Computer Speech and Language 94 (2025) 101806

Contents lists available at ScienceDirect

Computer Speech & Language

journal homepage: www.elsevier.com/locate/csl

Quantifying prediction uncertainties in automatic speaker verification systems

Miao Jing [a,*], Vidhyasaharan Sethu [a], Beena Ahmed [a], Kong Aik Lee [b]

[a] The University of New South Wales, Sydney, NSW, Australia
[b] The Hong Kong Polytechnic University, Hong Kong, China

ARTICLE INFO

Keywords:
Speaker verification
PLDA
Uncertainty
Hamiltonian Monte-Carlo
Stochastic gradient Langevin dynamics
Bayes-by-backprop

ABSTRACT

For modern automatic speaker verification (ASV) systems, explicitly quantifying the confidence for each prediction strengthens the system's reliability by indicating in which case the system is with trust. However, current paradigms do not take this into consideration. We thus propose to express confidence in the prediction by quantifying the uncertainty in ASV predictions. This is achieved by developing a novel Bayesian framework to obtain a score distribution for each input. The mean of the distribution is used to derive the decision while the spread of the distribution represents the uncertainty arising from the plausible choices of the model parameters. To capture the plausible choices, we sample the probabilistic linear discriminant analysis (PLDA) back-end model posterior through Hamiltonian Monte-Carlo (HMC) and approximate the embedding model posterior through stochastic Langevin dynamics (SGLD) and Bayes-by-backprop. Given the resulting score distribution, a further quantification and decomposition of the prediction uncertainty are achieved by calculating the score variance, entropy, and mutual information. The quantified uncertainties include the aleatoric uncertainty and epistemic uncertainty (model uncertainty). We evaluate them by observing how they change while varying the amount of training speech, the duration, and the noise level of testing speech. The experiments indicate that the behaviour of those quantified uncertainties reflects the changes we made to the training and testing data, demonstrating the validity of the proposed method as a measure of uncertainty.

1. Introduction

Automatic speaker verification (ASV) systems have been the focus of research for decades and have reached the state of being deployed commercially as part of identity authentication systems (Lee et al., 2013; Naika, 2018). Given a test utterance and a claimed identity, modern ASV systems compute a score that represents the similarity between the speech embeddings of that test utterance and an enrolled one. The decision to accept or reject the claim is then made by comparing this score to an appropriate threshold.

Current speaker verification answers "what is the likelihood that the input should be accepted/rejected". With state-of-the-art structures, the system can well discriminate different speakers and thus achieve low error rates over test datasets. The typical cases of acceptance and rejection are shown in ① and ③ in Fig. 1. However, while considering various interfering factors including background noises, short duration, channel distortions etc. (as in Fig. 1 ②), the performance of ASV degrades severely (Fan et al., 2020). Obtaining high-quality speaker embeddings under different conditions has been one of the major tasks in current research.

* Corresponding author.
E-mail address: z5182435@ad.unsw.edu.au (M. Jing).

https://doi.org/10.1016/j.csl.2025.101806
Received 22 June 2024; Received in revised form 22 December 2024; Accepted 15 April 2025
Available online 30 April 2025

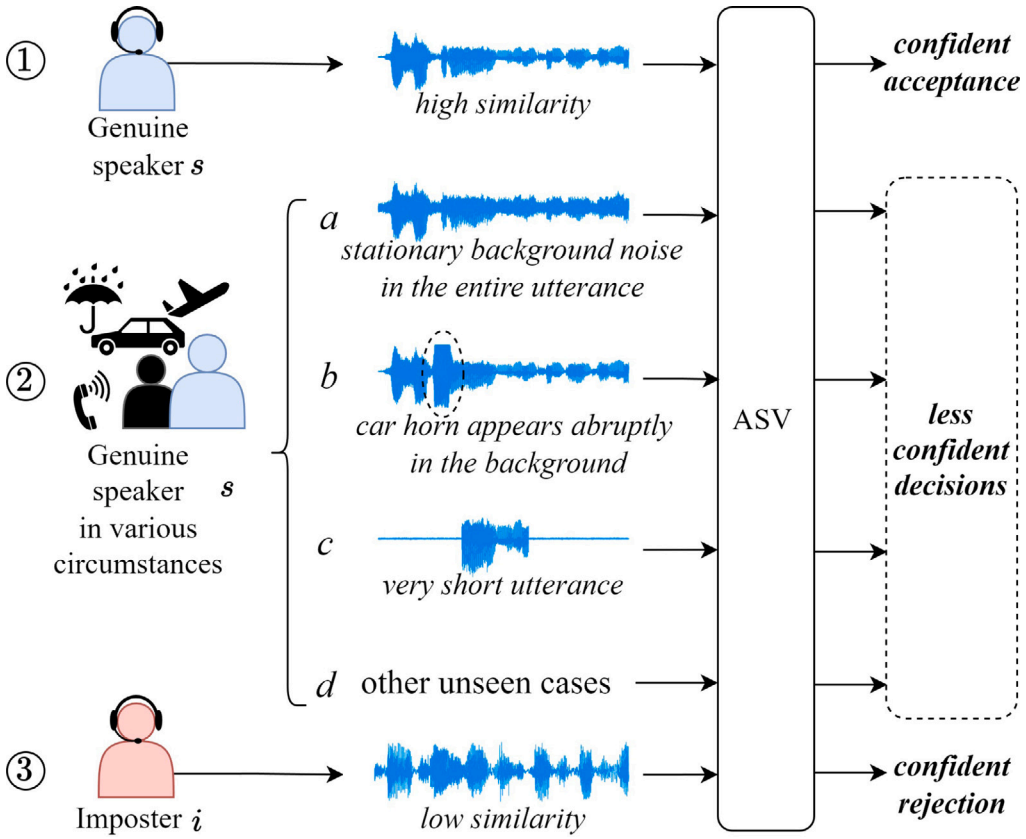


Fig. 1. ASV makes decisions in different conditions.

The endeavour to achieve this leads to successful modern ASV structures (Dehak et al., 2010; Snyder et al., 2018; Desplanques et al., 2020; Liu et al., 2023, 2024).

However, another research question also refers to the mentioned problem yet receives less attention, that is, "how confident can a system be with its decisions?" which is only partially answered by obtaining the likelihoods or posterior probabilities of acceptances/rejections. Current ASV tends to reject when conditions deteriorate such as a, b, c in ②, demonstrating its safety, but the real-world application requires making distinction between those cases since different responses need to be made for them. For example, in the scenario of unlocking a smartphone, frequently rejecting an authentic speaker will lead to bad user experience and thus the ASV needs to encourage the user to provide more convincing inputs, while in safety-demanding cases like accessing a banking system, ASV should be able to alert when there is an inappropriate access.

Conceptually, "speech samples recorded under bad acoustic conditions" (cases in ② of Fig. 1) and "intrinsically dissimilar speakers" (case ③ of Fig. 1) are in fact different situations. However, the current system might assign the same likelihoods of rejection for both and so, the system cannot decide which action should be taken. In addition, the traditional measure of confidence simply reuses what is being predicted (Campbell et al., 2005) or simply provides an overall assessment for the system like EER or min-DCF (Brümmer and Du Preez, 2006), which are also unable to show the difference between ② and ③. The underlying reason for this inability is that the prediction made by the current system cannot express "how likely" and "how confident" separately at the same time — they are conflated in a single score, as demonstrated in other machine learning systems in domains such as image classification and text-based sentiment analysis (Sato et al., 2018; Sheikholeslami et al., 2020).

Thus, to better address the problem of confidence, we consider quantifying the prediction uncertainty (uncertainty of the prediction) of the ASV system. There are numerous sources of uncertainty in ASV systems to influence their predictions, including aleatoric (irreducible) and epistemic (theoretically reducible) uncertainty (Senge et al., 2014; Vogt and Sridharan, 2008). Quantifying them can inform the decision of whether to trust the output of an ASV system or if additional data is needed before a confident decision can be made. It provides a view from the aspect of knowledge level (perceived by model), i.e., the ASV can show "what/how much knowledge the ASV decision is based on?". For example, the uncertainty level is low when in ① ③ and thus the decisions can be trusted, while in ②, the system shows high uncertainty so the predicted score may not be suitable for deriving the final decision. Taking a step further, in high uncertainty cases, by exploring which component (aleatoric or epistemic) is the dominating one, the system will also show the reason why there is such high uncertainty. For cases a, b in ② of Fig. 1, there might be high levels of aleatoric uncertainty since it is difficult to capture speaker information under noisy conditions, whereas in c, there might

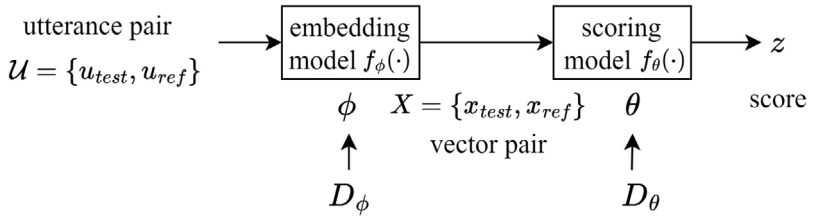


Fig. 2. The brief pipeline of a modern ASV.

be a high level of epistemic uncertainty since the information amount is limited due to the short length of the speech. Thus, if the uncertainty in ASV predictions can be quantified and decomposed, the system will be more valid for application purposes.

In this work, our primary objective is to extend the ASV such that it can continuously show its confidence level during operational stages. To achieve this, we propose to quantify uncertainty from the model's view showing how differently the model might perceive the input under various conditions. Our approach involves enumerating a set of plausible models trained from the same data and assessing the divergence in their predictions for the same input. We built a novel Bayesian framework to sample the model parameter space to get that set of models and used mutual information to measure the disagreement between possible predictions. This contrasts with the traditional system which only calculates a single similarity score and derives decisions based on it, since our framework results in multiple models and a score distribution. Aleatoric uncertainty is an inherent component of model prediction. Therefore, the mean of the score distribution is considered an estimation of aleatoric uncertainty whose accuracy benefits from averaging. Meanwhile, the spread of the distribution signifies the model uncertainty (epistemic). The framework achieves the dual purpose of expressing prediction and its confidence simultaneously. In detail, we build and evaluate our framework around the well-established x-vector PLDA system. The effectiveness of our approach is demonstrated by showing how the quantified uncertainties change in response to the varying speech variability in test utterances. Our work highlights that the quantified uncertainty shows the conditions of the test input, such as its length and the noise level, and they can be distinguished by checking if the dominant uncertainty is aleatoric or epistemic.

2. Background

2.1. Uncertainty arising from model parameters

In our work, we include the uncertainty from different model parameters into a novel Bayesian framework and quantify different uncertainties under it. To describe our framework, we illustrate the brief pipeline of modern ASV in Fig. 2

In the development stage, model parameters ϕ and θ are trained using datasets D_ϕ and D_θ respectively. During the test phase, the output similarity score z is calculated for the speech pair $\mathcal{U}$ formed by finding a reference utterance u_{ref} for the input u_{test} in consideration of the claimed speaker identity s, i.e., $\mathcal{U} = \{u_{test}, u_{ref}\}$. In the case when the model parameters ϕ and θ have uncertainty in their selections, the score z should also show uncertainty and we thus describe possible values for z using a distribution (conditioned on existing knowledge in training data D_ϕ and D_θ) denoted by:

$$p(z|\mathcal{U}, D_\phi, D_\theta) = \int \left(\int p(z|\mathcal{U}, \theta, \phi) p(\theta|D_\theta, \phi) d\theta \right) p(\phi|D_\phi) d\phi \tag{1}$$

where $p(\phi|D_\phi)$ is the model posterior of the embedding model; $p(\theta|D_\theta, \phi)$ is the model posterior of the scoring model. Notably, θ is usually conditioned on knowing ϕ because D_θ is obtained by forward passing D_ϕ through ϕ. In practice, uncertainty in $p(z|\mathcal{U}, D_\phi, D_\theta)$ is usually neglected since a point estimation for $p(z|\mathcal{U}, D_\phi, D_\theta)$ is used in current ASV systems for computational simplicity. In this work, we explore the uncertainty in this distribution.

Existing systems only use point estimation in place of $p(z|\mathcal{U}, D_\phi, D_\theta)$ by obtaining one sample of $p(\phi|D_\phi)$ and $p(\theta|D_\theta, \phi)$. This makes for computationally efficient implementations but ignores the uncertainty reflected in these distributions. Our Bayesian framework includes such uncertainty and uses it to predict how much confidence there is in each system prediction. In addition to the Bayesian framework, bootstrapping (Tibshirani and Efron, 1993), and conformal prediction (Angelopoulos and Bates, 2021) also study uncertainty in a similar way but the former fails to formulate model parameter space precisely and the latter only considers possible predictions.

2.2. Related work

Early research in the ASV field has identified speech variabilities as major sources of uncertainty and developed methods to address them (Mandasari et al., 2012; Kheder et al., 2015; Athulya and Sathidevi, 2017). Many efforts also have been made to address "uncertainty" problems such as domain mismatch or spoofing (Zhang et al., 2023; Chen et al., 2021; Süslü et al., 2021). The primary target of existing research is achieving higher accuracy, only taking uncertainty as data variation that degrades the system performance. The approaches mainly focus on either combating the uncertainties or incorporating them with extended models.

However, few of them provide a view of uncertainties from the model's perspective. There is also limited work in quantifying uncertainty as a standalone indicator showing the level of confidence presented by the model.

Existing works on ASV uncertainties can be broadly categorised into those that focus on the embedding procedure, and those that focus on the backend scoring. A number of approaches that fall in the first category assume the embedding $p(x|u)$ is Gaussian and seek to incorporate its parameters into a point estimate of the embedding vector (Kuzmin et al., 2022; Brümmer et al., 2018; Ribas and Vincent, 2019) or propagate it to an appropriate backend which can then incorporate it into scoring (Cumani et al., 2014; Stafylakis et al., 2013). Other related approaches include the use of Bayesian x-vectors (Li et al., 2020), which explore the uncertainty in $p(\phi|D_\phi)$ by applying Bayes-by-backprop (Blundell et al., 2015) to the first convolutional layers of the classic x-vector embedding network, combined with Bayesian model average (Fragoso et al., 2018) to improve the generalisability of the embedding. Finally, the recently presented Xi-vectors (Lee et al., 2021) improve speaker embedding, by incorporating frame-level data uncertainty into network training.

The focus of methods that fall in the second category is on backend model uncertainty. These include the Bayesian PLDA and a number of variants and improvements that utilise the information about the spread of $p(\theta|D_\theta)$. (Villalba and Brümmer, 2011) uses variational inference to approximate $p(\theta|D_\theta)$ and tries to marginalise it out when calculating the loglikelihood ratio. Another work provides analytical procedures for the Bayesian estimation of PLDA (Borgström, 2021), which is assisted by coordinate ascent variational inference, CAVI (Blei et al., 2017). The method results in a point estimation for $p(\theta|D_\theta)$, which maintains high accuracy under the fuzzy label condition in training data D_θ.

Our work is fundamentally different from the mentioned ones since our proposed novel framework uses exact sampling of $p(\theta|D_\theta)$ and approximations to $p(\phi|D_\phi)$ to quantitatively estimate uncertainties in the context of ASV systems. Instead of making the system robust against speech variabilities, we view uncertainty from the model and seek alignment between quantified uncertainties and the severity of speech variabilities in the input. The work mainly focuses on exploring the uncertainty possibly originating from problems such as speech variabilities or insufficient modelling, rather than improving the system accuracy by addressing those specific problems.

3. The proposed scheme

We propose a novel approach that replaces the single score used in the existing ASV paradigm with a distribution of scores, as shown in Fig. 3. Such distribution simultaneously provides similar information and confidence in the prediction. To obtain this score distribution, we apply sampling techniques to the components of the ASV system.

In the existing ASV shown in Fig. 3(a), the decision-making is based on a point estimation of the score. The deviation between the score and a threshold reflects the decision confidence. In contrast, as shown in Fig. 3(b), the involvement of model uncertainty results in a distribution of scores. The sharpness of the score distribution serves as the indicator of the confidence relating to the similarity score and the subsequent decision based on it. Notably, the quantified uncertainty is an auxiliary output and does not change the score calculation and the hard decision (acceptance/rejection).

ASV systems are usually not end-to-end since the back-end model is dependent on the embedding model. As usually in existing ASV systems, the embedding network $f_\phi(\cdot)$ is initially trained as a classifier to discriminate a set of speakers in dataset D_ϕ. Subsequently, the back-end PLDA model $f_\theta(\cdot)$ is trained using the embeddings D_θ extracted from D_ϕ using f_ϕ, i.e., $D_\theta = f_\phi(D_\phi)$. Therefore, we propose a hierarchical Bayesian framework to organise different model posterior distributions and calculate score distribution based on them, as shown in Fig. 4:

By realising the framework in Fig. 4, the score distribution $p(z|\mathcal{U}, D_\phi, D_\theta)$ we defined in Eq. (1) can be obtained. In Fig. 4, both $p(\phi|D_\phi)$ and $p(\theta|D_\theta, \phi)$ need to be approximated by existing Bayesian learning techniques such as variational inference (such as Bayes-by-backprop) (Blundell et al., 2015), Markov chain Monte-Carlo (MCMC) (Brooks et al., 2011), stochastic gradient Markov chain Monte-Carlo (SGMCMC) (Nemeth and Fearnhead, 2021), Monte-Carlo dropout (MCDP) (Gal and Ghahramani, 2016), etc. However, implementing the whole scheme requires substantial computational resources since the calculation of score distribution for a single input requires parallelling hundreds of models. In this work, we therefore only explore the effects when either $p(\phi|D_\phi)$ or $p(\theta|D_\theta, \phi)$ is marginalised, retaining another as a reliable point estimation. Thus, $p(z|\mathcal{U}, D_\phi, D_\theta)$ is approximated using the following equations:

$$p(z|\mathcal{U}, D_\phi, D_\theta) \approx \int p(z|\mathcal{U}, s, \hat{\theta}, \phi)p(\phi|D_\phi)d\phi \tag{2}$$

or

$$p(z|\mathcal{U}, D_\phi, D_\theta) \approx \int p(z|\mathcal{U}, s, \theta, \hat{\phi})p(\theta|D_\theta, \hat{\phi})d\theta \tag{3}$$

where $\hat{\theta}$ is the point estimation for PLDA model parameters conditioned on knowing embedding model ϕ, and $\hat{\phi}$ is the point estimation for embedding network parameters.

We interpret the uncertainty included in the score distribution using either score variance or mutual information. Both metrics reflect the variation among possible predictions and the level of agreement on them. To present these metrics which show uncertainty originating from a plausible set of models (epistemic), we use the notation U_M for them. The quantification by variance is simply given by:

$$U_M(\mathcal{U}) = Var(z) \tag{4}$$

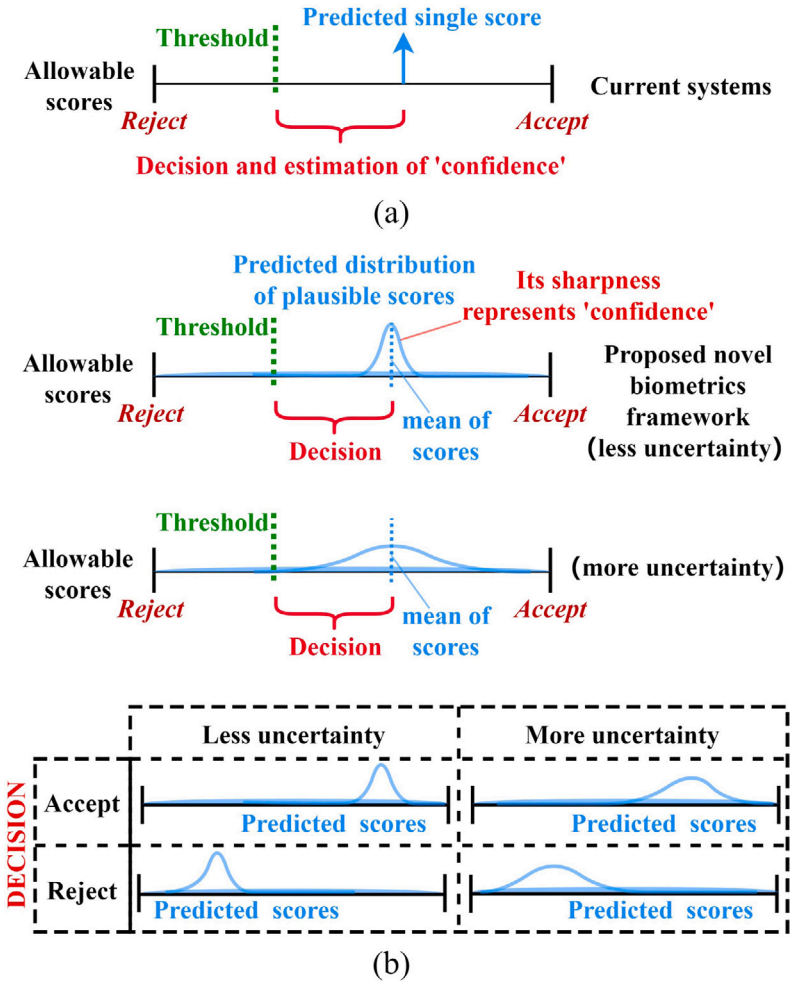


Fig. 3. The proposed scheme to quantify the uncertainties in ASV predictions.

To quantify uncertainty by mutual information, the scores need to be turned into probabilities of acceptance and rejection. This is still an open question discussed in score calibration works (Cumani, 2020). Here, based on the definition of loglikelihood ratio score given by the PLDA back-end, we use a simple shifted sigmoid function for the conversion:

$$\vec{p} = \left[P_{accept}, P_{reject}\right] = \left[\frac{1}{1+e^{-\delta}}, \frac{e^{-\delta}}{1+e^{-\delta}}\right] \tag{5}$$

where $\delta = z - \lambda$, the deviation between score z and the threshold λ. Finally, the mutual information between prediction and model parameters $I(\vec{p}; \theta | \mathcal{U})$ is calculated using the procedure as proposed by Malinin and Gales (2018):

$$\underbrace{U_M(\mathcal{U})}_{\text{epistemic uncertainty}} = \underbrace{H\left[E_{p(\theta,\phi|D_\phi)}\left[\vec{p}|\mathcal{U},\theta,\phi\right]\right]}_{\text{total uncertainty } U_T} - \underbrace{E_{p(\theta,\phi|D_\phi)}\left[H\left[\vec{p}|\mathcal{U},\theta,\phi\right]\right]}_{\text{aleatoric uncertainty } U_A} \tag{6}$$

where the total uncertainty U_T for input x is given by the entropy (denoted by H in Eq. (6)) of the prediction mean and the aleatoric component U_A is given by the averaged entropies of all predictions. Finally, U_M is used as a quantification for the epistemic component reflecting the degree of conflict between the predictions. According to information theory (Reza, 1994), the unit of U_T, U_A and U_M is Nat when the base of the logarithm in the entropy calculation is the natural constant e.

The decomposition of uncertainty (originally by Depeweg et al. (2017)) described in Eq. (6) can be further explained with an illustration shown in Fig. 5, which presents the case when such technique is applied to a simple binary classification task over 2-D space $[-10, 10]^2$.

In Fig. 5, we can clearly see the aleatoric component signifies the overlapping region between two classes of training samples while the epistemic component is more significant in areas where less training data is available. The intuition is that more training data better constrains the model, resulting in more similar predictions.

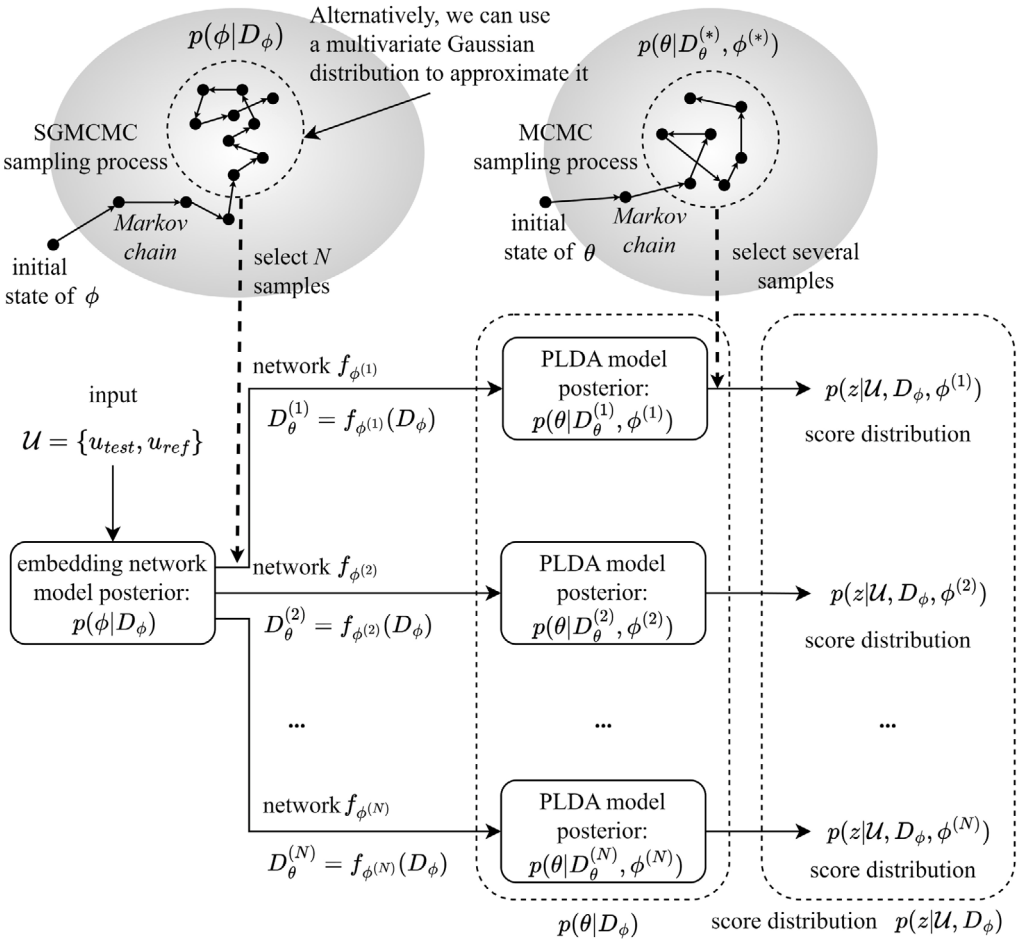


Fig. 4. Hierarchical framework to calculate score distribution: there are two model posterior distributions included: the one for embedding model $p(\phi|D_\phi)$ and the one for PLDA $p(\theta|D_\theta)$. The embedding model $p(\phi|D_\phi)$ can be approximated (sampled) using Bayesian learning, while $p(\theta|D_\theta)$ requires marginalising variable ϕ by having multiple posterior distributions with each one also being sampled using Bayesian learning. In this work, we evaluate $p(\phi|D_\phi)$ approximated by stochastic gradient Markov chain Monte-Carlo (SGMCMC) or a Gaussian distribution provided by variational inference, and $p(\theta|D_\theta)$ approximated by Markov chain Monte-Carlo (MCMC). The resulting multiple models finally provide a score distribution reflecting the uncertainty, which achieves our scheme described in Fig. 3.

4. Bayesian learning for ASV components

In this section, we present the formulation of model posterior distributions and describe the techniques we employ for sampling from them as the basic procedures to obtain score distribution.

4.1. Bayesian learning for generative Gaussian PLDA model

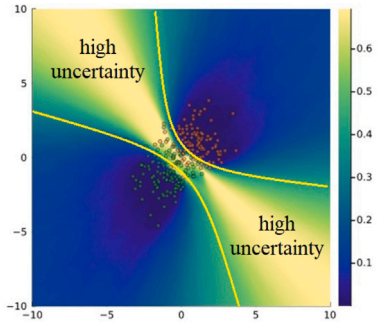
Gaussian PLDA model is a widely used parametric model which describes a two-step Gaussian generative process. It is used as the default back-end scoring model in ASV, which is applied after the embedding process. Due to its well-structured parameters, getting exact samples from PLDA model posterior, denoted as $p(\theta|D_\theta)$, can be achieved by employing the Hamiltonian Monte-Carlo (HMC) algorithm (Betancourt, 2017).

4.1.1. Formulation of the PLDA posterior distribution

The prerequisite step of applying sampling techniques to the PLDA model is to formulate its model posterior distribution. According to the Bayes rule, the log density of the posterior distribution of PLDA parameters θ can be written as:

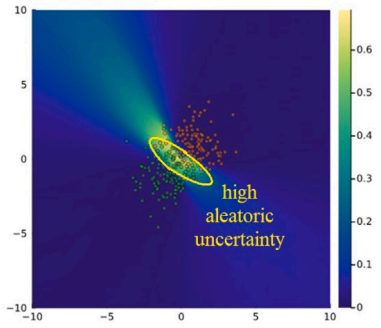
$$ln(p(\theta|D_\theta)) = Z + ln(p(D_\theta|\theta)) + ln(p(\theta)) \tag{7}$$

where, Z denotes the normalising constant, and θ denotes the PLDA model parameters which includes a global centre: m, between class covariance matrix $\mathbf{\Phi_b}$, and with-in class covariance matrix $\mathbf{\Phi_w}$. By offsetting the data by the global mean m, the parameter set



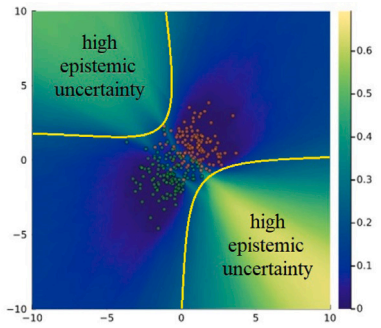
(a) Total Uncertainty U_T

In this binary classification task, a simple neural network model is trained to distinguish two clouds of samples (red and green ones). The heatmap represents the level of Entropy-based uncertainty at each position in the 2-D space. Under the Bayesian framework as presented in Eq.6, the model provides uncertain predictions when the input sample is from the bright-yellow regions (as specified in (a)).



(b) Aleatoric Uncertainty U_A

Aleatoric uncertainty can be estimated by calculating the Entropy of the averaged predictions (from different models). In (b), this value is high in the region where two sample clouds are overlapping, indicating the inherent difficulty in classifying green and red dots in that region.



(c) Epistemic (model) Uncertainty U_M

By subtracting aleatoric uncertainty from total uncertainty we can obtain epistemic uncertainty. It is high where the input resides in regions aside or far away from the data clouds, indicating the model does not have sufficient information to classify the input – a lack of knowledge.

Fig. 5. Decomposition of the total uncertainty into aleatoric and epistemic ones: Total uncertainty (a) = Aleatoric uncertainty (b) + Epistemic uncertainty (c). The high aleatoric uncertainty (i.e., known–unknown) represents the overlapping region between two classes and high epistemic uncertainty (i.e., unknown–unknown) represents regions where less training samples are provided. (The colour represents uncertainties)

is reduced to only the covariance matrices. i.e., $\theta = \{\mathbf{\Phi_b}, \mathbf{\Phi_w}\}$. D_θ represents the training data which consists of K speakers with each one providing C speaker embeddings:

$$D_\theta = \{x_1^{(1)}, x_2^{(1)} \dots x_1^{(2)}, x_2^{(2)} \dots x_1^{(K)}, x_2^{(K)}, \dots, x_c^{(K)}\} \tag{8}$$

In Eq. (9), Gaussian PLDA log-likelihood function is known according to Ioffe (2006):

$$\begin{aligned} &ln(p(D_\theta|\theta)) = const. - \tfrac{K}{2}\{ln(\mathbf{\Phi}_b + \tfrac{1}{c}\mathbf{\Phi}_w)\} + \\ &\tfrac{K}{2}\{tr((\mathbf{\Phi}_b + \tfrac{1}{c}\mathbf{\Phi}_w)^{-1}\mathbf{S_b}) + (c-1)ln|\mathbf{\Phi}_w| + c \cdot tr(\mathbf{\Phi}_w^{-1}\mathbf{S_w})\} \end{aligned} \tag{9}$$

where tr is the trace calculation; $\mathbf{S_w}$ and $\mathbf{S_b}$ are the within-class and between-class scatter matrices, given by:

$$\mathbf{S_w} = \frac{1}{N}\sum_k \sum_i (x^{(i)} - m_k)(x^{(i)} - m_k)^T \tag{10}$$

$$\mathbf{S_b} = \frac{1}{N}\sum_k n(m_k - m)(m_k - m)^T \tag{11}$$

In Eqs. (10) and (11), $x^{(i)}$ denotes the ith sample in class k; m_k denotes the centre of class k. Finally, to obtain the model prior, $p(\theta) = p(\mathbf{\Phi_b})p(\mathbf{\Phi_w})$, we assume Wishart distributions for $\mathbf{\Phi_b}$ and $\mathbf{\Phi_w}$, which are usually used as priors for positive-definite matrices.

$$\mathbf{\Phi_b} \sim Wishart(\upsilon_b, \mathbf{S_b}/\upsilon_b), \mathbf{\Phi}_w \sim Wishart(\upsilon_w, \mathbf{S_b}/\upsilon_w) \tag{12}$$

where υ_b and υ_w are the degrees of freedom deciding the sharpness of those Wishart distributions.

4.1.2. Sampling the PLDA posterior distribution using HMC

To sample the posterior, $p(\theta|D_\theta)$ as illustrated in Fig. 4, Hamiltonian Monte-Carlo (HMC) is chosen because it explores the exact posterior fast and is used as the ground truth when compared to others (Yao et al., 2019). HMC creates a chain of states, i.e. $\{\theta_0, \theta_1, \theta_2, \dots \theta_{t-1}, \theta_t \dots\}$, as a realisation of sampling. This requires building a transition $p(\theta_t|\theta_{t-1})$ which is obtained by period-by-period simulating a reversible physical process defined by Hamiltonian Dynamics:

$$\begin{cases} \frac{\partial q}{\partial t} = M^{-1}p \\ \frac{\partial p}{\partial t} = -\frac{\partial U}{\partial q} \end{cases} \tag{13}$$

where p is the momentum periodically generated from a diagonal multi-variate Gaussian distribution, and M is the mass matrix set to identity matrix I. We set the potential energy function $U(\theta)$ in HMC to $-ln(p(\theta|D_\theta))$, i.e., the negative log-density, and the kinetic energy function to $K(p) = \frac{1}{2}p^T M^{-1} p$. Then, the Hamiltonian total energy function is $H(\theta, p) = U(\theta) + K(p)$. We used the Leapfrog algorithm (Betancourt, 2017) (parameterised by step number l and step size ϵ) as the discretisation of Hamiltonian dynamics to propose a new state (θ_t, p_t) depending on the previous (θ_{t-1}, p_{t-1}). The proposed state was accepted with the Metropolis–Hastings(MH) ratio (Chib and Greenberg, 1995), in this case, it is $exp\left(H(\theta_t, p_t) - H(\theta_{t-1}, p_{t-1})\right)$ to ensure there is no energy loss during each simulation. Finally, this process results in a chain of states, and some are selected as the output. In the HMC algorithm, it strictly ensures an exact sampling of $p(\theta|D_\theta)$ due to the inclusion of the MH ratio. For the detailed algorithm, refer to Appendix A

4.2. Sampling the model posterior of embedding neural network

X-vector embedding vectors are extracted by a time-delay neural network (TDNN). The training of this network requires propagating millions of speech samples through it until it learns to discriminate thousands of speakers. In theory, its model posterior can still be exactly sampled using MCMC. However, the computational resource required is prohibitively large in this case, because while following the sampling process in 4.2.2, accurately proposing a new state from its previous requires calculating the full gradient based on the entire dataset. Thus, alternative methods that only require stochastic gradient estimates should be explored in this case.

4.2.1. Posterior distribution and stochastic gradient for neural network

We begin by formulating the posterior distribution for the embedding model. Consider the training data D_ϕ, for x-vector embedding model $f_\phi(\cdot)$ we assume a prior distribution for ϕ in the form of diagonal multivariate Gaussian distribution, $p(\phi) = \mathcal{N}(0, \sigma_p^2 \cdot I)$, where σ_p is the standard deviation of the Gaussian distribution, and I is the identity matrix. Let $x^{(i)}$ be the ith sample in D_ϕ with N_D samples in total. Then the likelihood function can be expressed as $p(D_\phi|\phi) \propto \prod_{i=1}^{N_D} p(x^{(i)}|\phi)$. Consequently, the log-posterior is given by:

$$ln(p(\phi|D_\phi)) = const. + \sum_{i=1}^{N_D} ln\left[p(x^{(i)}|\phi)\right] + ln\left[p(\phi)\right] \tag{14}$$

The full gradient of the log-posterior can also be given:

$$\nabla ln(p(\phi|D_\phi)) = \sum_{i=1}^{N_D} \nabla ln\left[p(x^{(i)}|\phi)\right] + \nabla ln\left[p(\phi)\right] \tag{15}$$

Since N_D can be as large as millions, sampling $p(\phi|D_\phi)$ using $\nabla ln(p(\phi|D_\phi))$ is not realistic. Thus, we turn to the stochastic gradient method. It uses an n-sample mini-batch of x to estimate the full gradient instead of exactly calculating it. The estimation to $\nabla ln(p(\phi|D_\phi))$ is formulated as:

$$\nabla ln(p(\phi|D_\phi)) \approx \frac{N_D}{n}\sum_{i=1}^{n} \nabla ln\left[p(x^{(i)}|\phi)\right] + \nabla ln\left[p(\phi)\right] \tag{16}$$

Popular Bayesian learning methods using such stochastic gradient (as in (16)) include stochastic gradient Hamiltonian Monte-Carlo (SGHMC) (Chen et al., 2014), stochastic gradient Langevin dynamics (SGLD) (Welling and Teh, 2011), Bayes-by-backprop etc. In our work, we use SGLD due to its simplicity in calculation. Additionally, Bayes-by-backprop is also applied in comparison with SGLD, showing the effect of using different techniques in the framework. Notably, although the calculation is simplified with the mentioned methods, SGHMC and SGLD cannot perform exact sampling of the posterior, and Bayes-by-backprop only provides a rough estimation for one mode of $p(D_\phi|\phi)$.

Table 1
Different settings to quantify predictive uncertainty in ASV.

Settings	Embedding network	PLDA back-end
Baseline	Point estimation by SGD	Point estimation
Setting 1	Point estimation by SGD	Ensemble by Bayesian PLDA
Setting 2	Ensemble by SGLD	Multiple point estimations
Setting 3	Ensemble by Bayes-by-backprop with early stopping	Multiple point estimations
Setting 4	Ensemble by Bayes-by-backprop with sufficient training	Multiple point estimations

4.2.2. Bayesian learning for TDNN using SGLD

When using SGLD, like the vanilla stochastic gradient descent (SGD), the model parameter ϕ is iteratively updated by:

$$\Delta\phi_{t+1} = \frac{\varepsilon_t}{2}\left(\frac{N_D}{n}\sum_{i=1}^{n}\nabla ln\left[p(x_i|\phi_t)\right] + \nabla ln\left[p(\phi_t)\right]\right) + \upsilon_t \tag{17}$$

where $\upsilon_t \sim \mathcal{N}(0, \varepsilon_t I)$ is the injected Gaussian randomness which encourages the state ϕ_t to explore the posterior landscape; ε_t is the learning rate which satisfies $\sum_t \varepsilon_t = \infty; \sum_t \varepsilon_t^2 < \infty$ to ensure the convergence of the algorithm. But in practice, ε_t is set to a small constant value. Finally, after training, a sub-sequence of $\{\phi_0, \phi_1, \phi_2, \ldots\ \phi_t, \ldots\}$ is used as the samples from $p(\phi|D_\phi)$.

4.2.3. Bayesian learning for TDNN using Bayes-by-backprop

Bayes-by-backprop is another widely used method for approximating the model posterior. It assumes a variational Gaussian distribution $q(\phi;\varphi_q)$ (parameterised by φ_q) as an approximation to $p(\phi|D_\phi)$. An estimation to parameter φ_q can be inferred by minimising the reverse KL-divergence between $p(\phi|D_\phi)$ and $q(\phi;\varphi_q)$:

$$\hat{\varphi_q} = \arg\min_{\varphi_q} L, where\ L = KL\left[q(\phi;\varphi_q)\parallel p(\phi|D_\phi)\right] \tag{18}$$

where $\varphi_q = \{\mu_q, \rho_q\}$; μ_q denotes the mean of the assumed Gaussian distribution and ρ_q is the reparametrised standard deviation ($\sigma_q = ln\left(1+e^{\rho_q}\right)$) of the assumed Gaussian distribution.

We optimise the KL-divergence in Eq. (18) stochastically by randomly extracting one model from $q(\phi;\varphi_q)$ and updating φ_q based on the gradients calculated. We assume ϵ is a standard normal variable, then one sample from $q(\phi;\varphi_q)$ can be expressed by $\Phi = \mu_q + ln\left(1+e^{\rho_q}\right)\cdot\epsilon$. By choosing a small value α as the learning rate, $\hat{\varphi_q}$ can be updated by SGD using the following formula:

$$\Delta\mu_q = -\alpha\cdot\frac{\partial L}{\partial\mu_q} = -\alpha\cdot\frac{\partial ln\left(p(\Phi|D_\phi)\right)}{\partial\Phi} \tag{19}$$

$$\Delta\rho_q = -\alpha\cdot\frac{\partial L}{\partial\rho_q} = -\alpha\cdot\frac{1}{1+e^{-\rho_q}}\left(\frac{1}{ln(1+e^{\rho_q})}+\epsilon\cdot\frac{\partial ln(p(\Phi|D_\phi))}{\partial\Phi}\right) \tag{20}$$

In Eq. (19) and (20), term $\partial ln(p(\Phi|D_\phi))/\partial\Phi$ is estimated by first choosing minibatch as in Eq. (16) and then executing backpropagation. After iterating the update procedure, $q(\phi;\hat{\varphi_q})$ can be taken as an approximation to the true posterior $p(\phi|D_\phi)$.

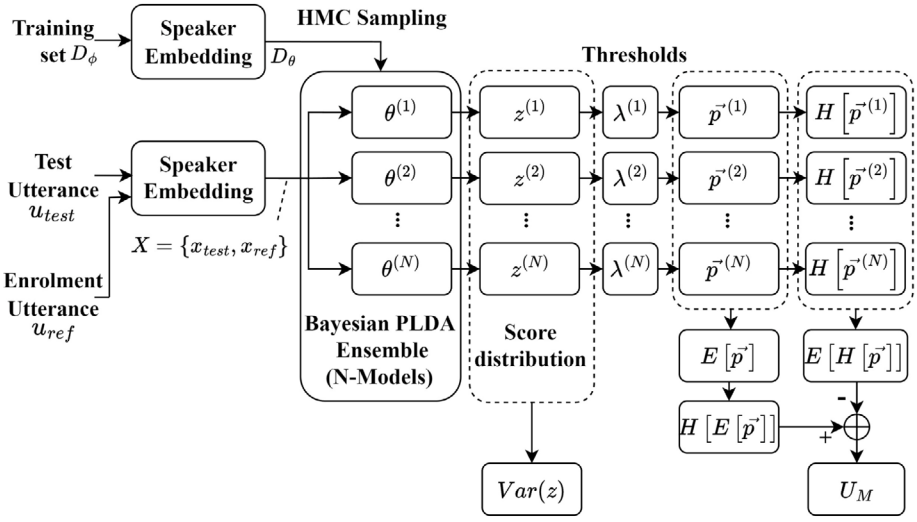
5. Experiment setups

The proposed approach to quantifying uncertainties in ASV was implemented as depicted in Fig. 6. Following Eqs. (2) and (3) in Section 3, we established two schemes by integrating either PLDA model posterior (Fig. 6(a)) or embedding model posterior (Fig. 6(b)) into x-vector ASV paradigm. The involvement of Bayesian learning results in multiple repeats of ASV systems with each one having "TDNN-PLDA-threshold" pattern. In the schemes depicted in Fig. 6, the threshold in each repeat is dedicated and is determined by suiting the evaluation set for an optimal EER. For a single test instance, each threshold is used to convert the according score into probabilities using the shifted sigmoid function described in Eq. (5).

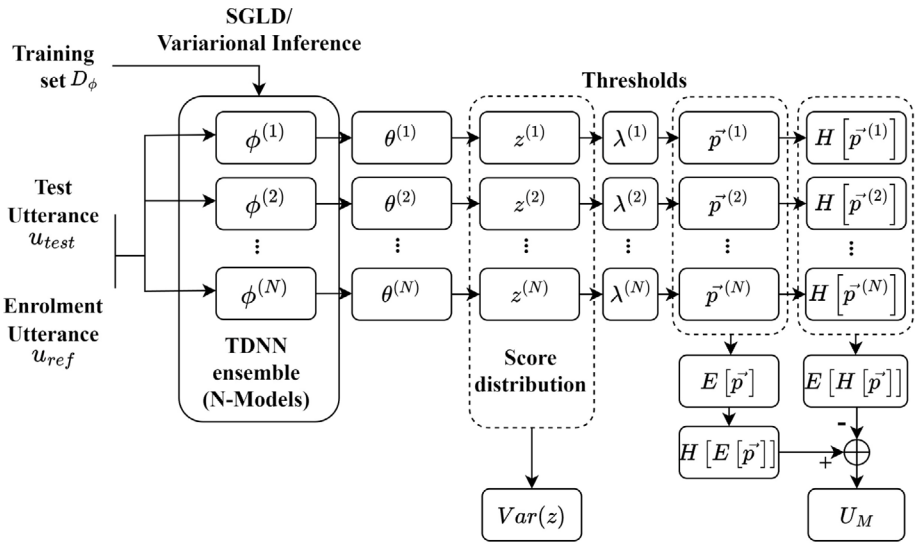
The estimated uncertainties may vary depending on factors such as which component of the system is sampled, the methods used for sampling, or the extent of Bayesian learning executed. Thus, we conducted several sets of experiments to evaluate the validity of the approach. We consider 4 settings developed around the baseline, as detailed in Table 1:

We expect reductions in the quantified uncertainties when more appropriate data (more sufficient and less variability) is available and vice versa. Thus, under settings 1∼4, we altered the duration of the utterances, or added noise to them to ascertain whether the estimated uncertainty reflects these changes to conditions. Under each condition, we calculated error rates, score variance, total predictive uncertainty U_T, expected data uncertainty U_A (aleatoric), predictive model uncertainty U_M (epistemic) to show the change. Specifically, we used unaltered utterances for training and enrolment but varied the duration (achieved by truncation) or noise levels (by adding noise) of the test utterances and estimated uncertainties for them. In these experiments, the system learned from clean and long utterances but was tested using "bad" inputs with "reduced/false knowledge". To show the averaged behaviour, we calculated the sum of uncertainty over all instances in the test dataset (i.e., $\sum Var(z)$, $\sum U_T$, $\sum U_A$ and $\sum U_M$).

Another consideration is about the choice for threshold because it directly influences the uncertainty calculation. Under settings 1∼4, we considered two options: under the original baseline conditions, finding the threshold when EER is achieved and fixed its



(a) This scheme exploits the sampling of PLDA model posterior. It is an implementation of Eq.3 in Section 3. The embedding network is the same as the one used as the baseline.



(b) This scheme exploits the sampling of the TDNN model posterior. It is an implementation of Eq.2 in Section 3. Several TDNN nets were obtained by Bayesian learning and each backend PLDA was trained using one of TDNNs in the ensemble. This means having N working ASVs simultaneously.

Fig. 6. An overview of the proposed methods to quantify ASV predictive uncertainties. In the test phase, model uncertainty for each input was estimated from the output score distributions. Score variance and entropy-based uncertainties $U_T = H[E[\vec{p}]]$, $U_A = E[H[\vec{p}]]$ and U_M (mutual information) were calculated.

value for subsequent tests; or else allowing the threshold to be changed to achieve EER over the test set under different conditions. The former indicates the case when ASV operates without being informed of the unknown condition while the latter indicates the case when ASV is informed of the condition change and is allowed to recalibrate itself with labelled test samples. The former is better associated with the real application scenario but suffers from miscalibration. The latter might provide better estimations of the uncertainty due to higher level of calibration, but it involves quantifying uncertainty by foreknowing the information of the unknown, which is not ideal in logic.

Specifically, under Setting 1, whether the quantified epistemic uncertainty behaves with the quality of the training data signifies if sampling the exact posterior is successful or not. To test this, we ran our quantification scheme depicted Fig. 6(a) several times using varying amounts of data for D_θ. The amount was controlled by the number of samples per class or the number of classes in the training dataset.

All the detailed setups regarding the baseline ASV system training, Bayesian learning for PLDA and TDNN, and the protocols for introducing duration and noise changes will be outlined in Section 5.1∼5.4:

5.1. Training baseline x-vector system on Voxceleb1 dataset

We trained and evaluated the ASV system using the Voxceleb1 dataset (Nagrani et al., 2020) which contains short speech utterances in two partitions, one for development and one for test. The development partition has 148642 utterances of 1211 speakers, and the test partition has 4874 utterances of 40 speakers. To evaluate our scheme, we used 37720 enrol-test pairs selected from the test partition according to the list' List of trial pairs - VoxCeleb1'.

We used x-vectors as the speaker embedding system prior to scoring. To train x-vector embedding model, we pre-processed the training utterances by extracting 24-dimensional MFCCs from them after a simple energy-based voice activity detector. MFCCs were then normalised over a sliding window of 1.5 s long. In commonly adopted settings, those normalised MFCCs should be randomly resampled such that each speaker had 3700∼5000 MFCC matrices with each one 2∼8s long. However, in this work, to stable the gradient and let the setup be more suitable for Bayesian learning, we fixed the length of the MFCC matrices to 3 s only. We omitted data augmentation to assess how ASV performs when test utterance contains speech variability rather than letting the ASV learn how to eliminate the impact of speech variabilities during training.

To optimise the TDNN model parameters, we used the ADAM optimiser with a learning rate of 5×10^{-5} and batch size of 100. The network was trained for 8∼10 epochs and x-vectors were extracted for all labelled speech in the development set. After extraction, the dimension of the x-vectors was reduced by LDA (linear discriminant analysis) from 512 to 200. Finally, those x-vectors with reduced dimensions were used to train a point estimation of PLDA which served as the back end. It was trained using the EM (expectation maximisation) algorithm (Ding, 2018) for 25 epochs.

5.2. Details for HMC sampling (Bayesian inference/learning)

To generate the PLDA model ensemble (refer Fig. 6(a)) we employed HMC sampling. We initialised with a random state and then ran 200 iterations with a step number of 5 and a step size of 0.001. To alleviate the effect of bad geometry on the posterior distribution, we used the delayed rejection HMC (DRHMC) (Modi et al., 2023) which allowed using of two different step sizes simultaneously in the Leapfrog algorithm. The step reduction factor for DRHMC was 5. During the sampling stage, we adjusted the step number to 60∼100 depending on the situation and set the step size to 0.01. DRHMC ran for 1500 iterations until convergence. Finally, we selected the ending 1000 states as the sampled PLDA models. The sampling process produces 2 chains alternately and took 3∼7 h.

To check the convergence (ergodicity of the chains), we used the produced 2 chains to calculate the potential scale reduction statistics (Gelman and Rubin, 1992) (known as $\hat{R}$) with an optimised splitting scheme (Vehtari et al., 2021). The recommended criterion is typically $\hat{R}-1>0.01$ for a single parameter. In our experiment, we used a looser criterion of 0.1 for all the 40200 parameters. For evaluation efficiency, we refined the 1000-sample ensemble to 100-sample by random sampling. This adjustment kept convergence and provided stable quantified uncertainties.

5.3. Details for TDNN training using SGLD and Bayes-by-backprop

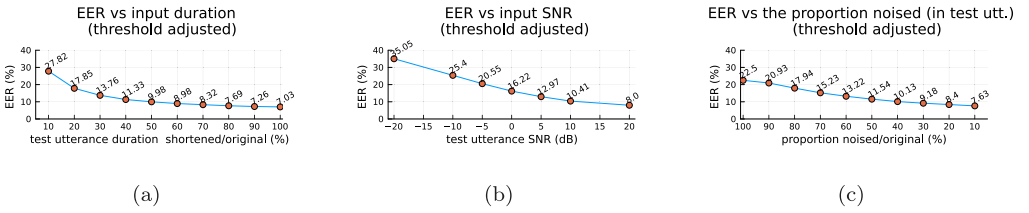
We created an ensemble of TDNN (refer Fig. 6(b)) by using either SGLD or Bayes-by-backprop methods. For both methods, we set a sharp prior $p(\phi)=\mathcal{N}(0,0.01\cdot I)$ to reduce the difficulty in inference. For the SGLD method, we kept using the training setup in Section 5.1 but replaced the ADAM optimiser using the update scheme in Eq. (17). The learning rate was fixed to 5×10^{-6}. We trained the network from scratch for more than 20 epochs and in each epoch, the parameters were regularly recorded. The networks in the TDNN ensemble were chosen from the last 20 networks saved during training.

For Bayes-by-backprop, we performed full inference for all the parameters of TDNN. However, updates described in Eq. (19) and (20) do not provide efficient improvements in the early stage of training. Thus, we followed the scheme used in (Izmailov et al., 2021). The centre μ_q was initialised using the parameters of the baseline directly, and as for ρ_q, we initialised it to −5.0. This enables the assumed Gaussian distribution to be centred at a proper point estimation in the beginning. In the subsequent training, it could gradually find how to spread and move itself to fit one mode of the TDNN model posterior. We continue using the training setups in Section 5.1 and further train the baseline using Eq. (19) and (20) to update the parameters for extra 10 epochs. The learning rate for μ_q was 0.0001 and the one for ρ_q was 0.001. After training, the TDNN models in posterior can be directly sampled from the variational Gaussian distribution $\mathcal{N}(\mu_q,(ln(1+e^{\rho_q}))^2)$.

5.4. Varying length and adding noise to test utterances

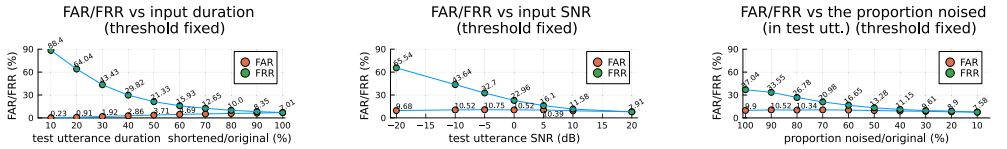
As previously mentioned, we evaluate the uncertainties by varying one speech variability. We chose the duration and the noise level of test utterances as two different factors that influence the uncertainty.

One straightforward way to change the duration of the test utterance is by extracting a consistent segment from it. We set the length of the segment to one of {10%, 20%, ... 100%} of the original duration and applied this extraction to all utterances in the test set. This process created 10 test sets with varying amounts of speech information. Notably, most of the test utterances in Voxceleb1 are between 2∼8 s long. Therefore, in the extreme case of 10%, the shortened utterances had lengths ranging from 0.2∼0.8 s.



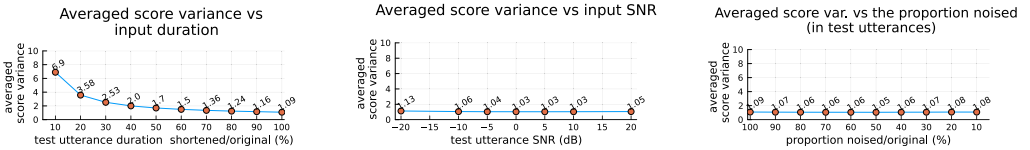
(a) (b) (c)

Fig. 7a. part 1: Subplots (a) (b) and (c) indicate that EER increases as shorter or noisier utterances are used as the test input. These plots were obtained as the system adjusted its threshold to fit the duration/noise changes.



(d) (e) (f)

Fig. 7b. Part 2: Subplots (d) (e) and (f) indicate that FRR decreases as shorter or noisier utterances are used as the test input, showing the security of the current system. This happens when the threshold is set fixed regardless of the duration and noise change in the test input.



(g) (h) (i)

Fig. 7c. Part 3: Subplots (g) shows that the average score variance increases as the input becomes shorter, while (h) and (i) shows that there is no significant change in score variance (being always close to 1.0) when the noise level of the input varies. This indicates that models tend to provide dispersive scores regarding shortened test inputs.

To vary the noise level, we added noise to either the entire utterance or a portion of it. These two methods represent two different scenarios: one simulates utterances collected in a consistently noisy environment, while the other simulates utterances disrupted by sudden noise signals. In this work, the process of adding noise is similar to data augmentation as described by Snyder et al. (2018), but with modifications to suit our task. First, the noise signals were selected only from the free noise partition in the MUSAN dataset (Snyder et al., 2015), and the utterances to be noised were from the Voxceleb1 test partition (including all the test utterances listed in Voxceleb1-O.txt on the Voxceleb official website:).

Secondly, when adding noise to the entire test speech utterance, we created seven new evaluation sets by choosing SNR from the set:{−20 dB, −10 dB, −5 dB, 0 dB, 5 dB, 10 dB, 20 dB}. Finally, when only a portion of the utterance is noised, we created nine evaluation sets by randomly adding noise to a segment of the original waveform, with that segment length chosen from {10%, 20%, 30%, . . . 90%} of the original length.

5.5. Evaluating uncertainties by sorting test utterance durations

Besides manually varying the noise level and the duration of the test input, there is also another interesting setting that can evaluate the quantified uncertainties. In this setting, we fix the length of the enrolled utterance and observe how the quantified uncertainties change over the original duration of the test utterance. To achieve this, we first modified the test data by fixing the length of each enrolled reference to 4 s, and then, plot the uncertainties over the test utterance length. In detail, from the Voxceleb-O test list, we selected test pairs whose enrolled utterance lengths are over 4 sec and truncated those enrolled utterances to 4 s For all the selected pairs, we measured the lengths of their test utterances and created a histogram to show how those lengths were distributed. In the last step, in each bin of the histogram we calculated the averaged score variance or the averaged entropy-based uncertainties (U_T, U_A and U_M) and plot them on top of the histogram. The graph provides how those uncertainties change over increasing input duration. If those uncertainties are well aligned with the input duration, there should be monotonic decreasing trends. Notably, in the entropy-based uncertainty calculation, the thresholds were not allowed to fit the modified test data since we treat this experiment as a variant of conditions.

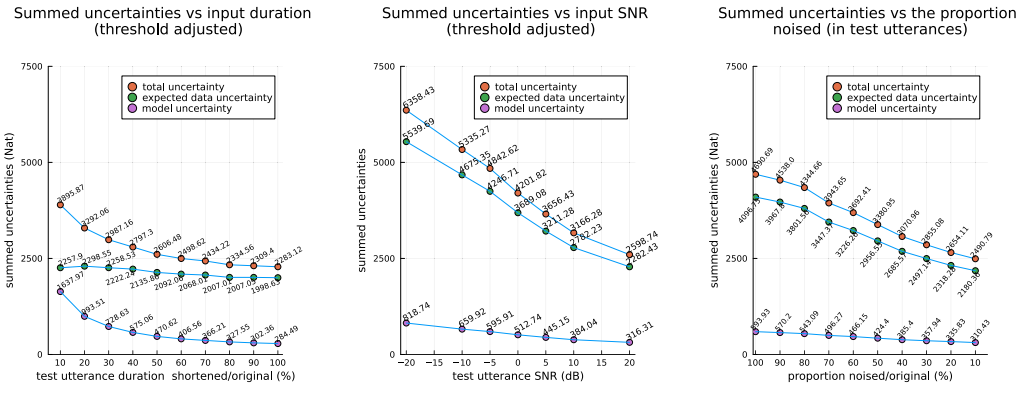


Fig. 7d. Part 4: In subplots (j) (k) and (l), it can be clearly seen that the model uncertainty (epistemic) corresponds to the change of duration since it decreases when longer utterances are used, while the expected data uncertainty (aleatoric) corresponds to the noise level in test utterances. The higher the noise level, the larger the aleatoric uncertainty there is. Compared to (j) (h) and (i), the model uncertainty quantified by mutual information weakly correlates to the overall EER in this case since the calculation includes the deviation between the score and the threshold. These plots are generated in the condition that the threshold is allowed to adapt to the changes in the test data.

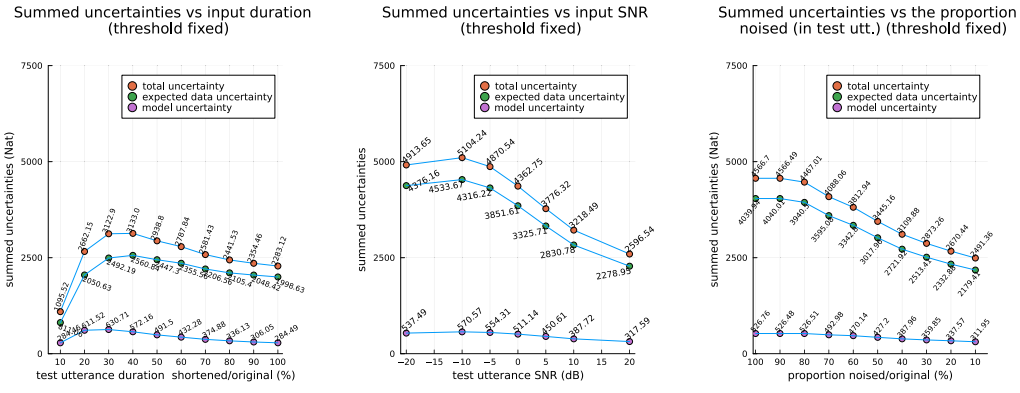


Fig. 7e. Part 5: In subplots (m) (n) and (o), the behaviour of model uncertainty (epistemic) and expected data uncertainty (aleatoric) is similar to that in (j) (k) and (l). However, the change in quantified model uncertainty is less significant. This is because these subplots are obtained without adjusting the threshold, and thus the uncertainty does not effectively reflect the duration and noise level in extreme cases.

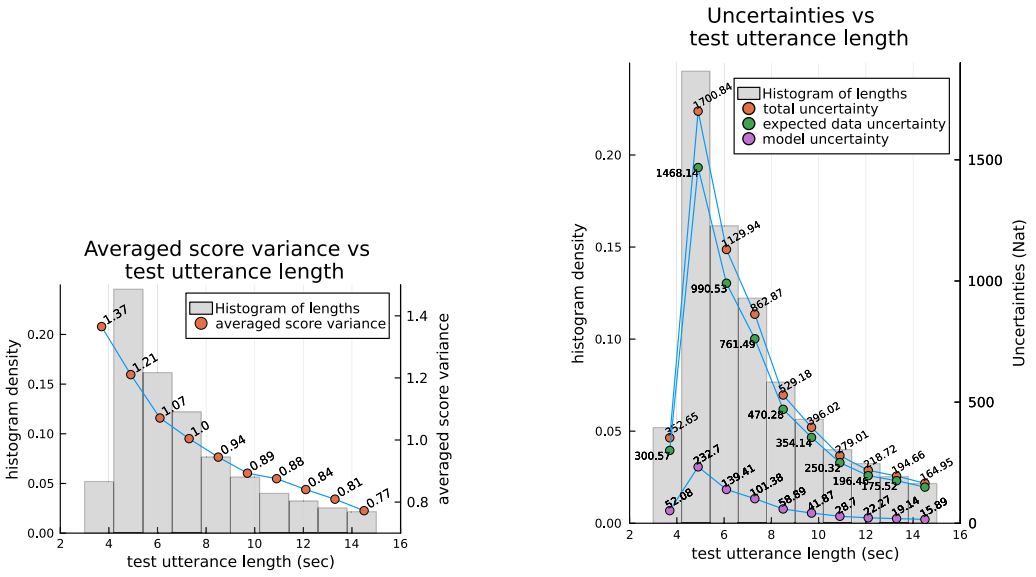
6. Results and discussions

In this section, we present the representative results obtained by running schemes described in Figs. 6(a) and 6(b).

6.1. The effect caused by the amount of training data

More training data better constrains the model and thus the lower model uncertainty there is. We check this for the PLDA model (under setting 1 in Table 1) to justify that the quantified values (epistemic uncertainty) truly represent the state of "lack of knowledge". We varied the number of training speakers or utterances per speaker as shown in Table 2. and calculated the sum of quantified epistemic uncertainty for all test instances in the dataset ($\sum U_M$).

Table 2 shows that the overall model uncertainty decreases significantly when more training data is provided. The EER, however, only changes slightly. While the reduction in model uncertainty (with more training data) is in itself not surprising, the results clearly indicate that the estimated model uncertainty better reflects the level of "knowledge" in the trained models. This can also be regarded as a justification for the correctness of the HMC sampling we have implemented since the posterior ought to be sharper when there is more training data.



(p) (q)

Fig. 7f. Part 6: For the additional setting described in Section 5.5, we show how the quantified uncertainties change over the original test utterance duration in (p)(q) of Fig. 7, In (p), we can see that the score variance aligns well with the test duration, showing a monotonic decrease over the test utterance length. Similar trends are also observable when the quantified uncertainties are entropy-based (shown in Fig. 7(q)). Typically, for all the entropy-based uncertainties, when the test input length is smaller than 4 sec, the uncertainty suddenly becomes very small, which is due to bad calibration. Finally, unlike the results given by (j)(m) in Fig. 7, all the entropy-based uncertainties are significantly affected by the test utterance duration, and this suggests that further investigation is needed to show the relationship between test utterance length and epistemic uncertainty.

Table 2
Estimated model uncertainty (by scheme in Fig. 6(a)) and EER resulting from sets of models trained using different amounts of training data (unit of uncertainty: Nat).

Changing the number of speakers			
#Speakers	806	606	406
$\sum U_M$ (Nat)	**284.82**	**333.20**	**405.10**
EER (%)	6.81 ± 0.03	6.81 ± 0.02	7.03 ± 0.05
Changing the number of utterances per speaker			
#Utterances	80	60	40
$\sum U_M$ (Nat)	**284.82**	**306.72**	**348.47**
EER (%)	6.80 ± 0.02	6.82 ± 0.02	6.88 ± 0.07

6.2. How uncertainties vary as duration and noise changes

Under different 4 settings in Table 1, we also observed how uncertainties change as duration and noise level change in the test input. In addition, we calculated different types of error rates and score variance in comparison to the varying uncertainties. Even though the schemes or techniques are different under different settings, similar trends are presented and thus we specify our findings mainly by analysing Figs. 7a–7f which is obtained by using HMC for sampling PLDA model posterior and running the scheme in Fig. 6(a) (under setting 1 in Table 1). It shows the change in error rates and uncertainties over different test utterance durations or noise levels.

The analyse of Fig. 7 can be summarised as follows: the current system has already been safe since it tends to reject more when the level of speech variabilities increases in the test data. The model uncertainty corresponds to the change of duration while the data uncertainty corresponds to the noise level. Finally, the uncertainties quantified rely on the proper degree of calibration, because if the threshold is not allowed to adjust for extreme conditions, the quantified uncertainties will not show the desired behaviour.

For the rest of the results obtained from settings 2~4, we show them in the Appendix (B.8, B.9, B.10). They present results that are similar to those in Fig. 7 in terms of the overall trend. This indicates that no matter which model is sampled, the behaviour of uncertainties does not largely change. Thus, it would be more desirable to first look at the model uncertainty in simple components of the system. However, some minor differences can still be found. As in Fig. B.9a and Fig. B.10a, Bayes-by-backprop further updates model so that it achieves EERs which are slightly lower than the baseline 6.72% (EER of 5.78% in Fig. B.9a, 5.63% in Fig. B.10a), while SGLD samples networks that are inferior to the baseline (EER of 6.92% in Fig. B.8a) due to the additional randomness involved. In addition, by comparing Fig. B.9 and Fig. B.10, more diverse predictions are made when the posterior is approximated

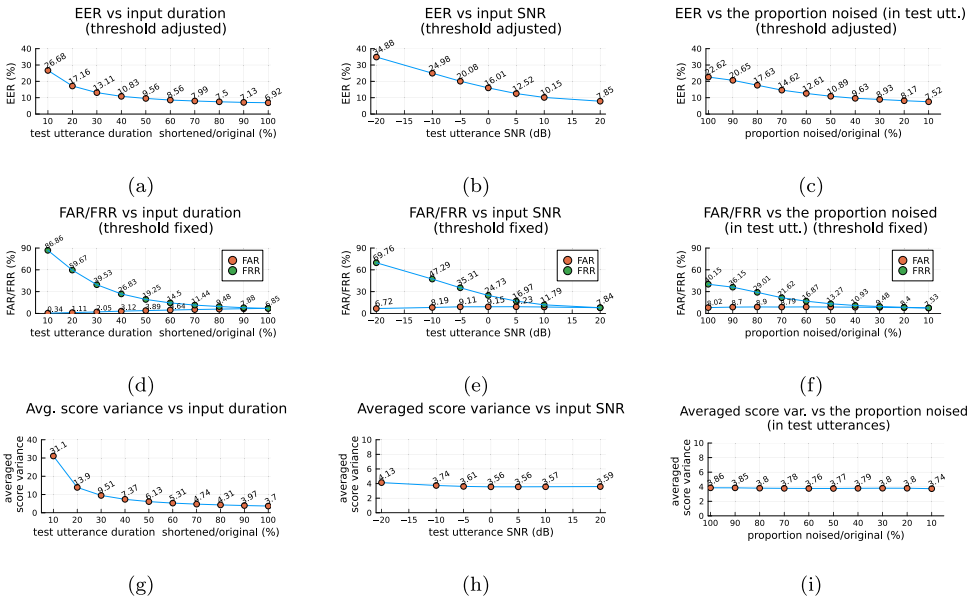


Fig. B.8a. Part 1: (a)–(i).

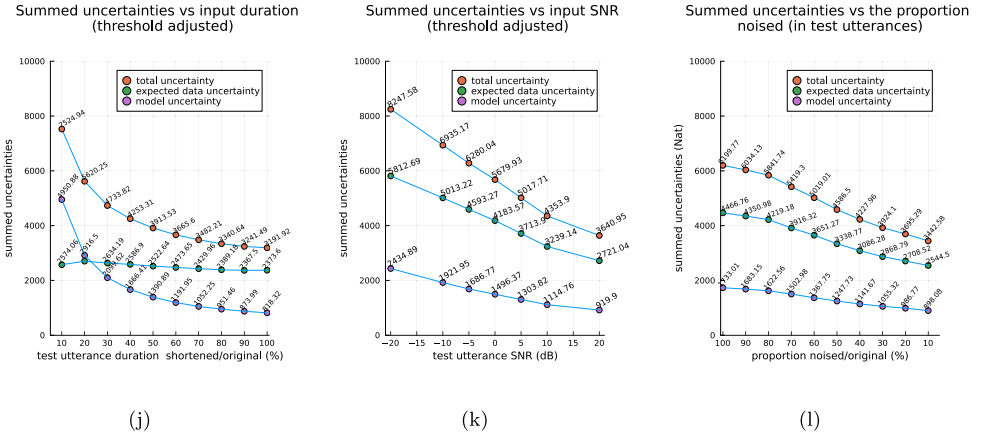


Fig. B.8b. Part 2: (j)(k)(l).

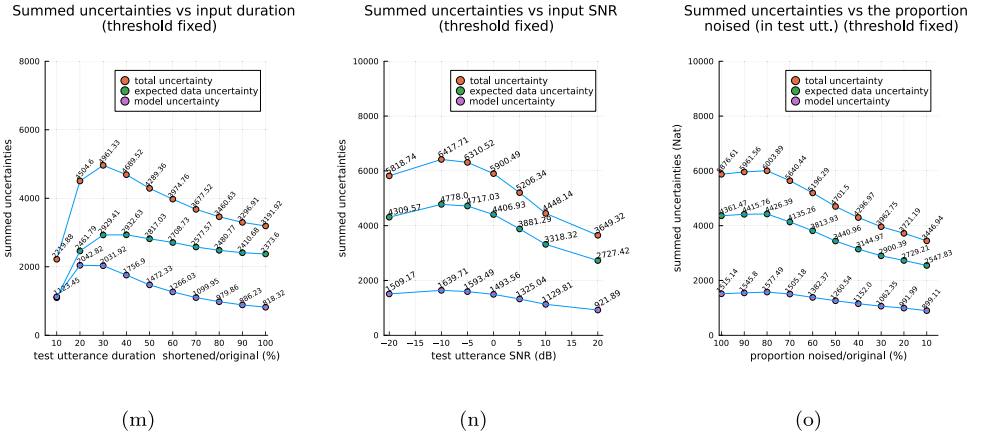


Fig. B.8c. Part 3: (m)(n)(o).

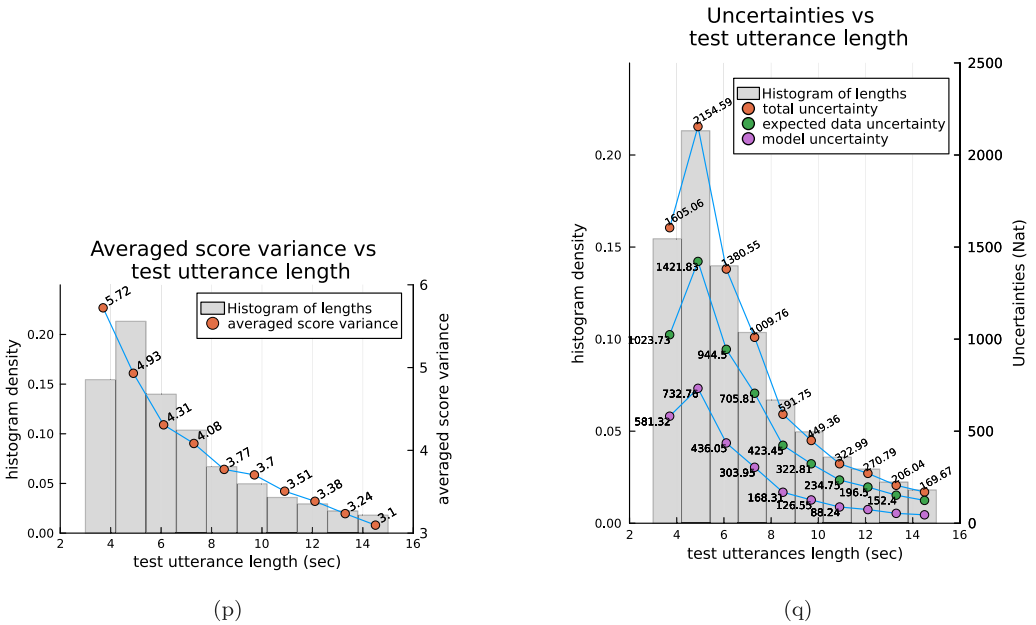


Fig. B.8d. Part 4: (p)(q).

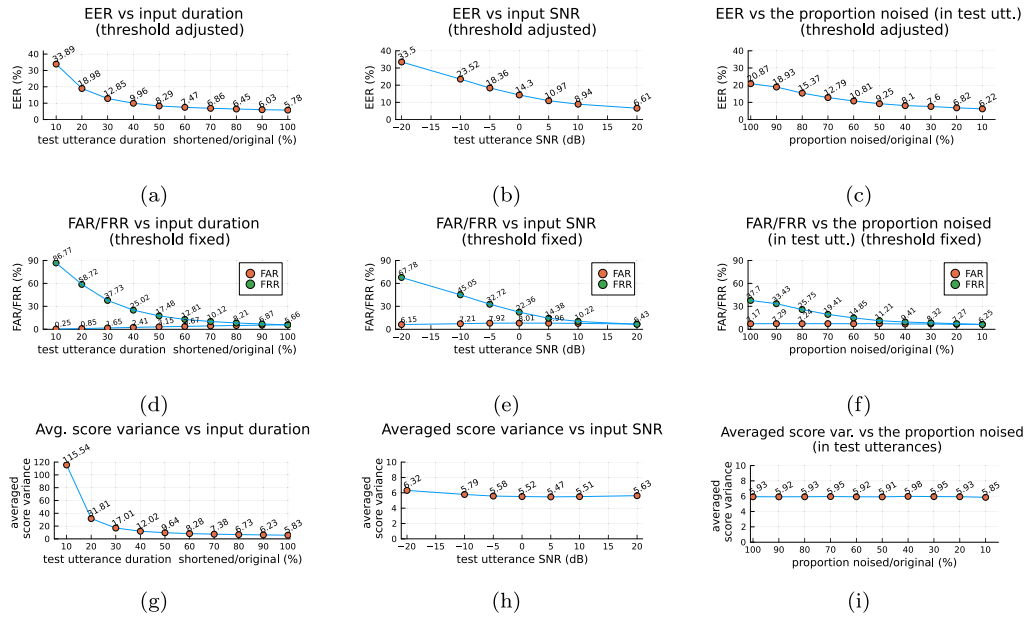


Fig. B.9a. Part 1: (a)–(i).

by conducting a longer training and hence the higher model uncertainties in Fig. 10. (E.g., averaged score variance is 115.51 when duration is 10% in Fig. B.9g, but in Fig. B.10g, it is 1239.22 under the same condition). This suggests that different sampling or approximation methods might estimate model uncertainty differently.

7. Conclusion

In this work, we have introduced a novel approach to quantify uncertainties in the modern speaker verification system. Exploiting approximate Bayesian techniques such as HMC, SGLD, and variational inference (Bayes-by-backprop), our method generates ensembles of ASV models, which allows us to estimate the uncertainties associated with ASV predictions. Through extensive

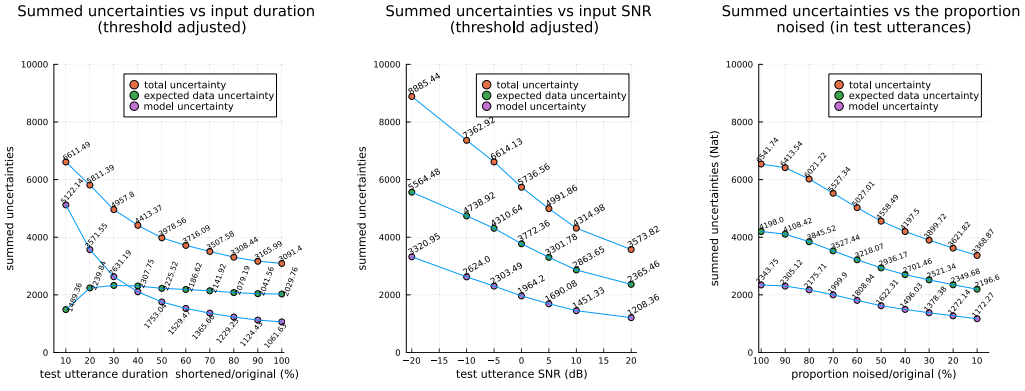


Fig. B.9b. Part 2: (j)(k)(l).

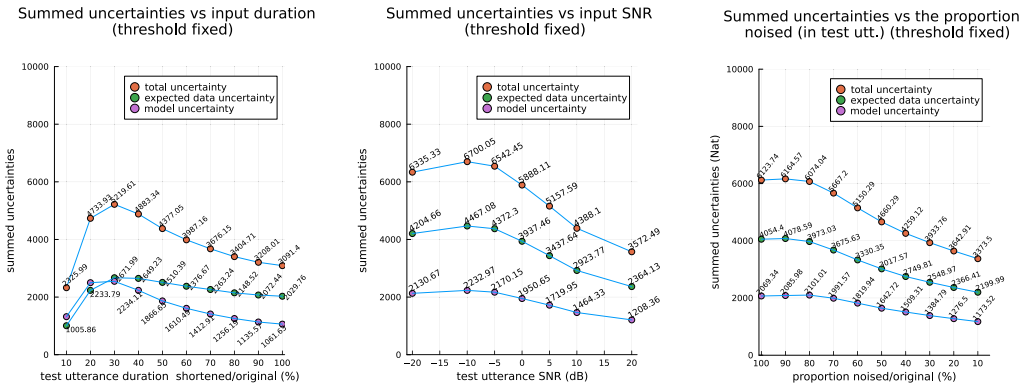


Fig. B.9c. Part 3: (m)(n)(o).

experiments conducted on the Voxceleb1 dataset, we validated that our approach provides a plausible measure of confidence in the decisions made by the ASV system. We assessed our method under various conditions, including varying amounts of training data, varying the duration of test utterances, and varying noise levels of test utterances. In the first case, we justified the reducible nature of the quantified epistemic uncertainty. In the latter two cases, our measure accurately reflected the impact of test utterance duration and noise on the system. The changes to the uncertainties reflected in the measure are also reflected in the EER of the system indicating that the proposed measure can be employed to decide whether to act on a prediction made by the speaker verification system or wait for more data and make another prediction, i.e., it is a measure of confidence that can be utilised when the system is operational, and the ground truth is not known. However, some aspects of the quantified uncertainty remain unexplored. For example, seeking calibration between uncertainty levels and error rates might lead to performance improvements but this has not been addressed in this work. In other words, the question of "how this uncertainty can be used to improve system performance" is left for future research.

CRediT authorship contribution statement

Miao Jing: Writing – original draft, Software, Methodology, Conceptualization. **Vidhyasaharan Sethu:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Beena Ahmed:** Writing – review & editing, Supervision, Methodology, Conceptualization. **Kong Aik Lee:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

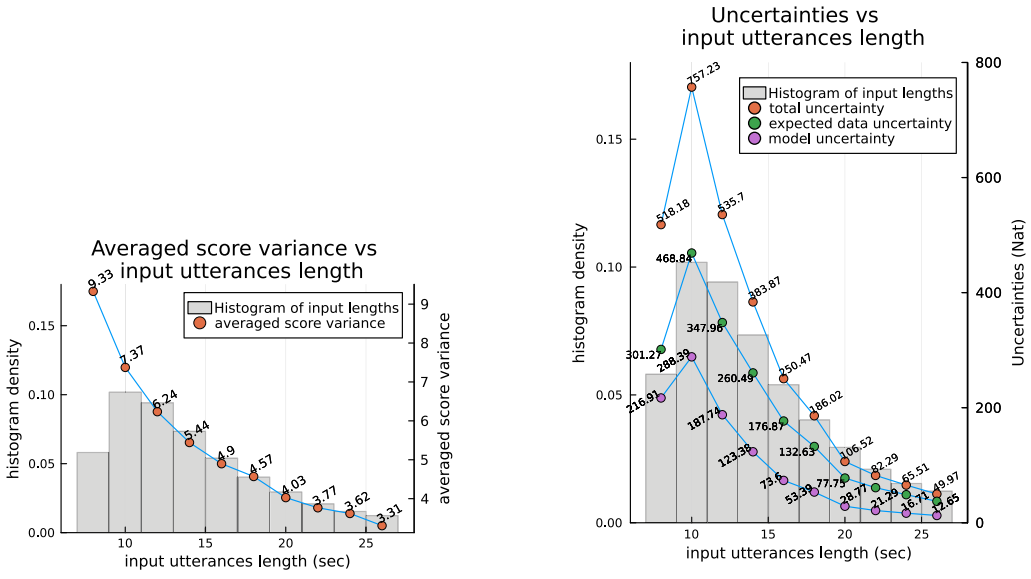


Fig. B.9d. Part 4: (p)(q).

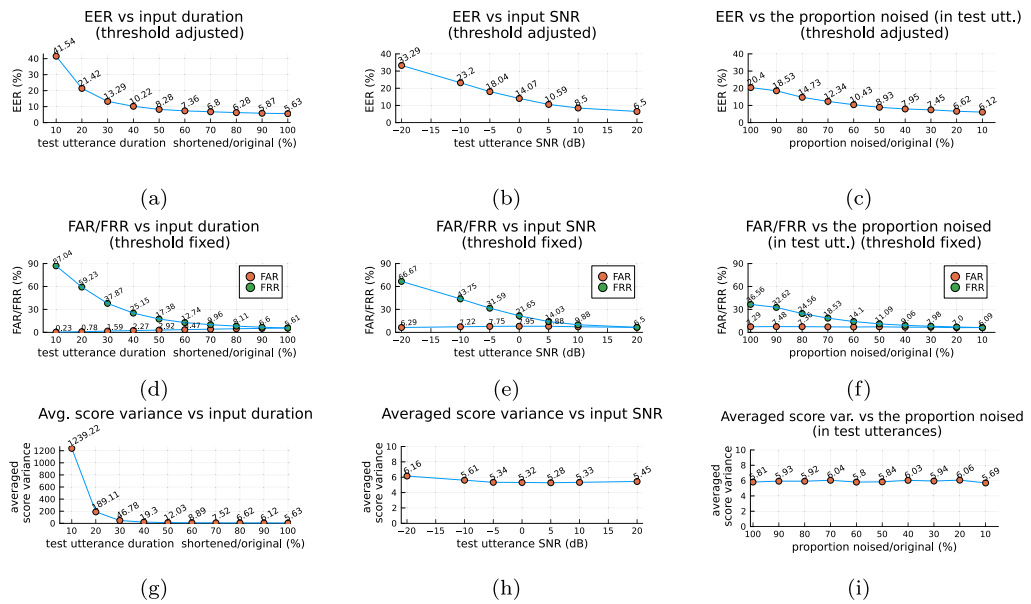


Fig. B.10a. Part 1: (a)–(i).

Acknowledgment

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

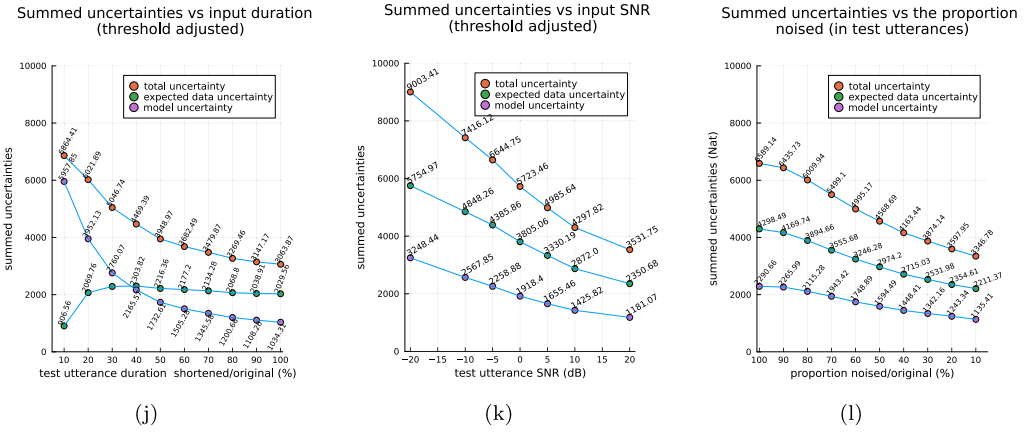


Fig. B.10b. Part 2: (j)(k)(l).

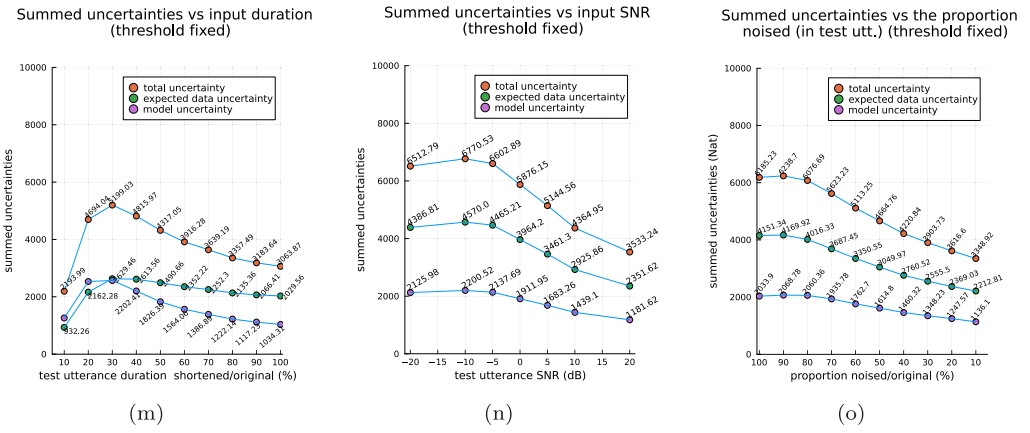


Fig. B.10c. Part 3: (m)(n)(o).

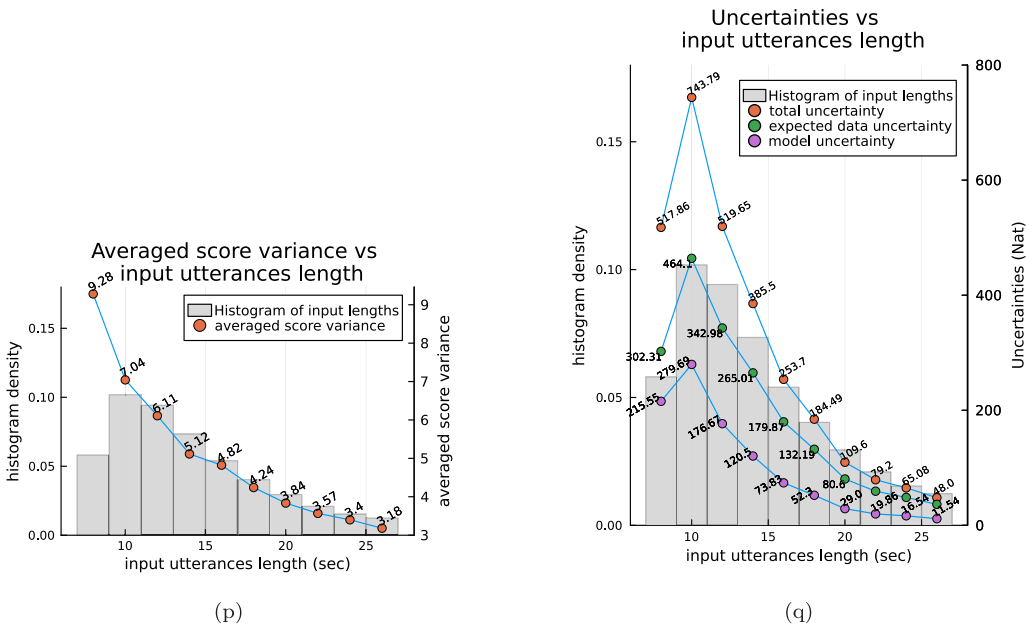


Fig. B.10d. Part 4: (p)(q).

Appendix A. Detailed Hamiltonian Monte-Carlo algorithm

The HMC algorithm is detailed as follows:

function HMC(starting state θ_0, number of steps l, and step size ϵ)
  for $t = 1, 2, 3...$
    sample momentum from Gaussian distribution: $\mathrm{p}_t \sim \mathcal{N}(0, \boldsymbol{I})$
    simulate Hamiltonian Dynamics using Leapfrog algorithm:
      $\theta_t^{(0)} = \theta_{t-1}$
      $\mathrm{p}_t^{(0)} := \mathrm{p}_t - \frac{\epsilon}{2}\nabla_\theta U(\theta_t^{(0)})$
    for $i = 1, 2, 3...l$:
      $\theta_t^{(i)} := \theta_t^{(i-1)} + \epsilon \boldsymbol{M}^{-1}\mathrm{p}_t^{(i-1)}$
      $\mathrm{p}_t^{(i)} := \mathrm{p}_t^{(i-1)} - \epsilon\nabla_\theta U(\theta_t^{(i)})$
    end
    $\mathrm{p}_t^{(l)} := \mathrm{p}_t^{(l)} - \frac{\epsilon}{2}\nabla_\theta U(\theta_t^{(l)})$
    $\hat{\theta}_t := \theta_t^{(l)}$
    Metropolis-Hasting's step to calculate acceptance ratio:
    $a(\hat{\theta}_t|\theta_{t-1}) = min\left\{1, exp\left(H(\hat{\theta}_t, \mathrm{p}_t^{(l)}) - H(\theta_{t-1}, \mathrm{p}_{t-1})\right)\right\}$
    sample $u \sim Uniform[0, 1]$
    if $u < a(\theta_t|\theta_{t-1})$: $\theta_t := \hat{\theta}_t$
    else: $\theta_t = \theta_{t-1}$
  end
  output: $\{\theta_0, \theta_1, \theta_2, \ldots\}$
end function

Appendix B. Other experiment results obtained by varying the settings (Fig. B.8–B.10)

The results collected under setting 2 in Table 1 are shown in Fig. B.8. In this case, multiple embedding nets are sampled using SGLD.

The results collected under setting 3 in Table 1 are shown in Fig. B.9. In this case, Bayes-by-backprop (5 epochs) is used to approximate the model posterior of the embedding net.

The results collected under setting 4 in Table 1 are shown in Fig. B.10. In this case, Bayes-by-backprop (10 epochs) is used to approximate the model posterior of the embedding net.

Similar to Fig. 7, all these figures present the change of error rates and uncertainties over the test utterance duration or noise level.

Data availability

Data will be made available on request.

References

Angelopoulos, A., Bates, S., 2021. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. http://dx.doi.org/10.48550/arXiv.2107.07511.

Athulya, M., Sathidevi, P., 2017. Mitigating effects of noise in forensic speaker recognition. In: International Conference on Wireless Communications, Signal Processing and Networking. WiSPNET, pp. 1602–1606. http://dx.doi.org/10.1109/WiSPNET.2017.8300031.

Betancourt, M., 2017. A conceptual introduction to hamiltonian Monte Carlo. https://arxiv.org/pdf/1701.02434.pdf.

Blei, D.M., Kucukelbir, A., McAuliffe, J.D., 2017. Variational inference: A review for statisticians. J. Amer. Statist. Assoc. 112 (518), 859–877. http://dx.doi.org/10.1080/01621459.2017.1285773.

Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D., 2015. Weight uncertainty in neural networks. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning, vol. 37, pp. 1613–1622, https://dl.acm.org/doi/10.5555/3045118.3045290.

Borgström, B.J., 2021. Bayesian estimation of plda in the presence of noisy training labels, with applications to speaker verification. IEEE/ACM Trans. Audio, Speech, Lang. Process. 30, 414–428. http://dx.doi.org/10.1109/TASLP.2021.3130980.

Brooks, S., Gelman, A., Jones, G., Meng, X.L., 2011. Handbook of Markov Chain Monte Carlo, first ed. http://dx.doi.org/10.1201/b10905.

Brümmer, N., Du Preez, J., 2006. Application-independent evaluation of speaker detection. Comput. Speech & Lang. 20 (2-3), 230–275. http://dx.doi.org/10.1016/j.csl.2005.08.001.

Brümmer, N., Silnova, A., Burget, L., Stafylakis, T., 2018. Gaussian meta-embeddings for efficient scoring of a heavy-tailed plda model. Speak. Lang. Recognit. Work. 349–356. http://dx.doi.org/10.21437/Odyssey.2018-49.

Campbell, W.M., Reynolds, D.A., Campbell, J.P., Brady, K.J., 2005. Estimating and evaluating confidence for forensic speaker recognition. IEEE Int. Conf. Acoust. Speech, Signal Process. 1, 717–720. http://dx.doi.org/10.1109/ICASSP.2005.1415214.

Chen, T., Fox, E., Guestrin, C., 2014. Stochastic gradient hamiltonian monte carlo. Int. Conf. Mach. Learn. 1683–1691, https://dl.acm.org/doi/pdf/10.5555/3454287.3454631.

Chen, Z., Wang, S., Qian, Y., 2021. Self-supervised learning based domain adaptation for robust speaker verification. In: IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, pp. 5834–5838. http://dx.doi.org/10.1109/ICASSP39728.2021.9414261.

Chib, S., Greenberg, E., 1995. Understanding the metropolis-hastings algorithm. Amer. Statist. 49 (4), 327–335. http://dx.doi.org/10.2307/2684568.

Cumani, S., 2020. On the distribution of speaker verification scores: Generative models for unsupervised calibration. IEEE/ACM Trans. Audio, Speech, Lang. Process. 29, 547–562. http://dx.doi.org/10.1109/TASLP.2020.3040103.
Cumani, S., Plchot, O., Laface, P., 2014. On the use of i-vector posterior distributions in plda. IEEE/ACM Trans. Audio, Speech, Lang. Process. 22, 846–857. http://dx.doi.org/10.1109/TASLP.2014.2308473.
Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P., 2010. Front-end factor analysis for speaker verification. IEEE Trans. Audio, Speech, Lang. Process. 19 (4), 788–798. http://dx.doi.org/10.1109/TASL.2010.2064307.
Depeweg, S., Hernandez-Lobato, J.M., Doshi-Velez, F., Udluft, S., 2017. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In: Proceedings of the 35th International Conference on Machine Learning, vol. 80, http://dx.doi.org/10.48550/arXiv.1710.07283.
Desplanques, B., Thienpondt, J., Demuynck, K., 2020. ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In: Interspeech 2020. pp. 3830–3834. http://dx.doi.org/10.21437/Interspeech.2020-2650.
Ding, K., 2018. A note on kaldi's plda implementation. https://arxiv.org/pdf/1804.00403.pdf.
Fan, Y., Kang, J., Li, L., Li, K., Chen, H., Cheng, S., Zhang, P., Zhou, Z., Cai, Y., Wang, D., 2020. Cn-celeb: a challenging Chinese speaker recognition dataset. In: IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, pp. 7604–7608. http://dx.doi.org/10.1109/ICASSP40776.2020.9054017.
Fragoso, T.M., Bertoli, W., Louzada, F., 2018. Bayesian model averaging: A systematic review and conceptual classification. Int. Stat. Rev. 86, 1–28. http://dx.doi.org/10.1111/insr.12243.
Gal, Y., Ghahramani, Z., 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: International Conference on Machine Learning. pp. 1050–1059, https://dl.acm.org/doi/10.5555/3045390.3045502.
Gelman, A., Rubin, D.B., 1992. Inference from iterative simulation using multiple sequences. Statist. Sci. 7,no.4, 457–472. http://dx.doi.org/10.1214/ss/1177011136.
Ioffe, S., 2006. Probabilistic linear discriminant analysis. In: Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Vol. Part IV 9. Graz, Austria, pp. 531–542. http://dx.doi.org/10.1007/11744085.
Izmailov, P., Vikram, S., Hoffman, M., Wilson, A., 2021. What are Bayesian neural network posteriors really like? In: International Conference on Machine Learning. pp. 4629–4640. http://dx.doi.org/10.48550/arXiv.2104.14421.
Kheder, W., Matrouf, D., Bonastre, J., Ajili, M., Bousquet, P., 2015. Additive noise compensation in the i-vector space for speaker recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, pp. 4190–4194. http://dx.doi.org/10.1109/ICASSP.2015.7178760.
Kuzmin, N., Fedorov, I., Sholokhov, A., 2022. Magnitude-aware probabilistic speaker embeddings. In: The Speaker and Language Recognition Workshop. pp. 1–8. http://dx.doi.org/10.21437/Odyssey.2022-1.
Lee, K., Ma, B., Li, H., 2013. Speaker verification makes its debut in smartphone. In: IEEE Signal Processing Society Speech and Language Technical Committee Newsletter.
Lee, K.A., Wang, Q., Koshinaka, T., 2021. Xi-vector embedding for speaker recognition. IEEE Signal Process. Lett. 28, 1385–1389. http://dx.doi.org/10.1109/LSP.2021.3091932.
Li, X., Zhong, J., Yu, J., Hu, S., Wu, X., Liu, X., H., M., 2020. Bayesian x-vector: Bayesian neural network-based x-vector system for speaker verification. In: The Speaker and Language Recognition Workshop. pp. 365–371. http://dx.doi.org/10.21437/Odyssey.2020-51.
Liu, T., Lee, K., Wang, Q., Li, H., 2023. Disentangling voice and content with self-supervision for speaker recognition. In: 37th Conference on Neural Information Processing Systems. NeurIPS 2023, http://dx.doi.org/10.48550/arXiv.2310.01128.
Liu, X., Sahidullah, M., Lee, K., Kinnunen, T., 2024. Generalizing speaker verification for spoof awareness in the embedding space. IEEE/ACM Trans. Audio, Speech, Lang. Process. http://dx.doi.org/10.1109/TASLP.2024.3358056.
Malinin, A., Gales, M., 2018. Predictive uncertainty estimation via prior networks. Adv. Neural Inf. Process. Syst. https://dl.acm.org/doi/10.5555/3327757.3327808.
Mandasari, M., McLaren, M., van Leeuwen, D., 2012. The effect of noise on modern automatic speaker recognition systems. In: IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, pp. 4249–4252. http://dx.doi.org/10.1109/ICASSP.2012.6288857.
Modi, C., Barnett, A., Carpenter, B., 2023. Delayed rejection hamiltonian monte carlo for sampling multiscale distributions. Bayesian Anal. Adv. Publ. (1-28), http://dx.doi.org/10.1214/23-ba1360.
Nagrani, A., Chung, J., Xie, W., Zisserman, A., 2020. Voxceleb: Large-scale speaker verification in the wild. Comput. Speech & Lang. 60–101027, http://dx.doi.org/10.1016/j.csl.2019.101027.
Naika, R., 2018. An overview of automatic speaker verification system. In: Intelligent Computing and Information and Communication: Proceedings of 2nd International Conference. ICICC 2017, pp. 603–610. http://dx.doi.org/10.1007/978-981-10-7245-1_59.
Nemeth, C., Fearnhead, P., 2021. Stochastic gradient markov chain monte carlo. J. Amer. Statist. Assoc. 116 (533), 433–450. http://dx.doi.org/10.1080/01621459.2020.1847120.
Reza, F.M., 1994. An Introduction to Information Theory. Courier Corporation.
Ribas, D., Vincent, E., 2019. An improved uncertainty propagation method for robust i-vector based speaker recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 6331–6335. http://dx.doi.org/10.1109/ICASSP.2019.8683132.
Sato, M., Suzuki, J., Shindo, H., Matsumoto, Y., 2018. Interpretable adversarial perturbation in input embedding space for text. In: 27th International Joint Conference on Artificial Intelligence. IJCAI 2018, pp. 4323–4330, https://dl.acm.org/doi/10.5555/3304222.3304371.
Senge, R., Bösner, S., Dembczyński, K., Haasenritter, J., Hirsch, O., Donner-Banzhoff, N., Hüllermeier, E., 2014. Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty. Inform. Sci. 255, 16–29. http://dx.doi.org/10.1016/j.ins.2013.07.030Getrightsandcontent.
Sheikholeslami, F., Jain, S., Giannakis, G., 2020. Minimum uncertainty-based detection of adversaries in deep neural networks. In: 2020 Information Theory and Applications Workshop. ITA, pp. 1–16. http://dx.doi.org/10.1109/ITA50056.2020.9244964.
Snyder, D., Chen, G., Povey, D., 2015. MUSAN: A music, speech, and noise corpus. http://dx.doi.org/10.48550/arXiv.1510.08484.
Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S., 2018. X-vectors: Robust dnn embeddings for speaker recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, pp. 5329–5333. http://dx.doi.org/10.1109/ICASSP.2018.8461375.
Stafylakis, T., Kenny, P., Ouellet, P., Perez, J., Kockmann, M., Dumouchel, P., 2013. Text-dependent speaker recognition using plda with uncertainty propagation. In: Interspeech. pp. 3684–3688. http://dx.doi.org/10.21437/Interspeech.2013-691.
Süslü, C., Eren, E., Demiroglu, C., 2021. Uncertainty assessment for detection of spoofing attacks to speaker verification systems using a Bayesian approach. Speech Commun. 137, 44–51. http://dx.doi.org/10.1016/j.specom.2021.12.003.
Tibshirani, R., Efron, B., 1993. An introduction to the bootstrap. Monogr. Statist. Appl. Probab. 57 (1), http://dx.doi.org/10.1201/9780429246593.
Vehtari, A., Gelman, A., Simpson, D., Carpenter, B., Burkner, P., 2021. Rank-normalization, folding, and localization: an improved r for assessing convergence of mcmc. Bayesian Anal. 16 (2), 667–718. http://dx.doi.org/10.1214/20-BA1221.
Villalba, J., Brümmer, N., 2011. Towards fully bayesian speaker recognition: integrating out the between-speaker covariance. In: Interspeech. pp. 505–508. http://dx.doi.org/10.21437/Interspeech.2011-142.
Vogt, R., Sridharan, S., 2008. Explicit modelling of session variability for speaker verification. Comput. Speech & Lang. 22 (1), 17–38. http://dx.doi.org/10.1016/j.csl.2007.05.003.
Welling, M., Teh, Y., 2011. Bayesian learning via stochastic gradient langevin dynamics. In: Proceedings of the 28th International Conference on Machine Learning. ICML-2011, pp. 681–688, https://dl.acm.org/doi/10.5555/3104482.3104568.
Yao, J., Pan, W., Ghosh, S., Doshi-Velez, F., 2019. Quality of uncertainty quantification for Bayesian neural network inference. In: Proceedings at the International Conference on Machine Learning: Workshop on Uncertainty & Robustness in Deep Learning. ICML, http://dx.doi.org/10.48550/arXiv.1906.09686.
Zhang, H., Wang, L., Lee, K.A., Liu, M., Dang, J., Meng, H., 2023. Meta-generalization for domain-invariant speaker verification. IEEE/ACM Trans. Audio, Speech, Lang. Process. 31, 1024–1036. http://dx.doi.org/10.1109/TASLP.2023.3244518.