

Vocoder drift in x-vector–based speaker anonymization

Michele Panariello, Massimiliano Todisco, Nicholas Evans

EURECOM, Sophia Antipolis, France

firstname.lastname@eurecom.fr

Abstract

State-of-the-art approaches to speaker anonymization typically employ some form of perturbation function to conceal speaker information contained within an x-vector embedding, then resynthesize utterances in the voice of a new pseudo-speaker using a vocoder. Strategies to improve the x-vector anonymization function have attracted considerable research effort, whereas vocoder impacts are generally neglected. In this paper, we show that the impact of the vocoder is substantial and sometimes dominant. The vocoder drift, namely the difference between the x-vector vocoder input and that which can be extracted subsequently from the output, is learnable and can hence be reversed by an attacker; anonymization can be undone and the level of privacy protection provided by such approaches might be weaker than previously thought. The findings call into question the focus upon x-vector anonymization, prompting the need for greater attention to vocoder impacts and stronger attack models alike.

Index Terms: speaker anonymization, automatic speaker verification, privacy

1. Introduction

The task of *speaker anonymization* broadly refers to the processing of speech recordings to conceal the speaker identity while preserving the linguistic and paralinguistic content. The topic has attracted increasing research interest in recent years, in particular through the VoicePrivacy Challenge [1, 2], which was founded in 2020 to define the problem, provide strong baselines, foster progress and identify research priorities. No matter what the application, anonymization should protect an appropriate trade-off between *privacy* and *utility*. Privacy can be estimated using automatic speaker verification (ASV) and an equal error rate (EER) metric to gauge the ability of an attacker to infer the true speaker identity. Utility is estimated using automatic speech recognition (ASR) and a word error rate (WER) metric which reflects the degree to which linguistic and paralinguistic content is preserved.

Most anonymization solutions are based upon original work [3] and upon the extraction and processing of three different representations [4]:

- a set of linguistic features produced by an ASR model;
- a representation of intonation and prosody, usually in the form of a fundamental frequency (F0) curve;
- an *x-vector*, namely a neural embedding which encodes the speaker identity [5].

To conceal the speaker identity, the x-vector is typically perturbed by means of an anonymization function, thereby obtaining a new *pseudo-speaker* embedding. The three components

are then fed to a waveform synthesis model (a *vocoder*) to produce an utterance in the voice of the pseudo-speaker.

The anonymization function used by two of the three VoicePrivacy baselines utilizes a *pool* of external x-vectors. The pseudo-speaker x-vector is derived from a subset of the furthest vectors in the pool from the input x-vector. Most VoicePrivacy participants focused predominantly upon improving the anonymization function to enhance privacy [6, 7, 8]. This focus can imply an assumption that no other processing stages contribute substantially to anonymization. We have found this not to be the case.

We report in this paper our work to observe and compare the relative impacts of a conventional x-vector anonymization function and a vocoder, two components of a state-of-the-art anonymization system [9]. We show that both components contribute to anonymization and that the contribution of the vocoder, which we refer to as the *vocoder drift*, is in some cases even greater than that of the anonymization function. We demonstrate that this phenomenon is also common to other popular vocoders. Collectively, they fail to provide the level of fine-grained control over the input/output x-vector space that would otherwise justify the focus within the community upon the anonymization function. Finally, we show that the vocoder drift can be learned by an attacker, knowledge which can be exploited in order to reverse the anonymization. Our findings corroborate other evidence [10] that the protection provided by such approaches to anonymization might be overestimated.

2. Relation to prior work

In this section we describe the typical, high-level structure of an x-vector–based speaker anonymization system (see Figure 1), along with relevant prior work. We then introduce our own setup which we used for all experiments reported in Section 3 and Section 4.

2.1. X-vector–based speaker anonymization

Let $\mathbf{s} \in \mathbb{R}^L$ be an input speech utterance of L samples. The input is first frame-blocked into a sequence of N frames and then decomposed into three separate representations comprising: an F0 curve $\mathbf{f} \in \mathbb{R}^N$ which is intended to encode intonation and prosody; a set of c -dimensional linguistic features $\mathbf{G} \in \mathbb{R}^{c \times N}$ which encode the spoken content (the text); an x-vector $\mathbf{x}_o \in \mathbb{R}^m$ which encodes the speaker identity, where subscript o denotes extraction from an *original* input utterance.

A vocoder model $V(\mathbf{f}, \mathbf{G}, \mathbf{x}_o)$ is trained to reconstruct input waveforms from the decomposition. Anonymization is achieved by replacing \mathbf{x}_o with a substitute so as to conceal the speaker identity, but by using the other components unchanged in order to preserve remaining speech attributes. The substit-

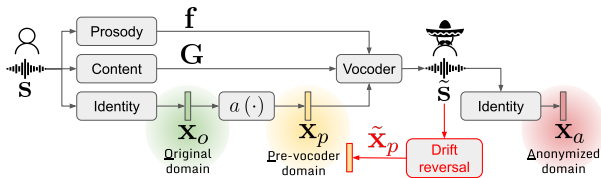


Figure 1: Overview of a conventional speaker anonymization system and the different x-vector domains. The block in red represents the vocoder drift reversal attack reported in Section 4.

tion is performed using an anonymization function $a(\mathbf{x}_o) = \mathbf{x}_p \in \mathbb{R}^m$ to perturb the original x-vector. An anonymized utterance \tilde{s} in the voice of a fictitious, pseudo-speaker determined by the anonymized x-vector \mathbf{x}_p is then synthesized according to $\tilde{s} = V(\mathbf{f}, \mathbf{G}, \mathbf{x}_p)$. The anonymized utterance should maintain the same linguistic and paralinguistic content as the original input signal. As discussed later, an additional x-vector \mathbf{x}_a can be extracted from \tilde{s} in order to measure privacy.

By convention, $a(\cdot)$ acts to create a new pseudo-speaker using speaker embeddings drawn from an external pool of x-vectors [1, 2, 3, 4, 9, 10]. Given an input \mathbf{x}_o , the K vectors within the pool that are furthest from \mathbf{x}_o according to some distance metric are selected and then, from among them, K^* vectors are chosen randomly and averaged to obtain \mathbf{x}_p . The design of this function has received considerable attention, with numerous works having investigated how its configuration, the choice of distance metric [11] and the strategy by which x-vectors are selected from the pool [11, 12] influence performance. The participants of the two VoicePrivacy Challenges held in 2020 and 2022 proposed different enhancements to $a(\cdot)$. They include the generation of pseudo-speaker embeddings using a generative adversarial network [6, 13] and adversarial noise [7], among others [8, 14]. None of the participants reported the influence of the vocoder. In this paper, we show that it too contributes to anonymization and that it can be responsible for a great deal of the privacy protection.

2.2. Our approach

Our approach is based on the pipeline described in [9].¹ The F0 curve is estimated using YAAPT [15]. The linguistic feature extractor is a HuBERT-based soft content encoder [16] and x-vectors are extracted using ECAPA-TDNN [17]. We experimented with three vocoders: the HiFi-GAN [18], originally used in [9]; the neural source filter (NSF) model [19] as used by baseline B1a of the VoicePrivacy Challenge held in 2022; a variation of the HiFi-GAN which uses a NSF model as generator, as used by baseline B1b of the same VoicePrivacy Challenge edition [2]. We use the conventional pool-based anonymization function $a(\cdot)$ described above with $K = 200$, $K^* = 100$, and a cosine distance metric.

Like most related work, we use the VoicePrivacy database and standard protocols [2]. The *LibriTTS-train-clean-100* dataset is used for vocoder training. The *LibriSpeech-test-clean* and *VCTK* datasets (decomposed into male and female subsets) are used for evaluation. The external pool of x-vectors is derived using the *LibriTTS-train-other-500* [20] dataset. Privacy is evaluated using ASV experiments comprising a set of enrollment utterances that an attacker attempts to match to a set of

¹Code available at github.com/eurecom-asp/vocoder-drift.

	target	drift		
		HiFi-GAN	NSF	HiFi-NSF
LibriSpeech (F)	1.3	0.62	0.91	0.97
LibriSpeech (M)	1.2	0.56	0.80	0.94
VCTK (F)	1.3	0.67	0.92	0.94
VCTK(M)	1.3	0.59	0.90	0.90

Table 1: Average target distance and drift for each vocoder and each test set of LibriSpeech and VCTK, separated by speaker sex. All cosine distances have a standard deviation between 0.05 and 0.10.

protected (anonymized) trial utterances. ASV is performed by scoring x-vectors with the cosine distance and without any additional backend processing [9, 17].

3. Vocoder drift

In this section, we introduce the notion of *vocoder drift* and report an investigation of its impact upon x-vector perturbation and privacy.

3.1. Definition

Figure 1 shows the three x-vectors used in this work. The first \mathbf{x}_o is extracted from the original utterance s (left in Figure 1). A second x-vector \mathbf{x}_a can be extracted from the anonymized utterance \tilde{s} (right). The third x-vector \mathbf{x}_p is the output of the anonymization function (middle). We denote the separate domains of \mathbf{x}_o , \mathbf{x}_a and \mathbf{x}_p as \hat{O} (original), \hat{A} (anonymized) and \hat{P} (pre-vocoder), respectively.

As described in Section 2, the majority of research has focused on improving the anonymization function $a(\cdot)$, the general hypothesis being that this component is primarily responsible for ensuring privacy. Intuitively, privacy is improved by increasing the difference between \mathbf{x}_o and \mathbf{x}_p , e.g. according to the cosine distance. With the focus being upon the anonymization function, there is an inherent, perhaps unrealistic assumption that the vocoder preserves this distance such that the difference, which we term as the *drift*, between the x-vectors at the input (\mathbf{x}_p) and that which can be extracted from the output (\mathbf{x}_a) is only modest. In this work, we seek to test this assumption.

We model the relationship between the \hat{P} and \hat{A} domains with a function $v(\mathbf{x}_p) = \mathbf{x}_a$. It allows us to define the trajectory of an x-vector through the whole anonymization system as $v \circ a : \mathbf{x}_o \mapsto \mathbf{x}_a$, where \circ denotes function composition.

3.2. X-vector perturbation

In seeking to quantify the impact of $v(\cdot)$ on the x-vector trajectory, we define two metrics. Let d be some distance measure over \mathbb{R}^m . We then define:

- $d(\mathbf{x}_o, \mathbf{x}_p)$ as the *target* distance, a measure of how far \mathbf{x}_o is perturbed away from its original position according to $a(\cdot)$;
- $d(\mathbf{x}_p, \mathbf{x}_a)$ as the *vocoder drift*, a measure of the shift between the input x-vector \mathbf{x}_p and that which can be extracted from the vocoder output \mathbf{x}_a , introduced by means of $v(\cdot)$.

Intuitively, it is desirable that $\text{drift} \ll \text{target}$, which means the anonymization system provides fine-grained control over the final position of \mathbf{x}_a : it is close to the targeted pseudo-speaker embedding \mathbf{x}_p . If this is not the case, then the x-vector trajectory is determined in considerable part by $v(\cdot)$; the x-vector extracted from the output \mathbf{x}_a is far from the target and the system does not provide fine-grained control over the x-vector space in \hat{A} .

	\hat{O} dom.	\hat{P} dom.	\hat{A} dom.		
			HiFi-GAN	NSF	HiFi-NSF
LibriSpeech (F)	0.54	2.51	15.0	17.9	16.2
LibriSpeech (M)	0.88	2.99	14.5	20.3	19.0
VCTK (F)	1.13	5.59	25.3	31.0	28.1
VCTK(M)	0.17	3.04	18.5	16.7	19.1

Table 2: Privacy protection of the x-vector domains at different stages of the anonymization pipeline (EER, %) on test sets of LibriSpeech and VCTK, separated by speaker sex.

We compute the average *drift* and *target* for each database subset and each vocoder: results are shown in Table 1. The *target* is in the order of 1.3 for all four subsets. The value of these distances lies in their *comparison* to estimates of the *drift* shown in the last three columns. For the HiFi-GAN vocoder, the drift is almost half the target distance. Lying between 0.8 and 0.97, the drift for the NSF and HiFi-NSF vocoders is substantially greater still, with drift distances almost as large as target distances. These results show that the control over the x-vector domain \hat{A} is potentially low and suggest that the x-vector anonymization and vocoder functions have an almost-comparable contribution to x-vector perturbation. It is still necessary, however, to explore their resulting impact upon *privacy*.

3.3. Impacts upon privacy

We follow the VoicePrivacy-defined approach to measure privacy impacts. We report a set of ASV experiments using different combinations of x-vectors. In all cases, privacy is measured using estimates of the EER. Enrollment and trial utterances are as defined by the VoicePrivacy protocol (see Section 2.2). There are several enrollment utterances per speaker. Individual x-vectors are extracted from each, averaged, and compared to a number of trial utterances. For each utterance, we extract the set of \mathbf{x}_o , \mathbf{x}_p and \mathbf{x}_a x-vectors. Each set of experiments is conducted three times, with each iteration using one of the three different x-vectors. Results using the set containing \mathbf{x}_o x-vectors (\hat{O} domain) provide a baseline. Those derived from the set of \mathbf{x}_p x-vectors (\hat{P} domain) provide an indication of the contribution to privacy of the anonymization function $a(\cdot)$. Results using final set containing \mathbf{x}_a x-vectors (\hat{A} domain) provide an indication of the contribution of the vocoder function $v(\cdot)$. Once again, we report results for the same experiment using all three vocoders.

Results are shown in Table 2, for the same database subsets as in Table 1. Baseline results for the \hat{O} domain show EERs of approximately 1%. In the \hat{P} domain, increases in the EER to between 2.5% and 5.6% indicate that the anonymization function delivers only a low level of privacy. In the \hat{A} domain, however, EERs are substantially higher for all three vocoders, if still far from providing perfect privacy (EERs of 50%). The comparison of results for \hat{P} and \hat{A} domains show that the vocoder plays a dominant role; most of the anonymization can be attributed to vocoder drift. We explored this phenomenon with t-SNE visualizations [21] of pooled x-vectors. Results are illustrated in Figure 2, which depicts a distribution of x-vectors for the male partition of the LibriSpeech dataset. In the \hat{P} domain, speaker clusters are still clearly distinguishable, while the bulk of the anonymization can be attributed to vocoder drift.

One could claim that these findings are neither surprising, nor cause for concern. There is no guarantee that the vocoder function $v(\cdot)$ is invertible in any way which would allow the

recovery of x-vector inputs \mathbf{x}_p in the \hat{P} domain. Since the attacker does not have access to the \hat{P} domain, but only to the \hat{A} domain, whether anonymization is attributed to the anonymization function or the vocoder function is of little consequence. In the next section, we disprove these arguments and show that an attacker *can* learn this function or, more specifically, how to undo it. Armed with the inverse function $v^{-1}(\cdot)$, an attacker can estimate an x-vector in the \hat{P} domain that corresponds to an x-vector in the \hat{A} domain and hence *reverse* the anonymization.

4. Drift-reversal attacks

In this section, we introduce *drift reversal*, a novel attack against anonymization systems.

4.1. Attacks on anonymization systems

Since speaker anonymization is a relatively new research topic, it is hardly surprising that little attention has been dedicated to attacks against it. Even so, the VoicePrivacy Challenge has explored the robustness of anonymization systems under a so-called *semi-informed* attack model [2]. Under this scenario, an attacker is aware of anonymization having been performed, and seeks to overcome it (break the anonymization) by using a similar system to generate anonymized data with which to train an ASV system. Evaluations using ASV systems trained using in-domain (similarly anonymized) data show the potential for attacks to circumvent anonymization. A more explicit approach is reported in [10] and can be used by an attacker to invert a complete anonymization system by means of a rotation matrix and to estimate speaker embeddings \mathbf{x}_o in the unprotected domain \hat{O} from protected x-vectors \mathbf{x}_a in \hat{A} . Our approach is different since we aim to explore the anonymization robustness when we revert only the *vocoder drift* to recover an estimate of \mathbf{x}_p in \hat{P} .

4.2. Definition and implementation

In the case that the bulk of the anonymization performance can be attributed to the vocoder function $v(\cdot)$ instead of the anonymization function $a(\cdot)$, a drift reversal attack can be mounted to undo most of the protection Let $\mathbf{s}^{(e)}$ be an original (i.e. unprotected) enrollment utterance. An attacker can derive a representation of this signal in the \hat{P} domain by extracting an x-vector $\mathbf{x}_o^{(e)}$ and then by computing $a(\mathbf{x}_o^{(e)}) = \mathbf{x}_p^{(e)}$. Now let $\tilde{\mathbf{s}}^{(t)}$ be an anonymized trial utterance with corresponding x-vector $\mathbf{x}_a^{(t)}$ in the \hat{A} domain. The attacker can estimate a representation in the \hat{P} domain $\mathbf{x}_p^{(t)}$ by reversing the vocoder drift, i.e. by computing $v^{-1}(\mathbf{x}_a^{(t)})$.

While the inverse function is not analytically tractable, the attacker can attempt to *learn* a function $g_\theta(\cdot) \approx v^{-1}(\cdot)$ using a database of training pairs $\{\mathbf{x}_{p_i}, \tilde{\mathbf{s}}_i\}_i$. Function g_θ can be learned using a neural network to map an anonymized utterance $\tilde{\mathbf{s}}$ to an approximation of the corresponding x-vector \mathbf{x}_p in \hat{P} . This can be achieved by optimizing the objective function

$$\min_{\theta} d(\mathbf{x}_p, g_\theta(\tilde{\mathbf{s}})) \quad (1)$$

where d is the cosine distance. Training pairs $\{\{\mathbf{x}_{p_i}, \tilde{\mathbf{s}}_i\}\}_i$ can be obtained by applying anonymization to *any* appropriate (even unlabeled) speech dataset.

Because function g_θ is effectively an x-vector extraction operation, we fine-tune a pretrained ECAPA-TDNN model to learn it. In line with the VoicePrivacy protocol, the model

Vocoder	Dataset	Unprotected	Lazy informed	Semi informed	Drift reversal
HiFi-GAN	LibriSpeech (F)	0.54	11.3	3.21	3.10
	LibriSpeech (M)	0.88	10.9	1.78	4.45
	VCTK (F)	1.13	19.2	13.3	7.53
	VCTK (M)	0.17	11.0	8.14	3.79
NSF	LibriSpeech (F)	0.54	15.3	1.92	5.05
	LibriSpeech (M)	0.88	13.8	1.94	6.04
	VCTK (F)	1.13	24.7	15.7	16.5
	VCTK (M)	0.17	9.81	12.3	10.7
HiFi-NSF	LibriSpeech (F)	0.54	12.6	4.01	4.23
	LibriSpeech (M)	0.88	14.7	2.23	4.90
	VCTK (F)	1.13	22.8	18.4	14.1
	VCTK (M)	0.17	13.5	11.7	11.1

Table 3: Performance of the proposed drift-reversal attack compared to a lazy-informed attack and a supervised semi-informed attack (EER, %) on the LibriSpeech and VCTK test sets.

is trained using the *LibriSpeech-train-clean-360* dataset, although approximately 3% of the data is set aside for validation purposes. Still in line with the VoicePrivacy protocol, anonymization is performed at the *speaker level*² in deriving \mathbf{x}_a for each enrollment and trial utterance, instead of at the utterance level. The network is fine-tuned for 3 epochs using Adam optimizer [22] with a learning rate of $5 \cdot 10^{-5}$ and a batch size of 8. Validation is performed every 200 iterations. Attacks are performed using the network for which the validation loss is lowest.

4.3. Evaluation

We compare the drift reversal attack to related VoicePrivacy *lazy-informed* and *semi-informed* attacks. For the former, the attacker compares enrollment and trial utterances which are both in the \hat{A} domain, but with an ASV model trained using data in the \hat{O} domain; other than by anonymizing the enrollment utterance, there is no compensation for operating upon anonymized data. The *semi-informed* attacker makes greater effort and uses an ASV system that is trained using an independent set of similarly-anonymized data. The latter is the default VoicePrivacy attack model.³ The *lazy-informed* attack is implemented using the original, pretrained ECAPA-TDNN for x-vector extraction. The *semi-informed* attack is performed using an ECAPA-TDNN model which is fine-tuned using AAM-softmax loss [23] and the same training settings as the drift reversal attack model.

Privacy evaluation results in terms of EER estimates are presented in Table 3 for each vocoder and each dataset. EER results for unprotected data (no anonymization) are shown in column 3 and provide a reference against which EERs for protected data can be compared. Results for the *lazy-informed* attack are shown in column 4 and show substantial privacy gains (higher EERs). This setting, however, gives a false sense of protection. Results for the *semi-informed* attack shown in column 5 show considerably lower privacy gains; by retraining the ASV

² \mathbf{x}_{p_i} is estimated once for each speaker i and the same \mathbf{x}_{p_i} is used for each utterance corresponding to the same speaker – see [2] for details.

³It could be argued that drift reversal is also a *semi-informed* attack, since it involves re-training a model on anonymized data (albeit unlabeled). However, for clarity, we use the term *semi-informed* to refer to the attack method used in the VoicePrivacy Challenge 2022.

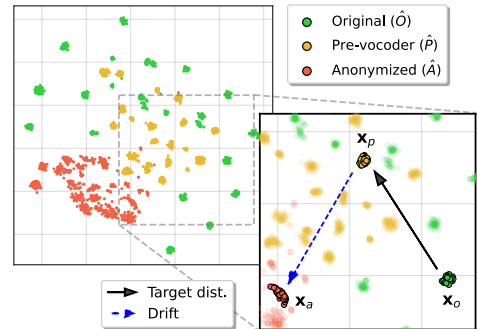


Figure 2: *t*-SNE visualization of the x-vector trajectory for LibriSpeech trial utterances (M) across the three x-vector domains (left). Focus on the trajectory of a single speaker (right). Best viewed in color.

system using similarly anonymized data, the attacker can undo the anonymization to some degree. Results for the drift reversal attack also show universally lower privacy gains compared to the lazy-informed attack. These results add to the evidence that the role played by vocoder drift in anonymization is substantial and is also a potential weakness that can be exploited by an adversary. The most powerful of the 3 attacks for each vocoder and dataset is highlighted in bold face and, for 5 of the 12 cases, the most powerful attack is drift reversal.

5. Conclusions

The work presented in this paper shows that, for the analyzed systems, the bulk of anonymization can be attributed not to the anonymization function of a conventional x-vector-based approach but, instead, to the vocoder function. The cause is vocoder drift, namely the substantial difference between an input x-vector and the x-vector which can be extracted from the vocoder output. This finding, while not necessarily surprising, calls into the question the research effort upon improving the anonymization function. One might wonder whether the design of different anonymization functions has any relevance at all, given that the position of the final x-vector is dominated by the vocoder drift, essentially nullifying the effort devoted to pseudo-speaker optimization. This finding should not discourage further work in the design of x-vector anonymization functions, however. Instead, it should encourage design toward more grounded criteria.

Drift-reversal attacks rely on the fact that the x-vectors fed to the vocoder, though allegedly anonymized, still have a low level of protection. This is the result of an over-deterministic anonymization function; similar x-vector inputs will produce similar x-vector outputs, thus producing trial and enrollment speaker embeddings which are close in the output domain, and thus easy to match as the same speaker, even when anonymized. That is the case for the pool-based anonymization function. Future work should investigate less deterministic anonymization functions to improve privacy directly in their output domain. Improvements to privacy in this domain will not only undoubtedly mitigate the risk of vocoder-drift-reversal attacks, but likely also that of *semi-informed* attacks, which might inadvertently learn to exploit the same kind of vulnerability during training.

6. References

- [1] N. Tomashenko, X. Wang, E. Vincent, J. Patino, B. M. L. Srivastava, P.-G. Noé, A. Nautsch, N. Evans, J. Yamagishi, B. O'Brien, A. Chanclu, J.-F. Bonastre, M. Todisco, and M. Maouche, "The voiceprivacy 2020 challenge: Results and findings," *Computer Speech & Language*, vol. 74, p. 101362, 2022.
- [2] N. Tomashenko, X. Wang, X. Miao, H. Nourtel, P. Champion, M. Todisco, E. Vincent, N. Evans, J. Yamagishi, and J.-F. Bonastre, "The voiceprivacy 2022 challenge evaluation plan," *arXiv preprint arXiv:2203.12468*, 2022.
- [3] F. Fang, X. Wang, J. Yamagishi, I. Echizen, M. Todisco, N. Evans, and J.-F. Bonastre, "Speaker Anonymization Using X-vector and Neural Waveform Models," in *Proc. 10th ISCA Speech Synthesis Workshop*, 2019, pp. 155–160.
- [4] B. M. L. Srivastava, N. Tomashenko, X. Wang, E. Vincent, J. Yamagishi, M. Maouche, A. Bellet, and M. Tommasi, "Design Choices for X-Vector Based Speaker Anonymization," in *Proc. Interspeech 2020*, 2020, pp. 1713–1717.
- [5] 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.
- [6] S. Meyer, P. Tilli, F. Lux, P. Denisov, J. Koch, and N. T. Vu, "Cascade of phonetic speech recognition, speaker embeddings gan and multispeaker speech synthesis for the VoicePrivacy 2022 Challenge," in *Proc. 2nd Symposium on Security and Privacy in Speech Communication*, 2022.
- [7] J. Yao, Q. Wang, L. Zhang, P. Guo, Y. Liang, and L. Xie, "NWPU-ASLP System for the VoicePrivacy 2022 Challenge," in *Proc. 2nd Symposium on Security and Privacy in Speech Communication*, 2022.
- [8] X. Chen, G. Li, H. Huang, W. Zhou, S. Li, Y. Cao, and Y. Zhao, "System description for Voice Privacy Challenge 2022," in *Proc. 2nd Symposium on Security and Privacy in Speech Communication*, 2022.
- [9] X. Miao, X. Wang, E. Cooper, J. Yamagishi, and N. Tomashenko, "Language-independent speaker anonymization approach using self-supervised pre-trained models," in *Proc. The Speaker and Language Recognition Workshop (Odyssey 2022)*, 2022, pp. 279–286.
- [10] P. Champion, T. Thebaud, G. Le Lan, A. Larcher, and D. Juvet, "On the invertibility of a voice privacy system using embedding alignment," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 191–197.
- [11] B. M. L. Srivastava, M. Maouche, M. Sahidullah, E. Vincent, A. Bellet, M. Tommasi, N. Tomashenko, X. Wang, and J. Yamagishi, "Privacy and utility of x-vector based speaker anonymization," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, Jun. 2022.
- [12] P. Champion, D. Juvet, and A. Larcher, "Evaluating x-vector-based speaker anonymization under white-box assessment," in *Speech and Computer*, A. Karpov and R. Potapova, Eds. Cham: Springer International Publishing, 2021, pp. 100–111.
- [13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., vol. 27. Curran Associates, Inc., 2014.
- [14] U. E. Gaznepoglu, A. Leschanowsky, and N. Peters, "VoicePrivacy 2022 system description: speaker anonymization with feature-matched f0 trajectories," in *Proc. 2nd Symposium on Security and Privacy in Speech Communication*, 2022.
- [15] K. Kasi and S. A. Zahorian, "Yet another algorithm for pitch tracking," in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 2002, pp. I-361–I-364.
- [16] B. van Niekerk, M.-A. Carboneau, J. Zaïdi, M. Baas, H. Seuté, and H. Kamper, "A comparison of discrete and soft speech units for improved voice conversion," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6562–6566.
- [17] B. Desplanques, J. Thienpondt, and K. Demuyne, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Proc. Interspeech 2020*, 2020, pp. 3830–3834.
- [18] J. Kong, J. Kim, and J. Bae, "Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 17 022–17 033.
- [19] X. Wang, S. Takaki, and J. Yamagishi, "Neural Source-Filter Waveform Models for Statistical Parametric Speech Synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 402–415, 2020.
- [20] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech," in *Proc. Interspeech 2019*, 2019, pp. 1526–1530.
- [21] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.
- [22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.
- [23] X. Xiang, S. Wang, H. Huang, Y. Qian, and K. Yu, "Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2019, pp. 1652–1656.