# Differentially Private Speaker Anonymization

Ali Shahin Shamsabadi*
The Alan Turing Institute
Vector Institute
a.shahinshamsabadi@turing.ac.uk

Brij Mohan Lal Srivastava
Université de Lille, Inria, CNRS,
Centrale Lille, UMR 9189 - CRIStAL,
F-59000 Lille, France

Aurélien Bellet
Université de Lille, Inria, CNRS,
Centrale Lille, UMR 9189 - CRIStAL,
F-59000 Lille, France

Nathalie Vauquier
Université de Lille, Inria, CNRS,
Centrale Lille, UMR 9189 - CRIStAL,
F-59000 Lille, France

Emmanuel Vincent
Université de Lorraine, CNRS, Inria,
LORIA, F-54000 Nancy, France

Mohamed Maouche
Université de Lille, Inria, CNRS,
Centrale Lille, UMR 9189 - CRIStAL,
F-59000 Lille, France

Marc Tommasi
Université de Lille, Inria, CNRS,
Centrale Lille, UMR 9189 - CRIStAL,
F-59000 Lille, France

Nicolas Papernot
Vector Institute
University of Toronto
Canada

## ABSTRACT

Sharing real-world speech utterances is key to the training and deployment of voice-based services. However, it also raises privacy risks as speech contains a wealth of personal data. Speaker anonymization aims to remove speaker information from a speech utterance while leaving its linguistic and prosodic attributes intact. State-of-the-art techniques operate by disentangling the speaker information (represented via a speaker embedding) from these attributes and re-synthesizing speech based on the speaker embedding of another speaker. Prior research in the privacy community has shown that anonymization often provides brittle privacy protection, even less so any provable guarantee. In this work, we show that disentanglement is indeed not perfect: linguistic and prosodic attributes still contain speaker information. We remove speaker information from these attributes by introducing differentially private feature extractors based on an autoencoder and an automatic speech recognizer, respectively, trained using noise layers. We plug these extractors in the state-of-the-art anonymization pipeline and generate, for the first time, private speech utterances with a provable upper bound on the speaker information they contain. We evaluate empirically the privacy and utility resulting from our differentially private speaker anonymization approach on the LibriSpeech data set. Experimental results show that the generated utterances retain very high utility for automatic speech recognition training and inference, while being much better protected against strong adversaries who leverage the full knowledge of the anonymization process to try to infer the speaker identity.

## KEYWORDS

speaker anonymization, differential privacy, automatic speech recognition, automatic speaker recognition, voice-based services, privacy

## 1 INTRODUCTION

Recent advances in automatic speech recognition (ASR) [4, 28] have enabled the deployment of voice-based services such as dictation, voice search and voice assistants in our daily life [16]. In these applications, speech is collected by service providers and third-party contractors[1] to process user queries (*inference*), but also before deployment to train ASR systems on real, diverse, annotated speech data (*training*) and thereby achieve state-of-the-art performance [40, 76]. However, speech data is very sensitive, not only through the linguistic content (what is being said) but first and foremost because it is a biometric modality that can identify the speaker [34]. Automatic speaker identification (ASI) and speaker verification (ASV) techniques can identify and distinguish speakers in large populations with low error [10, 32]. Malicious parties with access to the speech data of a victim can impersonate him/her or assemble fake recordings that he/she never said [12]. Disseminating speech data thus entails significant privacy and security risks. The last two years have seen the rise of privacy issues in the agenda of the speech processing community, as evidenced by the creation of a special interest group of the International Speech Communication Association,[2] the launch of the VoicePrivacy initiative [73], and ongoing efforts to understand the requirements of effective privacy preservation for speech data [54] in light of recent regulation [53].

In this work, we are interested in the problem of *speaker anonymization*, which aims to conceal the speaker's identity (privacy) while preserving the linguistic and prosodic (intonation, stress and rhythm) content as well as the diversity of speech (utility). This problem was the focus of the recent VoicePrivacy challenge [74]. A successful speaker anonymization approach enables users to freely share their speech data with service providers for both inference and training purposes, while concealing their identity. It is

*Most of this work was done during the internship of Ali Shahin Shamsabadi at Inria / University of Lille.


*Proceedings on Privacy Enhancing Technologies 2023(1), 98–114*

https://doi.org/10.56553/popets-2023-0007

[1] See, e.g., https://www.bbc.com/news/technology-31296188.
[2] https://www.spsc-sig.org

important to note that, as in the VoicePrivacy challenge, we seek to preserve all linguistic content and do not address the problem of protecting the personally identifiable information that it may contain (e.g., names, addresses, credit card numbers) — this could be done using privacy-preserving ASR methods [3], for instance. Speaker anonymization can benefit many real-world applications beyond ASR-based services. For instance, in France, legal cases can be broadcasted on TV provided that the speakers' voices are transformed so as to prevent re-identification while preserving the linguistic and prosodic content. Call centers which record customer calls are facing the same problem.

Speaker anonymization cannot be addressed by cryptographic solutions as they provide confidentiality (i.e., only the data owner can observe data) [84], not privacy (i.e., protecting what can be inferred about the speaker identity from speech data). For example, cryptographic solutions make human annotation impossible [73]. A naive approach to speaker anonymization consists in transcribing the speech into text using an ASR model followed by a text-to-speech (TTS) system that re-synthesizes speech from the text transcription. While this ASR+TTS approach perfectly conceals the speaker identity, it destroys the utility of speech in several ways: in addition to the incorrect linguistic content induced by (unavoidable) ASR errors, the original prosodic attributes (intonation, stress, and rhythm) are lost, and the variability of synthesized speech output by TTS is very limited, especially when the TTS system can only generate a few voices. Due to this limited diversity, ASR models trained on synthetic speech generated by TTS perform poorly when applied to real speech [15, 44].

By contrast, state-of-the-art speaker anonymization methods seek to separate the speaker identity information from the linguistic and prosodic content so as to generate speech where only the identity information has been removed [24, 68, 69, 73]. These methods rely on the extraction of three types of features from a speech recording: (i) a speaker embedding (typically an x-vector [66] extracted from an intermediate layer of an ASI model) which encodes the characteristics of the speaker's voice, (ii) a sequence of bottleneck (BN) features [81] (a low-dimensional phonetic representation extracted from an intermediate layer of an ASR model) that captures fine-grained linguistic information (as opposed to the possibly erroneous word sequence in the ASR+TTS approach), and (iii) a sequence of pitch features (i.e., the signal's fundamental frequency) which conveys prosodic information [30]. Speaker anonymization is then realized by re-synthesizing speech from the BN and pitch features of the original speech recording and a replaced speaker embedding corresponding to another (real or pseudo) speaker. While this general approach has been quite successful and achieves good practical performance [68], there remains a lot of room for improvement in protecting against concrete attacks [48]. In particular, the disentanglement of speaker information is not perfect: linguistic and prosodic features are known to contain residual identity information [24] which can propagate to the anonymized speech and be used by an adversary to re-identify speakers (see Section 6). Furthermore, the effectiveness of anonymization is evaluated only empirically: even if the evaluation is performed using state-of-the-art ASI or ASV techniques and takes into account some auxiliary information that the adversary may have [70], there is no guarantee that the resulting speech cannot be de-anonymized using better attacks.

In this paper, we propose to use ideas from $\varepsilon$-differential privacy ($\varepsilon$-DP) [20], a rigorous mathematical framework to quantify the information leakage of algorithms, to design more robust speaker anonymization techniques. Following the pipeline of state-of-the-art methods described above, we introduce differentially private pitch and BN feature extractors that can bound the risk of the speaker identity leaking through the prosodic and linguistic attributes used to re-synthetize speech. While it is easy to enforce DP by adding random noise directly to the original features, this naive approach destroys the linguistic and prosodic content that we wish to preserve. Instead, we carefully design machine learning-based pitch and BN feature extractors, and train them to retain the desired information while adding the necessary amount of noise to get DP guarantees via a Laplace noise layer. Specifically, for pitch, we introduce a novel autoencoder (an encoder-decoder network) that learns to reconstruct the input pitch at the output by optimizing an original reconstruction loss function designed to preserve the global pitch dynamics which conveys prosodic information (e.g., pitch increases when asking a question) while the noise perturbs the local variations that are more speaker-specific [2, 17, 49, 58]. For BN features, we train a deep ASR acoustic model to learn features that retain as much as possible the phonetic information needed to decode the linguistic content, while the noise helps to remove the residual speaker information. Regarding the speaker embedding, we simply choose a public x-vector (provided by the VoicePrivacy challenge [68]) randomly and independently of the input utterance so that it does not contain any information about the original speaker, following [69]. Plugging our private feature extractors into the full speaker anonymization pipeline, we obtain a differentially private version of the above state-of-the-art speaker anonymization approach.

Our approach satisfies a rigorous analytical DP guarantee that upper bounds the leakage of the speaker identity by the parameter $\varepsilon$ (the smaller $\varepsilon$, the stronger the privacy guarantee). Being a worst-case measure, DP however gives a conservative privacy guarantee: there is often a large gap with what adversaries can infer in practical settings [33, 35, 52]. To complement our analytical guarantees, we lower bound the leakage of the speaker identity by empirically evaluating the success of concrete adversaries designed to be as close to the worst case as possible given realistic knowledge. We conduct a two-step evaluation of empirical privacy and utility. First, we evaluate the ability of an adversary to re-identify a known speaker from pitch and BN features by training an ASI model directly on these features. Our results show that the features output by our proposed DP extractors provide much better protection than the standard features used in previous work. Second, plugging our feature extractors into the state-of-the-art speaker anonymization technique [69, 73], we show that we can generate speech utterances which empirically preserve better the privacy of speakers (even when using rather large values for $\varepsilon$) at only a small cost in utility. Here, utility is measured by the word error rate (WER) of an ASR system trained and tested on anonymized speech, and privacy by the equal error rate (EER) of an adversary that uses a state-of-the-art ASV system trained on a large corpus of anonymized utterances. Low WER and high EER indicate that the speech generated with our approach can be shared, stored, annotated and used to train ASR models for voice-based services, while protecting the speaker identity.

In summary, our contributions advance the state-of-the-art in speaker anonymization as follows:

- We empirically demonstrate that the BN and pitch features used by current speaker anonymization methods contain a lot of speaker information by mounting an attack in which we train ASI models directly on these features. Our attack achieves 97% and 37% accuracy on BN and pitch features, respectively, on the 921 speakers from the LibriSpeech data set.
- We introduce DP pitch and BN feature extractors that remove the speaker identity while preserving the linguistic and prosodic information. We show that our extractors provide analytical $\varepsilon$-DP guarantees. In addition to this, our DP extractors with $\varepsilon = 1$ reduce the accuracy of the above attack on BN and pitch to 14% and 5%, respectively. We show that our BN features can be shared instead of raw utterances to perform ASR training and inference with negligible effect on the WER. On LibriSpeech, our DP BN extractor with $\varepsilon = 1$ achieves 6% WER, compared to 5% with the original BN features.
- Finally, we synthesize speech from our DP extractors and compare against state-of-the-art speaker anonymization. We evaluate the empirical privacy by mounting an attack in which we train ASV models on anonymized utterances. Our results demonstrate that the state-of-the-art speaker anonymization pipeline still leaks speaker identity information. Remarkably, in addition to its analytical privacy guarantees, our approach provides much better empirical privacy while utility remains very high. On LibriSpeech, our DP speaker anonymization scheme achieves an EER and a WER of 30% and 7%, respectively, whereas the state-of-the-art scheme achieves an EER and a WER of respectively 15% and 5%.

## 2 BACKGROUND

In this section, we introduce background concepts in speech processing and differential privacy.

### 2.1 Speech Processing

Speech data contain speaker information as well as linguistic and prosodic attributes. State-of-the-art speaker anonymization relies on automatic speaker recognition, automatic speech recognition and pitch estimation for extracting speaker information, linguistic and prosodic attributes, respectively. We give an overview of these techniques, which will be used both in the design of our approach and in our empirical evaluation.

**Automatic speaker recognition** is the task of recognizing the speaker of a given speech utterance. Existing techniques can be categorized into automatic speaker verification (ASV), i.e., authenticating the identity claimed by the speaker, and automatic speaker identification (ASI), i.e., determining the identity within a set of known speakers [11]. ASI relies on training a neural network based speaker classifier on utterances from multiple speakers and later using that network to classify the identity of each test utterance as one of the known training identities. As opposed to the closed-set ASI task, ASV is an open-set (rejection/acceptance) task which comprises two successive phases: enrollment and authentication. In the former, a speaker embedding is extracted from one or more enrollment utterances spoken by the speaker whose identity is being claimed. The most popular embeddings called x-vectors [66] are obtained from an intermediate layer of a neural network trained to perform ASI. In the latter phase, the x-vector extracted from the utterance of an unknown speaker (called trial utterance) is compared with the x-vector of the speaker whose identity is being claimed. and a log-likelihood ratio score is computed by probabilistic linear discriminant analysis (PLDA) [41]. The ASV system then decides whether the trial utterance is from that speaker or not by comparing the obtained score with a threshold.

**Automatic speech recognition (ASR)** aims to convert an utterance into its textual content, also called transcription. We describe here the classical monolingual ASR architecture based on separate acoustic and language models, which was used both to extract BN features and to quantify utility in the VoicePrivacy challenge [74]. End-to-end neural ASR architectures and/or multilingual architectures could also be used without loss of generality [13]. The input to ASR is a sequence $\mathbf{O} = [\mathbf{o}_1, \ldots, \mathbf{o}_K]^\top \in \mathbb{R}^{K \times A}$ of length $K$ of acoustic feature vectors $\mathbf{o}_k \in \mathbb{R}^A$ derived from speech, e.g., Mel-frequency cepstral coefficients (MFCCs), and the output is the estimated word sequence $\hat{W}$. This problem can be formulated as [80]

$$\hat{W} = \text{argmax}_W \, P(W|\mathbf{O}). \tag{1}$$

In practice, it is infeasible to directly model the conditional distribution of the true word sequence given the acoustic features. Hence, using Bayes' rule, some independence assumptions and the fact that $\mathbf{O}$ is fixed, the ASR problem is reformulated as [46]

$$\begin{aligned} \hat{W} &= \underset{W}{\text{argmax}} \, P(\mathbf{O}|W)P(W)/P(\mathbf{O}) \\ \hat{W} &= \underset{W}{\text{argmax}} \, P(\mathbf{O}|W)P(W) \\ &= \underset{W}{\text{argmax}} \sum_{S,N} P(\mathbf{O}|S)P(S|N)P(N|W)P(W). \end{aligned} \tag{2}$$

Here $P(W)$ is the so-called language model that represents the prior distribution of word sequences, $P(N|W)$ is the lexicon which maps words to the corresponding phoneme sequences, $P(S|N)$ maps a phoneme sequence to the corresponding triphone (i.e., tied context-dependent phoneme) sequence $S = [S_1, \ldots, S_K]$, and $P(\mathbf{O}|S) \propto \prod_{k=1}^{K} P(S_k|\mathbf{O})/P(S_k)$ where the triphone posterior probabilities $P(S_k|\mathbf{O})$ are given by the so-called acoustic model and $P(S_k)$ is the prior probability of each triphone. These models are trained independently and composed together as a graph using finite state transducers.

**Bottleneck features.** An acoustic model trained for ASR can also be used for other tasks which rely on the phonetic content but do not require a word-level transcription, such as language identification [63] or keyword spotting [77]. In such cases, instead of using the acoustic model output (triphone posterior probabilities), a sequence of phonetic features called bottleneck (BN) features is extracted from an intermediate layer of the acoustic model [81] and used, possibly in combination with other features, as input to these tasks.

**Pitch estimation.** The fundamental frequency (called pitch and denoted as F0) is the frequency of oscillation of the vocal folds. The vocal folds are the flap-like organ at the upper end of the

trachea which controls the air stream emerging from the lungs. The range of pitch is determined by the physiological factors of the vocal folds, such as their mass and length, hence it depends on the speaker and is typically lower for male than female [9]. The pitch sequence governs the *intonation* of the spoken utterance. It is a key component of prosody (together with stress and rhythm) which determines the utterance expressiveness. It is important to note that pitch is the rate of vibration of vocal folds hence it is only defined for voiced phonemes such as /a/, /b/, /z/, etc. It is pointless to compute pitch for silence, noise or unvoiced regions of an utterance since there is no vibration of the vocal folds, hence it is conventionally zero at these locations. Pitch estimation is a difficult task due to erroneous observation of harmonics causing pitch doubling/halving [82]. It is also difficult to estimate the pitch when the quality of speech is distorted due to noise or channel effects. We use a fairly robust and widely used algorithm for pitch tracking called YAAPT [38].

## 2.2 Differential Privacy

Differential Privacy (DP) [20] provides a rigorous probabilistic way to quantify the privacy leakage of an information release process. DP also comes with strong mathematical properties and a powerful algorithmic framework [21]. For these reasons, DP and its variants have become the gold standard notion of privacy in machine learning and many other scientific fields. DP has also seen recent real-world deployments, notably by the US Census [1].

Two main trust models have been considered in DP. The central model assumes the presence of a trusted curator which collects raw data from data owners. In the local model [18, 39], each data owner obfuscates its data locally before sharing it. As we aim to design methods for speakers to anonymize their speech, we place ourselves in the local model. Formally, $\varepsilon$-differential privacy is defined as follows.

Definition 1 (Local Differential Privacy [18, 39]). *Let $\mathcal{A}$ be a randomized algorithm taking as input a data point in some space $\mathcal{X}$, and let $\varepsilon > 0$. We say that $\mathcal{A}$ is $\varepsilon$-differentially private ($\varepsilon$-DP) if for any $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ and any $S \subseteq \text{range}(\mathcal{A})$:*

$$\Pr[\mathcal{A}(\mathbf{x}) \in S] \leq e^{\varepsilon} \Pr[\mathcal{A}(\mathbf{x}') \in S],$$

*where the probabilities are taken over the randomness of $\mathcal{A}$.*

DP essentially requires that the probability of any output does not vary "too much" (as captured by $\varepsilon$) when changing the input. The smaller $\varepsilon$, the stronger the privacy guarantee. In our setting, a data point $\mathbf{x}$ will correspond to a speech utterance and $\mathcal{A}$ will be a speaker anonymization procedure that produces an anonymzed utterance as output. DP then ensures that any two arbitrary utterances will be indistiguinshable (to some extent), and therefore bounds the risk that an adversary observing the output can predict who spoke it.

DP possesses a number of desirable properties that we will use in our work. First, any function of an $\varepsilon$-DP algorithm remains $\varepsilon$-DP (*robustness to post-processing*). Second, one can easily keep track of the privacy guarantees across multiple analyses (*composition*). In particular, given $K$ algorithms that satisfy $\varepsilon$-DP, executing them on the same data and releasing their combined outputs is $K\varepsilon$-differentially private.

A standard way to design differentially private algorithms is based on output perturbation. In this work, we will rely on the so-called Laplace mechanism, which consists in adding Laplace noise calibrated to the $\ell_1$-sensitivity of the (non-private) function one would like to compute on the data.

Definition 2 (Laplace mechanism). *Let $f : \mathcal{X} \to \mathbb{R}^d$ and let the $\ell_1$-sensitivity of $f$ be defined as*

$$\Delta_1(f) = \max_{\mathbf{x},\mathbf{x}' \in \mathcal{X}} \|f(\mathbf{x}) - f(\mathbf{x}')\|_1.$$

*Let $\eta = [\eta_1, \ldots, \eta_d] \in \mathbb{R}^d$ be a vector where each $\eta_i \sim Lap(\Delta_1(f)/\varepsilon)$ is drawn from the centered Laplace distribution with scale $\Delta_1(f)/\varepsilon$. Then, $\mathcal{A}(\cdot) = f(\cdot) + \eta$ is $\varepsilon$-DP.*

In our work, we will use the Laplace mechanism to construct a differentially private transformation of speech utterances for speaker anonymization. It is important to note that DP enforces a stronger notion of privacy than what we aim to achieve: as explained above, it entails hiding the speaker identity but may also suppress other information that we wish to preserve. In particular, adding Laplace noise to raw speech destroys linguistic and prosodic information. Instead, we will show that applying DP at an intermediate layer of carefully designed feature extractors trained to preserve linguistic and prosodic information, we can successfully conceal the speaker identity while retaining the usefulness of anonymized utterances for ASR training and inference. This is in line with other recent work on using DP to hide specific attributes from image [43] and text data [8, 26, 45].

# 3 PROBLEM STATEMENT AND THREAT MODEL

Inspired by the VoicePrivacy challenge [73], we consider a scenario where speakers produce speech utterances that they would like to share with a voice-based service or donate to a public corpus. As discussed before, a raw speech utterance leaks identifying information embedded in the speaker's voice. To mitigate this, our goal is to design a speaker anonymization method which takes as input a speech utterance and satisfies the following properties:

(1) it outputs a speech waveform with the same length as the original speech waveform;
(2) it preserves as well as possible the phonetic and prosodic content of the original utterance (utility);
(3) it conceals as well as possible the identity of the speaker (privacy).

Property 1 allows human annotators to listen to the anonymized speech and annotate it, which is crucial for building useful corpora. The requirement that the speech rate should not be modified comes from the VoicePrivacy challenge [73]. Properties 2 and 3 are conflicting to a certain extent and lead to a classic *privacy-utility trade-off*.

On the one hand, utility can be measured by the performance of an ASR system trained and tested on anonymized utterances. On the other hand, quantifying the level of privacy requires to define a threat model. In this work, we consider adversaries that aim to verify whether a given speaker spoke a target anonymized utterance. We assume that the speaker anonymization method used to anonymize the utterance is public and thus fully known to the adversary. We further assume that adversaries have access to some

raw speech utterances from the hypothesized speaker as well as to a large public speech corpus with speaker labels.[3] With the above knowledge, an adversary can anonymize the public corpus with the same method that was used to protect the target utterance, and use the resulting data to train an anonymization-aware ASV system (see Section 2.1). This system can then be deployed to conduct the attack by comparing the target utterance to those from the hypothesized speaker. This attack scenario corresponds to the strongest attack model introduced in [71] (called "informed" adversaries therein).

In this work, we design a novel speaker anonymization method for which we provide both formal and empirical privacy guarantees, in the form of differential privacy and ASV error rates achieved by a concrete adversary, respectively.

## 4 EXISTING SPEAKER ANONYMIZATION METHODS

Due to legal and technological awareness in society, privacy-preserving data publishing approaches specific to speech data have recently gained prominence and several methodologies have been proposed for speaker anonymization [73]. In this section, we describe existing techniques and their limitations.

**Speech transformation**, also called voice transformation [72], refers to modifications of speech that aim to shift the perceived attributes of an utterance in a certain direction while leaving the linguistic content unchanged. Speech transformation is the simplest kind of speaker anonymization since it does not require large data sets for training machine learning models; instead, it relies on classic signal processing to modify speech parameters. Patino et al. [57] presented a speech transformation method for speaker anonymization that alters the spectral envelope, the smooth curve that follows the peaks of the spectrum for any given analysis frame and is governed by the shape of the vocal tract. Specifically, they altered the pole angles of the linear prediction (LP) spectral envelope via the McAdams coefficient [51]. Gupta et al. [29] improved this work by modifying both the pole angles and the pole radii of the LP spectral envelope. Although these parameter manipulations are perceptually reasonable, speaker information can be easily recovered by machine learning-based attacks, such as training an ASI or ASV system on transformed speech [71].

**Adversarial training** has been used for speaker anonymization by training neural networks to compute representations of speech that maximize the accuracy of a certain utility task while minimizing the accuracy of speaker identification [67]. For instance, Champion et al. [13] showed that the phonetic features extracted from a neural network trained for automatic speech recognition contain residual speaker information and used speaker adversarial training similar to [67] to remove it. A common criticism against adversarial methods is that they do not guarantee protection against other speaker identification attacks than the one implemented by the adversarial branch. In practice, it has been shown experimentally that they also fail against identification attacks following the same architecture as the adversarial branch, due to the fact that they do not generalize well to unseen speakers [67].

**Voice conversion** aims to modify the original speaker's voice (called *source*) such that it sounds like another speaker's voice (called *target*), while leaving the linguistic content unchanged. Bahmaninezhad et al. [6] performed speaker anonymization by converting the original speaker's voice into the average of all voices of the same gender. Pobar and Ipšić [59] pre-trained a set of speaker transformations for fixed source-target pairs and identified the source to select one of the corresponding transformations. Yoo et al. [79] presented a many-to-many CycleGAN variational autoencoder-based voice conversion method which encodes the identity of each speaker by a one-hot vector. These methods are hardly applicable in practice: they require many utterances the source speaker to be present in the training set while, in the context of anonymization, the source speaker is usually unknown at training time and limited to a single utterance at test time. Furthermore, except for [6], they use real speakers as targets, which can be seen as a form of voice spoofing and raises ethical concerns.

**Speech synthesis** techniques have also been proposed to relax the above requirement. For instance, Justin et al. [36] transcribed speech into a diphone sequence and re-synthesized it using a single target speaker. Speech synthesis methods suffer from three limitations. First, they still result in a limited set of target speakers or speaker transformations, which prevents the original speaker from choosing an arbitrary unseen speaker as the target. Second, using a real speaker's voice as the target raises ethical concerns. Third, the conversion of speech into a sequence of discrete tokens as in [36] is error-prone and destroys the prosodic information. These limitations motivate the goal of converting the original utterance into an arbitrary pseudo-speaker's voice without relying on a transcription step. This goal can be achieved using x-vector based anonymization, which is considered to be the current state-of-the-art [14, 23, 24, 31, 50, 68, 69, 73, 75]. The core idea is to extract the sequences of phonetic and prosodic features of the source utterance along with a single x-vector for the whole utterance, and to replace that x-vector by a target x-vector that does not correspond to a real speaker.[4] This target x-vector, along with the original phonetic and prosodic features, is provided as input to a neural source-filter (NSF) speech synthesizer [78] to produce anonymized speech.

In this paper, we choose x-vector based anonymization as our baseline, since it addresses the limitations of speech transformation, adversarial training, voice conversion, and classical speech synthesis techniques outlined above. Yet, it suffers from one remaining limitation: it assumes the identity markers of the source speaker to be concentrated in the x-vector extracted from the utterance, so that replacing it with the target x-vector is sufficient to remove them. In this work, we show that this assumption is incorrect (see Section 7.1) and provide an effective way to remove residual information about the source speaker and to obtain provable privacy guarantees.

[3] See e.g., the Librispeech data set: https://www.openslr.org/12

[4] Strategies where the choice of target x-vector does not depend on the input utterance was shown to give best performance in [69].

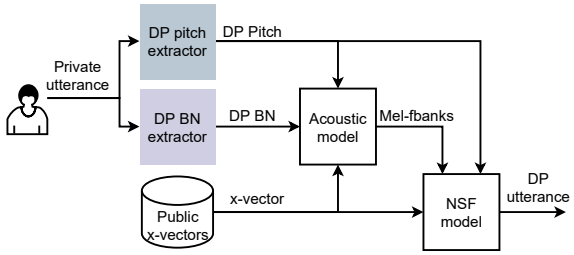


**Figure 1: Overview of our proposed speaker anonymization method. Our main contributions are the differentially private pitch and BN feature extractors (shown in color), which make the full pipeline differentially private.**

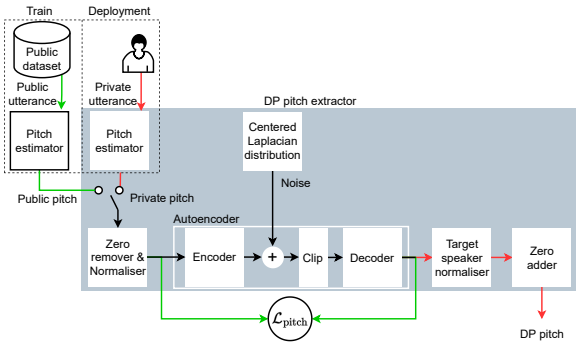


**Figure 2: Proposed DP pitch extractor. The convolutional autoencoder with a noise layer is trained using public pitch sequences and subsequently used to generate perturbed pitch sequences from private pitch sequences in a differentially private fashion. Black arrows show paths that are common to both training and deployment, while green and red arrows apply only to training or deployment, respectively.**

# 5 PROPOSED APPROACH

## 5.1 Overview

Our speaker anonymization approach is depicted in Figure 1. The overall pipeline is based on x-vector anonymization [24, 69] (see Section 4). Pitch and bottleneck (BN) features are first extracted from the input speech. These features, along with a public speaker embedding (x-vector) that corresponds to a different (pseudo) speaker, are then used to re-synthesize speech using acoustic[5] and NSF models. Note that the x-vector is chosen independently of the input utterance. Therefore, information about the input speaker can only leak through pitch and BN features. As we will see in our experiments (Figure 6), these features are actually very predictive of speaker identity. Our contribution is to design pitch and BN feature extractors that satisfy differential privacy so as to provably upper bound the amount of residual speaker information embedded in these features while preserving linguistic and prosodic content.[6] Crucially, the post-processing and composition properties of DP will guarantee that our full pipeline (from the input speech to the anonymized speech) also satisfies DP.

Our DP pitch extractor consists of a conventional pitch estimator followed by an autoencoder network with a Laplace noise layer trained to reconstruct the global pitch dynamics using a custom loss function. Our DP BN extractor is a deep ASR acoustic model, also with a Laplace noise layer, trained on speech utterances to estimate the corresponding word sequence. We use a public set of annotated speech utterances to train both extractors prior to deployment. We emphasize that our extractors are quite generic. They may be used in variants of x-vector based speaker anonymization [14, 23, 50, 75], or independently. For instance, we will empirically show that our DP BN features are sufficient to decode the linguistic content.

In the rest of this section, we describe our DP pitch and BN feature extractors in detail, and conclude by stating the DP guarantees for our full pipeline.

**Notations.** Let $\mathbf{x}$ be a speech utterance consisting of $K$ time frames. The value of $K$ depends on the duration of the utterance and the chosen frame rate (typically, 10 ms). The pitch sequence computed from $\mathbf{x}$ is a non-negative 1-dimensional sequence of length $K$, which we denote by a vector $\mathbf{p} \in \mathbb{R}_+^K$. The BN features extracted from $\mathbf{x}$ are an $M$-dimensional sequence of length $K$ that we denote by a matrix $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_K]^\top \in \mathbb{R}^{K \times M}$ where each $\mathbf{b}_k \in \mathbb{R}^M$. We denote by $W$ the ground truth text transcription of $\mathbf{x}$.

**Public data.** We assume that we have access to a *public* data set $\mathcal{X} = \{(\mathbf{x}_i, W_i)\}_{i=1}^N$ of $N$ annotated speech utterances to train our DP feature extractors. This training data set must be disjoint from the (private) data used at deployment, i.e., speakers have to be different in both datasets.

## 5.2 Differentially Private Pitch Extractor

As mentioned earlier, the global dynamics of the pitch sequence $\mathbf{p}$ for an utterance $\mathbf{x}$ conveys prosodic information, while its local variations are more specific to each speaker [2, 17, 49, 58]. We aim to learn a DP autoencoder $\mathcal{A}$ which takes as input a raw pitch sequence $\mathbf{p}$ computed by a conventional pitch estimator, and outputs a perturbed pitch sequence $\mathbf{p}^{\text{DP}}$ of the same length in which the identity information has been removed while most of the prosodic information is preserved. An obvious approach to obtain a DP autoencoder is to rely on input perturbation, i.e., to add Laplace noise directly to the raw pitch $\mathbf{p}$. However, this baseline largely destroys the time correlations that are indicative of prosody elements that we wish to preserve, as we show in our experiments.

Instead, we propose to learn a deep convolutional autoencoder with a noise layer. Below, we describe the architecture of our autoencoder, how it is trained, and finally how it can be deployed to anonymize pitch sequences. The block diagram of our complete DP pitch extractor is shown in Figure 2.

**Autoencoder architecture.** We propose to define $\mathcal{A} = \mathcal{D} \circ \mathcal{N}_p \circ \mathcal{E}$ as a fully convolutional autoencoder composed of an encoder $\mathcal{E}$, a noise layer $\mathcal{N}_p$ and a decoder $\mathcal{D}$. This encoder-decoder architecture, inspired by [65], has two important benefits in our context. First, a

[5] In the context of speech synthesis, the acoustic model generates a sequence of Mel-frequency spectra from the BN and pitch sequences, which is then transformed into a speech waveform by the NSF model. This acoustic model is distinct from that of ASR which, given an input sequence of acoustic features, estimates the corresponding triphone posterior probabilities.

[6] With a slight abuse of terminology, we will sometimes use *DP pitch* and *DP BN* to refer to features obtained with our DP extractors.

fully convolutional architecture enables us to deal with variable-length input and output sequences as the shape and size of the weights of each convolutional layer (kernel) are not affected by the size of the input and output of that layer. Second, convolutional layers are suitable to capture time dependencies in pitch sequences.

The encoder $\mathcal{E}$ maps an input pitch $\mathbf{p} \in \mathbb{R}^K$ to a latent representation $\mathbf{h} = \mathcal{E}(\mathbf{p}) \in [0,1]^{C \times K}$ through 3 convolutional layers (each with $C$ channels) with sigmoid activation functions.

In order for the autoencoder $\mathcal{A}$ to satisfy $\varepsilon$-DP for a given $\varepsilon > 0$, the encoder is followed by a noise layer $\mathcal{N}_p$ which adds centered Laplace noise to each entry of the latent representation $\mathbf{h}$ to generate a perturbed version $\mathbf{h}^{\mathsf{DP}} \in \mathbb{R}^{C \times K}$:

$$\mathbf{h}^{\mathsf{DP}} = \mathcal{N}_p(\mathbf{h}) = \mathbf{h} + \mathrm{Lap}(\Delta_1(\mathcal{E})/\varepsilon), \tag{3}$$

where $\Delta_1(\mathcal{E}) = \max_{\mathbf{p},\mathbf{p}'} \|\mathcal{E}(\mathbf{p}) - \mathcal{E}(\mathbf{p}')\|$ is the $\ell_1$-sensitivity of $\mathcal{E}$. While tightly bounding the sensitivity of neural networks can be challenging in general [56], here the use of the sigmoid activation allows us to easily bound $\Delta_1(\mathcal{E})$ since each entry of $\mathbf{h}$ belongs to $[0,1]$:

$$\Delta_1(\mathcal{E}) = C \times K \times 1 = CK. \tag{4}$$

This bound is tight enough in practice for the Laplace noise injected to the features not to be detrimental to utility, as long as the value of $\varepsilon$ remains reasonable (away from zero).

Finally, the decoder $\mathcal{D}$ takes as input the perturbed latent representation $\mathbf{h}^{\mathsf{DP}}$, deterministically clips each of its entries back to $[0,1]$ (which we found to help training to converge), and decodes it into a perturbed pitch sequence $\mathbf{p}^{\mathsf{DP}} = D(\mathbf{h}^{\mathsf{DP}}) \in \mathbb{R}^K$ through 3 convolutional layers (2 $C$-channel layers with sigmoid activation followed by 1 layer with linear activation).

**Training phase.** We train our autoencoder on a set of raw pitch sequences $\{\mathbf{p}_i \in \mathbb{R}^{K_i}\}_{i=1}^N$ computed from the speech waveforms $\mathbf{x}_i$ in the public data set $\mathcal{X}$ (using for instance the YAAPT estimator). The pitch sequences are pre-processed as follows. First, zero values are removed. Indeed, these values indicate silence or unvoiced phonemes and must be kept to zero so that silence remains silence, and every unvoiced phoneme remains the same unvoiced phoneme. For instance, replacing the zero pitch on phoneme */p/* by a nonzero pitch would transform it into a */b/*, which would harm utility. This operation does not affect privacy, since these values are equal to zero for all speakers and therefore do not convey identity information. It also makes it possible to account for variations of pitch across successive voiced phonemes. Table 1 shows statistics on the length of pitch sequences and the proportion of zero values in a data set used in our evaluation, and Figure 3 shows an example of raw pitch sequence. Finally, since pitch differs in range across speakers, the last step of pre-processing is to normalize each sequence to have zero mean and unit variance.

To preserve the prosodic content in the reconstructed pitch, we propose to train the autoencoder by minimizing the following loss:

$$\mathcal{L}_{\text{pitch}} = 1 - \sum_{i=1}^{N} \mathtt{Corr}(\mathbf{p}_i, \mathbf{p}_i^{\mathsf{DP}}), \tag{5}$$

where $\mathtt{Corr}(\cdot,\cdot)$ is the Pearson correlation coefficient. In contrast to the standard mean squared error loss, maximizing correlations between the original and reconstructed pitch makes reconstruction errors in the local variations of the pitch (which tend to be more speaker-specific) more costly than errors in the global dynamics of the sequence (which convey the global prosodic content of the utterance). As the autoencoder *must* suffer reconstruction errors due to the addition of noise, it will learn to preserve global dynamics as much as possible while sacrificing local variations, which is the desired behavior for speaker anonymization. This is illustrated in Figure 3.

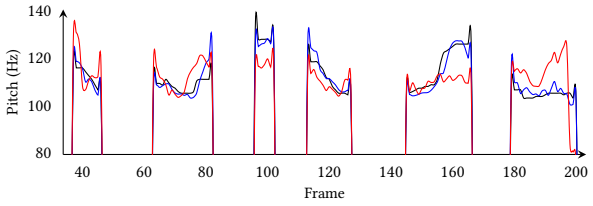


**Figure 3: Visualization of the original (non-private) pitch sequence (——) and noisy reconstructed pitch sequences obtained with our approach for $\varepsilon = 10$ (——) and $\varepsilon = 1$ (——). In general, our approach preserves the global dynamics of the original pitch sequence thanks to our correlation-based loss.**

**Table 1: YAAPT pitch statistics on the dev_clean subset of LibriSpeech.**

| | Min | Max | Avg | Std |
|---|---|---|---|---|
| Length $K$ | 147 | 3261 | 743 | 493 |
| Non-Zeros | 24% | 76% | 53% | 8% |

**Deployment phase.** Once the autoencoder $\mathcal{A}$ has been trained, we can use it to generate perturbed versions of the pitch sequence of any private utterance $\mathbf{x}$. Similarly to the training phase, we compute the pitch sequence $\mathbf{p}$ from $\mathbf{x}$, remove the zeros, normalize it, and push it to the autoencoder to obtain a perturbed pitch sequence $\mathbf{p}^{\mathsf{DP}}$. We then normalize $\mathbf{p}^{\mathsf{DP}}$ to match the mean $\mu_{\text{target}}$ and variance $\sigma_{\text{target}}$ of the pitch of a target (pseudo) speaker, where $\mu_{\text{target}}$ and $\sigma_{\text{target}}$ are computed over a public set of utterances from the target speaker. In the context of the full speaker anonymization pipeline of Figure 1, this normalization (which we call *pitch conversion*) makes the mean and variance of the perturbed pitch consistent with the choice of the target x-vector. Finally, we add the zero values back in their original positions in the sequence.

**Privacy guarantees.** By the Laplace mechanism (Definition 2), $\mathcal{N}_p \circ \mathcal{E}$ satisfies $\varepsilon$-DP, and so does the autoencoder $\mathcal{A} = \mathcal{D} \circ \mathcal{N}_p \circ \mathcal{E}$ by the post-processing property of DP.

## 5.3 Differentially Private BN Extractor

We now turn to the BN features, which are phonetic features that should be sufficient to decode the linguistic information. BN features are typically obtained as an intermediate layer of an ASR acoustic model. However, traditional BN features also contain residual speaker information as shown in our experiments (see Section 7.1). We propose to address this issue by adding a noise layer to the acoustic model, similarly to the approach used for pitch. The block diagram of our complete DP BN extractor is shown in Figure 4.

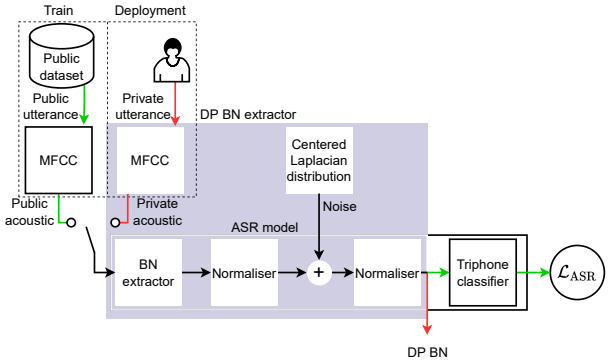


**Figure 4: Proposed DP BN extractor. The ASR acoustic model with a noise layer is trained on public utterances and subsequently used to generate perturbed BN features from private utterances in a DP fashion. Black arrows show paths that are common to both training and deployment, while green and red arrows apply only to training or deployment, respectively.**

**ASR model architecture.** We adapt here the widely used ASR acoustic model architecture and sequence-discriminative training criterion proposed in [62].[7] We view the ASR acoustic model $\mathcal{M} = \mathcal{T} \circ \mathcal{N}_B \circ \mathcal{B}$ as three sequential parts: a BN extractor $\mathcal{B}$, followed by a noise layer $\mathcal{N}_B$, and finally a triphone classifier $\mathcal{T}$. The BN extractor $\mathcal{B}$ takes as input a sequence of acoustic features $\mathbf{O} \in \mathbb{R}^{K \times A}$ extracted from a speech utterance $\mathbf{x}$ with $K$ frames and outputs a sequence of BN features $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_K]^\top \in \mathbb{R}^{K \times M}$:

$$\mathbf{B} = \mathcal{B}(\mathbf{O}). \tag{6}$$

The acoustic features $\mathbf{O}$ are the concatenation of 40-dimensional MFCCs, which are the most popular spectral features for speech processing, and 100-dimensional i-vectors [64] which help the acoustic model adapt to different speakers. Therefore, the per-frame dimensionality of these features is $A = 140$. The BN extractor $\mathcal{B}$ is composed of 17 factorized time delay neural network layers [60], which perform one-dimensional convolution operations to learn the temporal context present in the acoustic feature sequence $\mathbf{O}$.

We now describe our noise layer $\mathcal{N}_B$, which we use to hide speaker information and achieve differential privacy. Each frame-level BN feature vector $\mathbf{b}_k$ is $M$-dimensional with $M = 256$. Due to this high dimensionality, we enforce $\varepsilon$-DP at the frame level (from which we can deduce DP guarantees at the utterance level, as explained at the end of this section). Note that each frame-level BN feature vector $\mathbf{b}_k$ is not normalized. Therefore, our noise layer $\mathcal{N}_B$ first normalizes each $\mathbf{b}_k$ to have unit $\ell_1$-norm and then adds Laplace noise to their entries to generate a sequence of perturbed BN features $\mathbf{B}^{\text{DP}} = [\mathbf{b}_1^{\text{DP}}, \ldots, \mathbf{b}_K^{\text{DP}}]^\top \in \mathbb{R}^{K \times M}$:

$$\mathbf{B}^{\text{DP}} = \mathcal{N}_B(\mathbf{B}) = [\mathcal{N}_b(\mathbf{b}_1), \ldots, \mathcal{N}_b(\mathbf{b}_K)]^\top, \tag{7}$$

where $\mathcal{N}_b(\mathbf{b}) = \text{norm}_1(\text{norm}_1(\mathbf{b}) + \text{Lap}(2/\varepsilon))$ with $\text{norm}_1(\mathbf{c}) = \mathbf{c}/\|\mathbf{c}\|_1$. The scale of the centered Laplace noise comes from the application of the Laplace mechanism (Definition 2), where the $\ell_1$-sensitivity of the normalized frame-level BN features is bounded by $\max_{\mathbf{b},\mathbf{b}'} \|\mathbf{b} - \mathbf{b}'\|_1 \leq 2$ based on the fact that we normalize each BN feature vector $\mathbf{b}$ to have $\|\mathbf{b}\|_1 = 1$ and by triangle inequality $\|\mathbf{b} - \mathbf{b}'\|_1 \leq \|\mathbf{b}\|_1 + \|\mathbf{b}'\|_1 = 2$.

Note that we post-normalize the perturbed BN features to have unit $\ell_1$-norm, as we found this to improve training convergence.

Finally, the triphone classifier $\mathcal{T}$ takes the sequence of perturbed BN features $\mathbf{B}^{\text{DP}}$ as input and outputs the corresponding triphone log-posterior probabilities $\{P(S_k|\mathbf{B}^{\text{DP}})\}_{k=1}^N$ which represent the phonetic content of $\mathbf{x}$ (see Section 2.1). We refer to [62] for details on the architecture of $\mathcal{T}$.

**Training phase.** Our ASR acoustic model $\mathcal{M}$ is trained on acoustic features $\{\mathbf{O}_i\}_{i=1}^N$ extracted from the utterances $\{\mathbf{x}_i\}_{i=1}^N$ in the public annotated data set $\mathcal{X}$ and their corresponding transcriptions $\{W_i\}_{i=1}^N$. Our goal is that the model learns to keep sufficient information in the BN features to predict the linguistic content, while other information (in particular, speaker identity) is naturally lost due to noise addition. To this end, we minimize a cost function $\mathcal{L}_{\text{ASR}} = \mathcal{L}_{\text{MMI}} + 0.1 \cdot \mathcal{L}_{\text{CE}}$ composed of two terms. The dominant term, $\mathcal{L}_{\text{MMI}}$, is the lattice-free maximum mutual information (LF-MMI) [62] cost which aims to maximize the posterior probability of the ground truth word sequence $W_i$:

$$\mathcal{L}_{\text{MMI}} = -\sum_{i=1}^{N} \log \frac{P(\mathbf{O}_i|W_i)P(W_i)}{\sum_{W'} P(\mathbf{O}_i|W')P(W')}. \tag{8}$$

The numerator is the joint likelihood of the acoustic features $\mathbf{O}_i$ and the ground truth word sequence $W_i$, while the denominator is the likelihood of the acoustic features marginalized over all possible word sequences. The numerator is computed by summing over all triphone sequences corresponding to $W_i$ : $P(\mathbf{O}_i|W_i) = \sum_{S_i,N_i} P(\mathbf{O}_i|S_i)P(S_i|N_i)P(N_i|W_i)$ where $P(\mathbf{O}_i|S_i) \propto \prod_{k=1}^{K_i} P(S_{i,k}|\mathbf{B}_i^{\text{DP}})/P(S_{i,k})$, and $P(S_{i,k})$, $P(S_i|N_i)$, $P(N_i|W_i)$ and $P(W_i)$ are fixed as explained in Section 2.1. The numerator is computed in a similar way, except that the (intractable) sum over all possible word sequences with a word-level language model is approximated by a (tractable) sum over all possible phoneme sequences with a phoneme-level language model. The second term $\mathcal{L}_{\text{CE}}$ of the cost function is the frame-level cross-entropy loss between true and estimated triphone states, which acts as a regularizer [62]:

$$\mathcal{L}_{\text{CE}} = -\sum_{i=1}^{N}\sum_{k=1}^{K} \log P(S_{i,k}|\mathbf{B}_i^{\text{DP}}). \tag{9}$$

**Deployment phase.** Once the ASR acoustic model $\mathcal{M} = \mathcal{T} \circ \mathcal{N}_B \circ \mathcal{B}$ has been trained, we can use it to generate a sequence of perturbed BN features $\mathbf{B}^{\text{DP}} = \mathcal{N}_B \circ \mathcal{B}(\mathbf{O})$ for any private utterance $\mathbf{x}$ with acoustic features $\mathbf{O}$.

**Privacy guarantees.** By the Laplace mechanism, the frame-level mechanism $\mathcal{N}_b$ satisfies $\varepsilon$-DP. Note that this frame-level guarantee can be converted into a rigorous utterance-level guarantee using the composition property of DP. In particular, the BN extractor $\mathcal{N}_B \circ \mathcal{B}$ satisfies $\varepsilon'$-DP with $\varepsilon' = K\varepsilon$ for an utterance of length $K$. The utterance-level bound is thus looser by a factor of the utterance length $K$. Yet, we will see in our experiments that even when the *analytical* utterance-level privacy guarantee is rather weak (i.e.,

[7] We note that our approach can be easily applied to other acoustic models.

large $\varepsilon'$ due to large values of $K$), noise addition still has a significant effect on the *practical* protection of utterances against strong speaker identification and speaker verification attacks.

### 5.4 Privacy Guarantees for the Full Pipeline

Our full speaker anonymization pipeline (from an utterance to its anonymized version) can be seen as a composition of two DP mechanisms (the DP pitch and DP BN extractors) followed by post-processing steps that do not depend on the input utterance, see Figure 1. Therefore, we can directly obtain DP guarantees for the full pipeline by composing the DP guarantees of the DP pitch and BN extractors.

Formally, let the DP pitch extractor satisfy $\varepsilon_1$-DP and the DP BN extractor satisfy $\varepsilon_2$-DP (at the frame level). Then, by the composition property of DP, the combined output of these extractors satisfy $(\varepsilon_1 + K\varepsilon_2)$-DP for an utterance of length $K$. From the robustness to post-processing property of DP, the full pipeline also satisfies $(\varepsilon_1 + K\varepsilon_2)$-DP.

## 6 EMPIRICAL VALIDATION SETUP

In complement to the analytical privacy guarantees obtained in Section 5, we design an empirical evaluation of the privacy and utility of our approach, and compare against the state-of-the-art speaker anonymization scheme.

### 6.1 General Objectives

Our DP pitch and BN extractors aim to remove the residual speaker identity information from the prosodic and linguistic attributes of an utterance, which are then used in our DP speaker anonymization pipeline of Figure 1 to output utterances with rich prosodic and linguistic attributes. Therefore, our experiments consider the following major dimensions:

(1) How much identity information is retained within the original pitch, BN features and anonymized utterances?
(2) How well can our approach remove this residual speaker information from pitch, BN features and utterances, and protect against concrete attacks?
(3) How does our approach affect the utility of utterances, for ASR training and inference?

### 6.2 Data Set

We work with the LibriSpeech data set [55] used in the VoicePrivacy challenge [74]. It contains about 1,000 hours of English speech derived from audiobooks. LibriSpeech is partitioned into five subsets: train_clean_100, train_clean_360, train_other_500, dev_clean and test_clean. We also use a subset of LibriTTS [83] (a text-to-speech dataset derived from Librispeech) to train the speech synthesis component of the systems. We refer to Appendix A.1 for details on these subsets. Following the VoicePrivacy challenge [74], we consider the train_clean_100, train_clean_360, train_other_500 and LibriTTS subsets as public data set for training our DP extractors, the speech synthesis models, the attacks and the ASR models used for utility evaluation, and the test_clean subset as the private utterances seen in the deployment phase.

**Table 2: Competing anonymization methods, including different variants of our approach. KEYS – Anon: Anonymization; PC: Pitch Conversion; DP: Differential Privacy.**

| Name | DP pitch extractor | PC | DP BN extractor | x-vector anon |
|---|---|---|---|---|
| Anon | - | - | - | ✓ |
| Anon+PC | - | ✓ | - | ✓ |
| Anon+DP_BN (ours) | - | - | ✓ | ✓ |
| Anon+DP_Pitch (ours) | ✓ | ✓ | - | ✓ |
| Anon+DP (ours) | ✓ | ✓ | ✓ | ✓ |

### 6.3 Speaker Anonymization Methods under Comparison

We compare our differentially private speaker anonymization approach (Anon+DP) with the state-of-the-art x-vector based method (Anon) [24, 69]. To analyze the impact of each of our DP extractors separately, we also consider two partial instantiations of our method: i) Anon+DP_BN, a modification of Anon where the BN extractor is replaced by our DP BN extractor; ii) Anon+DP_Pitch, a modification of Anon where the pitch extractor is replaced by our DP pitch extractor followed by pitch conversion. Note that the original Anon method does not perform pitch conversion. Therefore, when evaluated against Anon+DP_Pitch and Anon+DP, we enhance Anon with pitch conversion (Anon+PC) for a fair comparison. The different variants are summarized in Table 2.

All systems are trained on feature sequences with a frame rate of 10 ms. For pitch estimation, we use YAAPT [38]. Our proposed DP pitch extractor is implemented in PyTorch and trained on train_clean_100. For each system, the BN extractor is trained using the Kaldi toolkit [61] on the combined train_clean_100 and train_clean_500 data subsets as in [73], and the speech synthesis component (acoustic and NSF models) is trained on the LibriTTS subset, as in [73]. In all systems and consistently with Figure 1, the target x-vector assigned to a private utterance is selected independently of that utterance, preventing any leakage of speaker identity information from this step.[8] We use a variant of the approach proposed by [69] where the set of public x-vectors is first clustered, then a dense cluster is selected, and finally the target x-vector is chosen as the average of randomly selected x-vectors from that cluster.[9] Additional details on this selection strategy and on training the speaker anonymization systems are in Appendix A.2.

### 6.4 Attacks

In addition to the rigorous analytical guarantees of our approach based on the DP analysis, we provide an empirical understanding of the privacy benefits of our approach. We quantify the empirical protection provided by the different speaker anonymization methods against concrete attacks that aim to re-identify a known

[8] Previous work alternatively considered speaker-level assignment (i.e., same target x-vector used for all utterances of a given speaker) [68, 74], but this introduces an undesirable dependence on the speaker identity. In Appendix B, we show that speaker-level assignment does not provide better empirical protection than utterance-level assignment, and discuss the choice of assignment strategy from the point of view of the attacker.
[9] The averaging step ensures that the target x-vector corresponds to a pseudo speaker, not a real speaker.

speaker or to link together two utterances from the same speaker. The adversaries have full knowledge of the anonymization scheme that was applied (including its parameters), and have access to a large public speech corpus (with speaker identities) that they can anonymize using the targeted scheme to train their attack. They also have access to "enrollment" utterances from the target speaker. This corresponds to the strongest attack model that has been considered in speaker anonymization [71], allowing us to perform an empirical privacy evaluation closer to the worst-case (maximum privacy leakage).

First, we perform speaker re-identification attacks directly on (utterance-level) pitch and BN features. Similar to [67], the attack is based on an ASI system which follows the classical speaker classification architecture [66] except that, instead of MFCCs as inputs, it is trained on pitch or BN features (with or without DP depending on the anonymization method under attack) extracted from the train_clean_360 data subset. This subset contains 921 speakers and is divided into train/validation/test splits such that 80% utterances of each speaker are used for training, 10% for validation and 10% for testing. The ASI system is trained over the train split, while the validation set is used for monitoring the generalization performance and early stopping in case of convergence. The adversary then uses the trained ASI system to predict the speaker of unseen utterances in the test split, among the known identities from the training set.

Second, we perform speaker linkage attacks on anonymized utterances. In line with the evaluation of the VoicePrivacy challenge [73], the attack is based on an ASV system (see Section 2.1) that follows the standard setup in Kaldi [61]. The ASV system (both x-vector extractor and PLDA) is trained on the train_clean_360 data subset, anonymized with the anonymization method under attack. Finally, the adversary uses the trained ASV system to compute a log-likelihood ratio score between a trial and an enrollment utterance, and decides whether they are from the same speaker by comparing the score with a threshold. The choice of this threshold affects the ratio between the false acceptance rate and the false rejection rate of the attack (see Section 6.5). More details on the implementation of these attacks can be found in Appendix A.3.

In line with previous DP studies [33, 35, 52], we expect that our approach may give better protection against such concrete attacks than what the analytical $\varepsilon$-DP guarantee suggests, because DP is a worst-case guarantee and enforces a stronger notion of privacy than concealing the speaker identity.

## 6.5 Performance Measures

**Empirical privacy measures.** We define empirical privacy measures based on the performance of the above attacks.

The empirical privacy $P_{ASI}$ achieved by pitch and BN extractors is measured by the error of the speaker re-identification attack, i.e., the proportion of utterances which are *not* assigned to the correct speaker by the ASI system, which varies between 0% (worst privacy) and 100% (best privacy). We report $P_{ASI}$ over the (unseen) test split.

The empirical privacy achieved by a full speaker anonymization pipeline is measured by the equal error rate (EER) $P_{ASV,e}$ and unlinkability $P_{ASV,l}$ of the speaker linkage attack. These two metrics are standard in speaker anonymization [47]. The EER is equal to the false acceptance rate and the false rejection rate at the threshold for which these two rates are equal [10], and it varies between 0% (worst privacy) and 50% (best privacy). In constrast, the unlinkability does not depend on the choice of a threshold: it measures the amount of overlap between the *distributions* of same-speaker and different-speaker scores. It is equal to 1 minus the linkability metric in [27, 47] and varies between 0 (worst privacy) and 1 (best privacy). The trial and enrollment sets used to compute $P_{ASV,e}$ and $P_{ASV,l}$ are constructed from the test_clean subset in the same way as done in the VoicePrivacy challenge [73].

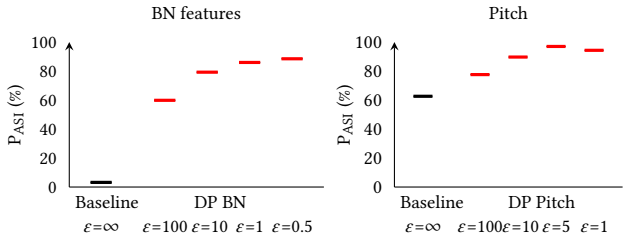


**Figure 5: Empirical privacy of original BN features versus our proposed DP BN features (left) and original pitch versus our proposed DP pitch (right) for different privacy budgets $\varepsilon$. Empirical privacy is assessed by the test error ($P_{ASI}$) of a speaker re-identification attack trained directly on BN features (left) and pitch (right) from the train_clean_360 split (921 speakers). Our DP extractors significantly reduce speaker information present in BN feature and pitch.**

**Utility measure.** We quantify the preservation of linguistic content of a speaker anonymization scheme by the accuracy of an ASR system trained and evaluated on anonymized utterances. The empirical utility $U_{ASR}$ is equal to 100 minus the word error rate (WER) of the ASR system, i.e., the percentage of word substitutions, deletions, and insertions compared to the number of words in the ground truth transcriptions. We train the evaluation ASR system on the train_clean_360 data subset following the Kaldi recipe for LibriSpeech [55], see Appendix A.4 for details.

In summary, the larger $P_{ASI}$, $P_{ASV,e}$ and $P_{ASV,l}$, the greater the privacy, and the larger $U_{ASR}$, the greater the utility.

# 7 EMPIRICAL RESULTS

Our direct evaluation of pitch and BN features in Section 7.1 shows that DP improves the empirical privacy of pitch and BN features with negligible effects on their utility. Our systematic study in Section 7.2 demonstrates the advantages of plugging our DP pitch and DP BN extractor *separately*, and above all *simultaneously*, on the privacy of the utterances over the previous state-of-the-art approach.

## 7.1 Privacy and Utility of Pitch and BN

**How much speaker identity information is retained withing pitch and BN features?** Figure 5 shows the empirical privacy of BN features (left) and pitch (right) in terms of the test error $P_{ASI}$ of speaker re-identification attacks (ASI systems trained directly on these features). First of all, the results clearly demonstrate that original pitch and BN features do retain a lot of speaker information, violating a key assumption supporting previous x-vector based

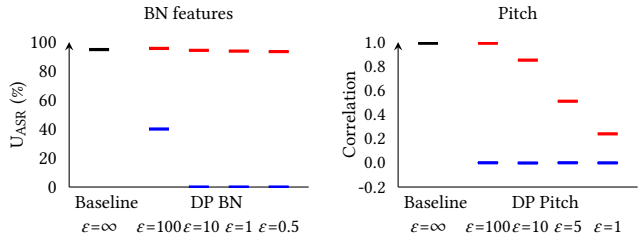


**Figure 6: Utility of original BN features versus our naive DP BN baseline and our proposed DP BN features (left), and original pitch versus our naive DP pitch baseline and proposed DP pitch (right) for different privacy budgets $\varepsilon$. The utility of BN features is assessed by the performance $U_{ASR}$ of an ASR system trained using the corresponding BN features. The utility of pitch is assessed by its correlation to the original pitch. The impact of our DP extractors on utility is negligible for BN features. It is more marked for pitch as $\epsilon$ gets small. Yet, utility is much better than with the naive DP baseline.**

speaker anonymization approaches. Note that BN features contain more identity information than pitches. For example, the $P_{ASI}$ of original BN features is 3.2%, meaning that the adversary can recognize the identity of speakers from original BN features with 96.8% accuracy (recall that there are 921 different speakers). In contrast, the $P_{ASI}$ of original pitch is 62%.

The empirical privacy of BN features and pitch improves significantly when using our DP extractors. For instance, using an analytical privacy budget of $\varepsilon = 1$ improves the empirical privacy $P_{ASI}$ from the baseline of 3.2% to above 86% for BN features, and from 62% to above 94% for pitch. Figure 5 also confirms that a better analytical privacy bound results in better empirical privacy. For example, the empirical privacy of DP BN features with $\varepsilon = 100$ is 60%, and it increases to 80% for $\varepsilon = 10$. We emphasize that although the reported analytical budget $\varepsilon$ for BN features is at the frame level, and thus translates (through the composition property of DP) into a large $\varepsilon'$ for the utterance level, the resulting noise appears to be sufficient in practice to protect well against re-identification attacks that exploit the entire sequence of BN features. We attribute this to the worst-case nature of DP (which often leads to large gaps between analytical and empirical privacy [33, 35, 52]) but also to the fact that our extractors are explicitly trained to preserve only the relevant linguistic and prosodic information.

**How does our approach affect the utility of pitch and BN features?** Figure 6 shows the impact of using our DP extractors on the utility of the resulting pitch and BN features. The utility of BN features is measured by the performance of an ASR model trained from the BN features, while the utility of pitch is measured by its correlation to the original pitch sequence. In addition to pitch and BN features obtained with our DP extractors, we also report the utility of naive DP baselines based on directly adding noise to the original features (see Appendix A.5 for details). Unsurprisingly, these baselines perform very poorly, essentially destroying the utility even for large values of $\varepsilon$. In contrast, our DP BN features induce a negligible drop in the performance of the ASR model. Therefore, our DP BN extractor outputs high-utility BN features that contain enough linguistic information to perform the transcription task, while effectively concealing speaker information as shown in Figure 5.

Regarding pitch, Figure 6 shows that the more noise injected in the pitch, the less correlation with the original pitch. While this is expected, we see that our approach is able to maintain high correlation with the original pitch sequence as long as $\varepsilon$ is not too small (in contrast to the naive baseline).

## 7.2 Privacy and Utility of Anonymized Speech

We now plug the pitch and BN extractors into the full speaker anonymization pipeline (Figure 1) to generate anonymized utterances, and empirically evaluate their privacy and utility. Note that for conciseness, we focus on $\varepsilon \in \{1, 10, 100\}$ in the following.

**Separate effect of DP pitch and DP BN.** We start by evaluating the effect of DP pitch and DP BN separately. Figure 7 shows the empirical privacy and utility of `Anon+DP_Pitch` for different $\varepsilon$ against the state-of-the-art approach `Anon+PC`. Decreasing $\varepsilon$ in the DP pitch extractor improves the protection of utterances anonymized with `Anon+DP_Pitch` against speaker linkage attacks, as reflected by both empirical privacy metrics $P_{ASV,e}$ and $P_{ASV,l}$. On the other hand, the impact on utility, as measured by the performance $U_{ASR}$ of the ASR model, is quite negligible.

Figure 8 compares the empirical privacy and utility of `Anon+DP_BN` for different $\varepsilon$ against `Anon`. Again, we see that utterances anonymized by `Anon+DP_BN` are better protected against speaker linkage attacks. The loss in utility (as measured by $U_{ASR}$) is slightly more noticeable than for `Anon+DP_Pitch`, which is expected since BN features contain most of the linguistic information. Nevertheless, the utility remains high: `Anon` achieves utility 94.69%, while `Anon+DP_BN` achieves 93.96%, 93.49% and 93.03% for privacy budgets of 100, 10 and 1 respectively.

All figures report the variation of the results due to the randomness in x-vector selection. In Appendix C, we report the variation due to the randomness of the noise, which we typically found to be of smaller magnitude.

**Evaluation of the complete system.** Finally, we compare our full DP speaker anonymization approach with both DP pitch and DP BN features (`Anon+DP`) to the state-of-the-art technique of the VoicePrivacy challenge (`Anon+PC`) [68, 73]. Table 3 reports the privacy and utility metrics achieved by our approach for different privacy budgets $\varepsilon$. In all cases, `Anon+DP` provides significantly better protection against speaker linkage attacks at a negligible cost in utility. Indeed, both empirical privacy metrics $P_{ASV,e}$ and $P_{ASV,l}$ nearly double for a very small loss in ASR performance (as measured by $U_{ASR}$). For example, our method `Anon+DP` with $\varepsilon = 100$ for both BN and pitch extractors achieves 24.22% EER, 0.57 unlinkability and 94.00% WER, whereas `Anon+PC` achieves 14.85% EER, 0.35 unlinkability and 94.64% WER. As desired, decreasing $\varepsilon$ (i.e., enforcing stronger analytical privacy guarantees) in `Anon+DP` improves the empirical privacy. For example, `Anon+DP` with $\varepsilon = 1$ increases the empirical privacy of `Anon+DP` with $\varepsilon = 10$ from 26.68% and 0.65 unlinkability to 29.98% and 0.70, with less than 1% degradation in WER. Comparing the results in Table 3 with those in Figures 7-8 also shows that reducing speaker information in *both* pitch and BN features provides a large gain in the privacy of anonymized utterances.

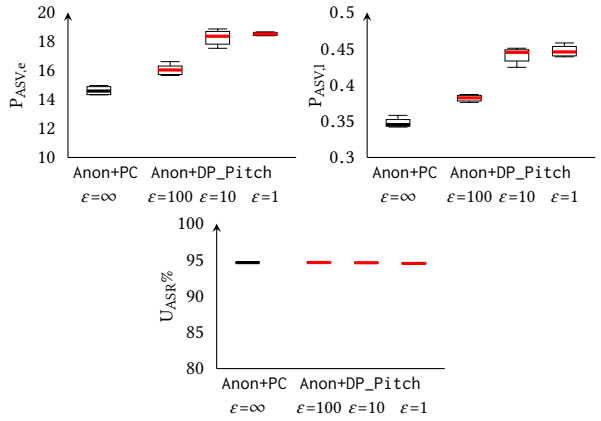


**Figure 7: Empirical privacy (top) and utility (bottom) of utterances anonymized with Anon+PC and our proposed Anon+DP_Pitch for different privacy budgets $\varepsilon$. Empirical privacy is measured by the EER ($P_{ASV,e}$) and unlinkability ($P_{ASV,l}$) of a speaker linkage attack, while utility is assessed by the performance $U_{ASR}$ of an ASR system trained on anonymized utterances. Boxplots are computed over 5 runs of x-vector selection. Plugging our DP pitch extractor in the state-of-the-art speaker anonymization pipeline improves privacy with almost no impact on utility.**

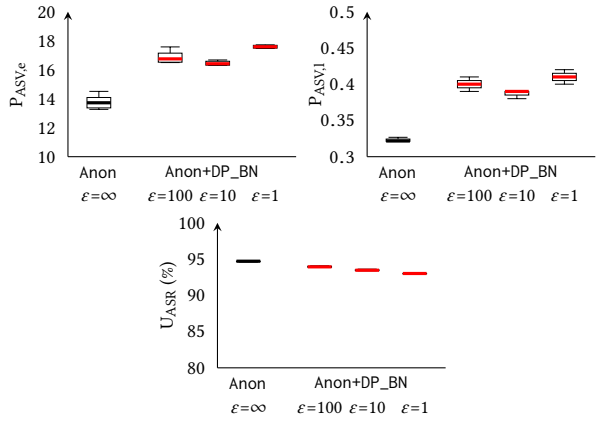


**Figure 8: Empirical privacy (top) and utility (bottom) of utterances anonymized with Anon and our proposed Anon+DP_BN for different privacy budgets $\varepsilon$. Empirical privacy is measured by the EER ($P_{ASV,e}$) and unlinkability ($P_{ASV,l}$) of a speaker linkage attack, while utility is assessed by the performance $U_{ASR}$ of an ASR system trained on anonymized utterances. Boxplots are computed over 5 runs of x-vector selection. Plugging our DP BN extractor in the state-of-the-art speaker anonymization pipeline improves privacy at a negligible loss in utility.**

These results fully validate our design choices and demonstrate the usefulness of our approach for speaker anonymization.

**Table 3: Empirical privacy and utility of our complete speaker anonymization method Anon+DP for different analytical privacy budgets $\varepsilon$ against the state-of-the-art approach Anon+PC. Results are computed across 5 runs of x-vector selection. The results show that our method allows to significantly improve the privacy of anonymized speech at a negligible cost in utility.**

| Method | Privacy: Analytical ($\varepsilon$) BN | Privacy: Analytical ($\varepsilon$) Pitch | Privacy: Empirical $P_{ASV,e}$ (%) | Privacy: Empirical $P_{ASV,l}$ | Utility: Empirical $U_{ASR}$ (%) |
|---|---|---|---|---|---|
| Anon+PC | ∞ | ∞ | 14.62 ± 0.25 | .35 ± .01 | 94.64 ± .06 |
| Anon+DP | 100 | 100 | 24.22 ± .44 | .57 ± .01 | 94.00 ± .10 |
| Anon+DP | 10 | 10 | 27.68 ± .25 | .65 ± .01 | 93.01 ± .07 |
| Anon+DP | 1 | 1 | 29.98 ± .76 | .70 ± .01 | 92.16 ± .05 |

**Table 4: Privacy and utility of the Anon+PC baseline and our Anon+DP scheme on female and male subpopulations.**

| Method | Privacy: Analytical ($\varepsilon$) BN | Privacy: Analytical ($\varepsilon$) Pitch | Privacy: $P_{ASV,e}$ Female | Privacy: $P_{ASV,e}$ Male | Utility: $U_{ASR}$ Female | Utility: $U_{ASR}$ Male |
|---|---|---|---|---|---|---|
| Anon+PC | ∞ | ∞ | 15.87 | 13.09 | 94.25 | 94.24 |
| Anon+DP | 10 | 10 | 28.51 | 26.34 | 92.40 | 92.56 |

**Variation across speakers.** We investigate how the privacy-utility trade-off varies across (subpopulations) of speakers. Note that LibriSpeech contains an imbalanced number of male/female speakers and each speaker has various number of utterances, which may cause some variations across sex and speaker. Table 4 and Figure 9 compare the per-sex and per-speaker performance of our approach with the Anon+PC baseline. The results show that the privacy-utility trade-offs are similar across sex for both our method and the baseline. We observe stronger variations of privacy and utility across speakers for both Anon+PC and Anon+DP. Yet, the gains provided by our approach are clear: while the distribution of utility across speakers is similar for both methods, Anon+DP "shifts up" the distribution of privacy: for instance, the worst privacy protection across speakers with Anon+DP is roughly the same as the median privacy protection with Anon+PC. Similarly, Anon+DP protects half of the speakers better than what Anon+PC provides for the best protected speaker. Finally, while differential privacy can have a disparate impact on the utility across different subpopulations in certain settings [5], we note that it does not seem to be the case for our approach compared to the baseline Anon+PC. Indeed, the magnitude of the variations (across sex and speaker) remain roughly the same across both methods.

## 8 DISCUSSION AND FUTURE WORK

In this paper, we proposed a differentially private speaker anonymization approach which can be used to share speech utterances for training or deployment of voice-based services while concealing the speaker's identity. More specifically, we revisited the disentanglement of speaker (x-vector), linguistic (BN features) and prosodic (pitch) information used in state-of-the-art speaker anonymization techniques so as to analytically bound the speaker information contained in pitch and BN features using carefully designed extractors and differential privacy. Plugging our proposed

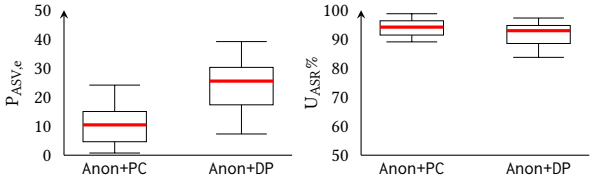


**Figure 9: Variation in the privacy and utility across speakers for the Anon+PC baseline and our Anon+DP scheme. Boxplots are computed over the 29 speakers of the enrollment set.**

DP pitch and BN extractors in a speaker anonymization pipeline, we are able to re-synthesize utterances in a differentially private fashion. We also empirically demonstrated that our approach provides significant gains in practical privacy protection against strong attacks while maintaining a high level of utility.

Below, we discuss some limitations of our approach and promising directions for future work.

**Tightness of analytical privacy guarantees.** In our empirical study, we observed that our approach provides high protection against concrete attacks despite the looseness of the analytical DP guarantee at the utterance-level. Recall that the $\varepsilon$ we report for BN features is frame-level and should be multiplied by the utterance length to obtain an utterance-level guarantee. Our analytical guarantees are even weaker if we consider the DP guarantee at the user level (i.e., adding the budget of all utterances contributed by a given user). While this gap between empirical and analytical privacy is expected and commonly observed in DP literature [33, 35, 52], we briefly discuss below several ways in which it could be reduced.

An obvious solution to get stronger analytical utterance-level privacy guarantees is simply to train DP BN features with a smaller $\varepsilon$. However, we could not manage to make the training converge to a satisfactory value with $\varepsilon < 0.5$ (e.g., for $\varepsilon = 0.1$), presumably because the signal to noise ratio becomes too small. As an alternative, recent tools from DP theory may be leveraged to bound the analytical privacy budget more tightly for utterance and user-level DP. Instead of using simple composition which makes the utterance-level budget $\varepsilon'$ grow linearly with the utterance length $K$, one can resort to advanced composition [21, 37] to achieve $\varepsilon'$ of order $\sqrt{K} \cdot \varepsilon$ at the cost of achieving slightly weaker $(\varepsilon, \delta)$-DP (we illustrate this in Appendix D). One may also reduce the length of utterances by slicing them into shorter segments, which was recently shown to preserve utility in x-vector based speaker anonymization [48]. When the anonymized corpus is large, since the adversary does not know which user submitted which utterance, we can further benefit from privacy amplification by shuffling [7, 22, 25], which gives a reduction of the privacy budget of order $1/\sqrt{N}$ where $N$ is the total number of utterances shared by all users.

Despite these potential improvements, we stress that there is a fundamental gap between the notion of DP we enforce and what we are truly after (we want to remove information about speaker identity, whereas DP seeks to make any two utterances sufficiently indistinguishable). We believe that the careful design of appropriate relaxations of DP that would better capture the objective of speaker anonymization constitutes an interesting challenge.

**Utility measures.** Another interesting future direction is to consider better utility measures for anonymized utterances and pitch. In this work, we demonstrated the utility of utterances through ASR performance. While listening to a few samples suggests that the anonymized speech is intelligible to humans, this could be confirmed and quantified through subjective evaluations. Regarding pitch, we used the correlation with the original pitch as a proxy to measure the preservation of prosodic content. Nonetheless, further experimentation is required to better quantify utility, for instance by training a network to predict certain prosodic attributes from pitch.

**Speech rate.** Our anonymization scheme preserves the speech rate as this is a requirement of the VoicePrivacy challenge [73]. However, a small amount of speaker information might be leaked through the speech rate. An interesting future direction is to address this potential privacy leakage without harming the utility of utterances. This is a challenging problem because the duration of some phonemes must remain unchanged to preserve utility and prosody, while the duration of silences and other phonemes may be changed more or less depending on their position inside words and the utterance.

## ACKNOWLEDGMENTS

This work was supported in part by the French National Research Agency under project DEEP-PRIVACY (ANR-18-CE23-0018) and by the European Union's Horizon 2020 Research and Innovation Program under Grant Agreements No. 825081 COMPRISE and No. 952215 TAILOR. Experiments presented in this paper were partially carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see https://www.grid5000.fr). Ali Shahin Shamsabadi and Nicolas Papernot were also partially supported by CIFAR and the DARPA GARD program. Finally, Ali Shahin Shamsabadi acknowledges the partial support from The Alan Turing Institute.

## REFERENCES

[1] John M. Abowd. The U.S. census bureau adopts differential privacy. In *Proceedings of the ACMInternational Conference on Knowledge Discovery & Data Mining (KDD)*, London, UK, August 2018.

[2] André Adami, Radu Mihaescu, Douglas A. Reynolds, and John J. Godfrey. Modeling prosodic dynamics for speaker recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Hong Kong, April 2003.

[3] Shimaa Ahmed, Amrita Roy Chowdhury, Kassem Fawaz, and Parmesh Ramanathan. Preech: A system for privacy-preserving speech transcription. In *Proceedings of the USENIX Security Symposium*, Virtual Event, August 2020.

[4] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse H. Engel, Linxi Fan, Christopher Fougner, Awni Y. Hannun, Billy Jun, Tony Han, Patrick LeGresley, Xiangang Li, Libby Lin, Sharan Narang, Andrew Y. Ng, Sherjil Ozair, Ryan Prenger, Sheng Qian, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Chong Wang, Yi Wang, Zhiqian Wang, Bo Xiao, Yan Xie, Dani Yogatama, Jun Zhan, and Zhenyao Zhu. Deep speech 2 : End-to-end speech recognition in english and mandarin. In *Proceedings of the International Conference on Machine Learning (ICML)*, New York City, NY, USA, June 2016.

[5] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. In *Proceedings of the Advances in Neural Information Processing (NeurIPS)*, 2019.

[6] Fahimeh Bahmaninezhad, Chunlei Zhang, and John H. L. Hansen. Convolutional neural network based speaker de-identification. In *Proceedings of Odyssey, The Speaker and Language Recognition Workshop*, Les Sables d'Olonne, France, June 2018.

[7] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. The privacy blanket of the shuffle model. In *Proceedings of the Annual International Cryptology Conference*

*(CRYPTO)*, Santa Barbara, CA, USA, August 2019.

[8] Ghazaleh Beigi, Kai Shu, Ruocheng Guo, Suhang Wang, and Huan Liu. I am not what i write: Privacy preserving text representation learning. *arXiv preprint arXiv:1907.03189*, 2019.

[9] Monique Biemans. The effect of biological gender (sex) and social gender (gender identity) on three pitch measures. *Linguistics in the Netherlands*, 15(1):41–52, 1998.

[10] Frédéric Bimbot, Jean-François Bonastre, Corinne Fredouille, Guillaume Gravier, Ivan Magrin-Chagnolleau, Sylvain Meignier, Téva Merlin, Javier Ortega-Garcia, Dijana Petrovska-Delacrétaz, and Douglas A. Reynolds. A tutorial on text-independent speaker verification. *EURASIP Journal on Advances in Signal Processing*, 2004(4):430–451, 2004.

[11] Frédéric Bimbot, Jean-François Bonastre, Corinne Fredouille, Guillaume Gravier, Ivan Magrin-Chagnolleau, Sylvain Meignier, Téva Merlin, Javier Ortega-Garcia, Dijana Petrovska-Delacrétaz, and Douglas A. Reynolds. A tutorial on text-independent speaker verification. *EURASIP Journal on Advances in Signal Processing*, 2004(4):430–451, 2004.

[12] Ferdinand Brasser, Tommaso Frassetto, Korbinian Riedhammer, Ahmad-Reza Sadeghi, Thomas Schneider, and Christian Weinert. VoiceGuard: Secure and private speech processing. In *Proceedings of the conference of the International Speech Communication Association (Interspeech)*, Hyderabad, India, September 2018.

[13] Pierre Champion, Denis Jouvet, and Anthony Larcher. Speaker information modification in the VoicePrivacy 2020 toolchain. Technical report, hal-02995855, November 2020.

[14] Pierre Champion, Denis Jouvet, and Anthony Larcher. A study of F0 modification for x-vector based speech pseudonymization across gender. In *Proceedings of the AAAI Workshop on Privacy-Preserving Artificial Intelligence*, Virtual Event, February 2021.

[15] Zhehuai Chen, Andrew Rosenberg, Yu Zhang, Gary Wang, Bhuvana Ramabhadran, and Pedro J. Moreno. Improving speech recognition using GAN-based speech synthesis and contrastive unspoken text selection. In *Proceedings of the conference of the International Speech Communication Association (Interspeech)*, Virtual Event, October 2020.

[16] Leigh Clark, Philip R. Doyle, Diego Garaialde, Emer Gilmartin, Stephan Schlögl, Jens Edlund, Matthew P. Aylett, João P. Cabral, Cosmin Munteanu, Justin Edwards, and Benjamin R. Cowan. The state of speech in HCI: trends, themes and challenges. *Interacting Computers*, 31(4):349–371, 2019.

[17] Najim Dehak, Pierre Dumouchel, and Patrick Kenny. Modeling prosodic features with joint factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):2095–2103, 2007.

[18] John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *54th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2013.

[19] Delbert Dueck. *Affinity propagation: clustering data by passing messages*. Citeseer, 2009.

[20] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Theory of Cryptography Conference (TCC)*, New York, NY, USA, March 2006.

[21] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[22] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, San Diego, California, USA, January 2019.

[23] Fernando M. Espinoza-Cuadros, Juan M. Perero-Codosero, Javier Antón-Martín, and Luis A. Hernández-Gómez. Speaker de-identification system using autoencoders and adversarial training. *arXiv preprint arXiv:2011.04696*, 2020.

[24] Fuming Fang, Xin Wang, Junichi Yamagishi, Isao Echizen, Massimiliano Todisco, Nicholas Evans, and Jean-Francois Bonastre. Speaker anonymization using x-vector and neural waveform models. In *Proceedings of the ISCA Speech Synthesis Workshop*, Vienna, Austria, September 2019.

[25] Vitaly Feldman, Audra McMillan, and Kunal Talwar. Hiding Among the Clones: A Simple and Nearly Optimal Analysis of Privacy Amplification by Shuffling. Technical report, arXiv:2012.12803, 2020.

[26] Oluwaseyi Feyisetan, Borja Balle, Thomas Drake, and Tom Diethe. Privacy- and utility-preserving textual analysis via calibrated multivariate perturbations. In *Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM)*, Houston, TX, USA, February 2020.

[27] Marta Gomez-Barrero, Javier Galbally, Christian Rathgeb, and Christoph Busch. General framework to evaluate unlinkability in biometric template protection systems. *IEEE Transactions on Information Forensics and Security*, 13(6):1406–1420, 2017.

[28] Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, BC, Canada, May 2013.

[29] Priyanka Gupta, Gauri P. Prajapati, Shrishti Singh, Madhu R. Kamble, and Hemant A. Patil. Design of voice privacy system using linear prediction. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, Auckland, New Zealand, December 2020.

[30] Carlos Gussenhoven et al. *The phonology of tone and intonation*. Cambridge University Press, 2004.

[31] Yaowei Han, Sheng Li, Yang Cao, Qiang Ma, and Masatoshi Yoshikawa. Voice-indistinguishability: Protecting voiceprint in privacy-preserving speech data release. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, London, UK, July 2020.

[32] John H. L. Hansen and Taufiq Hasan. Speaker recognition by machines and humans: A tutorial review. *IEEE Signal Processing Magazine*, 32(6):74–99, 2015.

[33] Matthew Jagielski, Jonathan R. Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private sgd? In *Proceedings of the Advances in Neural Information Processing (NeurIPS)*, Virtual Event, December 2020.

[34] Anil Jain, Lin Hong, and Sharath Pankanti. Biometric identification. *Communications of the ACM*, 43(2):90–98, 2000.

[35] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *Proceedings of the USENIX Security Symposium*, Santa Clara, CA, USA, August 2019.

[36] Tadej Justin, Vitomir Struc, Simon Dobrisek, Bostjan Vesnicer, Ivo Ipsic, and France Mihelic. Speaker de-identification using diphone recognition and speech synthesis. In *Proceedings of the IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, Ljubljana, Slovenia, May 2015.

[37] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International conference on machine learning*, pages 1376–1385. PMLR, 2015.

[38] Kavita Kasi and Stephen A. Zahorian. Yet another algorithm for pitch tracking. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Orlando, Florida, USA, May 2002.

[39] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. What Can We Learn Privately? In *49th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2008.

[40] Kazuya Kawakami, Luyu Wang, Chris Dyer, Phil Blunsom, and Aäron van den Oord. Learning robust and multilingual speech representations. In *Proceedings of the Findings of the Association for Computational Linguistics (EMNLP)*, Virtual Event, November 2020.

[41] Patrick Kenny. Bayesian speaker verification with heavy-tailed priors. In *Proceedings of Odyssey, The Speaker and Language Recognition Workshop*, Brno, Czech Republic, June 2010.

[42] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, May 2015.

[43] Mathias Lécuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, San Francisco, CA, USA, May 2019.

[44] Jason Li, Ravi Gadde, Boris Ginsburg, and Vitaly Lavrukhin. Training neural speech recognition systems with synthetic speech augmentation. *arXiv preprint arXiv:1811.00707*, 2018.

[45] Lingjuan Lyu, Xuanli He, and Yitong Li. Differentially private representation for NLP: formal guarantee and an empirical study on privacy and fairness. In *Proceedings of the Findings of the Association for Computational Linguistics (EMNLP)*, Virtual Event, November 2020.

[46] Vimal Manohar. *Semi-supervised training for automatic speech recognition*. PhD thesis, The Johns Hopkins University, 2019.

[47] Mohamed Maouche, Brij Mohan Lal Srivastava, Nathalie Vauquier, Aurélien Bellet, Marc Tommasi, and Emmanuel Vincent. A comparative study of speech anonymization metrics. In *Proceedings of the conference of the International Speech Communication Association (Interspeech)*, Virtual Event, October 2020.

[48] Mohamed Maouche, Brij Mohan Lal Srivastava, Nathalie Vauquier, Aurélien Bellet, Marc Tommasi, and Emmanuel Vincent. Enhancing Speech Privacy with Slicing. preprint, 2021.

[49] Leena Mary and Bayya Yegnanarayana. Extraction and representation of prosodic features for language and speaker recognition. *Speech Communication*, 50(10):782–796, 2008.

[50] Candy Olivia Mawalim, Kasorn Galajit, Jessada Karnjana, and Masashi Unoki. X-vector singular value modification and statistical-based decomposition with ensemble regression modeling for speaker anonymization system. In *Proceedings of the conference of the International Speech Communication Association (Interspeech)*, Virtual Event, October 2020.

[51] Stephen Edward McAdams. *Spectral Fusion, Spectral Parsing and the Formation of Auditory Images*. PhD thesis, Stanford University, 1984.

[52] Milad Nasr, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlini. Adversary instantiation: Lower bounds for differentially private machine learning. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*,

San Francisco, CA, USA, May 2021.
[53] Andreas Nautsch, Catherine Jasserand, Els Kindt, Massimiliano Todisco, Isabel Trancoso, and Nicholas W. D. Evans. The GDPR & speech data: Reflections of legal and technology communities, first steps towards a common understanding. In *Proceedings of the conference of the International Speech Communication Association (Interspeech)*, Graz, Austria, September 2019.
[54] Andreas Nautsch, Abelino Jiménez, Amos Treiber, Jascha Kolberg, Catherine Jasserand, Els Kindt, Héctor Delgado, Massimiliano Todisco, Mohamed Amine Hmani, Aymen Mtibaa, Mohammed Ahmed Abdelraheem, Alberto Abad, Francisco Teixeira, Driss Matrouf, Marta Gomez-Barrero, Dijana Petrovska-Delacrétaz, Gérard Chollet, Nicholas W. D. Evans, and Christoph Busch. Preserving privacy in speaker and speech characterisation. *Computer Speech and Language*, 58:441–480, 2019.
[55] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, Queensland, Australia, April 2015.
[56] Nicolas Papernot, Abhradeep Thakurta, Shuang Song, Steve Chien, and Úlfar Erlingsson. Tempered sigmoid activations for deep learning with differential privacy. In *Proceedings of the conference on Artificial Intelligence (AAAI)*, Virtual Event, February 2021.
[57] Jose Patino, Natalia Tomashenko, Massimiliano Todisco, Andreas Nautsch, and Nicholas Evans. Speaker anonymisation using the McAdams coefficient. *arXiv preprint arXiv:2011.01130*, 2020.
[58] Barbara Peskin, Jirí Navrátil, Joy S. Abramson, Douglas A. Jones, David Klusácek, Douglas A. Reynolds, and Bing Xiang. Using prosodic and conversational features for high-performance speaker recognition: report from JHU ws'02. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Hong Kong, April 2003.
[59] Miran Pobar and Ivo Ipsic. Online speaker de-identification using voice transformation. In *Proceedings of the International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, May 2014.
[60] Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohammadi, and Sanjeev Khudanpur. Semi-orthogonal low-rank matrix factorization for deep neural networks. In *Proceedings of the conference of the International Speech Communication Association (Interspeech)*, Hyderabad, India, September 2018.
[61] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. The kaldi speech recognition toolkit. In *Proceedings of the IEEE workshop on automatic speech recognition and understanding*, Waikoloa, HI, USA, December 2011.
[62] Daniel Povey, Vijayaditya Peddinti, Daniel Galvez, Pegah Ghahremani, Vimal Manohar, Xingyu Na, Yiming Wang, and Sanjeev Khudanpur. Purely sequence-trained neural networks for ASR based on lattice-free MMI. In *Proceedings of the conference of the International Speech Communication Association (Interspeech)*, San Francisco, CA, USA, September 2016.
[63] Denis Jouvet Raphaël Duroselle, Md Sahidullah and Irina Illina. Modeling and training strategies for language recognition systems. In *Proceedings of the conference of the International Speech Communication Association (Interspeech)*, Brno, Czech Republic, September 2021.
[64] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny. Speaker adaptation of neural network acoustic models using i-vectors. In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, Olomouc, Czech Republic, December 2013.
[65] Ali Shahin Shamsabadi, Francisco Sepúlveda Teixeira, Alberto Abad, Bhiksha Raj, Andrea Cavallaro, and Isabel Trancoso. Foolhd: Fooling speaker identification by highly imperceptible adversarial disturbances. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, June 2021.
[66] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust DNN embeddings for speaker recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, April 2018.
[67] Brij Mohan Lal Srivastava, Aurélien Bellet, Marc Tommasi, and Emmanuel Vincent. Privacy-preserving adversarial representation learning in ASR: reality or illusion? In *Proceedings of the conference of the International Speech Communication Association (Interspeech)*, Graz, Austria, September 2019.
[68] Brij Mohan Lal Srivastava, Mohamed Maouche, Md Sahidullah, Emmanuel Vincent, Aurélien Bellet, Marc Tommasi, Natalia Tomashenko, Xin Wang, and Junichi Yamagishi. Privacy and utility of x-vector based speaker anonymization. preprint, April 2021.
[69] Brij Mohan Lal Srivastava, Natalia A. Tomashenko, Xin Wang, Emmanuel Vincent, Junichi Yamagishi, Mohamed Maouche, Aurélien Bellet, and Marc Tommasi. Design choices for x-vector based speaker anonymization. In *Proceedings of the conference of the International Speech Communication Association (Interspeech)*, Virtual Event, October 2020.
[70] Brij Mohan Lal Srivastava, Nathalie Vauquier, Md. Sahidullah, Aurélien Bellet, Marc Tommasi, and Emmanuel Vincent. Evaluating voice conversion-based privacy protection against informed attackers. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, May 2020.
[71] Brij Mohan Lal Srivastava, Nathalie Vauquier, Md. Sahidullah, Aurélien Bellet, Marc Tommasi, and Emmanuel Vincent. Evaluating voice conversion-based privacy protection against informed attackers. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, May 2020.
[72] Yannis Stylianou. Voice transformation: A survey. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, April 2019.
[73] Natalia A. Tomashenko, Brij Mohan Lal Srivastava, Xin Wang, Emmanuel Vincent, Andreas Nautsch, Junichi Yamagishi, Nicholas W. D. Evans, Jose Patino, Jean-François Bonastre, Paul-Gauthier Noé, and Massimiliano Todisco. Introducing the VoicePrivacy initiative. In *Proceedings of the conference of the International Speech Communication Association (Interspeech)*, Virtual Event, October 2020.
[74] Natalia A. Tomashenko, Xin Wang, Emmanuel Vincent, Jose Patino, Brij Mohan Lal Srivastava, Paul-Gauthier Noé, Andreas Nautsch, Nicholas W. D. Evans, Junichi Yamagishi, Benjamin O'Brien, Anaïs Chanclu, Jean-François Bonastre, Massimiliano Todisco, and Mohamed Maouche. The VoicePrivacy 2020 Challenge: Results and findings. *Computer Speech & Language*, 2022.
[75] Henry Turner, Giulio Lovisotto, and Ivan Martinovic. Speaker anonymization with distribution-preserving x-vector generation for the VoicePrivacy Challenge 2020. *arXiv preprint arXiv:2010.13457*, 2020.
[76] Zoltán Tüske, Pavel Golik, David Nolden, Ralf Schlüter, and Hermann Ney. Data augmentation, feature combination, and multilingual neural networks to improve ASR and KWS performance for low-resource languages. In *Proceedings of the conference of the International Speech Communication Association (Interspeech)*, Singapore, September 2014.
[77] Ewald van der Westhuizen, Herman Kamper, Raghav Menon, John Quinn, and Thomas Niesler. Feature learning for efficient ASR-free keyword spotting in low-resource languages. *Computer Speech & Language*, 71:101275, 2022.
[78] Xin Wang, Shinji Takaki, and Junichi Yamagishi. Neural source-filter waveform models for statistical parametric speech synthesis. *IEEE ACM Trans. Audio Speech Lang. Process.*, 28:402–415, 2019.
[79] In-Chul Yoo, Keonnyeong Lee, Seong-Gyun Leem, Hyunwoo Oh, BongGu Ko, and Dongsuk Yook. Speaker anonymization for personal information protection using voice conversion techniques. *IEEE Access*, 8:198637–198645, 2020.
[80] Dong Yu and Li Deng. *Automatic Speech Recognition*. Springer, 2016.
[81] Dong Yu and Michael L. Seltzer. Improved bottleneck features using pretrained deep neural networks. In *Proceedings of the conference of the International Speech Communication Association (Interspeech)*, Florence, Italy, August 2011.
[82] Stephen A Zahorian and Hongbing Hu. A spectral/temporal method for robust fundamental frequency tracking. *The Journal of the Acoustical Society of America*, 123(6):4559–4571, 2008.
[83] Heiga Zen, Rob Clark, Ron J. Weiss, Viet Dang, Ye Jia, Yonghui Wu, Yu Zhang, and Zhifeng Chen. Libritts: A corpus derived from librispeech for text-to-speech. In *Proceedings of the conference of the International Speech Communication Association (Interspeech)*, Graz, Austria, September 2019.
[84] Shi-Xiong Zhang, Yifan Gong, and Dong Yu. Encrypted speech recognition using deep polynomial networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, Taiwan, April 2019.

# A DETAILS ON EXPERIMENTAL SETUP

In this section, we give additional details on our experimental setup and implementations.

## A.1 Details on the Data Set

The five subsets of LibriSpeech data set [55], and the subset of LibriTTS [83] used in our experiments, are detailed in Table 5. Note that we do not use the dev_clean subset.

## A.2 Details on Speaker Anonymization Systems

**DP pitch autoencoder.** We implement our DP pitch autoencoder in PyTorch and train it on train_clean_100, using a mini-batch size of 1 due to the variable sequence length. We use the Adam optimizer [42] with a learning rate of $1e^{-3}$, a weight decay of $1e^{-5}$, and a dropout of $1e^{-3}$, similarly to [65].

**Target x-vector selection.** In all speaker anonymization systems, the target x-vector for each utterance is selected as follows. First, we cluster all public x-vectors using the Affinity Propagation algorithm [19] and PLDA [41] (obtaining 80 clusters in total). Second, we randomly select one cluster from the 10 largest clusters. Third, we randomly select half of the members of the dense cluster to introduce further randomness in the choice of x-vector.[10] Finally, we average the selected candidate x-vectors to obtain the target x-vector. This selection strategy is very similar to the "dense" strategy proposed in [69], but makes the choice of x-vector completely independent from the input utterance so that this step does not leak any information about the source speaker.

## A.3 Additional Details on Attacks

We give additional details on the implementations of the ASI and ASV systems which form the basis of our attacks.

**ASI system.** The ASI system follows the classical TDNN speaker classification architecture in Kaldi [66]: it is composed of 5 TDNN layers after the input layer, followed by a statistical pooling layer which computes the mean and standard deviation over all the frames to get the utterance-level context. This layer is followed by 2 TDNN layers and finally a softmax output layer. This system is trained on pitch or BN features, with or without $\varepsilon$-DP depending on the system under attack. In the case of pitch, silent regions are removed using energy-based voice activity detection before training the ASI. We train the ASI system over the training split with 15 epochs using a mini-batch size of 64.

**ASV system.** The ASV system, i.e., both the x-vector extractor and PLDA, follows the classical ASV recipe in Kaldi [61]. The x-vector extractor (a TDNN with MFCCs as inputs) and PLDA are trained over the train_clean_360 data set, anonymized using exactly the same method and parameters as used for anonymizing the utterances that the adversary wants to attack. The target x-vector selection used by the attack is the same as the one used by the speaker anonymization system (see Appendix A.2). The ASV system is then used to compute PLDA scores between trial and an enrollment utterances using x-vectors extracted from these utterances.

[10] We noticed that selecting less than 50% of a cluster negatively affects utility.

**Table 5: Statistics of the different subsets of the LibriSpeech (top row) and LibriTTS (bottom row) data sets.**

| | Subset | Size (hrs.) | #speakers | | | #utterances |
|---|---|---|---|---|---|---|
| | | | Female | Male | Total | |
| Librispeech | train-clean_100 | 100.6 | 125 | 126 | 251 | 28,539 |
| | train-clean_360 | 363.6 | 439 | 482 | 921 | 104,014 |
| | train-other_500 | 496.7 | 564 | 602 | 1,166 | 148,688 |
| | dev_clean | 5.4 | 20 | 20 | 40 | 2,703 |
| | test_clean | 5.4 | 20 | 20 | 40 | 2,620 |
| LibriTTS | train-clean_100 | 54 | 123 | 124 | 247 | 33,236 |

**Table 6: Effect of x-vector selection strategy used in anonymization and in the attack on the empirical privacy of speech anonymized with the state-of-the-art Anon+PC. Results are computed across 5 runs of x-vector selection.**

| X-vector Selection Strategy | | Empirical Privacy | |
|---|---|---|---|
| Anonymization | Attack | $P_{ASV,e}$ (%) | $P_{ASV,l}$ |
| speaker-level | speaker-level | 45.90 ± 1.86 | 0.86 ± 0.04 |
| | utterance-level | 15.58 ± 0.23 | 0.37 ± 0.01 |
| utterance-level | speaker-level | 44.32 ± 0.67 | 0.91 ± 0.01 |
| | utterance-level | 14.62 ± 0.25 | 0.35 ± 0.01 |

## A.4 Details on the ASR Evaluation Model

The training procedure and architecture of the ASR system used to compute the utility metric $U_{ASR}$ is similar to the one used to extract BN features in Section 5.3, except that we do not use any noise layer after the BN extractor: the BN features extracted from anonymized utterances are directly fed to the triphone classifier to compute the loss $\mathcal{L}_{ASR}$. We train a different ASR system for each speaker anonymization scheme (e.g., for each value of $\varepsilon$) so that the ASR model can adapt to each scheme. During decoding, we use a large trigram language model $P(W)$ available at the openslr website.[11]

## A.5 Naive DP Baselines

We explain below how the naive input perturbation DP baselines evaluated in Figure 6 are implemented.

**Naive DP pitch baseline.** This baseline consists in adding DP noise directly to the normalized pitches (zero mean and unit std). As pitch values are unbounded, we artificially bound them by clipping in a range. We select the range $[-4, +4]$, which includes most values while being sufficiently narrow to keep the sensitivity small.

**Naive DP BN baseline.** This baseline consists in applying the noise layer $\mathcal{N}_B$ (which normalizes the features, adds noise and normalizes again, see Eq. 7) to the original BN features used by the state-of-the-art speaker anonymization method Anon and re-training the triphone classifier on top of these fixed noisy features. Note that, unlike our proposed approach, this naive baseline does not re-train the BN extractor $\mathcal{B}$. Fine-tuning the original triphone classifier or re-training it from scratch yields similar results.

[11] http://openslr.org/11/


**Figure 10: Empirical privacy (top) and utility (bottom) of utterances anonymized with our proposed Anon+DP_Pitch for different privacy budgets $\varepsilon$. Empirical privacy is measured by the EER ($P_{ASV,e}$) and unlinkability ($P_{ASV,l}$) of a speaker linkage attack, while utility is assessed by the performance $U_{ASR}$ of an ASR system trained on anonymized utterances. Unlike in Figure 7, boxplots are computed over 5 runs of DP noise addition. The results show that the variations due to the randomness of noise in our DP pitch extractor are small.**

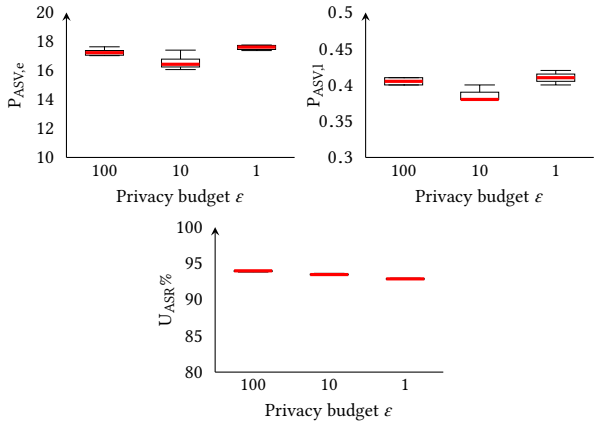


**Figure 11: Empirical privacy (top) and utility (bottom) of utterances anonymized with our proposed Anon+DP_BN for different privacy budgets $\varepsilon$. Empirical privacy is measured by the EER ($P_{ASV,e}$) and unlinkability ($P_{ASV,l}$) of a speaker linkage attack, while utility is assessed by the performance $U_{ASR}$ of an ASR system trained on anonymized utterances. Unlike in Figure 8, boxplots are computed over 5 runs of DP noise addition. The results show that the variations due to the randomness of the noise in our DP BN extractor are small.**

## B SPEAKER VERSUS UTTERANCE-LEVEL X-VECTOR SELECTION

In our experiments, we use an utterance-level x-vector assignment strategy for all anonymization schemes, as well as for the design of our attacks (recall that attackers use the targeted anonymization scheme to anonymize the data they use to train their attack). Utterance-level assignment chooses a potentially different target x-vector for each utterance. Our specific utterance-level assignment strategy has the advantage of making the choice of x-vector independent of the speaker identity, thereby avoiding any leakage at this step. However, prior work also considered speaker-level assignment for anonymization (i.e., using the same target x-vector for all utterances of a given speaker) [68, 74]. This introduces a dependence on the speaker identity that may leak information and would need to be accounted for in a DP analysis.

In this section, we investigate the impact of such design choices empirically. Table 6 shows the empirical privacy for all possible combinations of assignment strategies in the anonymization scheme (here, we focus on Anon+PC) and the attack. We can draw two main conclusions: (i) speaker and utterance-level assignments provide the same level of protection against the best attack, and (ii) perhaps surprisingly, utterance-level assignment is the best choice for attacks, even when the anonymization scheme uses speaker-level assignment. This has led some prior work to largely overestimate the privacy protection provided by x-vector based speaker anonymization with speaker-level assignment, as recently observed in [48]. We empirically found that the poor performance of speaker-level based attack is due to the ASI model overfitting the training data.

Overall, these results validate our choice of using utterance-level assignment for both the anonymization schemes and the attacks.

## C EFFECT OF THE RANDOMNESS OF DP NOISE

Figure 10 and Figure 11 show the variations in privacy and utility due to the randomness of the noise in our DP extractors, rather than due to the randomness in the x-vector selection as in Figure 7 and Figure 8. We see that the variations due to the randomness of the noise are quite small (and typically smaller than those due to the randomness in x-vector selection).

## D USING ADVANCED COMPOSITION

We illustrate here how we can obtain tighter analytical privacy guarantees at the utterance-level by leveraging advanced composition theorems [21, 37]. Specifically, Table 7 reports the utterance-level DP guarantees for utterances with different length using the composition theorem of [37, Theorem-3.4]. The results show that advanced composition can significantly improve the bound on the analytical privacy budget at the utterance level in comparison with using simple composition, especially for long utterances.

**Table 7: Utterance-level DP guarantees using simple composition and advanced composition [37] for frame-level $\varepsilon = 0.5$. KEY– $K$: number of frames in an utterance.**

| Composition | Utterance-level privacy budget $\varepsilon$ | | | |
|---|---|---|---|---|
| | $K = 100$ | $K = 500$ | $K = 1000$ | $K = 10,000$ |
| **Simple** | 50 | 250 | 500 | 5,000 |
| **Advanced** with $\delta = 10^{-5}$ | 36 | 114 | 198 | 1,464 |