Vocoder drift compensation by x-vector alignment in speaker anonymisation

Michele Panariello, Massimiliano Todisco, Nicholas Evans

EURECOM, Sophia Antipolis, France

firstname.lastname@eurecom.fr

Abstract

For the most popular x-vector–based approaches to speaker anonymisation, the bulk of the anonymisation can stem from vocoding rather than from the core anonymisation function which is used to substitute an original speaker x-vector with that of a fictitious pseudo-speaker. This phenomenon can impede the design of better anonymisation systems since there is a lack of fine-grained control over the x-vector space. The work reported in this paper explores the origin of so-called vocoder drift and shows that it is due to the mismatch between the substituted x-vector and the original representations of the linguistic content, intonation and prosody. Also reported is an original approach to vocoder drift compensation. While anonymisation performance degrades as expected, compensation reduces vocoder drift substantially, offers improved control over the x-vector space and lays a foundation for the design of better anonymisation functions in the future.

Index Terms: anonymisation, pseudonymisation, privacy, vocoder drift, automatic speaker verification

1. Introduction

Speaker anonymisation broadly refers to the task of processing speech recordings to conceal the identity of the speaker while preserving linguistic and paralinguistic content. Recently, the topic has attracted notable research interest, particularly through the VoicePrivacy Challenge [1, 2, 3], first launched in 2020 to define the task and to encourage the development of more effective speaker anonymisation techniques. According to the VoicePrivacy Challenge Evaluation Plan [3], the evaluation of a speaker anonymisation solution is based upon estimates of the trade off between *privacy* (protection of the speaker identity) and *utility* (how well the remaining signal content is preserved). The former is estimated by the ability of an attacker to use automatic speaker verification (ASV) to infer the original speaker identity and is measured in terms of equal error rate (EER). The latter is estimated using the word error rate (WER) of an automatic speech recognition (ASR) system as a proxy for utility.

Currently, the better-performing anonymisation solutions reflect the processing pipeline described in [4] and rely upon an initial decomposition of the input signal into the following three components:

- a set of features representing the linguistic content of the signal, typically in the form of ASR;
- a component representing intonation and prosody, normally a fundamental frequency (F0) curve;
- a neural embedding representing the identity of the speaker, usually an *x-vector*.

To obfuscate the speaker identity, an *anonymisation function* is applied to the x-vector embedding, thereby obtaining a new embedding which represents the voice of a fictitious *pseudo-speaker*. The three components are subsequently fed to a *vocoder* model which synthesises a waveform with the same spoken content and prosody as the original input audio, but in the voice of the substitute pseudo-speaker.

For effective anonymisation, the pseudo-speaker's voice should sound *different* to that of the original speaker. For the majority of anonymisation systems proposed to date, this requirement is fulfilled by maximising some measure of the distance between the chosen pseudo-speaker embedding and the original speaker embedding. The most popular anonymisation function to date uses a *pool* of external x-vectors [2, 3, 4, 5, 6, 7]. The pseudo-speaker embedding is obtained by averaging a random subset of the furthest x-vectors in the pool from the x-vector of the original speaker. More refined methods of pseudo-speaker selection, e.g. based upon the use of singular-value decomposition [8] and generative adversarial networks [9], have also been explored. However, in our previous work [10], we showed that anonymisation performance is influenced by more than just the role of the anonymisation function. The vocoder also plays a role and its impact is comparable to, or even dominates that of the anonymisation function. We termed this phenomenon *vocoder drift*.

While one interpretation of these observations is that vocoder drift contributes positively to anonymisation, and is hence a benefit, another is that it implies a lack of fine-grained control over the x-vector space and that this lack of control in turn impedes the design of effective x-vector anonymisation functions. Moreover, vocoder drift can be learned and reversed to undermine anonymisation safeguards [10].

With the work reported in this paper, we have sought to understand the cause of vocoder drift and how it can be reduced in order to improve control over the x-vector trajectory and full anonymisation process. We show that the cause of drift is related to the mismatch between the distribution of x-vectors used for vocoder training and the distribution after anonymisation. Such a mismatch can be compensated for during anonymisation by aligning the input and output x-vectors of the vocoder via gradient descent.

2. Relation to prior work

In this section, we describe the typical structure of an x-vector–based anonymisation solution, the concept of drift, and other relevant, prior work. We also describe the system we used for the experiments reported in Sections 3 and 4.

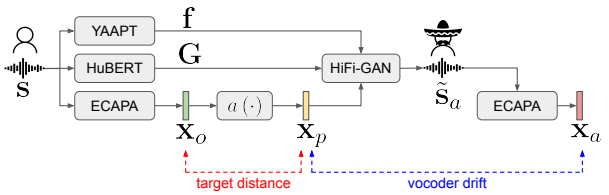

Figure 1: *Diagram of the speaker anonymisation system used in this work. The dashed arrows indicate which x-vectors are used to compute* target distance *and* vocoder drift.

2.1. X-vector–based anonymisation

A graphical overview of a typical x-vector–based approach to anonymisation is shown in Figure 1. Let $\mathbf{s}$ be a speech signal which we seek to anonymise and from which we extract the following components: an F0 curve $\mathbf{f} \in \mathbb{R}^N$, where N is the number of frames into which $\mathbf{s}$ is split; a set of linguistic features $\mathbf{G} \in \mathbb{R}^{c \times N}$, where c is the feature dimension; an x-vector $f(\mathbf{s}) = \mathbf{x}_o \in \mathbb{R}^m$, where m is the embedding dimension and $f(\cdot)$ is the embedding extraction function. The x-vector is duplicated once for each frame, resulting in a matrix $\mathbf{X}_o \in \mathbb{R}^{m \times N}$. The set of features are then concatenated into a final matrix of dimension $(1+c+m) \times N$ and fed to a vocoder model to produce an utterance $\tilde{\mathbf{s}} = \mathrm{V}(\mathbf{f}, \mathbf{G}, \mathbf{X}_o)$.[1]

The vocoder is trained in a self-supervised fashion to reconstruct the original signal. At test time, an input utterance is anonymised by substituting the original speaker embedding $\mathbf{x}_o$ with a pseudo-speaker embedding $\mathbf{x}_p$, which is obtained by means of an anonymisation function $a(\mathbf{x}_o) = \mathbf{x}_p$. The anonymised utterance $\tilde{\mathbf{s}}_a$ is synthesised as $\tilde{\mathbf{s}}_a = \mathrm{V}(\mathbf{f}, \mathbf{G}, \mathbf{X}_p)$, and a further x-vector $\mathbf{x}_a = f(\tilde{\mathbf{s}}_a)$ can then be extracted from it. Thus, as a result of anonymisation, the speaker identity follows an x-vector trajectory, from $\mathbf{x}_o$ to $\mathbf{x}_p$ and then $\mathbf{x}_a$.

2.2. Vocoder drift

In [10], we sought to understand the degree to which the anonymisation function and the speech synthesis procedure impact upon the x-vector trajectory from $\mathbf{x}_o$ to $\mathbf{x}_a$. We did so by measuring how much the x-vector is perturbed during these two steps of the anonymisation pipeline.

The anonymisation function controls the shift from $\mathbf{x}_o$ to $\mathbf{x}_p$. Given a distance metric d, we define $d(\mathbf{x}_o, \mathbf{x}_p)$ as the *target distance*: this quantity is set by the system designer and indicates the desired perturbation which is applied to the original speaker embedding to give the pseudo-speaker embedding. As a result of synthesis, $\mathbf{x}_p$ is further perturbed by the vocoder, giving $\mathbf{x}_a$. We term $d(\mathbf{x}_p, \mathbf{x}_a)$ the *vocoder drift*. To provide fine-grained control over the x-vector space, the impact of drift should be as small as possible in the total trajectory of an individual x-vector. In other words, ideally, $d(\mathbf{x}_o, \mathbf{x}_p) \gg d(\mathbf{x}_p, \mathbf{x}_a)$.

Our work reported in [10] shows that the impact of the anonymisation function and vocoder are comparable and that, in some cases, the bulk of the anonymisation is delivered by the vocoder, not the anonymisation function. In this work, we propose a technique to compensate for vocoder drift.

[1] Henceforth, a bold lowercase $\mathbf{x}$ refers to a single x-vector, while an uppercase $\mathbf{X}$ of the same subscript represents the matrix constructed from the same x-vector duplicated N times.

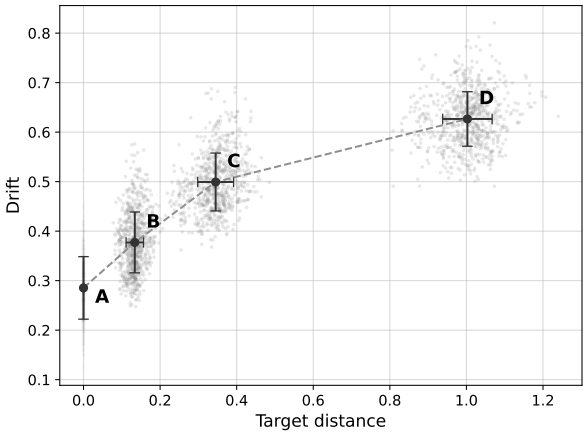

Figure 2: *Vocoder drift plotted against target distance for the LibriSpeech dataset and for female speakers. Dots and bars represent mean and standard deviation of each of the following experimental setups: (A) $\lambda = 0$; (B) $\lambda = 1/3$; (C) $\lambda = 1/2$; (D) $\lambda = 1$.*

2.3. System setup

For all experiments reported in this paper, we use a system which, except for the use of different vocoders, is the same as that described in [10], which is itself inspired by original work in [7]. The F0 contour $\mathbf{f}$ and the linguistic features $\mathbf{G}$ are produced with YAAPT [11] and a HuBERT-based soft content encoder [12], respectively. All x-vectors are extracted with ECAPA-TDNN [13], and the vocoder model is a HiFi-GAN [14]. The anonymisation function $a(\cdot)$ is the same x-vector pool-based averaging function described in Section 1. Given an input $\mathbf{x}_o$, the K x-vectors furthest from it are selected from the pool. K^* of them are then randomly chosen and averaged to obtain $\mathbf{x}_p$. We set $K = 200$, $K^* = 100$, and use the cosine distance metric as in [10]. Following the VoicePrivacy Challenge 2022 [3] protocol, the external x-vector pool is derived from *LibriTTS-train-other-500* [15], and the evaluation sets are derived from the *LibriSpeech-test-clean* [16] and VCTK [17] (split into female and male sub-partitions) datasets. For consistency with $a(\cdot)$, the target distance and vocoder drift are also measured in terms of the cosine distance.

3. The cause of vocoder drift

In this section, we describe what we believe to be the source of vocoder drift and present a set of experiments which validate our hypothesis.

3.1. Feature mismatch

As illustrated in Section 2.1, the vocoder model is trained in self-supervised fashion to reconstruct input signals $\mathbf{s}$ at the output. While, ideally, input components $\mathbf{f}$, $\mathbf{G}$ and $\mathbf{x}_o$ should be disentangled from one another – so that none contains any information that is also contained in any other – there is no explicit incentive in the training criterion of any of the three extraction models which would encourage the learning of disentangled representations. Previous work has confirmed that the representations are indeed *entangled* to some extent. For example, results in [18, 19] show that speaker-related information, normally captured in $\mathbf{x}_o$, can leak into linguistic representa-

Table 1: *Average target distance and drift (without and with compensation) for the four different VoicePrivacy Challenge 2022 data subsets and for four different values of λ.*

	$\lambda = 0$ (copy-synthesis)			$\lambda = 1/3$			$\lambda = 1/2$			$\lambda = 1$ (normal anon.)		
	target	drift	drift (compens.)	target	drift	drift (compens.)	target	drift	drift (compens.)	target	drift	drift (compens.)
LibriSpeech (F)	0	0.29	0.047	0.13	0.38	0.049	0.35	0.50	0.054	1.0	0.63	0.052
LibriSpeech (M)	0	0.27	0.047	0.11	0.35	0.048	0.31	0.48	0.051	1.0	0.65	0.052
VCTK (F)	0	0.29	0.049	0.11	0.36	0.051	0.30	0.49	0.084	1.0	0.69	0.082
VCTK (M)	0	0.29	0.049	0.08	0.35	0.049	0.26	0.46	0.062	1.1	0.79	0.078

tions $\mathbf{G}$. The vocoder can hence learn to rely on such mutual dependencies between input features in learning how it should reconstruct $\tilde{\mathbf{s}}$.

Through anonymisation, original speaker embeddings $\mathbf{x}_o$ are substituted by pseudo-speaker embeddings $\mathbf{x}_p$, and used by the vocoder to reconstruct a speech signal using the F0 curve $\mathbf{f}$ and linguistic features $\mathbf{G}$ extracted from the input speech signal corresponding to x-vector $\mathbf{x}_o$. The new pseudo-speaker embedding will hence not *match* any speaker-related information contained in $\mathbf{f}$ and $\mathbf{G}$. This results in a *mismatch* with the data distribution learned by the vocoder at training time. It is our hypothesis that this mismatch is the source of vocoder drift.

We verified our hypothesis with an experiment in which we anonymised a set of utterances $\mathbf{s}$ and computed original x-vectors $\mathbf{x}_o$ and corresponding pseudo-speaker embeddings $a(\mathbf{x}_o) = \mathbf{x}_p$. Then, rather than synthesising new waveforms according to the usual approach $\mathrm{V}(\mathbf{f}, \mathbf{G}, \mathbf{X}_p)$, we compute instead $\mathrm{V}(\mathbf{f}, \mathbf{G}, \mathbf{X}_i)$, where $\mathbf{x}_i$ is an interpolation between $\mathbf{x}_o$ and $\mathbf{x}_p$:

$$\mathbf{x}_i = \mathbf{x}_o + \lambda(\mathbf{x}_p - \mathbf{x}_o) \tag{1}$$

The parameter $\lambda \in [0, 1]$ acts to control the distance between $\mathbf{x}_i$ and either $\mathbf{x}_o$ or $\mathbf{x}_p$. In line with definitions presented in Section 2.2, we term $d(\mathbf{x}_o, \mathbf{x}_i)$ the *target distance*. The target distance can be interpreted to reflect the mismatch between the speaker embedding that would naturally complement $\mathbf{f}$ and $\mathbf{G}$ and the embedding received by the vocoder. By adjusting λ, we conducted a set of anonymisation experiments with increasing target distances, i.e. higher values of λ, equivalent to increasing feature mismatch. For each experiment, we also measure the resulting vocoder drift. A positive correlation between drift and target distance would then suggest that vocoder drift does indeed have some dependency on the mismatch between vocoder input features.

3.2. Experiments and results

We conducted experiments with values of $\lambda = \{0, 1/3, 1/2, 1\}$. In the case of $\lambda = 0$, (1) reduces to $\mathbf{x}_i = \mathbf{x}_o$, which corresponds to the absence of anonymisation (i.e. $a(\cdot)$ is not applied): the system performs copy-synthesis. Conversely, in the case of $\lambda = 1$, (1) reduces to $\mathbf{x}_i = \mathbf{x}_p$: the pseudo-speaker embedding is employed during synthesis as with usual anonymisation. Values of $\lambda = 1/3$ and $1/2$ correspond to different interpolations between $\mathbf{x}_o$ and $\mathbf{x}_p$. We measured the target distance and vocoder drift for all four configurations.

Results are reported in Table 1, which shows the target distance and drift in the first two columns of each set of results for each value of λ. Results are shown separately for LibriSpeech and VCTK datasets and for male and female subsets in both cases. A degree of positive correlation between λ and both the target distance and drift is apparent. For $\lambda = 0$, the target distance is always 0 (since $\mathbf{x}_i = \mathbf{x}_o$) and the drift is consistently in the order of 0.28. For $\lambda = 1/3$, the target distance increases to an average of 0.1 and the drift to an average of 0.36. Both the target distance and drift increase further for higher values of λ: such a correlation is evident when plotting the two metrics against one another for a whole data partition and different values of λ, as in Figure 2. These results show that, the greater the degree of mismatch between input features, the greater is the vocoder drift. This in turn implies that greater target distances incur less control over the x-vector space. However, but not surprisingly, for copy-synthesis when $\lambda = 0$, the drift is still substantial. For this configuration, there is no mismatch in the input features; those used for reconstruction are exactly those extracted from the input signal. This suggests that a component of the drift stems from the intrinsic nature of the waveform reconstruction process. In the following, we report an approach to compensate for the vocoder drift.

4. Drift compensation

Vocoder drift, while advantageous in terms of anonymisation [10], can be undesirable in that it prevents fine-grained control over the x-vector space. Because the impact of the vocoder upon the x-vector space can dominate that of the anonymisation function, this lack of control impedes the design of better anonymisation functions. Hence, even if lower vocoder drift might initially degrade anonymisation performance, it may deliver better control over the x-vector space and then be beneficial to the future development of better anonymisation functions. In this section, we introduce a new technique for vocoder drift compensation. It is based upon the iterative *alignment* of $\mathbf{x}_a$ to $\mathbf{x}_p$ at inference time.

4.1. X-vector alignment

Our goal is to adjust the matrix $\mathbf{X}_i$ so as to reduce the mismatch to $\mathbf{G}$ and $\mathbf{f}$ in order then to reduce vocoder drift. This adjustment can be formulated as an optimisation problem:

$$\mathbf{X}_i^* = \underset{\mathbf{X}_i}{\operatorname{argmin}}\ d\Big(\underbrace{f\big(\mathrm{V}(\mathbf{f}, \mathbf{G}, \mathbf{X}_i)\big)}_{\mathbf{x}_a},\ \mathbf{x}_p \Big) \tag{2}$$

where d is again the cosine distance. In essence, we seek to adjust $\mathbf{X}_i$ so as to minimise the cosine distance between $\mathbf{x}_p$ (the x-vector vocoder input) and $\mathbf{x}_a$ (the x-vector extracted from its output). The resulting, optimised matrix $\mathbf{X}_i^*$ is then used to synthesise an anonymised utterance $\tilde{\mathbf{s}}_a^*$, whose drift-compensated x-vector we denote as $\mathbf{x}_a^*$. We optimise the objective function directly at inference time via gradient descent. With this approach, the drift can be arbitrarily reduced by any desired amount, at the cost of proportionately increasing the computation time required to synthesise the anonymised waveform.

4.2. Experiments and results

We optimise (2) at the utterance level using Adam [20] with a learning rate of $5\mathrm{e}{-3}$. Optimisation runs for a maximum of 150

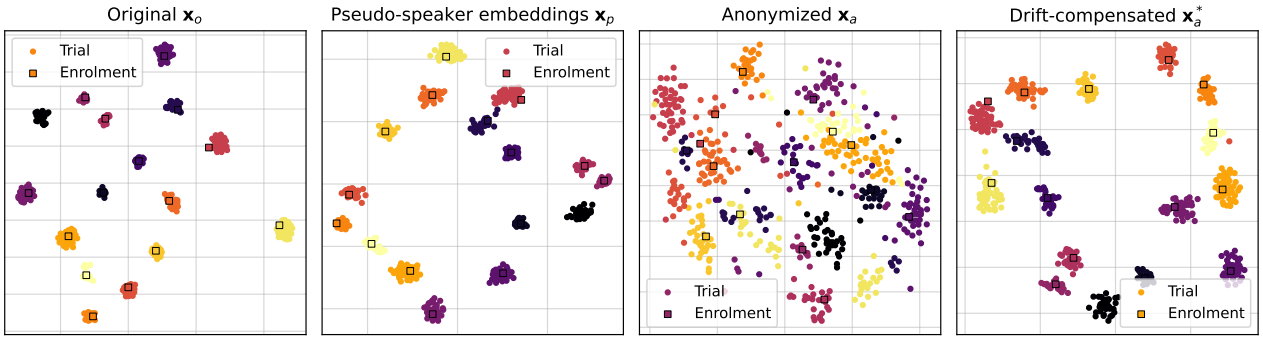

Figure 3: *t-SNE visualisations of four different x-vector spaces and embeddings for the enrolment and trial utterances of the LibriSpeech dataset and female speakers. Different colours correspond to different speakers. From left to right: original x-vectors* $\mathbf{x}_o$*, pseudo-speaker embeddings* $\mathbf{x}_p$*, anonimised embeddings* $\mathbf{x}_a$*, anonymised and drift-compensated embeddings* $\mathbf{x}_a^*$*.*

steps, but stops earlier if the drift falls below 0.05 (set arbitrarily to reduce processing time). The impact of drift compensation is then observed by repeating the experiments described in Section 3 but with $\mathbf{X}_i$ replaced by drift compensated versions $\mathbf{X}_i^*$ and by observing the reduction in vocoder drift.

Results are shown in the third columns of each block in Table 1. Drift compensation reduces the vocoder drift for all values of λ. For $\lambda = \{0, 1/3\}$, 150 optimisation steps are generally sufficient for the drift to reach the lower bound of 0.05, for all datasets. This is also the case for the LibriSpeech dataset for $\lambda = \{1/2, 1\}$. For the VCTK dataset, we obtain drift values of approximately 0.07 — slightly higher than LibriSpeech, yet still considerably lower than the initial vocoder drift. Informal listening tests show that drift compensation introduces no discernible degradation to speech quality — any differences are negligible to the point that signals generated with and without drift compensation are difficult to tell apart.

4.3. Impact upon privacy protection

If the vocoder drift is responsible for the bulk of anonymisation performance, and if drift compensation performs as intended, then the application of drift compensation is expected to result in degraded anonymisation performance. We performed a set of ASV experiments to observe the impact. Experiments were conducted according to the protocol described in the VoicePrivacy Challenge 2022 evaluation plan [3]. For each dataset, the experiment is run four times, each time using one of the set of x-vectors ($\mathbf{x}_o$, $\mathbf{x}_p$, $\mathbf{x}_a$, $\mathbf{x}_a^*$) for each utterance. The results are reported in Table 2.

As expected, low EERs for x-vectors $\mathbf{x}_o$ increase for $\mathbf{x}_p$ and even more noticeably for $\mathbf{x}_a$, indicating the dominant impact of the vocoder upon anonymisation. This is especially evident in VCTK partitions, likely because of a domain mismatch with the HiFi-GAN vocoder which, in accordance with the VoicePrivacy 2022 protocol, is trained on *LibriTTS-train-clean-100*. EERs for x-vectors $\mathbf{x}_a^*$ are close to those of $\mathbf{x}_p$, indicating successful vocoder drift compensation. This result can also be observed visually in Figure 3, which shows t-SNE visualisations [21] of all four x-vector embeddings for the LibriSpeech dataset and female speakers (both trial and enrolment utterances). The effect of drift is clearly visible upon the comparison of the visualisations for $\mathbf{x}_p$ and $\mathbf{x}_a$: in the latter, embeddings are notably more dispersed. The visualisation for $\mathbf{x}_a^*$ shows that drift compensation reduces the dispersion, giving compact clusters once more.

Table 2: *ASV results (EER, %) for VoicePrivacy 2022 test sets, using the same set of different x-vector speaker embeddings as in Figure 3.*

	$\mathbf{x}_o$	$\mathbf{x}_p$	$\mathbf{x}_a$	$\mathbf{x}_a^*$ (comp.)
LibriSpeech (F)	0.54	2.51	15.0	2.75
LibriSpeech (M)	0.88	2.99	14.5	3.34
VCTK (F)	1.13	5.59	25.3	9.20
VCTK(M)	0.17	3.04	18.5	5.23

5. Conclusions

This paper shows that the mismatch between the representations of linguistic information, intonation and prosody and a substitute pseudo-speaker embedding is a source of vocoder drift – the difference between a target x-vector and that which can be extracted from the synthesised output of popular approaches to speaker anonymisation.

While beneficial to anonymisation, vocoder drift can nonetheless be undesirable: it reduces fine-grained control over the x-vector space and hence impedes optimisation of the core anonymisation function. Experiments show that a novel approach to compensate for vocoder drift through the iterative adjustment of pseudo-speaker embeddings to linguistic, intonation and prosodic components is effective in reducing the drift. As expected, however, the loss of vocoder drift degrades anonymisation performance. This result adds further weight to our previous findings that vocoder drift plays a substantial, but only superficial role in anonymisation; the vocoder drift can be learned and undone, or reversed by an adversary.

The anonymisation function remains to be of paramount importance since its impact cannot be, or is at least much more difficult to reverse. The design of better anonymisation functions should hence remain a focus in future work. The alleviation of extraneous influences coming from vocoder drift delivers better control over the x-vector space and hence better potential to design more effective anonymisation functions in the future. This does not preclude the study of disentangled representations or other vocoder schemes which might also offer complementary opportunities to reduce drift and improve control over the x-vector space.

6. References

[1] N. Tomashenko, B. M. L. Srivastava, X. Wang, E. Vincent, A. Nautsch, J. Yamagishi, N. Evans, J. Patino, J.-F. Bonastre, P.-G. Noé, and M. Todisco, "Introducing the VoicePrivacy Initiative," in *Proc. Interspeech 2020*, 2020, pp. 1693–1697.

[2] 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230822000080

[3] 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," 2022.

[4] 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.

[5] P. Champion, D. Jouvet, and A. Larcher, "A study of F0 modification for x-vector based speech pseudo-anonymization across gender," in *The Second AAAI Workshop on Privacy-Preserving Artificial Intelligence (PPAI).*, 2020.

[6] 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.

[7] 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.

[8] C. O. Mawalim, K. Galajit, J. Karnjana, S. Kidani, and M. Unoki, "Speaker anonymization by modifying fundamental frequency and x-vector singular value," *Computer Speech & Language*, vol. 73, p. 101326, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230821001194

[9] S. Meyer, P. Tilli, P. Denisov, F. Lux, J. Koch, and N. T. Vu, "Anonymizing speech with generative adversarial networks to preserve speaker privacy," in *2022 IEEE Spoken Language Technology Workshop (SLT)*, 2023, pp. 912–919.

[10] M. Panariello, M. Todisco, and N. Evans, "Vocoder drift in x-vector-based speaker anonymization," in *Proc. Interspeech 2023*, 2023.

[11] 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.

[12] B. van Niekerk, M.-A. Carbonneau, 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.

[13] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification," in *Proc. Interspeech 2020*, 2020, pp. 3830–3834.

[14] 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.

[15] 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.

[16] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[17] J. Yamagishi, C. Veaux, and K. MacDonald, "CSTR VCTK Corpus: English multi-speaker corpus for CSTR Voice Cloning Toolkit (version 0.92)," 2019. [Online]. Available: https://datashare.ed.ac.uk/handle/10283/3443

[18] C. Pierre, A. Larcher, and D. Jouvet, "Are disentangled representations all you need to build speaker anonymization systems?" in *Proc. Interspeech 2022*, 2022, pp. 2793–2797.

[19] A. S. Shamsabadi, B. M. L. Srivastava, A. Bellet, N. Vauquier, E. Vincent, M. Maouche, M. Tommasi, and N. Papernot, "Differentially private speaker anonymization," *Proceedings on Privacy Enhancing Technologies*, vol. 1, pp. 98–114, 2023.

[20] 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.

[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.