Spoofing Speaker Verification With Voice Style Transfer And Reconstruction Loss

Thomas Thebaud, Gaël Le Lan, Anthony Larcher

Spoofing Speaker Verification With Voice Style Transfer And Reconstruction Loss

Thomas Thebaud
Orange
Cesson-Sévigné, France
0000-0001-8953-7872

Gaël Le Lan
Orange
Cesson-Sévigné, France
0000-0002-1493-5777

Anthony Larcher
LIUM
Le Mans, France
0000-0003-4398-0224

***Abstract*—In this paper we investigate a template reconstruction attack against a speaker verification system. A stolen speaker embedding is processed with a zero-shot voice-style transfer system to reconstruct a Mel-spectrogram containing as much speaker information as possible. We assume the attacker has a black box access to a state-of-the-art automatic speaker verification system. We modify the AutoVC voice-style transfer system to spoof the automatic speaker verification system.**

We find that integrating a new loss targeting embedding reconstruction and optimizing training hyper-parameters significantly improves spoofing. Results obtained for speaker verification are similar to other biometrics, such as handwritten digits or face verification. We show on standard corpora (*VoxCeleb* and *VCTK*) that the reconstructed Mel-spectrograms contain enough speaker characteristics to spoof the original authentication system.

***Index Terms*—x-vector, Zero-shot voice style transfer, Automatic Speaker Verification**

I. Introduction

There is a rising interest of ethical concerns about machine learning systems, especially for personal data protection. Most biometric authentication systems [1] use personal data such as voice recordings [2], face images [3], fingerprints [4] or handwritten digits [5] to extract identity characteristics of their users as high dimensional vectors, using deep feature extractors. Those vectors, usually named embeddings or templates, are stored in a database when a user registers during the enrollment phase. In the context of this paper, embeddings and templates are equivalent terms: high-dimensional vectors extracted by neural networks. Then for the authentication phase, new data from the user requiring access is collected and processed by the same feature extractor to compute trial embeddings. The computation and analysis of a score between trial and enrollment embeddings gives an estimation of the legitimacy of the user. Because enrollment embeddings are stored and transferred between devices, they could be vulnerable to theft, therefore they represent a potential breach in the system's security and a risk for the users' data safety. While template protection mechanisms exist [6], [7], we suppose here that the embeddings are not protected.

Mai et al. [3] investigate template reconstruction attacks on face images that require a black box access to the feature extractor. This attack uses a comparison feature extractor and a generator to reconstruct the face images from deep face templates. Our long term goal is to similarly design a template reconstruction attack on text-independent voice-based authentication systems, but without access to the feature extractor (as performed on handwritten digits in [8]). First, we need to prove such an attack is even feasible with a black box access to the extractor. Due to the nature of speech and the design of modern speaker verification feature extractors (temporal pooling layers [9]), it is yet to prove speaker embeddings [10] contain sufficient information to generate spoofing utterances. In this paper we propose to perform a template reconstruction attack with a black-box access to the feature extractor. We investigate a solution for the generation of spoofing speech utterances, only focused on the speaker's identity and not on the linguistic content of the utterances. The *fastResNet34* feature extractor architecture of [11] is chosen to extract the speaker identity features from Mel-spectrograms. For the generation of spoofing samples, we use a modified version of AutoVC [12], a zero-shot non-parallel voice conversion system that was designed for voice style transfer: disentangle identity and linguistic contents to generate Mel-spectrograms as uttered by a given speaker.

The main contributions of this paper are:

1) The use of a voice style transfer system [12] to reconstruct Mel-spectrograms related to an utterance-related speaker embedding.
2) The improvement of this system for spoofing attacks, thanks to an enhanced reconstruction loss.
3) The use of the voice style transfer system on speech embeddings extracted using a state of the art feature extractor [11].
4) The proposition of an evaluation protocol for template reconstruction attacks on text-independent speaker verification systems.

In section II, we expose related works about template reconstruction attacks and voice style transfer systems. Section III presents the experimental data. Section IV introduces the threat model and the attack scenario that we propose, while section V presents implementation details about the systems used in that scenario. In section VI we present the experiments and their associated results. Finally, section VII concludes and discusses possible future works.

II. Related Work

A. Template reconstruction attacks

Modern biometrics rely on neural network based templates to encode the identity of a user, as well as other features, such as linguistic content [10] for speech analysis. [3], [8], [13] expose vulnerabilities of such authentication systems by reconstructing face images, handwritten digits and iris images from templates. Our present work is inspired from [3], a template reconstruction attack that uses stolen templates and the associated black-box feature extractor, also referred to as the **encoder**. An artificially generated set of deep face templates is used to train a **decoder** to reconstruct faces images from their associated embeddings. Used on stolen face templates, this **decoder** succeeds in approximating the original face images. Such images used in a spoofing setting achieve 87.37% of Attack Success Rate for a 1% False Acceptation Rate threshold.

Our work differs from [3] as we work with speech, on text-independent automatic speaker verification ($\mathbf{ASV}$) systems. It was already shown that $\mathbf{ASV}$ systems can be fooled with adversarial examples [14]. Contrary to face images, speech utterances carry not only identity characteristics of the speaker, but also linguistic content. Due to the temporal pooling layer of modern speaker verification front end [9], most of the content information is lost during feature extraction [10]: this is not surprising considering linguistic content is not so discriminative between speakers. Therefore it is not possible to regenerate speech from embeddings as efficiently as face images without any prior linguistic content information.

In our work, we do not use artificially generated data to train our **decoder**, but real data, from a distinct set of speakers available to the attacker. In the case of voice biometrics, the speech **encoder** is a text-independent $\mathbf{ASV}$ system (see section V-A), and as the **decoder** a suitable voice style transfer system (see section V-B). While we choose to focus only on the identity characteristics in this work, we will have to consider the problem of speech intelligibility for the reconstruction of audible speech in future works.

B. Zero-shot voice style transfer

Voice conversion [15] consists in transforming a speech utterance of a speaker to another utterance that sounds like another speaker with its linguistic content preserved [16]. The first voice conversion systems were **parallel** [17]–[19], meaning they used audio sequences from different speakers with the same linguistic content. Then, **non-parallel** systems started to get more attention [20]–[23], trained on samples containing different sentences for different speakers. One major difficulty of the voice conversion task is to generalize well to unseen speakers, to produce efficient conversion from and/or to unseen speakers. Recently, few **zero-shot** voice style transfer ($\mathbf{VST}$) systems emerged [12], [24], [25], **zero-shot** meaning they can transfer the voice from and to previously unseen speakers. The authors of [12] view the **voice conversion** problem as a **voice style transfer** problem, where the vocal qualities can be regarded as styles, and speakers as domains.

AutoVC [12] works as an auto-encoder that disentangles identity from linguistic content of Mel-spectrograms, to reconstruct segments uttered by a source speaker in the style of a target speaker represented by a speaker embedding.

To perform a template reconstruction attack, we need a decoder able to reconstruct original Mel-spectrograms from templates of a target speaker. **AutoVC** theoretically allows to generate spoofing Mel-spectrograms by taking any source Mel-spectrogram and reconstructing it with the style of a target speaker, a speaker never seen by the decoder during its training.

In this paper, we optimize **AutoVC** for speaker verification spoofing. In particular, we extend the reconstruction loss for embedding stability, we work with state-of-the-art speaker embeddings and we use utterance-related speaker embeddings, instead of one single average embedding per speaker.

III. Data

Experimental data is composed of files from three different datasets: *VoxCeleb 1* [26], *VoxCeleb 2* [27] and *VCTK* [28], presented in table I.

TABLE I
DATASETS USED.

Dataset	Role	Speakers
VoxCeleb 1 [26]	$\mathbf{Template}$	1251
VoxCeleb 2 [27]	$\mathbf{ASV_{train/valid/test}}$	5994
VCTK [28]	$\mathbf{VST_{train/valid/test}}$	110

The *VoxCeleb 2* [27] and *VCTK* [28] sets are split to constitute training, validation and testing subsets to train respectively the $\mathbf{ASV}$ and $\mathbf{VST}$ systems. The speech utterances of 10% of the speakers are selected for evaluation. 10% of the speech utterances of the remaining speakers constitute the validation sets, and the remaining utterances are used for training. *VoxCeleb 1* [26] is fully used as a testing set. The table II details the different subsets used[1].

TABLE II
DETAILED SPLIT OF THE DIFFERENT SETS USED.

Datasets	From	Speakers	Files	alias
$\mathbf{Template}$	*VoxCeleb 1* [26]	1251	148642	**T**
$\mathbf{ASV_{train}}$	*VoxCeleb 2* [27]	5395	939766	
$\mathbf{ASV_{valid}}$	*VoxCeleb 2* [27]	5395	104419	
$\mathbf{ASV_{test}}$	*VoxCeleb 2* [27]	600	1547	
$\mathbf{VST_{train}}$	*VCTK* [28]	100	34733	**A**
$\mathbf{VST_{valid}}$	*VCTK* [28]	100	3860	
$\mathbf{E_{VST_{Test}}}$	*VCTK* [28]	10	3671	

For any given dataset of speech utterances $\mathbf{S}$, let $\mathbf{M_S}$ be the set of Mel-spectrograms computed from its files:

$$\mathbf{M_S} := \{m_i^u \mid \forall u, i \in [1, N_{uttr}] \times [1, N_{speaker}]\} \quad (1)$$

[1]Exact distribution available on https://github.com/Dretse/Spoofing_speaker_verif_datasets

m_i^u being the u^{th} Mel-spectrogram of the dataset, uttered by the i^{th} speaker.

Let $\mathbf{E_S}$ be the associated embedding set, computed using the black box **ASV** system:

$$\mathbf{E_S} := \{e_i^u = \mathbf{ASV}(m_i^u) \mid \forall m_i^u \in \mathbf{M_S}\} \quad (2)$$

Let $\widehat{\mathbf{M_S}}$ be the Mel-spectrogram spoofing set generated by the **VST** system:

$$\widehat{\mathbf{M_S}} := \{\widehat{m_i^v} = \mathbf{VST}(e_i^u, m_j^v) \mid (e_i^u, m_j^v) \in \mathbf{E_S} \times \mathbf{M_A}\} \quad (3)$$

Finally, let $\widehat{\mathbf{E_S}}$ be the associated spoofing embedding set, computed using the black box **ASV** system:

$$\widehat{\mathbf{E_S}} := \{\widehat{e_i^u} = \mathbf{ASV}(\widehat{m_i^u}) \mid \forall \widehat{m_i^u} \in \widehat{\mathbf{M_S}}\} \quad (4)$$

In the previous equations, $\mathbf{S}$ can be replaced by any of the datasets from the table II. For better readability, we will refer to the **Template** set (resp. the $\mathbf{VST_{train}}$ set) as the $\mathbf{T}$ set (resp. the $\mathbf{A}$ set).

IV. Proposed attack scenario

A. Threat model

We propose a template reconstruction attack of a text-independent **ASV** system [11]. As detailed in section II-A, we follow the threat model proposed for deep face templates in [3], but change the encoder and decoder because of the different nature of the data.

We suppose the attacker has an unlimited black box access to the encoder, an **ASV** feature extractor, meaning he can compute a speaker embedding for the Mel-spectrogram of his choice. This **ASV** system was trained beforehand on the $\mathbf{ASV_{train}}$ set (see table II)

Using the notations developed in the section III, we suppose the attacker stole the embedding set $\mathbf{E_T}$, but doesn't have access to its counter part $\mathbf{M_T}$. The attacker also has access to external sets of speech utterances from distinct speakers, from which he can compute Mel-spectrograms and associated embeddings, respectively the $\mathbf{M_A}$ and $\mathbf{E_A}$ datasets.

B. The attack scenario

As the attacker, we aim to generate Mel-spectrograms $\widehat{m_i^u} \in \widehat{\mathbf{M_T}}$ to spoof the **ASV** system, thanks to the stolen $\mathbf{E_T}$ set. To achieve this, we train a zero-shot **VST** system with $\mathbf{M_A}$ and $\mathbf{E_A}$ to reconstruct Mel-spectrograms as if they were uttered by another speaker represented by a known embedding. We use both **ASV** and **VST** systems, detailed in section V, and databases of speech utterances that are detailed in section III. This attack scenario is composed of three steps (illustrated by yellow numbers in the figure 1), where the attacker:

1) Computes the embeddings $\mathbf{E_A}$ from the Mel-spectrogram database $\mathbf{M_A}$ using the black box **ASV** system.
2) Trains a **VST** system with the Mel-spectrograms set $\mathbf{M_A}$ and the embedding set $\mathbf{E_A}$.
3) Uses the trained **VST** system to generate spoofing Mel-spectrograms $\widehat{\mathbf{M_T}}$ from source Mel-spectrograms $\mathbf{M_A}$, targeting the speaker embeddings of $\mathbf{E_T}$.

For the evaluation, we compute the embeddings $\widehat{\mathbf{E_T}}$ from the spoofing Mel-spectrograms $\widehat{\mathbf{M_T}}$ and we score them against the stolen embeddings $\mathbf{E_T}$ using cosine similarity.

Those steps are illustrated with colored numbers in the figure 1. The **ASV** system is black because the attacker can only access it as a black box, and $\mathbf{M_T}$ is masked because the attacker does not have any access to it. The steps 1-3 are sufficient to execute the attack, but the steps 4-5 allow us to evaluate its efficiency.

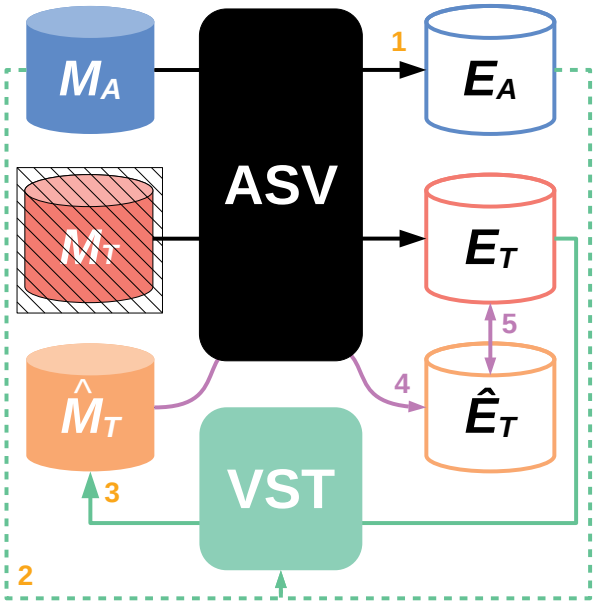

Fig. 1. Illustration of the systems and data used. The **VST** system is represented in green. Numbered steps of the attack are in yellow, evaluation steps are in purple. Schematic best viewed in color.

V. Implementation details

A. Automatic Speaker Verification

The **ASV** system is based on the *fastResNet34* architecture of [11]. It is designed to process a given Mel-spectrogram $\mathbf{m_i^u}$, u^{th} utterance of a speaker $\mathbf{i}$ into an embedding $\mathbf{e_i^u}$ of dimension $N = 256$, as noted in 5.

$$\mathbf{ASV}(\mathbf{m_i^u}) = \mathbf{e_i^u} \mid \mathbf{e_i^u} \in \mathbb{R}^N \quad (5)$$

The **ASV** is trained beforehand on $\mathbf{ASV_{train}}$ and the encoder weights are frozen for the experiments. It achieves an EER of 0.85% on the $\mathbf{E_A}$ set, and an EER of 2.31% on the $\mathbf{E_T}$ set.

B. Voice Style Transfer

The **VST** is inspired from the architecture described in [12]. Figure 2 details the complete processing pipeline for its training. It works as an Auto-Encoder.

The encoder processes the concatenation of a Mel-spectrogram $\mathbf{m_i^u}$ and its associated embedding $\mathbf{e_i^u}$, representing a source speaker $\mathbf{i}$. It compresses them into a carefully chosen bottleneck to filter out the identity part of the Mel-spectrogram and only let the linguistic content through.

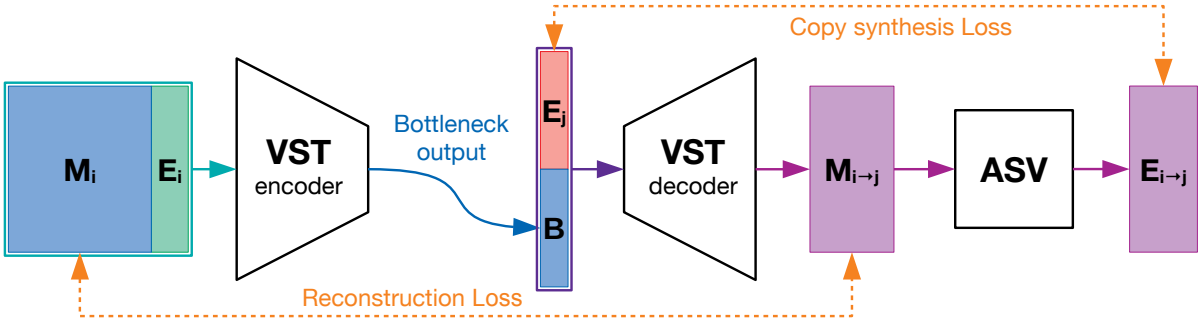

Fig. 2. Schematic of the training of the **VST** system. Losses are in Orange (copy synthesis loss is used for same source and target speakers, aka i=j). Schematic best viewed in color.

Then the decoder takes the concatenation of bottleneck layer output and the target embedding $\mathbf{e_j^v}$ of the speaker $\mathbf{j}$, and computes a Mel-spectrogram of same dimensions as the input ($\widehat{\mathbf{m}}_{\mathbf{i->j}}^{\mathbf{u}}$, as shown in equation 6).

$$\begin{aligned}\widehat{\mathbf{m}}_{\mathbf{i->j}}^{\mathbf{u}} &= \mathbf{VST}(\mathbf{m_i^u}, \mathbf{e_i^u}, \mathbf{e_j^v}) \\ &= \mathbf{VST}(\mathbf{m_i^u}, \mathbf{ASV}(\mathbf{m_i^u}), \mathbf{e_j^v}) \\ &= \mathbf{VST}(\mathbf{m_i^u}, \mathbf{e_j^v})\end{aligned} \tag{6}$$

Originally [12], the average embedding of a speaker was used ($\mathbf{e_i} := \frac{1}{N_{uttr}} \sum_{u=1}^{N_{uttr}} \mathbf{e_i^u}$), both for the source and target embedding. In our implementation, we use the embedding associated with each particular Mel-spectrogram ($\mathbf{e_i^u}$). This slight change underlines its practical use as a decoder: using only a given embedding from a given speaker $\mathbf{j}$ and a Mel-spectrogram $\mathbf{m_i^u}$ to reconstruct an output as if it was uttered by that target speaker.

Similarly to [12], our **VST** system is trained using the copy synthesis technique: using same source and target speakers and Mel-spectrograms, corresponding to equation (6) with $i = j$. When source and target speakers are identical, the target Mel-spectrogram is supposed to be identical to the source one, so we use the Mean Squared Error (MSE) between both as a first loss to minimize, as shown in equation 7.

$$\mathcal{L}_{spect} = ||\widehat{\mathbf{m}}_{\mathbf{i->i}}^{\mathbf{u}} - \mathbf{m_i^u}||_2 \tag{7}$$

This loss guides the system to reconstruct realistic Mel-spectrograms.

The output Mel-spectrogram ($\widehat{\mathbf{m}}_{\mathbf{i->j}}^{\mathbf{u}}$) is, after training, supposed to contain the linguistic content of the source spectrogram $\mathbf{m_i^u}$ with only the identity of the target speaker $\mathbf{j}$. A way to figure out the identity associated to the output Mel-spectrogram is to compute its associated embedding $\widehat{\mathbf{e}}_{\mathbf{i->j}}^{\mathbf{u}}$ and compare it to $\mathbf{e_j^u}$. $\widehat{\mathbf{e}}_{\mathbf{i->j}}^{\mathbf{u}}$ is supposed to represent the target speaker $\mathbf{j}$ in the embedding space. Thus, to help the **VST** system achieve this correct representation, we propose to minimize the MSE between $\widehat{\mathbf{e}}_{\mathbf{i->j}}^{\mathbf{u}}$ and $\mathbf{e_j^u}$, as shown in equation 8.

$$\mathcal{L}_{emb} = ||\widehat{\mathbf{e}}_{\mathbf{i->j}}^{\mathbf{u}} - \mathbf{e_i^u}||_2 \tag{8}$$

The training aims to minimize the global loss $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_{spect} + \mathcal{L}_{emb} \tag{9}$$

Results in the use of the Voice Style Transfer system for the template reconstruction attack are detailed in the section VI-C.

VI. Protocol Experiments

A. Metrics

We chose to use three metrics to evaluate the efficiency of the **VST** system, all applied to the spoofing embeddings $\widehat{\mathbf{E}_\mathbf{T}}$.

If we consider the source identity of a given spoofing embedding $\widehat{\mathbf{e}}_{\mathbf{i->j}}^{\mathbf{u}}$ (here the speaker $\mathbf{i}$), we can compute the EER_{source}. For this EER computation, the cosine score between every pair of distinct embeddings (as defined in equation 10) is a match if $i = a$ and non-match otherwise.

$$score(\widehat{\mathbf{e}}_{\mathbf{i->j}}^{\mathbf{u}}, \widehat{\mathbf{e}}_{\mathbf{a->b}}^{\mathbf{v}}) = \frac{\widehat{\mathbf{e}}_{\mathbf{i->j}}^{\mathbf{u}} \cdot \widehat{\mathbf{e}}_{\mathbf{a->b}}^{\mathbf{v}}}{||\widehat{\mathbf{e}}_{\mathbf{i->j}}^{\mathbf{u}}|| \cdot ||\widehat{\mathbf{e}}_{\mathbf{a->b}}^{\mathbf{v}}||} \tag{10}$$

Then, considering the target identity of the spoofing embeddings, we can compute the EER_{target} with the cosine score between every pair of distinct embeddings (as defined in 10), being a match if $j = b$ and non-match otherwise.

The analysis of those two metrics already gives a good understanding of the efficiency of the **VST** system. A low EER_{source} means that we can still discriminate the Mel-spectrogram outputs according to the source identity, thus the system has not learned how to properly disentangle the identity part of the source Mel-spectrograms from the content part. A low EER_{target} means that the outputs have a distribution in the embedding space that can be well split according to their target identities. However, a low EER_{target} does not assure that the reconstructed embeddings $\widehat{\mathbf{E}_\mathbf{T}}$ are close to the original ones $\mathbf{E_T}$, which is the goal of a template reconstruction attack.

To address this issue, we introduce an other metric: the **Attack Success Rate** (**ASR**). We want to evaluate the reconstructed embeddings $\widehat{\mathbf{E}_\mathbf{T}}$, to know how much of them would be accepted as produced by the target user if scored by the authentication system against the original embeddings $\mathbf{E_T}$. To compute the **ASR**, we first set a threshold that gives a False Acceptation Rate of $\mathbf{n}$% on the $\mathbf{E_T}$ set. The Attack Success Rate for a $\mathbf{n}$% threshold is referred to as $\mathbf{ASR_n}$. If $\mathbf{n}$ is equal to the **EER** of the $\mathbf{E_T}$ set, then we write: $\mathbf{ASR_{EER}}$. The original EERs for both test sets being around 1%, we will

compute Attack Success Rates for 1% and 0.1% ($\mathbf{ASR_1}$ and $\mathbf{ASR_{0.1}}$). Every metric is computed on the test sets $\mathbf{E_T}$ and $\mathbf{E_{VST_{Test}}}$.

B. Configurations explored

All systems are trained with a learning rate of 10^{-4}. We use the Adam optimizer [29] and train our **VST** system over 500 epochs (around 40 hours on a single *GTX 1080* GPU). The Mel-spectrograms are extracted from random 2 seconds segments of *.wav* files sampled in 16kHz. Mel-spectrograms are computed with the Sidekit toolkit [30], using 64ms Hanning windows with a 16ms shift, for 80 Mel filters, frequencies between 90 and 7600Hz.

Before using the complete **VST** system, the attack is performed with the targeted speaker embeddings only, in section VI-C1, equivalent as using only the **VST** decoder, to underline the need of its encoder part and the source Mel-spectrogram.

Then, the **VST** system is trained as suggested in [12], to constitute a baseline attack with performances shown in the first line of table III. This AutoVC [12] system is trained with the initial hyper-parameters proposed (batch size of **2**, bottleneck of **16**), using the average embeddings for each speaker.

Then we explored the hyper-parameters space to get the best results (notably batch size of 64 for the rest of the configurations) and trained the **VST** system using the embedding associated with each Mel-spectrogram instead of the average embedding of each speaker, as explained in V-B. The results for the attack using that optimized **VST** system are presented in the line 2 of the table III. To measure the impact of the bottleneck dimension, we compiled that experiment for bottlenecks from 1 to 128, to show the side effects for too wide or narrow ones. The results are presented in the section VI-C2. Then, we added our new loss on embedding reconstruction to that optimal configuration (line 3). A new search over the local hyper-parameters space did not give better results after adding this new loss.

C. Results

1) Decoder only: To measure the impact of the information transmitted through the bottleneck, we train a **VST** system with a bottleneck of **0**, meaning we only trained the **VST** decoder. This is equivalent to perform an adversarial attack [14] on the black-box **ASV** system. After training, the system got an EER_{source} of 48.15% and an EER_{target} of 18.51% on the $\mathbf{E_{VST_{Test}}}$ set. Those results show that reconstructing accurate original Mel-spectrograms from speaker embeddings alone is not possible. Thus the use of a source Mel-spectrogram in a **VST** setting plays an important role in our attack.

2) Bottleneck comparisons: The figure 3 shows comparative results for different bottlenecks than the one used previously (**16**), to show the evolution of the results for extremely wide and narrow bottlenecks. We can see the impact of a too large bottleneck where the EER_{target} skyrockets while the EER_{source} drops quickly, meaning the network keeps too much information about the source speakers, so it does not care about target speakers information for Mel-spectrogram reconstruction.

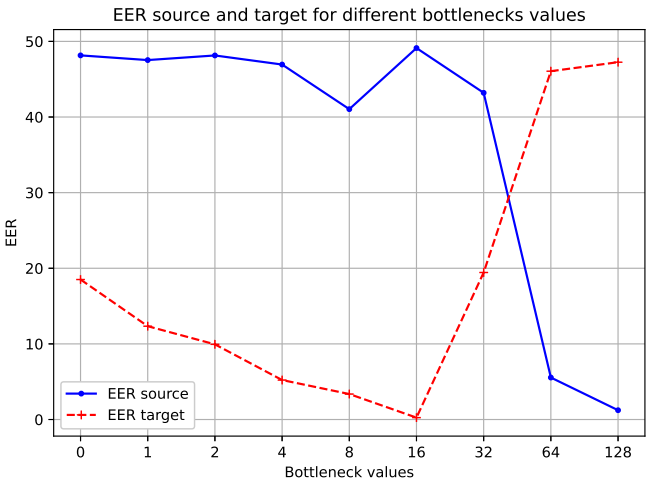

Fig. 3. Graph of the EER_{source} and EER_{target} for different bottlenecks values, computed on the $\mathbf{E_{VST_{Test}}}$ set.

3) The optimal scenario: The results for the different configurations are exposed in the table III. EER_{target} and EER_{source} are computed on the $\mathbf{E_{VST_{Test}}}$ set, while ASR_1 and $ASR_{0.1}$ are computed on the $\mathbf{E_{VST_{Test}}}$ and the $\mathbf{E_T}$ sets. Sets and metrics are respectively presented in the sections III and VI-A. In this table, we can see that the original AutoVC [12] configuration is not fine-tuned for the purpose of this attack, because of the EER and ASR results shown line 1. Once we improve the hyper-parameters (line 2), we obtain better EER results (lower EER_{target} and higher EER_{source}) but not enough for the attack to work correctly (11.27% on a 10 speakers dataset is close to random). Adding the reconstruction loss (line 3), we succeed in getting a functional template reconstruction attack, with up to 99.74% ASR_1 (99.04% $ASR_{0.1}$) on the *VCTK* test dataset and 60.07% ASR_1 (0.93% $ASR_{0.1}$) on the *VoxCeleb 1* dataset.

4) Related works comparisons: Finally, in the table III, we compared our results to other templates reconstruction attacks [3], [8]. Line 4 presents the best type-II attack result from [3]. The type-II attack consists in scoring different templates from the same speaker against each other, which is how we evaluated the previous configurations. Line 5 presents the performances obtained using a black box access to the feature extractor in [8] (from table 1). The result line 5 is given for an ASR_{EER}, meaning for a threshold at the EER, as the EER of the handwritten digit verification system is much higher than 1% [5].

D. Further exploration

1) Mel-spectrograms visualisation: In the figure 4, we plot spoofing Mel-spectrograms targeting embeddings from the $\mathbf{E_{VST_{test}}}$ and the $\mathbf{E_T}$ sets, from random source Mel-spectrograms belonging to $\mathbf{M_A}$. Both were generated by our

TABLE III

TABLE OF THE EER AND ASV RESULTS FOR DIFFERENT EXPERIMENTS. VST BASELINE IS FOR UNMODIFIED **VST** SYSTEM FROM [12]. LINES 2 AND 3 REFER TO THE SYSTEM IMPROVED FOLLOWING SECTION V-B DETAILS. LINES 4 AND 5 COMPARE TO KNOWN TEMPLATE RECONSTRUCTION ATTACKS WITH SIMILAR CONDITIONS ON OTHER BIOMETRIES [3], [8]. FOR LINES 1 TO 4 $ASR_{EER} \approx ASR_1$.

		Rec. loss	Dataset	EER_{source}	EER_{target}	ASR_{EER}	$ASR_{0.1}$
1	Baseline [12]		$\mathbf{E}_{\mathbf{VST_{Test}}}$	38.27%	26.23%	0.23%	0.08%
			$\widehat{\mathbf{E}_{\mathbf{T}}}$	45.50%	9.51%	0.01%	0%
2	Optimized AutoVC		$\mathbf{E}_{\mathbf{VST_{Test}}}$	30.24%	23.50%	11.27%	1.75%
			$\widehat{\mathbf{E}_{\mathbf{T}}}$	37.24%	6.48%	0.63%	0%
3	Optimized AutoVC	✓	$\mathbf{E}_{\mathbf{VST_{Test}}}$	**49.12%**	**0.25%**	**99.74%**	**99.04%**
			$\widehat{\mathbf{E}_{\mathbf{T}}}$	**46.15%**	**1.10%**	**60.07%**	**0.93%**
	Comparisons					ASR_{EER}	$ASR_{0.1}$
4	ASR from [3] (Type-II)		Face images			87.37%	58.05%
5	ASR from [8]	✓	Handwritten digits			87.48%	

best **VST** system. We can see that the Mel-spectrogram

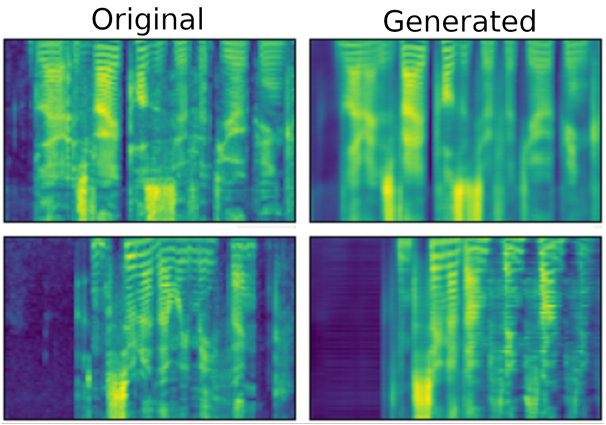

Fig. 4. Original and generated Mel-spectrograms, for target embeddings from the $\mathbf{E}_{\mathbf{VST_{test}}}$ (line 1) and the $\mathbf{E_T}$ (line 2) sets, using random source Mel-spectrograms from $\mathbf{M_A}$.

generated using a target embedding from $\mathbf{E}_{\mathbf{VST_{test}}}$ (top right in the figure) is fully reconstructed, with some blurred artifacts. However, the Mel-spectrogram on the bottom right contains repetitive patterns towards the end, a common error we found on almost all Mel-spectrograms generated using target embeddings from the $\mathbf{E_T}$ set. This difference probably explains the performance differences between both sets: the system is less efficient for a dataset out of the training domain.

2) Limitations: While the proposed approach achieves interesting Attack Success Rates, we underline the following:

- The attack is performed on Mel-spectrograms, a vocoder should be used to complete the attack with waveforms.
- Performances significantly drop when executed on a out of domain test set. This issues could be addressed using additional datasets for **VST** training (such as *VoxCeleb*).
- **VST** training makes it dependant to the **ASV** system, cross system attacks are worth investigating (e.g. training the **VST** system on a different feature extractor than the one targeted for the attack).
- We only consider 2 seconds speech segments for our experimental work, while in actual ASV systems, embeddings are computed on variable length utterances.

VII. CONCLUSION AND FUTURE WORK

This paper introduces a template reconstruction attack on an automatic text-independent speaker verification system [11], with a black box access to the feature extractor, in the event of a voice embeddings database theft. The attack aims to reconstruct Mel-spectrograms from a stolen embedding set in order to spoof the **ASV** system. This attack is inspired by the template reconstruction attack on face images of [3]. The main difference here being that speech utterances not only carry the identity of the speaker (extracted by the **ASV**, contained in the stolen embeddings), but also a linguistic content, where the stolen embeddings are supposed to contain mainly [10] the identity part. To reconstruct realistic Mel-spectrograms, we use a zero-shot voice style transfer system [12] that takes a Mel-spectrogram from any speaker and a voice embedding from a target speaker, and generates a spoofing Mel-spectrogram with the same linguistic content as the input, but as if it was uttered by the speaker of the represented by the target embedding. In order to improve the results on this task, we added a reconstruction loss on the embeddings computed from the reconstructed Mel-spectrograms.

We evaluate our attack on two test datasets, containing users distinct from the training and validation sets of the **ASV** and **VST** systems. On the first one ($\mathbf{E\_ASV_{test}}$, in domain data for the **VST** system), we spoof the speaker verification system with an attack success rate of up to 99.74% (for a false acceptation rate on the original embeddings at 1%). On the second one ($\mathbf{E_T}$, in domain data for the **ASV** system), we spoof the system for up to 60.07% of attack success rate. This difference is probably linked to the different recording conditions of the two sets.

These results show that template protection strategies for voice-based biometrics are required. The emergence of deep representations in biometric systems show that privacy issues need to be properly addressed. One promising outcome of the present work could be the use of zero-shot voice style transfer methods for speaker anonymization. Another one could be to help improve voice anti-spoofing mechanisms.

In the future, we plan to go beyond Mel-spectrogram reconstruction and focus on the intelligibility of reconstructed speech.

REFERENCES

[1] A. K. Jain, K. Nandakumar, and A. Nagar, "Biometric template security," *EURASIP Journal on advances in signal processing*, vol. 2008, pp. 1–17, 2008.

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

[3] G. Mai, K. Cao, P. C. Yuen, and A. K. Jain, "On the reconstruction of face images from deep face templates," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 5, pp. 1188–1202, 2018.

[4] W. Yang, S. Wang, J. Hu, G. Zheng, and C. Valli, "Security and accuracy of fingerprint-based biometrics: A review," *Symmetry*, vol. 11, no. 2, p. 141, 2019.

[5] G. Le Lan and V. Frey, "Securing smartphone handwritten pin codes with recurrent neural networks," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2612–2616.

[6] A. Lumini and L. Nanni, "An improved biohashing for human authentication," *Pattern recognition*, vol. 40, no. 3, pp. 1057–1065, 2007.

[7] A. Kong, K.-H. Cheung, D. Zhang, M. Kamel, and J. You, "An analysis of biohashing and its variants," *Pattern recognition*, vol. 39, no. 7, pp. 1359–1368, 2006.

[8] T. Thebaud, G. Le Lan, and A. Larcher, "Handwritten digits reconstruction from unlabelled embeddings," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 2540–2544.

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

[10] D. Raj, D. Snyder, D. Povey, and S. Khudanpur, "Probing the information encoded in x-vectors," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 726–733.

[11] J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," *arXiv preprint arXiv:2003.11982*, 2020.

[12] K. Qian, Y. Zhang, S. Chang, X. Yang, and M. Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5210–5219.

[13] J. Galbally, A. Ross, M. Gomez-Barrero, J. Fierrez, and J. Ortega-Garcia, "Iris image reconstruction from binary templates: An efficient probabilistic approach based on genetic algorithms," *Computer Vision and Image Understanding*, vol. 117, no. 10, pp. 1512–1525, 2013.

[14] F. Kreuk, Y. Adi, M. Cisse, and J. Keshet, "Fooling end-to-end speaker verification with adversarial examples," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 1962–1966.

[15] D. G. Childers, K. Wu, D. Hicks, and B. Yegnanarayana, "Voice conversion," *Speech Communication*, vol. 8, no. 2, pp. 147–158, 1989.

[16] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," *Journal of the Acoustical Society of Japan (E)*, vol. 11, no. 2, pp. 71–76, 1990.

[17] K. Kobayashi, T. Hayashi, A. Tamamori, and T. Toda, "Statistical voice conversion with wavenet-based waveform generation." in *Interspeech*, 2017, pp. 1138–1142.

[18] L.-J. Liu, Z.-H. Ling, Y. Jiang, M. Zhou, and L.-R. Dai, "Wavenet vocoder with limited training data for voice conversion." in *Interspeech*, 2018, pp. 1983–1987.

[19] K. Chen, B. Chen, J. Lai, and K. Yu, "High-quality voice conversion using spectrogram-based wavenet vocoder." in *Interspeech*, 2018, pp. 1993–1997.

[20] K. Qian, Z. Jin, M. Hasegawa-Johnson, and G. J. Mysore, "F0-consistent many-to-many non-parallel voice conversion via conditional autoencoder," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6284–6288.

[21] H. Kameoka, T. Kaneko, K. Tanaka, and N. Hojo, "Acvae-vc: Non-parallel many-to-many voice conversion with auxiliary classifier variational autoencoder," *arXiv preprint arXiv:1808.05092*, 2018.

[22] ——, "Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 266–273.

[23] Y. Saito, Y. Ijima, K. Nishida, and S. Takamichi, "Non-parallel voice conversion using variational autoencoders conditioned by phonetic posteriorgrams and d-vectors," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5274–5278.

[24] Y. Rebryk and S. Beliaev, "Convoice: Real-time zero-shot voice style transfer with convolutional network," *arXiv preprint arXiv:2005.07815*, 2020.

[25] S. Yuan, P. Cheng, R. Zhang, W. Hao, Z. Gan, and L. Carin, "Improving zero-shot voice style transfer via disentangled representation learning," *arXiv preprint arXiv:2103.09420*, 2021.

[26] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[27] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *INTERSPEECH*, 2018.

[28] C. Veaux, J. Yamagishi, K. MacDonald *et al.*, "Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit," *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 2016.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] A. Larcher, K. A. Lee, and S. Meignier, "An extensible speaker identification sidekit in python," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5095–5099.