

We present related works of the attacks on voice authentication systems and voice synthesis using face information.

Attacks on Voice Authentication Systems. There are two categories of attacks compromising voice authentication systems. First, an *adversarial attack* exploits the vulnerabilities of machine learning algorithms of the authentication system to produce erratic predictions [17–19, 36, 52, 58, 60]. However, for an *adversarial attack* to be successful, one must know either the implementation details of the authentication system or the victim's voice samples. On the contrary, *Foice* does not require this knowledge. Second, a *voice deepfake attack* leverages deep learning models to generate synthetic voice recordings that sound like the victim's voice [50, 57]. *Foice* is motivated by the *voice deepfake attack*, but only utilizes a single face image of the victim.

Voice Synthesis using Face Information. Recent studies investigate the correlation between face and voice. A family of works investigates the possibility of reconstructing voice from the associated face [28, 33, 37, 56, 59], and vice versa [25, 30, 44]. However, face images lack some essential voice features, making accurate face-to-voice conversion extremely challenging. Previous methods mostly rely on voice cues in the face, but we propose a new method to generate missing voice features and our approach significantly improves voice reconstruction performance compared to prior methods (see §A.3). Furthermore, prior works fail to underscore the **security implications**. For example, *Face2Speech* [28], *FaceVC* [37], *SP-FaceVC* [56] and *Face-TTS* [33] evaluate audio quality of synthetic voice recordings and how well the recordings match the corresponding face images. However, none of these works evaluates the similarity between the speaker's ground truth and synthesized voices while the similarity is the key to attacking voice authentication systems. Lastly, another family of works investigates the use of video to generate voice recordings [59]. *Foice*, on the other hand, only utilizes a single face image.

8 Conclusion

We present *Foice*, a novel pervasive and scalable voice deepfake attack that uniquely leverages only ***a single face image of the victim***. Hence, *Foice* addresses the limitation of state-of-the-art voice deepfake attacks that require the victim's voice samples, which might not always be readily available. By exploiting the correlation between facial and voice features, which originate from physiological structures, we design *Foice* to synthesize voice recordings. We demonstrate the feasibility of *Foice* through comprehensive real-world experiments, involving ten offline participants and an online dataset of 1,029 unique individuals, and testing on eight state-of-the-art voice authentication systems, such as WeChat and Microsoft Azure. We urge the research community and vendors of voice authentication systems to be alert to this new threat and to develop corresponding countermeasures.

9 Acknowledgement

We thank our shepherd and the reviewers for their insightful feedback. This work is partially funded by NUS WBS E-252-00-0019- 03 and the National Research Foundation of Korea (NRF), funded by the Ministry of Science and ICT (MSIT) under grant RS-2023-00277848. Jun Han is the corresponding author of this work.

References

[1] SV2TTS Implementation. https://github.com/CorentinJ/Real-Time-Voice-Cloning, 2019.

[2] Call Bixby with your voice. https://www.samsung.com/us/support/answer/ANS00076751/, 2023.

[3] DeepSpeaker GitHub Implementation. https://github.com/philipperemy/deep-speaker, 2023.

[4] Face-TTS GitHub Implementation. https://github.com/naver-ai/facetts, 2023.

[5] Gender Recognition By Voice. https://github.com/x4nth055/gender-recognition-by-voice, 2023.

[6] HSBC Voice ID. https://www.us.hsbc.com/customer-service/voice/, 2023.

[7] iFlytek Voiceprint Recognition. https://www.xfyun.cn/service/isv, 2023.

[8] Microsoft Azure Speaker Recognition. https://azure.microsoft.com/en-us/products/ai-services/speaker-recognition, 2023.

[9] Siri. https://www.apple.com/siri/, 2023.

[10] Teach Google Assistant to recognize your voice with Voice Match. https://support.google.com/assistant/answer/9071681, 2023.

[11] VGGVox GitHub Implementation. https://github.com/linhdvu14/vggvox-speaker-identification, 2023.

[12] Voice Authentication for WhatsApp. https://www.idrnd.ai/voice-biometrics-for-whatsapp/, 2023.

[13] WeChat's Voiceprint. https://kf.qq.com/touch/wxappfaq/150819uqYnUR150819YzINVb.html?platform=15, 2023.

[14] Muhammad Ejaz Ahmed, Il-Youp Kwak, Jun Ho Huh, Iljoo Kim, Taekkyung Oh, and Hyoungshick Kim. Void: A fast and light voice liveness detection system. In *USENIX Security*, 2020.

[15] Marc Arnela, Rémi Blandin, Saeed Dabbaghchian, Oriol Guasch, Francesc Alías, Xavier Pelorson, Annemie Van Hirtum, and Olov Engwall. Influence of lips on the production of vowels based on finite element simulations and experiments. *The Journal of the Acoustical Society of America*, 2016.

[16] Domna Bilika, Nikoletta Michopoulou, Efthimios Alepis, and Constantinos Patsakis. Hello me, meet the real me: Audio deepfake attacks on voice assistants. *arXiv preprint arXiv:2302.10328*, 2023.

[17] Guangke Chen, Sen Chenb, Lingling Fan, Xiaoning Du, Zhe Zhao, Fu Song, and Yang Liu. Who is real bob? adversarial attacks on speaker recognition systems. In *S&P*, 2021.

[18] Guangke Chen, Yedi Zhang, Zhe Zhao, and Fu Song. Qfa2sr: Query-free adversarial transfer attacks to speaker recognition systems. *arXiv preprint arXiv:2305.14097*, 2023.

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

[20] Akash Chintha, Bao Thai, Saniat Javid Sohrawardi, Kartavya Bhatt, Andrea Hickerson, Matthew Wright, and Raymond Ptucha. Recurrent convolutional structures for audio spoof and video deepfake detection. *IEEE Journal of Selected Topics in Signal Processing*, 2020.

[21] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622*, 2018.

[22] Ross Cutler, Ando Saabas, Tanel Parnamaa, Marju Purin, Hannes Gamper, Sebastian Braun, Karsten Sørensen, and Robert Aichner. Icassp 2022 acoustic echo cancellation challenge. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022.

[23] Peter B Denes and Elliot Pinson. *The speech chain*. Macmillan, 1993.

[24] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *CVPR workshops*, 2019.

[25] Amanda Cardoso Duarte, Francisco Roldan, Miquel Tubau, Janna Escur, Santiago Pascual, Amaia Salvador, Eva Mohedano, Kevin McGuinness, Jordi Torres, and Xavier Giro-i Nieto. Wav2pix: Speech-conditioned face generation using generative adversarial networks. In *ICASSP*, 2019.

[26] Grant Fairbanks. Voice and articulation drillbook. 1960.

[27] Asif A Ghazanfar and Drew Rendall. Evolution of human vocal production. *Current biology*, 2008.

[28] Shunsuke Goto, Kotaro Onishi, Yuki Saito, Kentaro Tachibana, and Koichiro Mori. Face2speech: Towards multi-speaker text-to-speech synthesis using an embedding vector predicted from a face image. In *INTERSPEECH*, 2020.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[30] Zhenhou Hong, Jianzong Wang, Wenqi Wei, Jie Liu, Xiaoyang Qu, Bo Chen, Zihang Wei, and Jing Xiao. When hearing the voice, who will come to your mind. In *2021 International Joint Conference on Neural Networks*, 2021.

[31] Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu, et al. Transfer learning

from speaker verification to multispeaker text-to-speech synthesis. *Advances in neural information processing systems*, 2018.

[32] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 1988.

[33] Jiyoung Lee, Joon Son Chung, and Soo-Whan Chung. Imaginary voice: Face-styled diffusion model for text-to-speech. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2023.

[34] Zeyu Lei, Yuhong Nan, Yanick Fratantonio, and Antonio Bianchi. On the insecurity of sms one-time password messages against local attackers in modern mobile devices. In *Network and Distributed Systems Security Symposium*, 2021.

[35] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu. Deep speaker: an end-to-end neural speaker embedding system. *arXiv preprint arXiv:1705.02304*, 2017.

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

[37] Hsiao-Han Lu, Shao-En Weng, Ya-Fan Yen, Hong-Han Shuai, and Wen-Huang Cheng. Face-based voice conversion: Learning the voice behind a face. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2021.

[38] Hafiz Malik. Securing voice-driven interfaces against fake (cloned) audio attacks. In *IEEE Conference on Multimedia Information Processing and Retrieval*, 2019.

[39] Dieter Maurer and Theodor Landis. Role of bone conduction in the self-perception of speech. *Folia phoniatrica*, 1990.

[40] Yan Meng, Jiachun Li, Matthew Pillari, Arjun Deopujari, Liam Brennan, Hafsah Shamsie, Haojin Zhu, and Yuan Tian. Your microphone array retains your identity: A robust voice liveness detection system for smart speakers. In *USENIX Security*, 2022.

[41] Gabriel Mittag, Babak Naderi, Assmaa Chehadi, and Sebastian Möller. Nisqa: A deep cnn-self-attention model for multidimensional speech quality prediction with crowdsourced datasets. *arXiv preprint arXiv:2104.09494*, 2021.

[42] Peter B Mueller. The aging voice. In *Seminars in speech and language*. © 1997 by Thieme Medical Publishers, Inc., 1997.

[43] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*, 2017.

[44] Tae-Hyun Oh, Tali Dekel, Changil Kim, Inbar Mosseri, William T Freeman, Michael Rubinstein, and Wojciech Matusik. Speech2face: Learning the face behind a voice. In *CVPR*, 2019.

[45] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *IEEE international conference on acoustics, speech and signal processing*, 2015.

[46] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson. Autovc: Zero-shot voice style transfer with only autoencoder loss. In *International Conference on Machine Learning*, 2019.

[47] Harriet MJ Smith, Andrew K Dunn, Thom Baguley, and Paula C Stacey. Concordant cues in faces and voices: Testing the backup signal hypothesis. *Evolutionary Psychology*, 2016.

[48] Sreenivas Sremath Tirumala, Seyed Reza Shahamiri, Abhimanyu Singh Garhwal, and Ruili Wang. Speaker identification features extraction methods: A systematic review. *Expert Systems with Applications*, 2017.

[49] Ingo R Titze and Daniel W Martin. Principles of voice production, 1998.

[50] Henry Turner, Giulio Lovisotto, and Ivan Martinovic. Attacking speaker recognition systems with phoneme morphing. In *24th European Symposium on Research in Computer Security*, 2019.

[51] Janwillem Van den Berg. Myoelastic-aerodynamic theory of voice production. *Journal of speech and hearing research*, 1958.

[52] Q Wang, P Guo, and L Xie. Inaudible adversarial perturbations for targeted attack in speaker recognition. arxiv 2020. *arXiv preprint arXiv:2005.10637*.

[53] Qian Wang, Xiu Lin, Man Zhou, Yanjiao Chen, Cong Wang, Qi Li, and Xiangyang Luo. Voicepop: A pop noise based anti-spoofing system for voice authentication on smartphones. In *IEEE INFOCOM Conference on Computer Communications*, 2019.

[54] Yuanda Wang, Hanqing Guo, Guangjing Wang, Bocheng Chen, and Qiben Yan. Vsmask: Defending against voice synthesis attack via real-time predictive perturbation. *arXiv preprint arXiv:2305.05736*, 2023.

[55] Timothy Wells, Thom Baguley, Mark Sergeant, and Andrew Dunn. Perceptions of human attractiveness comprising face and voice cues. *Archives of sexual behavior*, 2013.

[56] Shao-En Weng, Hong-Han Shuai, and Wen-Huang Cheng. Zero-shot face-based voice conversion: Bottleneck-free speech disentanglement in the real-world scenario. In *AAAI*, 2023.

[57] Emily Wenger, Max Bronckers, Christian Cianfarani, Jenna Cryan, Angela Sha, Haitao Zheng, and Ben Y Zhao. " hello, it's me": Deep learning-based speech synthesis attacks in the real world. In *CCS*, 2021.

[58] Xinghui Wu, Shiqing Ma, Chao Shen, Chenhao Lin, Qian Wang, Qi Li, and Yuan Rao. Kenku: Towards efficient and stealthy black-box adversarial attacks against asr systems. In *USENIX Security*, 2023.

[59] Zhihan Yang, Zhiyong Wu, Ying Shan, and Jia Jia. What does your face sound like? 3d face shape towards voice. 2023.

[60] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, Xiaofeng Wang, and Carl A Gunter. Commandersong: A systematic approach for practical adversarial voice recognition. In *USENIX Security*, 2018.

A Appendix

A.1 Implementation

We demonstrate the effectiveness of *Foice* in extracting voice features from the face image via a proof-of-concept implementation. We adopt the existing *Synthesizer* structure and only focus on implementing *Face-dependent Voice Feature Extractor* and *Face-independent Voice Feature Generator*. Figure 16 depicts *Foice*'s model structure. More details can be found in the GitHub repository https://github.com/SeCATrity/Foice.

A.2 Why does *Foice* Work?

In this section, we provide empirical evidence for **Observation 1** and **Observation 2** (see §4.3 and §4.4).

A.2.1 Observation 1: *Face-dependent* Features Extracted

We demonstrate the efficacy of *Foice* in extracting *face-dependent* feature vector from a single face image. Specifically, we investigate whether this vector reveals the **gender** information of the speaker. We utilize vector morphing, a commonly used technique to prove the existence of semantic information in the feature vector. By interpolating *face-dependent* vectors between two gender groups (i.e., male and female) and generating corresponding synthetic recordings, we investigate the voice gender of these recordings (i.e., whether the recording sounds like a male or a female voice).

Vector Morphing. Specifically, we take the weighted average of a *face-dependent* voice feature vector $VFeat_{ori}$ from the original gender (e.g., male) and a feature vector $VFeat_{opp}$ from the opposite gender (e.g., female):

$$F_{morphed} = (1-\omega) \times F_{ori} + \omega \times F_{opp},$$

where $\omega \in [0,1]$ denotes the morphing coefficient. A higher ω pulls $F_{morphed}$ closer to the opposite gender. For example, $\omega = 1$ results in a synthetic recording that sounds like the opposite gender, while $\omega = 0$ results in a recording that sounds like the original gender. Then, we utilize the state-of-the-art voice gender classifier [5] to conduct gender classification on the generated synthetic voice recordings.

Results. Figure 17 depicts the classification accuracy computed from the original and predicted gender label across different morphing coefficients. We observe that the accuracy (i.e., male, female, and overall) decreases smoothly with the increase of ω. These results suggest that the interpolated vector (i.e., the vector between the two gender groups) also captures semantic voice gender information. We highlight that we can derive voice gender information from a single face image with a proof-of-concept implementation (see §A.1). More accurate *face-dependent* voice features could be captured with a carefully designed structure (see §6).

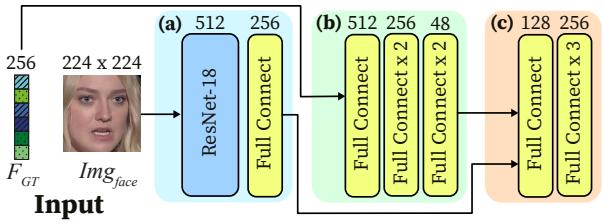


Figure 16: Figure depicts *Foice*'s deep-learning models for training. (a) depicts the *Encoder* and *Converter*. (b) depicts the *Bottleneck*, and (c) depicts the *Reconstructor*.

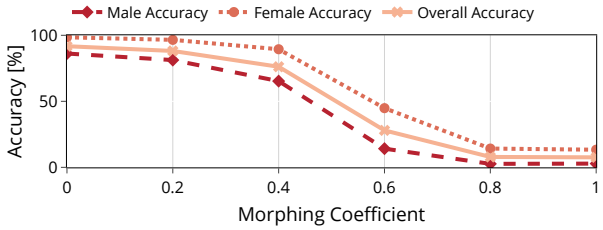
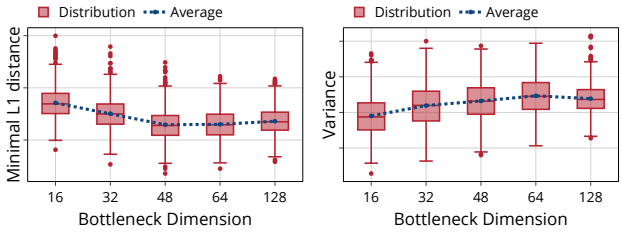


Figure 17: Figure depicts the empirical results of voice morphing. Specifically, the gender classification accuracy decreases as the morphing coefficient increases, indicating that the face-dependent voice features contain gender information.

A.2.2 Observation 2: Adjusting Bottleneck Dimension

Recall that we can find an optimal bottleneck dimension such that the search space contains only *face-independent* features (see Figure 8). We investigate the impact of varying bottleneck dimensions on (1) the size of the search space and (2) whether the search space contains a sufficiently similar *face-independent* feature vector to the victim's ground truth.



(a) Embedding similarity (b) Embedding diversity

Figure 18: (a) depicts the similarity of the voice features in the search space across varying dimensions. (b) depicts the voice features' variations and the search space size across varying dimensions.

Method	Overall Success Rate	
	VGGVox	DeepSpeaker
Face-TTS	0.97%	3.79%
Foice w/o Generator	10.9%	29.1%
Foice	67.6%	87.7%

Table 4: Table compares the attack performance of *Foice* with Face-TTS. VGGVox and DeepSpeaker are evaluated using their optimal threshold depicted in Table 2.

Experiment Setup. We experiment on VoxCeleb2 [21]. For each speaker in the dataset, we utilize the generated search space and the *face-dependent* feature vector to reconstruct 100 voice feature vectors, $\{F_{recon_1}, \ldots, F_{recon_{100}}\}$ (see §4.4). Since the *face-dependent* feature vector for each speaker is fixed by the input image, the space of the reconstructed voice feature vector exhibits the property of the search space. Then, we compute the L1 distance between each F_{recon_i} and the corresponding ground truth F_{GT}. For each speaker, we utilize the minimal L1 distance to evaluate the capability of a search space in producing a close enough F_{recon_i}. To evaluate the search space size, we use the standard deviation of the L1 distances to measure the variance of the search space.

$$Dist_{min} = \min_{i \in [1,100]} (\|F_{recon_i} - F_{GT}\|_1),$$

$$Var = \sqrt{\frac{\sum_{i=1}^{100} (F_{recon_i} - F_{GT})^2}{100}}$$

Lower $Dist_{min}$ means the attacker can find a voice feature vector that closely resembles the ground truth voice feature within 100 searching attempts. At the same time, larger Var indicates a larger variance in the search space. Hence, more samples are needed to find the target F_{GT}.

Results. Figure 18 depicts how different dimensions of the bottleneck affect the minimal L1 distance and the variance of the search space. We observe in Figure 18(a) that when the dimension increases, $Dist_{min}$ first decreases. This is because the search space contains more *face-independent* voice features such that the attacker can find voice feature vectors increasingly similar to the victim's ground truth. However, $Dist_{min}$ saturates when the bottleneck dimension reaches optimal because all the *face-independent* features are captured. In contrast, Figure 18(b) depicts an increasing trend in Var when increasing the dimension, indicating an increasing search space size. We attribute this to the increasing amount of voice features, including the *face-dependent* and the *face-independent*, encoded in the search space. In particular, dimension 48 yields a relatively small search space containing voice feature vectors sufficiently close to the ground truth. Hence, *Foice* selects the optimal dimension of 48.

A.3 Comparison with the Related Work

We compare *Foice* against Face-TTS [33], the latest closely related work. Like *Foice*, Face-TTS takes as input a text transcription and a face image, and generates a speech sample using a diffusion model conditioned on the input face image to model speaker characteristics. For evaluation, we utilize the official implementation and pre-trained weights provided by the authors [4], ensuring consistency and reproducibility in our experiments. In addition, we evaluate *Foice* without the Generator (see §4.4), which only relies on the *face-dependent* features for voice synthesis, to demonstrate the effectiveness of *Foice* in extracting *face-dependent* voice features from the face image. In total, our experiments target 1,029 speakers from the VoxCeleb1 dataset on VGGVox and DeepSpeaker. We choose the optimal threshold of 0.5 and 0.6 for VGGVox and DeepSpeaker, respectively. We evaluate the attack performance of *Foice* and Face-TTS using the *Overall Success Rate* (see §5.1.4), which represents the proportion of speakers in the dataset that can be attacked. We summarize the results in Table 4. Without the Generator, *Foice* achieves a success rate significantly higher, by an order of magnitude, than Face-TTS on both VGGVox and DeepSpeaker. With the Generator, *Foice* achieves an overall success rate that exceeds that of Face-TTS by 60 times on VGGVox and 30 times on DeepSpeaker. **These results demonstrate that *Foice* outperforms the state-of-the-art method by (i) extracting more relevant and accurate *face-dependent* voice features from the face image; (ii) and generating supplementary voice features that augment the *face-dependent* feature.**

A.4 Ethics

This study is approved by our university's Institutional Review Board (IRB). We carefully designed the experiments to protect the privacy of our participants. The photos and voice recordings collected are anonymized and securely stored on our servers. We use dummy accounts and the voices enrolled are deleted immediately after the completion of the experiments. We have responsibly disclosed the identified vulnerabilities to the affected company to ensure timely remediation.