

1

## 2           **Forensic voice comparison – Human-supervised-automatic approach**

3

### 4   **Authors**

#### 5   Geoffrey Stewart Morrison

6   Forensic Data Science Laboratory, Aston University

7   Forensic Evaluation Ltd

8   geoff-morrison@forensic-evaluation.net

9

#### 10   Philip Weber

11   Forensic Data Science Laboratory, Aston University

12   p.weber1@aston.ac.uk

13

#### 14   Ewald Enzinger

15   Forensic Data Science Laboratory, Aston University

16   Eduworks Corporation

17   ewald-enzinger@forensic-evaluation.net

18

#### 19   Beltrán Labrador

20   AUDIAS – Audio, Data Intelligence and Speech, Escuela Politécnica Superior, Universidad Autónoma de Madrid

21   beltran.labrador@uam.es

22

#### 23   Alicia Lozano-Díez

24 AUDIAS – Audio, Data Intelligence and Speech, Escuela Politécnica Superior, Universidad Autónoma de Madrid

25 alicia.lozano@uam.es

26

27 **Daniel Ramos**

28 AUDIAS – Audio, Data Intelligence and Speech, Escuela Politécnica Superior, Universidad Autónoma de Madrid

29 daniel.ramos@uam.es

30

31 **Joaquín González-Rodríguez**

32 AUDIAS – Audio, Data Intelligence and Speech, Escuela Politécnica Superior, Universidad Autónoma de Madrid

33 joaquin.gonzalez@uam.es

34

35 **Keywords**

36 [10–15 keywords, listed alphabetically]

37 automatic speaker recognition

38 calibration

39 forensic speaker comparison

40 forensic speaker identification

41 forensic speaker recognition

42 forensic speech science

43 forensic voice comparison

44 likelihood ratio

45 validation

46 x-vector

47

## 48 **Abstract**

49 [50–100 words]

50 The human-supervised-automatic analytical approach to forensic voice comparison in  
51 conjunction with the likelihood-ratio interpretive framework is described. Practitioner  
52 tasks are described, including adoption of relevant hypotheses for the case, assessment  
53 of the conditions of the questioned-speaker and known-speaker recordings in the case,  
54 and selection of data representing the relevant population and reflecting the conditions  
55 for the case. Software tools are also described. An example is provided of a forensic-  
56 voice-comparison system based on state-of-the-art automatic-speaker-recognition  
57 technology. Also described are the calibration and validation of that system using a  
58 benchmark dataset reflecting the conditions of a real forensic case.

59

## 60 **Key points**

61 [short bulleted list of key points]

- 62 • likelihood-ratio framework
  - 63 ○ hypotheses
  - 64 ○ relevant population
  - 65 ○ common-source likelihood-ratio model
  - 66 ○ recording conditions
  - 67 ○ training and validation data
- 68 • software tools based on state-of-the-art automatic-speaker-recognition  
69 technology
  - 70 ○ diarization and voice activity detection

- 71 ○ feature extraction
- 72 ○ x-vector extraction
- 73 ○ dimension reduction and mismatch compensation
- 74 ○ calculation of uncalibrated likelihood ratio
- 75 ○ calibration
- 76 ● validation

77

78

## 79 **Acknowledgements**

80 The writing of this entry was supported by Research England's Expanding Excellence  
81 in England Fund as part of funding for the Aston Institute for Forensic Linguistics  
82 2019–2023.

## 83 **1 Introduction**

84 The reader is assumed to already have some familiarity with the following topics, to  
85 the level which may be found in general introductions to forensic voice comparison:  
86 analytical approaches to forensic voice comparison, including the human-supervised-  
87 automatic approach; interpretive frameworks applied to forensic voice comparison,  
88 including the likelihood-ratio framework; and validation of forensic-evaluation  
89 systems, including validation of systems that output likelihood ratios.

90 The human-supervised-automatic analytical approach combined with the likelihood-  
91 ratio interpretive framework is described in greater detail than is usual in general  
92 introductions to forensic voice comparison. Recent reviews of automatic-speaker-  
93 recognition technology include Lee et al. (2020), Matějka et al. (2020), and Villalba et  
94 al. (2020). Morrison et al. (2020) provides an overview of the application of automatic-  
95 speaker-recognition technology to forensic voice comparison. Application of more  
96 recent automatic-speaker-recognition technology is described in Weber et al. (2022a,  
97 2022b).

98 For concreteness, in the current entry an example human-supervised-automatic  
99 forensic-voice-comparison system is described. The example is based on an alpha  
100 version of the core software tools of the E<sup>3</sup> Forensic Speech Science System (E<sup>3</sup>FS<sup>3</sup>).  
101 E<sup>3</sup>FS<sup>3</sup> is being developed by the Forensic Data Science Laboratory at Aston University,  
102 with contributions from AUDIAS – Audio, Data Intelligence and Speech at  
103 Universidad Autónoma de Madrid, and with additional contributions from multiple  
104 other research laboratories and operational forensic laboratories. E<sup>3</sup>FS<sup>3</sup> is being  
105 developed for both research and casework use, and includes open-code software tools.  
106 The E<sup>3</sup>FS<sup>3</sup> software tools are designed to be flexible and provide the user with various  
107 options. For simplicity, the example focuses on a single set of options. Although details  
108 may vary, the example system is broadly similar to other state-of-the-art systems. Also  
109 for concreteness, the discussion of protocols is based on those used by Forensic  
110 Evaluation Ltd. Protocols used by other forensic-service providers may vary.

111 Descriptions are provided of:

112 • hypotheses and data, including:

113 ○ the adoption of the relevant hypotheses for a case,

114 ○ the assessment of the conditions of the questioned-speaker and known-  
115 speaker recordings in the case, and

116 ○ the selection of data representing the relevant population and reflecting  
117 the conditions for the case

118 • the core software tools of the example system

119 • a benchmark validation of the example system

120

## 121 2 Hypotheses and data

### 122 2.1 Hypotheses

123 State-of-the-art human-supervised-automatic forensic-voice-comparison systems  
124 calculate common-source likelihood ratios for which the same-speaker versus  
125 different-speaker hypotheses are:

126  $H_1$ : the speakers on the questioned-speaker and the known-speaker recordings are  
127 the same speaker

128 versus

129  $H_2$ : the speakers on the questioned-speaker and the known-speaker recordings are  
130 not the same speaker but two different speakers each selected at random from the  
131 relevant population

### 132 2.2 Relevant population

133 The relevant population is the population from which the questioned speaker could  
134 potentially have come if they were not the known speaker. The relevant population can  
135 usually be restricted to either male or female speakers who speak a particular language  
136 with a particular accent (Morrison et al., 2016). By listening, it is usually clear to non-  
137 experts such as judges and jury members whether the speaker on the questioned-  
138 speaker recording is male or female, what language they are speaking, and broadly  
139 what accent of that language they are speaking. This, however, is not always the case,  
140 for example, it may be unclear whether the speaker on the questioned-speaker  
141 recording is male or female. If any of these things are disputed, then they may become  
142 issues requiring forensic evaluation. Usually, a forensic practitioner, in consultation  
143 with the instructing party, can define a relevant population to adopt for the case. In  
144 their casework report, the forensic practitioner should clearly state what population  
145 they have adopted, and they may request that the instructing party specify the relevant  
146 population in their letter of instruction. The choice of relevant population is a subjective  
147 judgment, and it can potentially be disputed. If the judge at an admissibility hearing or  
148 the judge or jury during trial is not convinced that the population adopted is a  
149 reasonable relevant population for the case, then the likelihood ratio that the forensic  
150 practitioner calculates will not be meaningful for the case – it will be answering a  
151 question about a different population than the one that the judge or jury considers  
152 relevant for the case.

### 153 **2.3 Common-source likelihood-ratio model**

154 In general, a common-source likelihood-ratio model has the form given in Equation (1)  
155 in which  $\Lambda$  is the likelihood ratio,  $f(x_q, x_k|M)$  is a joint probability-density function,  
156  $x_q$  and  $x_k$  are feature vectors characterizing the speech of the speaker of interest on the  
157 questioned-speaker and known-speaker recordings respectively (in our example system  
158 these are x-vectors), and  $M_s$  and  $M_d$  are the same-speaker and different-speaker models  
159 respectively.

160 (1)

$$161 \quad \Lambda = \frac{f(x_q, x_k | M_s)}{f(x_q | M_d) f(x_k | M_d)}$$

162 In order to train (or adapt) the statistical models that calculate the likelihood ratio, the  
163 practitioner must use data from recordings of speakers sampled from the population  
164 that has been adopted as the relevant population for the case. If the data are not  
165 sufficiently representative of the relevant population for the case, then the likelihood  
166 ratio calculated will answer a different question than the one defined by the stated  
167 hypotheses. In their report, the practitioner should describe the data that they use for  
168 training. Whether the data are sufficiently representative of the relevant population is  
169 a subjective judgment, and it can potentially be disputed. Likewise, data used for  
170 validating the system must be sufficiently representative of the relevant population for  
171 the case. Usually, a single dataset intended to be representative of the relevant  
172 population will be obtained or selected, and that dataset will then be divided into a  
173 training set and a validation set. If these data are not actually representative of the  
174 relevant population, no amount of empirical testing will reveal this (Morrison, 2021).

## 175 **2.4 Recording conditions**

176 In addition to the data used for training and validation being sufficiently representative  
177 of the relevant population for the case, they must also be sufficiently reflective of the  
178 conditions of the questioned-speaker and known-speaker recordings in the case.  
179 Hansen & Bořil (2018) provide a taxonomy of sources of speaker-intrinsic and speaker-  
180 extrinsic variability. Common speaker-intrinsic conditions or speaking styles include  
181 normal vocal effort and raised vocal effort. Slight to moderate raised vocal effort often  
182 occurs when a speaker is in a noisy environment or is communicating over a poor-  
183 quality telecommunications channel. Shouting and whispering are obvious extreme  
184 examples of speaking styles. More extreme speaking styles, such as whispering, have  
185 more negative effects on the performance of automatic-speaker-recognition technology

186 than do less extreme speaking styles, such as slight to moderate raised vocal effort  
187 (Kelly & Hansen, 2021). Speaking one language on one recording and another  
188 language on another recording would also be a change in speaking style. A speaker's  
189 speech varies from occasion to occasion, and variation tends to be greater across longer  
190 time intervals between recordings. Very long time intervals between when the  
191 questioned-speaker and known-speaker recordings were made should be accounted for  
192 in the statistical modeling process (Morrison & Kelly, 2019). Common speaker-  
193 extrinsic conditions include different types and volumes of background noise, different  
194 amounts of reverberation, different distances from the speaker to the microphone,  
195 transmission of the speech signal through telecommunications systems that include  
196 bandpass filters and lossy compression, and recordings being saved using lossy  
197 compression. The length of the speech of the speaker of interest on a recording is also  
198 a condition. If high-quality recordings of speakers in suitable speaker-intrinsic  
199 conditions are available, speaker-extrinsic conditions can potentially be simulated by  
200 adding noise, convolving the audio signal with filters, and compressing and  
201 decompressing the signal. Data from longer recordings can easily be truncated to reflect  
202 shorter recordings.

203 Assessing speaker-intrinsic conditions will usually require listening to the questioned-  
204 speaker and known-speaker recordings (see comments on listening in the Diarization  
205 and VAD subsection below). Some speaker-extrinsic conditions such as signal-to-noise  
206 ratio can be quantitatively analyzed. More sophisticated channel-characterization tools  
207 may provide additional information such as information relating to what codecs may  
208 have been applied to a recording. Through the instructing party, the practitioner should  
209 also make enquiries as to the technical properties of systems used to make the casework  
210 recordings, e.g., a recording may be received as "pulse code modulation" (PCM, the  
211 standard uncompressed encoding for audio recordings), but it may have originally been  
212 saved in a lossy format and then exported to PCM. The instructing party will not  
213 usually have such technical information to hand, and the enquiry will usually have to  
214 be passed on to the person or team responsible for installing and maintaining the

215 recording equipment at the organization that supplied the recording.

## 216 **2.5 Training and validation data**

217 There are often mismatches between the conditions of the questioned-speaker  
218 recording and the conditions of the known-speaker recording. For each speaker in the  
219 training and validation datasets, there should be at least one recording that reflects the  
220 conditions of the questioned-speaker recording and at least one recording that reflects  
221 the conditions of the known-speaker recording. In their report, the practitioner should  
222 describe the conditions of the questioned-speaker and known-speaker recordings for  
223 the case, and how they obtained, selected, or simulated recordings that reflect the  
224 conditions for the case. The choice of training and validation data is a subjective  
225 judgment, and whether the data are sufficiently reflective of the conditions of the case  
226 can potentially be disputed.

227 The relevant population and the conditions of the questioned-speaker and known-  
228 speaker recordings can be highly variable from case to case. State-of-the-art automatic-  
229 speaker-recognition technology can produce good results over a range of different  
230 conditions, but the major impediment to conducting forensic casework is availability  
231 of training and validation data representing the populations and reflecting the  
232 conditions of specific cases. Sometimes, new recordings can be made representing the  
233 population and reflecting the conditions for a specific case, but this is usually not  
234 practical – it depends on how easy it is to collect data for the specific population and  
235 specific set of conditions, the time available, and the budget available. Substantial  
236 investment is needed to build databases covering populations and conditions that are  
237 anticipated to occur in a substantial proportion of future cases. If the number of such  
238 databases increases and they cover a wider range of populations and conditions, then it  
239 will become practical to perform evaluations in a larger proportion of the cases for  
240 which forensic voice comparison is requested.

241 Research is needed to assess the effects of varying speaker-intrinsic and speaker-

242 extrinsic conditions, and of varying population. If changes in some conditions result in  
243 substantial changes in likelihood-ratio output, then this will inform practitioners that in  
244 future casework they should use data that closely reflect those conditions and not  
245 substitute data in one condition for data in another. If changes in some conditions result  
246 in negligible changes in likelihood-ratio output, then this will inform practitioners that  
247 in future casework they can substitute data in one condition for data in another, thus  
248 making it easier to obtain data that are sufficiently reflective of the conditions for a  
249 case. The results of such research will also inform future data-collection efforts.

250 The poorer the quality and the shorter the duration of casework recordings, and the  
251 greater the mismatch in recording conditions between questioned-speaker and known-  
252 speaker recordings, the poorer the performance of the forensic-voice-comparison  
253 system is expected to be. In principle, however, there is no minimum threshold below  
254 which forensic voice comparison cannot be performed. As long as suitable training and  
255 validation data can be obtained, the system can be trained and validated under  
256 conditions reflecting those of the case. A decision can subsequently be made about  
257 whether the performance of the system under those conditions is good enough to  
258 proceed to use it to compare the questioned-speaker and known-speaker recordings for  
259 the case. In practice, before training and validating the system, based on existing  
260 research and validation literature, the practitioner may be able to advise the instructing  
261 party in broad terms about the expected level of performance. If the level of  
262 performance is expected to be poor, a decision may be made to not proceed with  
263 training and validation (and not to proceed with data collection if it would be needed  
264 for training and validation).

265

### 266 **3 Software tools**

#### 267 **3.1 System architecture**

268 The high-level architecture of the example system's core software tools is presented in

269 Figure 1. It consists of the following stages:

270 1. speaker diarization and voice-activity detection (VAD)

271 2. feature extraction

272 3. x-vector extraction

273 4. dimension reduction and mismatch compensation using linear discriminant  
274 analysis (LDA)

275 5. calculation of uncalibrated likelihood ratios (scores) using probabilistic linear  
276 discriminant analysis (PLDA)

277 6. calibration

278

279 <Figure 1 about here>

280 **Figure 1.** High-level architecture for the example human-supervised-automatic  
281 forensic-voice-comparison system's core software tools.

282

283 Data from the questioned-speaker recording and data from the known-speaker  
284 recording are processed in parallel through Stages 1–4. Stages 5 and 6 operate on data  
285 from pairs of recordings. Recordings used for training and validating the system (not  
286 shown in Figure 1) are processed in the same manner as the data from the questioned-  
287 speaker and known-speaker recordings. Terminologically: Stage 1 (diarization and  
288 VAD) are key parts of preprocessing; Stage 3 (x-vector extraction) constitutes the  
289 frontend model; and Stages 4–6 (LDA, PLDA, and calibration) constitute the backend  
290 models.

291 We describe each stage of the system in its own subsection below.

## 292 **3.2 Diarization and VAD**

293 Audio recordings received for forensic evaluation often include the speech of more  
294 than one speaker on the same recording channel. Speaker diarization is the process of  
295 dividing a recording into sections spoken by different speakers. Usually, only one  
296 speaker on a recording is of interest. In this situation, the speaker-diarization task is to  
297 find the sections of the recording that correspond to speech of the speaker of interest.  
298 Usually, the different speakers on a recording sound sufficiently different from each  
299 other that this is a trivial task for a forensic practitioner to perform manually. If this is  
300 not the case, then the identity of the speaker at various points in the recording may be  
301 a question requiring forensic evaluation. For manual diarization, the practitioner uses  
302 a software tool that visually presents the waveform, allows the practitioner to select  
303 and listen to sections of the recording, and allows them to add markers and labels  
304 indicating the sections that contain speech of the speaker of interest. The practitioner  
305 should listen in a quiet environment using reference headphones. Marking and labeling  
306 can be performed using any one of multiple commercial or freeware software tools  
307 designed for general use, e.g., Audition, Sound Forge, Audacity, or Praat. The example  
308 system includes SoundLabeller, a marking and labeling tool that is designed  
309 specifically for this task.

310 A protocol can be adopted whereby one practitioner performs the diarization task, a  
311 second practitioner checks the results, and a pre-specified process to resolve any  
312 disagreements is then used. A protocol to reduce the potential for cognitive bias is to  
313 have one practitioner diarize the questioned-speaker recording and another practitioner  
314 diarize the known-speaker recording. That way, no individual practitioner auditorily  
315 compares the questioned-speaker and known-speaker recordings, so no practitioner can  
316 form a subjective judgment as to whether the two recordings are recordings of the same  
317 speaker or not. Strictly following both protocols would require a total of four  
318 practitioners. If this is impractical, a compromise would be to have a long time interval  
319 (e.g., at least several weeks) between when any individual practitioner (e.g., the

320 checker) listens to the questioned-speaker recordings and the known-speaker  
321 recordings.

322 Practitioners may manually diarize the questioned-speaker and known-speaker  
323 recordings for a case, but this may be impractical if there are a large number of  
324 recordings or if the recordings are very long. Manual diarization is unlikely to be  
325 practical for the large numbers of recordings used for training and validating the  
326 forensic-voice-comparison system. Automatic diarization is itself a form of automatic  
327 speaker recognition. The example system uses the automatic-diarization method that  
328 performed the best in the DIHARD'19 diarization challenge, the VBx algorithm (Diez  
329 et al., 2019, 2020a; Landini et al., 2020, 2022). (All the data for the benchmark  
330 validation described below were supplied already diarized.)

331 VAD is the detection and selection of sections of the recording that contain speech, as  
332 opposed to silence, background noise only, or transient noises. VAD is a prerequisite  
333 for diarization, but it is also required to find the sections of a recording containing  
334 speech when there is only one speaker on a recording channel. Although VAD could  
335 be performed manually, automatic VAD is preferred in order to obtain consistent  
336 results. If a practitioner performs manual diarization, they should mark the start of a  
337 section of speech a little early and the end a little late. Automatic VAD will then further  
338 truncate the manually marked section.

339 Simple automatic VAD methods are based only on the intensity of the signal in the  
340 recording, but these methods do not perform well when there is background noise on  
341 the recording, as is common in casework recordings. More sophisticated automatic  
342 VAD methods attempt to distinguish speech sounds from non-speech sounds.  
343 Supervised methods require labeled training data, and tend to not perform well if they  
344 are applied to recordings whose conditions differ from those of the training data.  
345 Unsupervised methods do not require labeled training data, and are more robust to  
346 changes in conditions. Unsupervised methods can achieve similar levels of  
347 performance to supervised methods when the latter are trained and tested on the same

348 conditions (Kinnunen et al., 2016; Nautsch et al., 2016; Tan et al., 2020).

349 The example system uses the rVAD-fast algorithm (Tan et al., 2020). This algorithm  
350 applies two noise-removal processes: The first process attempts to remove transient  
351 noises, and the second process attempts to remove background noise (this is  
352 background-noise removal for the purpose of performing VAD, features used for  
353 forensic voice comparison are extracted from the unmodified audio signal). The next  
354 stage in the algorithm searches for voiced speech sounds using a spectral flatness  
355 detector (which is faster than fundamental-frequency detection, which was used in the  
356 original rVAD algorithm). In order to also include voiceless sounds, the sections of the  
357 recording identified as containing voiced sounds are extended both before and after by  
358 60 frames (600 ms). The final stage uses heuristics based on the energy differences  
359 between frames to select frames deemed to be speech.

### 360 **3.3 Feature extraction**

361 Until recently, “mel-frequency cepstral coefficients” (MFCCs; Davis & Mermelstein,  
362 1980) were the most commonly used features for automatic-speaker-recognition  
363 systems, but “log-mel-filterbank features” have been found to be more effective for x-  
364 vector systems (Alam et al., 2020; Landini et al., 2020; Lee et al., 2020). The example  
365 system uses the implementation of log mel filterbanks described in Young et al. (2015,  
366 §3.1.5).

367 The steps for extracting log-mel-filterbank features are described below, see also  
368 Figure 2 in which the numbers correspond to the numbered steps below.

- 369 1. The speech signal is multiplied by a bell-shaped window. In our example system,  
370 this is a Hamming window with a duration of 25 ms, i.e., 200 samples if the  
371 sampling frequency of the recording is 8 kHz.
- 372 2. The power spectrum of the windowed signal is calculated using a discrete Fourier  
373 transform (DFT). The power spectrum consists of the squares of the magnitudes

374 of the components of the Fourier series (phase information is discarded). For  
375 computational efficiency, the example system uses a 512 point fast Fourier  
376 transform.

377 3. The power spectrum is multiplied by each filter in a filterbank. These are a series  
378 of triangular shaped filters that are equally spaced in the mel-frequency scale. The  
379 example system uses 40 filters that together cover the frequency range 0–4 kHz.  
380 Each filter has a 50% overlap with each of its neighbors.

381 4. For each filter in the filterbank, the logarithm is taken of the result of multiplying  
382 the power spectrum by that filter. This creates a set of 40 values that are output as  
383 a vector of log-mel-filterbank features.

384 The window is advanced through the recording. In the example system, it is advanced  
385 by 10 ms (80 samples). Each time-interval covered by a window is called a frame, and  
386 in the example system there is a 60% overlap between adjacent frames. Steps 1 through  
387 4 are then repeated to produce another vector of log-mel-filterbank features. The  
388 window is repeatedly advanced until feature vectors have been extracted from all  
389 sections of the recording corresponding to the speech of the speaker of interest. A series  
390 of feature vectors arranged in chronological order will be referred to as a feature matrix.

391 <Figure 2 about here>

392 **Figure 2.** Procedure for the calculation of log-mel-filterbank feature vectors. The  
393 numbers correspond to the numbered steps in the main text. DFT = discrete Fourier  
394 transform. (This figure is adapted from Morrison et al., 2020.)

395

## 396 3.4 x-vector extraction

### 397 3.4.1 Overview

398 With one feature vector extracted every 10 ms, i.e., 100 feature vectors extracted per

399 second, and recordings potentially including from several seconds to several minutes  
400 of speech of the speaker of interest, the number of feature vectors extracted per  
401 recording can range from several hundred to tens of thousands. x-vector extraction  
402 converts the feature vectors from a recording into a single x-vector. An x-vector has  
403 the same length irrespective of the duration of the speech of the speaker of interest on  
404 the recording. In addition, x-vector extraction is designed so that, in the  
405 multidimensional space of the x-vectors, x-vectors extracted from different recordings  
406 of the same speaker will be close to each other whereas x-vectors extracted from  
407 recordings of different speakers will be far from each other, i.e., x-vectors have small  
408 within-speaker variability and large between-speaker variability. When x-vectors are  
409 input to subsequent models that calculate likelihood ratios, those models can therefore  
410 produce likelihood-ratio values ranging from much smaller than 1 to much larger than  
411 1 (log-likelihood-ratio values ranging from much less than 0 to much more than 0).

412 x-vectors are extracted using a deep neural network (DNN), which is an artificial neural  
413 network that has multiple layers between the input and output layers. The DNN is  
414 trained using a large number of recordings from each of a large number of speakers.  
415 The speaker-intrinsic and speaker-extrinsic conditions of the recordings should be  
416 diverse, and the speakers should also be diverse. That way, the DNN has the  
417 opportunity to learn about both within-speaker variability and between-speaker  
418 variability. The DNN has one output node for each speaker in the training set. If the  
419 DNN were being used to make a decision as to which of the speakers from the training  
420 set was speaking on a recording, the output node with the highest activation would  
421 correspond to the speaker with the highest posterior probability. In forensic voice  
422 comparison, however, the purpose is not to classify the incoming recording as  
423 belonging to one of the speakers on which the system was already trained, the purpose  
424 is to calculate a likelihood ratio for the comparison of recordings of the questioned  
425 speaker and the known speaker, neither of whom was used to train the system.  
426 Therefore, rather than using the output layer of the DNN, the activations of the nodes  
427 in a pre-final layer of the DNN are instead used as the values of an x-vector. Because

428 the x-vector layer of the DNN is prior to the output layer, rather than capturing  
429 information about the training speakers in particular, it is more abstract and captures  
430 information about properties that can distinguish speakers from one another in general.

431 Key for successful training of a DNN x-vector extractor is to use large amounts of  
432 training data. The data should consist of tens of recordings in diverse speaker-intrinsic  
433 and speaker-extrinsic conditions from each of thousands of diverse speakers. The data  
434 used to train the DNN are not intended to represent the particular population or reflect  
435 the particular conditions of the case under consideration. Once it has been trained,  
436 however, a DNN can be used to extract x-vectors from recordings that do represent and  
437 reflect the populations and conditions specific to a case.

### 438 **3.4.2 Time-delay DNNs**

439 Compared to the current state of the art, the architecture of DNNs initially used for x-  
440 vector extraction was relatively simple. Figure 3 provides a simplified schematic of the  
441 architecture of a DNN based on the description in Snyder et al. (2017). The squares in  
442 the bottom row of the figure represent feature vectors. For the “frame level” of the  
443 DNN, only the time dimension of the feature vectors is shown. Each layer of the frame  
444 level includes a parallel set of nodes and connections for each step in the frequency  
445 dimension of the feature vectors. The second row from the bottom of the figure  
446 represents the input layer of the DNN. The circles represent nodes. Each node is  
447 connected to multiple feature vectors from adjacent time steps. The “activation” of a  
448 node (the value that is passed to the next layer of the DNN), is a function of the  
449 weighted sum of the input values to that node (the function used is often non linear,  
450 and different functions may be used for different layers in the DNN). The weights can  
451 be thought of as properties of the connections feeding into the node of interest from  
452 nodes in the previous layer (or, for the input layer, from the feature matrix) – a higher  
453 weight is associated with a stronger connection (the analogy is with the synapses  
454 between biological neurons). Progressing up through the frame level of the DNN, each  
455 layer combines information from nodes corresponding to multiple time steps in the

456 preceding layer until the time dimension is collapsed to a single node in the time  
457 dimension (not shown in Figure 3, there is still a node for each step in the frequency  
458 dimension).

459 <Figure 3 about here>

460 **Figure 3.** Simplified schematic of the architecture of a time-delay DNN used for x-  
461 vector extraction. (This figure is adapted from Morrison et al., 2020.)

462

463 The frame level of the DNN combines information from a total of 15 time steps, 150  
464 ms if feature vectors are extracted every 10 ms. Recordings of speakers of interest are  
465 longer than 150 ms. Advancing one time step at a time, all the feature vectors from the  
466 speaker of interest on a recording are sequentially presented as input to the DNN, and  
467 the “statistics-pooling layer” calculates the mean and standard-deviation values of the  
468 activations of the nodes in the immediately preceding layer. There is one mean node  
469 and one standard-deviation node for each frequency-step node in the immediately  
470 preceding layer. The circles in the “segmental level” of the figure represent additional  
471 layers of nodes that process information output by the statistics-pooling layer. This is  
472 a fully-connected feed-forward network (within the network, every node in a layer is  
473 connected to every node in the immediately preceding layer and every node in the  
474 immediately following layer). Prior to the output layer, there is a bottleneck layer (it  
475 has fewer nodes than the preceding layer or the following layer), and the activations of  
476 the nodes in this layer are used as the values of an x-vector.

477 To train the DNN, the weights are randomly initialized, a recording is presented, and  
478 the weights are adjusted to increase the relative level of activation of the output node  
479 corresponding to the speaker on the recording. This is repeated multiple times with tens  
480 of recordings from each of thousands of speakers.

481 To extract an x-vector, a recording is presented to the DNN, and the resulting

482 activations of the nodes of the “x-vector layer” are used as the values of the x-vector.  
483 The output layer of the DNN is not used.

### 484 3.4.3 Residual networks (ResNets)

485 Current state-of-the-art x-vector extraction uses more complex DNNs called Residual  
486 Networks (ResNets; He et al., 2016). The example system uses a variant of the  
487 ResNet34 architecture described in Chung et al. (2020a, 2020b). The details of sizes of  
488 layers etc. in the following paragraphs are those of the example system.

489 Each input-layer node of the ResNet receives input from a square “patch” of feature  
490 values which covers 7 time steps by 7 frequency steps in the feature matrix, see Figure  
491 4. There is one input-layer column for each time step in the feature matrix and one  
492 input-layer row for every other frequency step in the feature matrix – the “stride” is 1  
493 in the time dimension and 2 in the feature dimension. The length of the input-layer  
494 rows,  $T$ , is 400. The length of the input-layer columns,  $F$ , is 20 (the length of each  
495 feature vector is 40). A node in the input layer nominally corresponds to the feature-  
496 matrix time and frequency step that is in the center of the  $7 \times 7$  patch. If the center of a  
497 patch is near the edge of the feature matrix (in time or frequency steps), the part of the  
498 patch that extends beyond the edge of the feature matrix feeds in values of 0 (the feature  
499 matrix is padded with zeros). The set of connection weights between each node in the  
500 input layer and its corresponding patch of feature values is called a “kernel”. The same  
501 kernel is used for all input-layer nodes (the kernel is convolved with the matrix of  
502 feature values). Additional kernels are created by initializing the connections with  
503 different sets of weights. Each additional kernel is used for all nodes in an additional  
504 input layer that is parallel to the first input layer. Each parallel input layer creates a  
505 “channel” which is parallel to the other channels. The number of input channels,  $C$ , is  
506 16.

507 <Figure 4 about here>

508 **Figure 4.** Simplified schematic of the feature vectors and the input layer of a ResNet

509 DNN used for x-vector extraction. Only one channel is shown. (This figure is  
510 reproduced from Weber et al., 2022a, and Weber et al., 2022b.)

511

512 Figure 5 provides a simplified schematic of the architecture of the ResNet used for x-  
513 vector extraction by the example system. The ResNet consists of a series of “groups”,  
514 each group consists of a series of “blocks”, and each block consists of a series of layers.

515 <Figure 5 about here>

516 **Figure 5.** Simplified schematic of the architecture of a ResNet DNN used for x-vector  
517 extraction. (This figure is reproduced from Weber et al., 2022a, and Weber et al.,  
518 2022b.)

519

520 Figure 6 provides a simplified schematic of the architecture of a block. The first and  
521 second layer of each block are similar to the input layer of the ResNet in that each node  
522 receives input from a patch of nodes in the immediately preceding layer. These patches  
523 cover 3 time steps by 3 frequency steps by the full number of channels ( $3 \times 3 \times C$ ). In  
524 Groups 2 and 3, the stride for the first layer of the first block is 2 for both the time and  
525 frequency dimensions (hence the size of both these dimensions is halved). For the  
526 second layer of the first block in each of Group 2 and 3, and for both the first and  
527 second layers of all other blocks in all groups, the stride is 1 in each dimension (hence  
528 the size of both these dimensions is unchanged). For the first layer of each of Groups  
529 2 through 4, two kernels are applied to the output of the previous group. This results in  
530 a doubling of the number of channels. The sizes of dimensions  $T$  and  $F$  and the number  
531 of channels  $C$  within each group are provided in Table 1.

532 <Figure 6 about here>

533 **Figure 6.** Simplified schematic of the architecture of a block within a ResNet DNN

534 used for x-vector extraction. Only one channel is shown. The “input” is the last layer  
535 of the previous block. (This figure is reproduced from Weber et al., 2022a, and Weber  
536 et al., 2022b.)

537

538 **Table 1.** Sizes of the dimensions of the components and subcomponents of the ResNet  
539 DNN used by the example system for x-vector extraction.

540 <Table 1 about here>

541

542 After its first two layers, each block has a one-dimensional “squeeze-excitation  
543 network”. Each node in the input layer of this network calculates the mean value of all  
544 the nodes in the previous layer belonging to a single channel, e.g., in the first block  
545 there are 16 channels therefore there are 16 nodes in the input layer of the squeeze-  
546 excitation network. The network then has a bottleneck layer and an output layer. The  
547 output layer has the same number of nodes as the input layer, i.e., one per channel. The  
548 activations of the nodes in the output layer are used to weight the channels relative to  
549 one another. This focuses “attention” on the channels that are more useful for  
550 distinguishing speakers from one another (Cai et al., 2018).

551 For each channel, the output of a block is the elementwise summation of the channel-  
552 weighted output of the block’s second layer and of the original input to the block. If  
553 there is a difference in the number of time or frequency steps or the number of channels  
554 between the previous block and the current block, in order to be able to perform the  
555 elementwise summation, the input to the block is processed through a set of kernels  
556 that alter its dimensions to match those of the current block. This set of kernels is  
557 independent of other sets of kernels. The addition of the input to a block to what would  
558 otherwise be its output is the “residual” that give ResNets their name.

559 Figure 7 provides a simplified schematic of the final stages of the ResNet, which

560 consist of a “statistics-pooling block”, an “x-vector layer”, and an “output layer”. For  
561 each channel, the first layer of the statistic-pooling block collapses the frequency  
562 dimension by calculating the mean of the column of frequency values corresponding  
563 to each time step of the immediately preceding layer. This results in a two-dimensional  
564  $T \times C$  layer of nodes.

565 After the first layer of the statistic-pooling block, similar to the squeeze-excitation  
566 networks in earlier blocks, there is a one-dimensional channel-attention layer which is  
567 the same length as the number of channels. Unlike the squeeze-excitation networks,  
568 the channel-attention layer is a single layer and is only connected to a higher layer, it  
569 does not have input from a lower layer. The activations of the nodes in the channel-  
570 attention layer are therefore learned during training, and thereafter are fixed. The  
571 activations of the nodes in the channel-attention layer are used to weight the channels  
572 of the statistic-pooling block’s first layer. The result, the second layer, is a one-  
573 dimensional layer that is the same length as the number of time steps, and in which the  
574 activation of each node is a function applied to the weighted sum of the activations of  
575 the first layer’s nodes at the corresponding time step. The activations of the nodes in  
576 the second layer are then used to weight the time steps of the statistic-pooling block’s  
577 first layer. The result, the third layer, is a one-dimensional layer that is the same length  
578 as the number of channels, and in which the activation of each node is the weighted  
579 sum of the activations of the first layer’s nodes for the corresponding channel. The third  
580 layer is the output layer of the statistics-pooling block. It has combined information  
581 from across both time and frequency.

582 The output layer of the statistics-pooling block is fully connected to the x-vector layer.  
583 The x-vector layer of the example system has 512 nodes. The x-vector layer is fully  
584 connected to the ResNet’s output layer. The output layer has one node for each speaker  
585 in the training data.

586

587 <Figure 7 about here>

588 **Figure 7.** Simplified schematic of the final stages of a ResNet DNN used for x-vector  
589 extraction. The final stages include the pooling block, the x-vector layer, and the output  
590 layer. Multiple channels are shown. The “input” are the last layers of the previous block  
591 from each of the multiple channels. “ $\times$ ” indicates matrix multiplication. (This figure is  
592 reproduced from Weber et al., 2022a, and Weber et al., 2022b.)

593

594 For extraction of x-vectors, recordings longer (or shorter) than 400 feature vectors can  
595 be presented to the ResNet. Since the same kernels are used for each node in the input  
596 layer, the number of nodes in the rows of the input layer can be increased (or decreased)  
597 to accommodate the number of feature vectors, and the kernels simply repeated for  
598 each node – no retraining is needed to accommodate the different number of feature  
599 vectors. The number of time steps in higher layers can likewise be increased (or  
600 decreased) without the need for retraining. The statistics-pooling block collapses the  
601 time dimension (and the frequency dimension) so that the final one-dimensional layers  
602 of the ResNet have the same numbers of nodes irrespective of the number of feature  
603 vectors and the number of time steps in earlier layers.

### 604 **3.5 LDA**

605 From Stage 6 onward, the data used for training or adapting the backend models should  
606 be representative of the relevant population for the case and should reflect the  
607 conditions of the questioned-speaker and known-speaker recordings for the case,  
608 including any mismatch in conditions between the questioned-speaker and known-  
609 speaker recordings.

610 Linear discriminant functions (see Klecka, 1980) are used for mismatch-compensation  
611 and to reduce the number of dimensions of the x-vector. This process is referred to in  
612 the automatic-speaker-recognition literature as LDA. The example system uses the

613 algorithm described in Hastie et al. (2009, §4.3), and the x-vectors are reduced from  
614 512 to 120 dimensions.

615 The number of recordings available for training that represent the relevant population  
616 for a case and that reflect the conditions of the questioned-speaker and known-speaker  
617 recordings for a case is usually relatively low. In addition to using x-vectors from  
618 recordings that actually reflect the population and conditions for the case (in-domain  
619 data), x-vectors from a large number of non-case-specific recordings from a large  
620 number of speakers (out-of-domain data) can be adapted to simulate this population  
621 and these conditions. The correlation alignment (CORAL) algorithm (Sun et al., 2017;  
622 Alam et al., 2018) linearly shifts and scales the out-of-domain data so that their total  
623 covariance matrix (within-speaker plus between-speaker covariance matrix) matches  
624 that estimated from the in-domain data. The example system uses the CORAL  
625 algorithm described in Alam et al. (2018). The original in-domain data plus the adapted  
626 data are then used to calculate the linear discriminant functions. An alternative method  
627 would be to use the CORAL+ algorithm to adapt the PLDA model (Lee et al., 2019).

628 The in-domain and adapted x-vectors of the training data are transformed using the  
629 linear discriminant functions. The x-vectors that will be used for calibration and  
630 validation are transformed using the same linear discriminant functions.

### 631 **3.6 PLDA**

632 The post-LDA in-domain and adapted x-vectors are used to train a model that in the  
633 automatic-speaker-recognition literature is known as PLDA (Prince & Elder, 2007;  
634 Kenny, 2010; Brümmer & de Villiers, 2010; Sizov et al., 2014). The example system  
635 implements the two-covariance variant of PLDA described in Brümmer & de Villiers  
636 (2010). This is the same as the common-source likelihood-ratio model described in  
637 Ommen & Saunders (2021) and the multivariate normal procedure described in Aitken  
638 & Lucy (2014). For ease of explanation, a univariate version of the two-covariance  
639 PLDA is described below. The analogous multivariate version is then presented.

640 Prior to training the PLDA model, to better fit the assumptions of the model, the post-  
 641 LDA x-vectors in the training data are centered, “whitened” (i.e., rotated and scaled so  
 642 that for the entire training set the variance in each dimension is 1 and the covariance  
 643 between dimensions is 0, note that these are transformations based on the between-  
 644 plus-within-source covariance matrix, not either of the within-source or between-  
 645 source covariance matrices alone), then scaled to unit length in the Euclidian  
 646 multidimensional space (García-Romero & Espy-Wilson, 2011). The x-vectors that  
 647 will be used for calibration and validation are transformed using the centering and  
 648 whitening functions derived from the training data, and then scaled to unit length.

649 In general, a common-source likelihood-ratio model has the form given in Equation (1)  
 650 above. The two-covariance PLDA model assumes Gaussian distributions for same-  
 651 speaker and different-speaker models  $M_s$  and  $M_d$ , and assumes that all speakers have  
 652 the same within-speaker variance, see Equation (2), in which  $\lambda$  is an uncalibrated  
 653 likelihood-ratio value,  $f(x|\mu, \sigma^2)$  is a Gaussian probability-density function  
 654 (parametrized using mean and variance),  $x_q$  and  $x_k$  are the questioned-speaker and  
 655 known-speaker (post-LDA post-centering-whitening-and-scaling) x-vectors  
 656 respectively,  $\mu_r$  is the relevant-population mean,  $\mu_i$  and  $\mu_j$  are means for arbitrary  
 657 individual speakers, and  $\sigma_w^2$  and  $\sigma_b^2$  are the within-speaker variance and the between-  
 658 speaker variance respectively.  $\sigma_w^2$ , and  $\sigma_b^2$  are estimated using the training data.  $\sigma_w^2$  is  
 659 estimated as the pooled-within-speaker variance. Since the training data are centered,  
 660  $\mu_r = 0$ .

661 (2)

$$\begin{aligned}
 662 \quad \lambda &= \frac{\int f(x_q|\mu_i, \sigma_w^2) f(x_k|\mu_i, \sigma_w^2) f(\mu_i|\mu_r, \sigma_b^2) d\mu_i}{\int f(x_q|\mu_i, \sigma_w^2) f(\mu_i|\mu_r, \sigma_b^2) d\mu_i \int f(x_k|\mu_j, \sigma_w^2) f(\mu_j|\mu_r, \sigma_b^2) d\mu_j} \\
 663 \quad &= \frac{f\left(\begin{bmatrix} x_q \\ x_k \end{bmatrix} \middle| \begin{bmatrix} \mu_r \\ \mu_r \end{bmatrix}, \begin{bmatrix} \sigma_w^2 + \sigma_b^2 & \sigma_b^2 \\ \sigma_b^2 & \sigma_w^2 + \sigma_b^2 \end{bmatrix}\right)}{f(x_q|\mu_r, \sigma_w^2 + \sigma_b^2) f(x_k|\mu_r, \sigma_w^2 + \sigma_b^2)}
 \end{aligned}$$

$$664 \quad = \frac{f\left(\begin{matrix} [x_q] \\ [x_k] \end{matrix} \middle| \begin{matrix} [\mu_r] \\ [\mu_r] \end{matrix}, \begin{bmatrix} \sigma_w^2 + \sigma_b^2 & \sigma_b^2 \\ \sigma_b^2 & \sigma_w^2 + \sigma_b^2 \end{bmatrix}\right)}{f\left(\begin{matrix} [x_q] \\ [x_k] \end{matrix} \middle| \begin{matrix} [\mu_r] \\ [\mu_r] \end{matrix}, \begin{bmatrix} \sigma_w^2 + \sigma_b^2 & 0 \\ 0 & \sigma_w^2 + \sigma_b^2 \end{bmatrix}\right)}$$

665 The numerator of Equation (2) integrates over all possible values for individual-  
 666 speaker means given the between-speaker distribution, with the constraint that  $x_q$  and  
 667  $x_k$  come from the same speaker. The denominator of Equation (2) integrates over all  
 668 possible values for individual-speaker means given the between-speaker distribution,  
 669 but does so independently for  $x_q$  and for  $x_k$ . The solutions to the integrals can be  
 670 expressed as bivariate Gaussian distributions in which for the same-speaker model (the  
 671 numerator model) the covariances equal the between-speaker variance,  $\sigma_b^2$ , but for the  
 672 different-speaker model (the denominator model) the covariances are zero. This  
 673 reflects the logic that the values of x-vectors from different recordings of the same  
 674 speaker are expected to be correlated, but the values of x-vectors from recordings of  
 675 different speakers are not expected to be correlated. This is graphically represented in  
 676 Figure 8, in which the different-speaker model has round contours, but the same-  
 677 speaker model has elliptical contours with their major axes in the direction of the  
 678 positively correlated diagonal.

679 <Figure 8 about here>

680 **Figure 8.** Graphical representation of the calculation of a likelihood ratio using a  
 681 univariate two-covariance PLDA model. (This figure is adapted from Morrison et al.,  
 682 2020.)

683

684 The multivariate version of the two-covariance PLDA model is provided in Equation  
 685 (3), in which  $f(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$  is a multivariate Gaussian probability-density function, and the  
 686 scalar features, means, and variances of Equation (2) are replaced by their analogous  
 687 feature vectors, mean vectors, and covariance matrices.

688 (3)

$$689 \quad \lambda = \frac{f\left(\begin{bmatrix} \mathbf{x}_q \\ \mathbf{x}_k \end{bmatrix} \middle| \begin{bmatrix} \boldsymbol{\mu}_r \\ \boldsymbol{\mu}_r \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_w + \boldsymbol{\Sigma}_b & \boldsymbol{\Sigma}_b \\ \boldsymbol{\Sigma}_b & \boldsymbol{\Sigma}_w + \boldsymbol{\Sigma}_b \end{bmatrix}\right)}{f(\mathbf{x}_q | \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_w + \boldsymbol{\Sigma}_b) f(\mathbf{x}_k | \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_w + \boldsymbol{\Sigma}_b)}$$

690

691 **3.7 Calibration**

692 The output of the two-covariance PLDA model was described as an uncalibrated  
 693 likelihood ratio. This is because the model requires estimation of a large number of  
 694 parameter values using a limited amount of data. For the example system, it requires  
 695 the estimation of two covariance matrices in a 120 dimension space, i.e., a total of  
 696 14,520 parameter values. The output of the PLDA is therefore not expected to be well  
 697 calibrated. In the automatic-speaker-recognition literature, the logarithms of the  
 698 uncalibrated likelihood ratios are called scores, but note that they are not similarity  
 699 scores (Morrison & Enzinger, 2018; Neumann & Ausdemore, 2020; Neumann et al.,  
 700 2020).

701 A calibration model is trained using a dataset which will be called a calibration set.  
 702 This is a dataset that was not used to train earlier stages of the system and that should  
 703 be representative of the relevant population for the case and should reflect the  
 704 conditions of the questioned-speaker and known-speaker recordings for the case. Each  
 705 speaker in the calibration set should have at least one recording reflecting the  
 706 conditions of the questioned-speaker recording and at least one recording reflecting the  
 707 conditions of the known-speaker recording. From the calibration set, pairs of  
 708 recordings are constructed such that one member of each pair reflects the questioned-  
 709 speaker conditions and the other member reflects the known-speaker conditions. A  
 710 large number of pairs should be constructed in which the two recordings in each pair  
 711 come from the same speaker, and a large number of pairs should be constructed in  
 712 which the two recordings in each pair come from different speakers. These recordings

713 are processed through Stages 1–5 of our example system, resulting in a set of scores  
714 from same-speaker pairs and a set of scores from different-speaker pairs. These scores  
715 are univariate, and are used to train a parsimonious calibration model. For the  
716 parsimonious univariate model, the number of parameter values to be estimated is small  
717 compared to the amount of training data, hence the output of the model is well  
718 calibrated. A commonly used model is logistic regression (Pigeon et al., 2000;  
719 González-Rodríguez et al., 2007; Morrison, 2013, 2021). Logistic regression fits a  
720 linear model in the log-likelihood-ratio space, which only requires the estimation of  
721 two parameter values, an intercept  $\beta_0$  and a slope  $\beta_1$ , see Equation (4).

722 (4)

$$723 \quad \log(\Lambda) = \beta_0 + \beta_1 \log(\lambda)$$

724 The example system uses regularized logistic regression as described in Morrison &  
725 Poh (2018). (This model is fitted using a regularized version of the conjugate-gradient  
726 method.) Regularization reduces the slope,  $\beta_1$ , with the result that the calibrated log  
727 likelihood ratio is closer to the neutral value of 0 than would otherwise be the case (the  
728 likelihood-ratio value is closer to 1). This reduces the probability of overstating the  
729 strength of evidence in either direction. For the validations conducted using the  
730 example system, the regularization weight was set to be equivalent to 1 pseudo-speaker  
731 (see Morrison & Poh, 2018, for details).

732 An additional step that some systems use before calibration is score normalization.  
733 Adaptive Symmetric Norm (AS-Norm; Cumani et al., 2011) is currently the standard  
734 method for score normalization. Score normalization was not used for the validations  
735 conducted using the example system.

## 736 **4 Validation**

### 737 **4.1 Data and training**

738 The performance of the example system was validated on a benchmark dataset

739 (*forensic\_eval\_01*) that reflects the conditions of a forensic case. The benchmark  
740 dataset and validation protocols are described in Morrison & Enzinger (2016a). The  
741 speakers are adult male Australian-English speakers. The questioned-speaker condition  
742 reflects a 46 s long landline-telephone call, with background babble noise, saved using  
743 lossy compression. The known-speaker condition reflects a 126 s long interview  
744 recorded in a reverberant room, with background ventilation-system noise. The  
745 durations just stated are for the amount of speech of the speaker of interest after semi-  
746 automatic diarization but before applying VAD. The questioned-speaker-condition and  
747 known-speaker-condition recordings were recorded on different occasions separated  
748 by approximately a week or more. Each speaker in the test set was recorded on at least  
749 two occasions. The test set consists of a total of 223 recordings from 61 speakers, 61  
750 in questioned-speaker condition and 162 in known-speaker condition, allowing for the  
751 construction of 111 same-speaker pairs of recordings and 6720 different-speaker pairs  
752 of recordings (from 3660 pairs of speakers). The dataset also includes a training set  
753 consisting of a total of 423 recordings from 105 speakers (191 recordings in  
754 questioned-speaker condition and 232 in known-speaker condition).

755 The x-vector extractor was trained using approximately 1 million recordings total from  
756 approximately 6 thousand speakers from the VoxCeleb2 database (Chung et al., 2018;  
757 Nagrani et al., 2020). As out-of-domain data for CORAL, approximately 30 thousand  
758 recordings total were used from approximately 2.7 thousand speakers from the  
759 SRE2018 Test dataset (Greenberg et al., 2020). As in-domain data for LDA and PLDA  
760 training, the *forensic\_eval\_01* training set was used.

761 For training the calibration model and for validation, the *forensic\_eval\_01* test set was  
762 used. To avoid training and testing on the same data (and in accordance with the  
763 recommendations in the “Consensus on validation of forensic voice comparison”;  
764 Morrison et al., 2021), leave-one-speaker-out / leave-two-speakers-out cross-validation  
765 was used: In a cross-validation loop in which the score to be calibrated was a same-  
766 speaker score, e.g., a recording of speaker A compared to another recording of speaker

767 A, all scores that resulted from comparisons in which one or both members of the pair  
768 was a recording of speaker A were excluded from the data used to train the calibration  
769 model (leave-one-speaker-out). In a cross-validation loop in which the score to be  
770 calibrated was a different-speaker score, e.g., a recording of speaker A compared to a  
771 recording of speaker B, all scores that resulted from comparisons in which one or both  
772 members of the pair was a recording of speaker A or a recording of speaker B were  
773 excluded from the data used to train the calibration model (leave-two-speakers-out).

774 Prior to use, if not already in this format, all recordings were converted to 8 kHz  
775 sampling rate 16 bit quantization PCM.

## 776 4.2 Results

777 A Tippett plot showing validation results is presented in Figure 9. The same-speaker  
778 and different-speaker curves have relatively shallow slopes, indicating good  
779 performance, and they cross near a log-likelihood-ratio value of 0, indicating good  
780 calibration. The Tippett plot indicates that the validation results would support  
781 likelihood-ratio values into the thousands in favor of the same-speaker hypothesis and  
782 into the tens of thousands in favor of the different-speaker hypothesis ( $\log_{10}$  likelihood  
783 ratios beyond +3 and -4 respectively).

784

785 <Figure 9 about here>

786 **Figure 9.** Tippett plot of the results of validating the example system ( $E^3FS^3\alpha$ ) on the  
787 *forensic\_eval\_01* dataset. (This figure is adapted from Weber et al., 2022b.)

788

789 A virtual special issue of the journal *Speech Communication*, reports on the validation  
790 of several systems using the *forensic\_eval\_01* dataset. A summary of results is  
791 presented in Morrison & Enzinger (2019). Table 2 presents an extract of the  $C_{llr}$  results

792 from the virtual special issue, plus the  $C_{lr}$  result for the example system. The  $C_{lr}$  value  
793 for the example system was 0.208. The lower the  $C_{lr}$  value, the better the performance  
794 of the system. A system that gave no information and always responded with a  
795 likelihood ratio of 1 irrespective of the input would have a  $C_{lr}$  value of 1. In terms of  
796  $C_{lr}$ , the example system performed equally as well as the best-performing system from  
797 the virtual special issue, Phonexia SID-BETA4 (Jessen et al., 2019).

798

799 **Table 2.**  $C_{lr}$  values from the best-performing version of each system validated in the  
800 *Speech Communication* virtual special issue (Morrison & Enzinger, 2019), plus the  $C_{lr}$   
801 result for the example system ( $E^3FS^3\alpha$ ).

802 <Table 2 about here>

803

### 804 4.3 Discussion

805 The validation results could be used to decide whether the example system should be  
806 used to calculate and submit to court a likelihood ratio for comparison of a questioned-  
807 speaker recording and a known-speaker recording in a case for which the  
808 *forensic\_eval\_01* dataset represented the relevant population and reflected the  
809 conditions for that case. For cases involving other populations and conditions,  
810 validations with data representing those populations and reflecting those conditions  
811 would need to be conducted before deciding whether the system could be used and the  
812 results submitted to courts.

813 Since the publication of the virtual special issue, improvements may have been made  
814 to the actively-developed systems included in Table 2 (Nuance, Phonexia, and  
815 VOCALISE), and it may be that the newer versions of these systems would obtain  
816 better results.

817

## 818 **5 Conclusion**

819 The human-supervised-automatic analytical approach to forensic voice comparison in  
820 conjunction with the likelihood-ratio interpretive framework has been described. The  
821 description included practitioner tasks, including adoption of the relevant hypotheses  
822 for the case, the assessment of the conditions of the questioned-speaker and known-  
823 speaker recordings in the case, and the selection of data representing the relevant  
824 population and reflecting the conditions for the case. It also included an example  
825 forensic-voice-comparison system based on state-of-the-art automatic-speaker-  
826 recognition technology, and validation of that system using a benchmark dataset  
827 reflecting the conditions of a real forensic case.

828

## 829 **6 Relevant webpages**

830 E<sup>3</sup> Forensic Speech Science System (E<sup>3</sup>FS<sup>3</sup>)

831 <https://e3fs3.forensic-voice-comparison.net/>

832 Virtual special issue of the journal *Speech Communication*: “Multi-laboratory  
833 evaluation of forensic voice comparison systems under conditions reflecting those of a  
834 real forensic case (forensic\_eval\_01)”

835 [https://www.sciencedirect.com/journal/speech-communication/special-](https://www.sciencedirect.com/journal/speech-communication/special-issue/10KTJHC7HNM)  
836 [issue/10KTJHC7HNM](https://www.sciencedirect.com/journal/speech-communication/special-issue/10KTJHC7HNM)

837

## 838 **7 References**

839 Aitken, C.G.G. and Lucy, D. (2004). Evaluation of trace evidence in the form of  
840 multivariate data. *Applied Statistics* **53**, 109–122.

841 (<http://dox.doi.org/10.1046/j.0035-9254.2003.05271.x>) [Corrigendum: (2004)]

- 842           **53**, 665–666. <http://dox.doi.org/10.1111/j.1467-9876.2004.02031.x>]
- 843 Alam, J., Bhattacharya, G. and Kenny, P. (2018). Speaker verification in mismatched  
844 conditions with frustratingly easy domain adaptation. *Proceedings of Odyssey*  
845 *2018: The speaker and language recognition workshop*, pp. 176–180.  
846 (<https://doi.org/10.21437/Odyssey.2018-25>)
- 847 Alam, J., Boulianne, G., Burget, L., Dahmane, M., Díez Sánchez, M., Lozano-Díez,  
848 A., Glembek, O., St-Charles, P., Lalonde, M., Matejka P., Mizera, P., Monteiro,  
849 J., Mosner, L., Noisieux, C., Novotný, O., Plchot, O., Rohdin, J., Silnova, A.,  
850 Slavicek, J., Stafylakis, T., Wang, S. and Zeinali, H. (2020). Analysis of ABC  
851 submission to NIST SRE 2019 CMN and VAST challenge. *Proceedings of*  
852 *Odyssey 2020: The speaker and language recognition workshop*, pp. 289–295.  
853 (<https://doi.org/10.21437/Odyssey.2020-41>)
- 854 Brümmer, N. and de Villiers, E. (2010). The speaker partitioning problem.  
855 *Proceedings of Odyssey 2010: The speaker and language recognition workshop*,  
856 pp. 194–201. ([https://www.isca-](https://www.isca-speech.org/archive_open/odyssey_2010/od10_034.html)  
857 [speech.org/archive\\_open/odyssey\\_2010/od10\\_034.html](https://www.isca-speech.org/archive_open/odyssey_2010/od10_034.html))
- 858 Cai, W., Chen, J. and Li, M. (2018). Exploring the encoding layer and loss function  
859 in end-to-end speaker and language recognition system. *Proceedings of Odyssey*  
860 *2018: The speaker and language recognition workshop*, pp. 74–81.  
861 (<https://10.21437/Odyssey.2018-11>)
- 862 Chung, J.S., Nagrani, A. and Zisserman, A. (2018). VoxCeleb2: Deep speaker  
863 recognition. *Proceedings of Interspeech*, pp. 1086–1090.  
864 (<https://doi.org/10.21437/Interspeech.2018-1929>)
- 865 Chung, J.S., Huh, J., Mun, S., Lee, M., Heo, H.S., Choe, S., Ham, C., Jung, S., Lee,  
866 B.-J. and Han, I. (2020a). In defence of metric learning for speaker recognition.  
867 *Proceedings of Interspeech*, pp. 2977–2981.  
868 (<https://doi.org/10.21437/Interspeech.2020-1064>)

- 869 Chung, J.S., Huh, J. and Mun, S. (2020b). Delving into VoxCeleb: Environment  
870 invariant speaker recognition. *Proceedings of Odyssey 2020: The speaker and*  
871 *language recognition workshop*, pp. 349–356.  
872 (<https://doi.org/10.21437/Odyssey.2020-49>)
- 873 Cumani, A., Batzu, P.D., Colibro, D., Vair, C., Laface, P. and Vasilakakis, V. (2011)  
874 Comparison of speaker recognition approaches for real applications.  
875 *Proceedings of Interspeech*, pp. 2365–2368. ([https://isca-](https://isca-speech.org/archive/interspeech_2011/i11_2365.html)  
876 [speech.org/archive/interspeech\\_2011/i11\\_2365.html](https://isca-speech.org/archive/interspeech_2011/i11_2365.html))
- 877 Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for  
878 monosyllabic word recognition in continuously spoken sentences. *IEEE*  
879 *Transactions on Acoustics, Speech, and Signal Processing* **28**, 357–366.  
880 (<https://doi.org/10.1109/TASSP.1980.1163420>)
- 881 Diez, M., Burget, L., Wang, S., Rohdin, J. and Černocký H. (2019). Bayesian HMM  
882 based x-vector clustering for speaker diarization. *Proceedings of Interspeech*, pp.  
883 346–350. (<http://doi.org/10.21437/Interspeech.2019-2813>)
- 884 Diez, M., Burget, L., Landini, F. and Černocký J. (2020a). Analysis of speaker  
885 diarization based on Bayesian HMM with eigenvoice priors. *IEEE/ACM*  
886 *Transactions on Audio, Speech, and Language Processing* **28**, 355–368.  
887 (<https://doi.org/10.1109/TASLP.2019.2955293>)
- 888 García-Romero D. and Espy-Wilson C.Y. (2011). Analysis of i-vector length  
889 normalization in speaker recognition systems. *Proceedings of Interspeech*, pp.  
890 249–252. (<https://doi.org/10.21437/Interspeech.2011-53>)
- 891 González-Rodríguez, J., Rose P., Ramos, D., Toledano, D.T. and Ortega-García, J.  
892 (2007). Emulating DNA: Rigorous quantification of evidential weight in  
893 transparent and testable forensic speaker recognition. *IEEE Transactions on*  
894 *Speech and Audio Processing* **15**, 2104–2115.  
895 (<https://doi.org/10.1109/TASL.2007.902747>)

- 896 Greenberg, C., Sadjadi, O., Singer, E., Walker, K., Jones, K., Wright, J. and Strassel,  
897 S. (2020). 2018 NIST Speaker Recognition Evaluation Test Set (LDC2020S04).  
898 Linguistic Data Consortium. (<https://catalog.ldc.upenn.edu/LDC2020S04>)
- 899 Hansen, J.H.L. and Bořil, H. (2018). On the issues of intra-speaker variability and  
900 realism in speech, speaker, and language recognition tasks. *Speech*  
901 *Communication* **10**, 94–108. (<https://doi.org/10.1016/j.specom.2018.05.004>)
- 902 Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The elements of statistical*  
903 *learning: Data mining, inference and prediction* (2nd edn.). New York:  
904 Springer.
- 905 He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image  
906 recognition. *Proceedings of the IEEE Conference on Computer Vision and*  
907 *Pattern Recognition (CVPR)*, pp. 770–778.  
908 (<https://doi.org/10.1109/CVPR.2016.90>)
- 909 Jessen, M., Bortlík, J., Schwarz, P. and Solewicz, Y.A. (2019). Evaluation of  
910 Phonexia automatic speaker recognition software under conditions reflecting  
911 those of a real forensic voice comparison case (forensic\_eval\_01). *Speech*  
912 *Communication* **111**, 22–28. (<https://doi.org/10.1016/j.specom.2019.05.002>)
- 913 Kelly, F. and Hansen, H.L. (2021). Analysis and calibration of Lombard effect and  
914 whisper for speaker recognition. *IEEE Transactions on Audio, Speech, and*  
915 *Language Processing* **29**, 927–942.  
916 (<https://ieeexplore.ieee.org/document/9330795>)
- 917 Kenny, P. (2010). Bayesian speaker verification with heavy tailed priors. *Proceedings*  
918 *of Odyssey 2010: The Speaker and Language Recognition Workshop*, paper 14.  
919 ([https://www.isca-speech.org/archive\\_open/odyssey\\_2010/od10\\_014.html](https://www.isca-speech.org/archive_open/odyssey_2010/od10_014.html))
- 920 Kinnunen, T., Sholokhov, A., el Khoury, E., Thomsen, D.A.L., Sahidullah, M. and  
921 Tan, Z-H. (2016). HAPPY team entry to NIST OpenSAD challenge: A fusion of

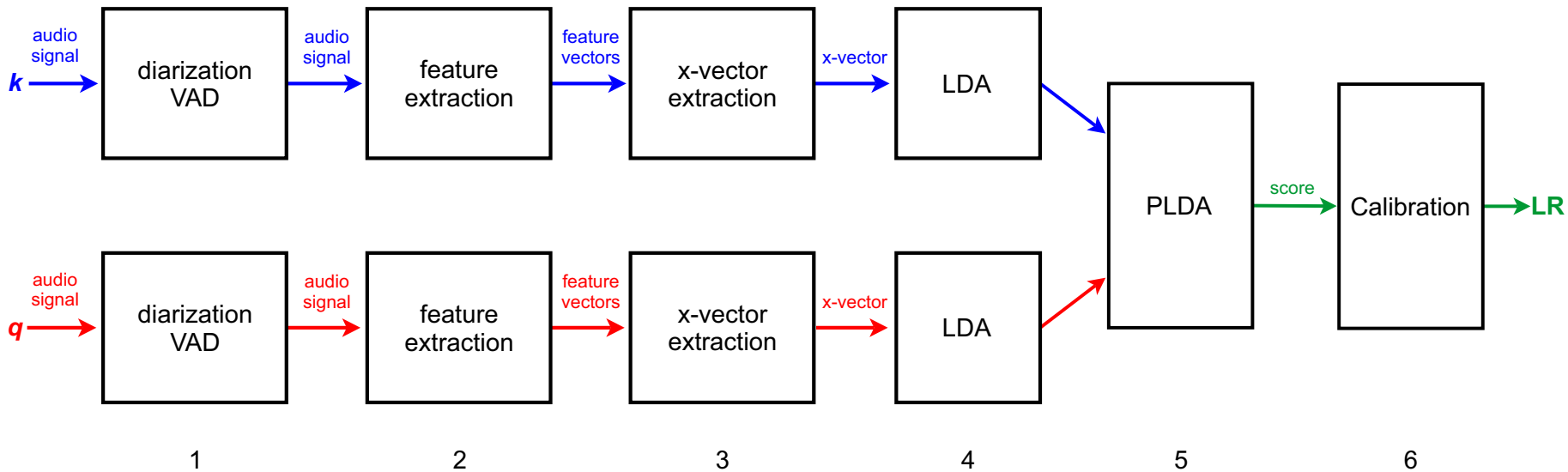
- 922 short-term unsupervised and segment i-vector based speech activity detectors.  
923 *Proceedings of Interspeech*, pp. 2992–2996.  
924 (<https://doi.org/10.21437/Interspeech.2016-1281>)
- 925 Klecka, W.R. (1980). *Discriminant analysis*. Beverly Hills, CA: Sage.
- 926 Landini, F., Wang, S., Díez, M., Burget, L., Matejka, P., Zmolíková, K., Mosner, L.,  
927 Silnova, A., Plchot, O., Novotný, O., Zeinali, H. and Rohdin J. (2020a). BUT  
928 system for the second DIHARD speech diarization challenge. *Proceedings of the*  
929 *IEEE International Conference on Digital Signal Processing (ICASSP)*, pp.  
930 6529–6533. (<https://doi.org/10.1109/ICASSP40776.2020.9054251>)
- 931 Landini, F., Profant, J., Díez, M. and Burget, L. (2022). Bayesian HMM clustering of  
932 x-vector sequences (VBx) in speaker diarization: theory, implementation and  
933 analysis on standard tasks. *Computer Speech & Language* **71**, 101254.  
934 (<https://doi.org/10.1016/j.csl.2021.101254>)
- 935 Lee, K.A., Wang, Q. and Koshinaka T. (2019). The CORAL+ algorithm for  
936 unsupervised domain adaptation of PLDA. *Proceedings of the IEEE*  
937 *International Conference on Digital Signal Processing (ICASSP)*, pp. 5821–  
938 5825. (<https://doi.org/10.1109/ICASSP.2019.8682852>)
- 939 Lee, K.A., Yamamoto, H., Okabe, K., Wang, Q., Guo, L., Koshinaka, T., Zhang, J.  
940 and Shinoda, K. (2020). NEC-TT system for mixed-bandwidth and multi-  
941 domain speaker recognition. *Computer Speech & Language* **61**, 101033.  
942 (<https://doi.org/10.1016/j.csl.2019.101033>)
- 943 Matějka, P., Plchot, O., Glembek, O., Burget, L., Rohdin, J., Zeinali, H., Mošner, L.,  
944 Silnova, A., Novotný, O., Díez, M. and Černocký, J.H. (2020). 13 years of  
945 speaker recognition research at BUT, with longitudinal analysis of NIST SRE.  
946 *Computer Speech & Language* **63**, 101035.  
947 (<https://doi.org/10.1016/j.csl.2019.101035>)
- 948 Morrison, G.S. (2013). Tutorial on logistic-regression calibration and fusion:

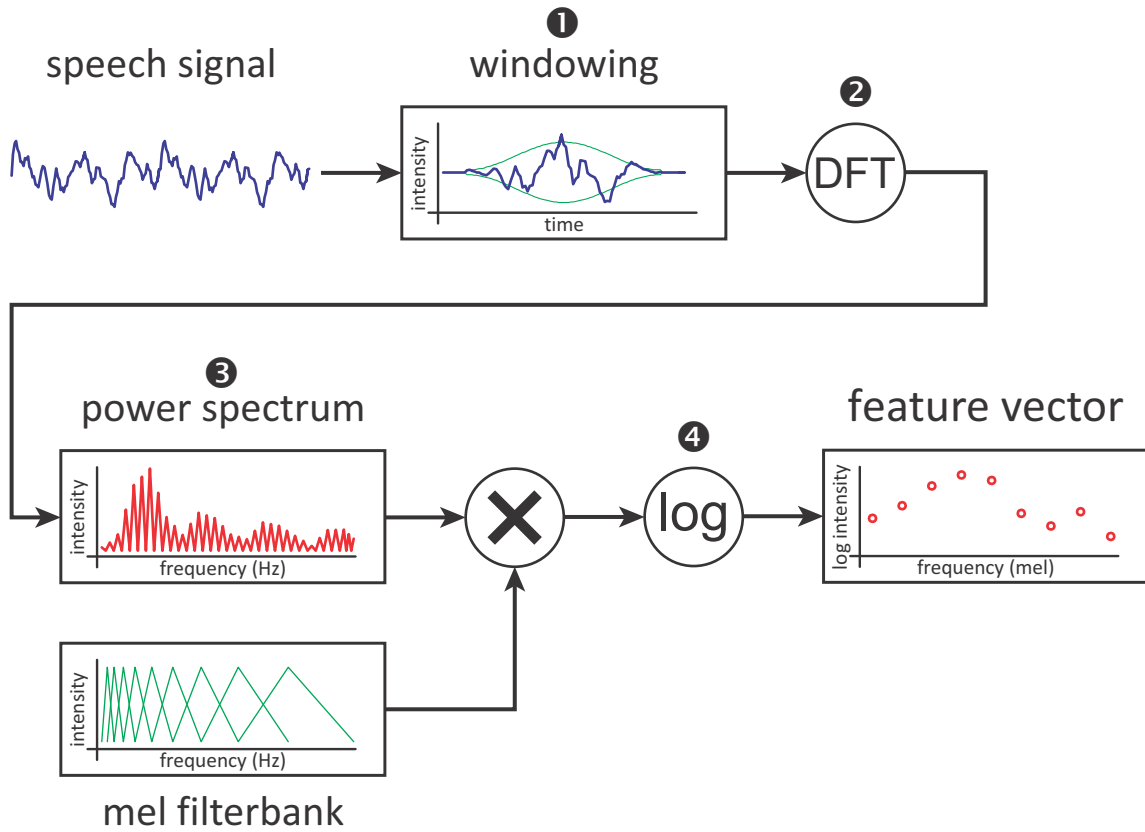
- 949 converting a score to a likelihood ratio. *Australian Journal of Forensic Sciences*  
950 **45**, 173–197. (<http://dx.doi.org/10.1080/00450618.2012.733025>)
- 951 Morrison, G.S. (2021). In the context of forensic casework, are there meaningful  
952 metrics of the degree of calibration? *Forensic Science International: Synergy* **3**,  
953 100157. (<https://doi.org/10.1016/j.fsisyn.2021.100157>)
- 954 Morrison, G.S. and Enzinger, E. (2016). Multi-laboratory evaluation of forensic voice  
955 comparison systems under conditions reflecting those of a real forensic case  
956 (forensic\_eval\_01) - Introduction. *Speech Communication* **85**, 119–126.  
957 (<http://dx.doi.org/10.1016/j.specom.2016.07.006>)
- 958 Morrison, G.S. and Enzinger, E. (2018). Score based procedures for the calculation of  
959 forensic likelihood ratios – Scores should take account of both similarity and  
960 typicality. *Science & Justice* **58**, 47–58.  
961 (<http://dx.doi.org/10.1016/j.scijus.2017.06.005>)
- 962 Morrison, G.S. and Enzinger, E. (2019). Multi-laboratory evaluation of forensic voice  
963 comparison systems under conditions reflecting those of a real forensic case  
964 (forensic\_eval\_01) - Conclusion. *Speech Communication* **112**, 37–39.  
965 (<https://doi.org/10.1016/j.specom.2019.06.007>)
- 966 Morrison, G.S., Enzinger, E., Hughes, V., Jessen, M., Meuwly, D., Neumann, C.,  
967 Planting, S., Thompson, W.C., van der Vloed, D., Ypma, R.J.F., Zhang, C.,  
968 Anonymous, A. and Anonymous, B. (2021). Consensus on validation of forensic  
969 voice comparison. *Science & Justice* **61**, 229–309.  
970 (<https://doi.org/10.1016/j.scijus.2021.02.002>)
- 971 Morrison, G.S., Enzinger, E., Ramos, D., González-Rodríguez, J. and Lozano-Díez,  
972 A. (2020). Statistical models in forensic voice comparison. In Banks, D.L.,  
973 Kafadar, K., Kaye, D.H. and Tackett, M. (eds.) *Handbook of Forensic Statistics*,  
974 pp. 451–497. Boca Raton, FL: CRC. (<https://doi.org/10.1201/9780367527709>)

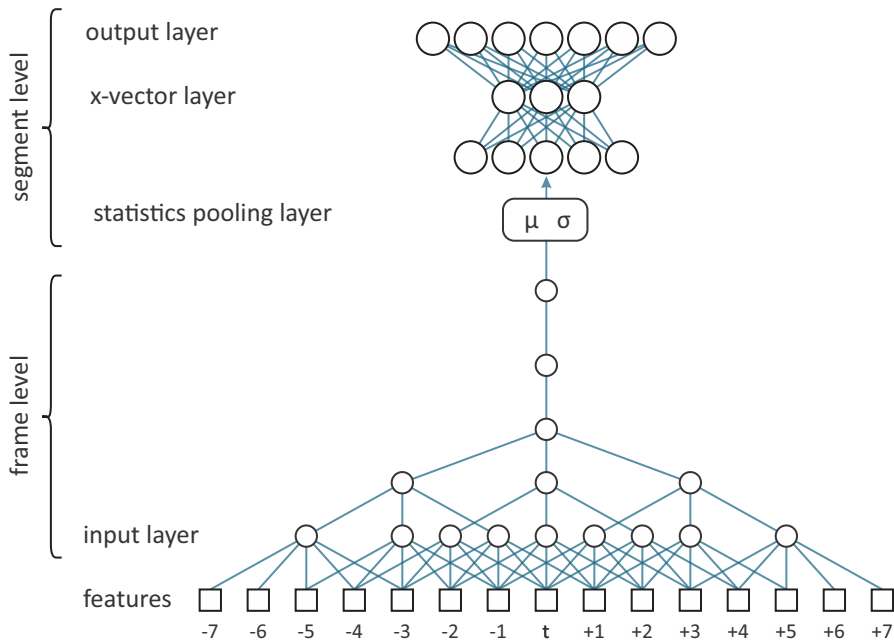
- 975 Morrison, G.S., Enzinger, E. and Zhang, C. (2016). Refining the relevant population  
976 in forensic voice comparison – A response to Hicks et alii (2015) The  
977 importance of distinguishing information from evidence/observations when  
978 formulating propositions. *Science & Justice* **56**, 492–497.  
979 (<http://dx.doi.org/10.1016/j.scijus.2016.07.002>)
- 980 Morrison, G.S. and Kelly, F. (2019). A statistical procedure to adjust for time-interval  
981 mismatch in forensic voice comparison. *Speech Communication* **112**, 15–21.  
982 (<https://doi.org/10.1016/j.specom.2019.07.001>)
- 983 Morrison, G.S. and Poh, N. (2018). Avoiding overstating the strength of forensic  
984 evidence: Shrunk likelihood ratios / Bayes factors. *Science & Justice* **58**, 200–  
985 218. (<http://dx.doi.org/10.1016/j.scijus.2017.12.005>)
- 986 Nagrani, A., Chung, J.S., Xie, W. and Zisserman, A. (2020). Voxceleb: Large-scale  
987 speaker verification in the wild. *Computer Speech and Language* **60** 101027.  
988 (<https://doi.org/10.1016/j.csl.2019.101027>)
- 989 Nautsch, A., Bamberger, R. and Busch, C. (2016). Decision robustness of voice  
990 activity segmentation in unconstrained mobile speaker recognition  
991 environments. *Proceedings of the International Conference of the Biometrics  
992 Special Interest Group (BIOSIG)*, pp. 1–7.  
993 (<https://doi.org/10.1109/BIOSIG.2016.7736916>)
- 994 Neumann, C. and Ausdemore, M. (2020). Defence against the modern arts: The curse  
995 of statistics – Part II: ‘Score-based likelihood ratios’. *Law, Probability and Risk*  
996 **19**, 21–42. (<http://dx.doi.org/10.1093/lpr/mgaa006>)
- 997 Neumann, C., Hendricks, J. and Ausdemore, M. (2020). Statistical support for  
998 conclusions in fingerprint examinations. In Banks, D., Kafadar, K., Kaye, D.H.,  
999 Tackett, M. (eds.) *Handbook of forensic statistics*, pp. 277–324. Boca Raton,  
1000 FL: CRC. (<https://doi.org/10.1201/9780367527709>)

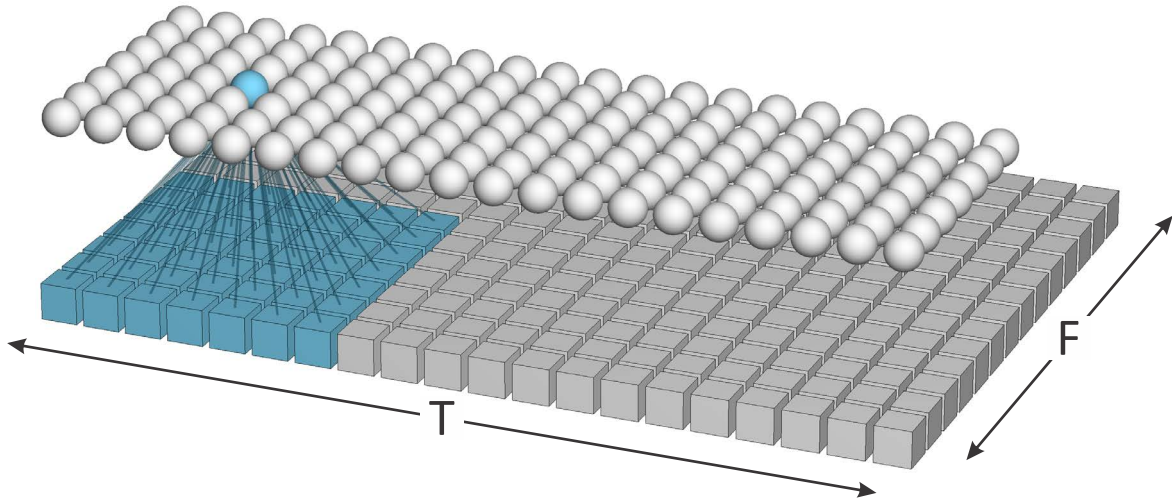
- 1001 Ommen, D.M. and Saunders, C.P. (2021). A problem in forensic science highlighting  
1002 the differences between the Bayes factor and likelihood ratio. *Statistical Science*  
1003 **36**, 344–359. (<https://doi.org/10.1214/20-STS805>)
- 1004 Pigeon, S., Druyts, P. and Verlinde, P. (2000). Applying logistic regression to the  
1005 fusion of the NIST'99 1-speaker submissions. *Digital Signal Processing* **10**,  
1006 237–248. (<https://doi.org/10.1006/dspr.1999.0358>)
- 1007 Prince, S.J.D. and Elder, J.H. (2007). Probabilistic linear discriminant analysis for  
1008 inferences about identity. *Proceedings of the IEEE 11th International*  
1009 *Conference on Computer Vision*, pp. 1–8.  
1010 (<https://doi.org/10.1109/ICCV.2007.4409052>)
- 1011 Sizov, A., Lee, K.A. and Kinnunen, T. (2014). Unifying probabilistic linear  
1012 discriminant analysis variants in biometric authentication. In Fränti, P., Brown,  
1013 G., Loog, M., Escolano, F. and Pelillo, M. (eds.) *Structural, syntactic, and*  
1014 *statistical pattern recognition*, pp. 464–475. Berlin: Springer.  
1015 ([https://doi.org/10.1007/978-3-662-44415-3\\_47](https://doi.org/10.1007/978-3-662-44415-3_47))
- 1016 Snyder, D., García-Romero, D., Povey, D. and Khudanpur, S. (2017). Deep neural  
1017 network embeddings for text-independent speaker verification. *Proceedings of*  
1018 *Interspeech*, pp. 999–1003. (<https://doi.org/10.21437/Interspeech.2017-620>)
- 1019 Sun, B., Feng, J. and Saenko, K. (2017). Correlation alignment for unsupervised  
1020 domain adaptation. In: Csurka G. (ed.) *Domain adaptation in computer vision*  
1021 *applications. Advances in computer vision and pattern recognition*. Cham:  
1022 Springer. ([https://doi.org/10.1007/978-3-319-58347-1\\_8](https://doi.org/10.1007/978-3-319-58347-1_8))
- 1023 Tan, Z., Sarkar, A.K. and Dehak, N. (2020). rVAD: An unsupervised segment-based  
1024 robust voice activity detection method. *Computer Speech & Language* **59**, 1–21.  
1025 (<https://doi.org/10.1016/j.csl.2019.06.005>)
- 1026 Villalba, J., Chen, N., Snyder, D., García-Romero, D., McCree, A., Sell, G.,

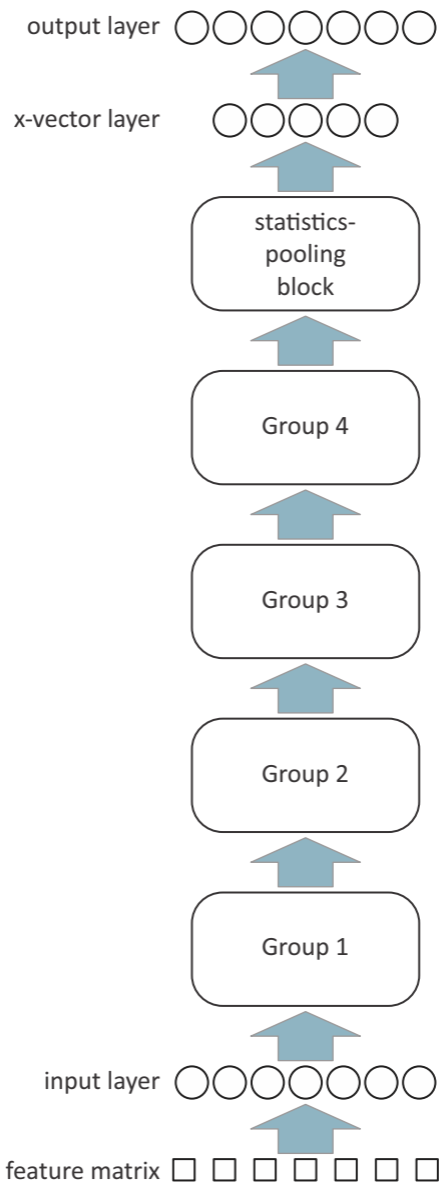
- 1027 Borgstrom, J., García-Perera, L.P., Richardson, F., Dehak R., Torres-  
1028 Carrasquillo, P.A. and Dehak, N. (2020). State-of-the-art speaker recognition  
1029 with neural network embeddings in NIST SRE18 and Speakers in the Wild  
1030 evaluations. *Computer Speech & Language* **60**, 101026.  
1031 (<https://doi.org/10.1016/j.csl.2019.101026>)
- 1032 Weber, P., Enzinger, E., and Morrison, G.S. (2022a) E<sup>3</sup> forensic speech science  
1033 system (E<sup>3</sup>FS<sup>3</sup>): Technical report on design and implementation of software  
1034 tools. (<https://forensic-voice-comparison.net/E3FS3/>)
- 1035 Weber, P., Enzinger, E., Labrador, B., Lozano-Díez, A., Ramos, D., González-  
1036 Rodríguez, J. and Morrison G.S. (2022b). Validation of the alpha version of the  
1037 E<sup>3</sup> forensic speech science system (E<sup>3</sup>FS<sup>3</sup>) core software tools. *Forensic Science*  
1038 *International: Synergy* **4**, 100223. (<https://doi.org/10.1016/j.fsisyn.2022.100223>)
- 1039 Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X. (A.), Moore, G.,  
1040 Odell, J., Ollason, D., Povey, D., Ragni, A., Valtchev, V., Woodland, P. and  
1041 Zhang, C., (2015). The HTK book. Cambridge University Engineering  
1042 Department. (<https://htk.eng.cam.ac.uk/>)

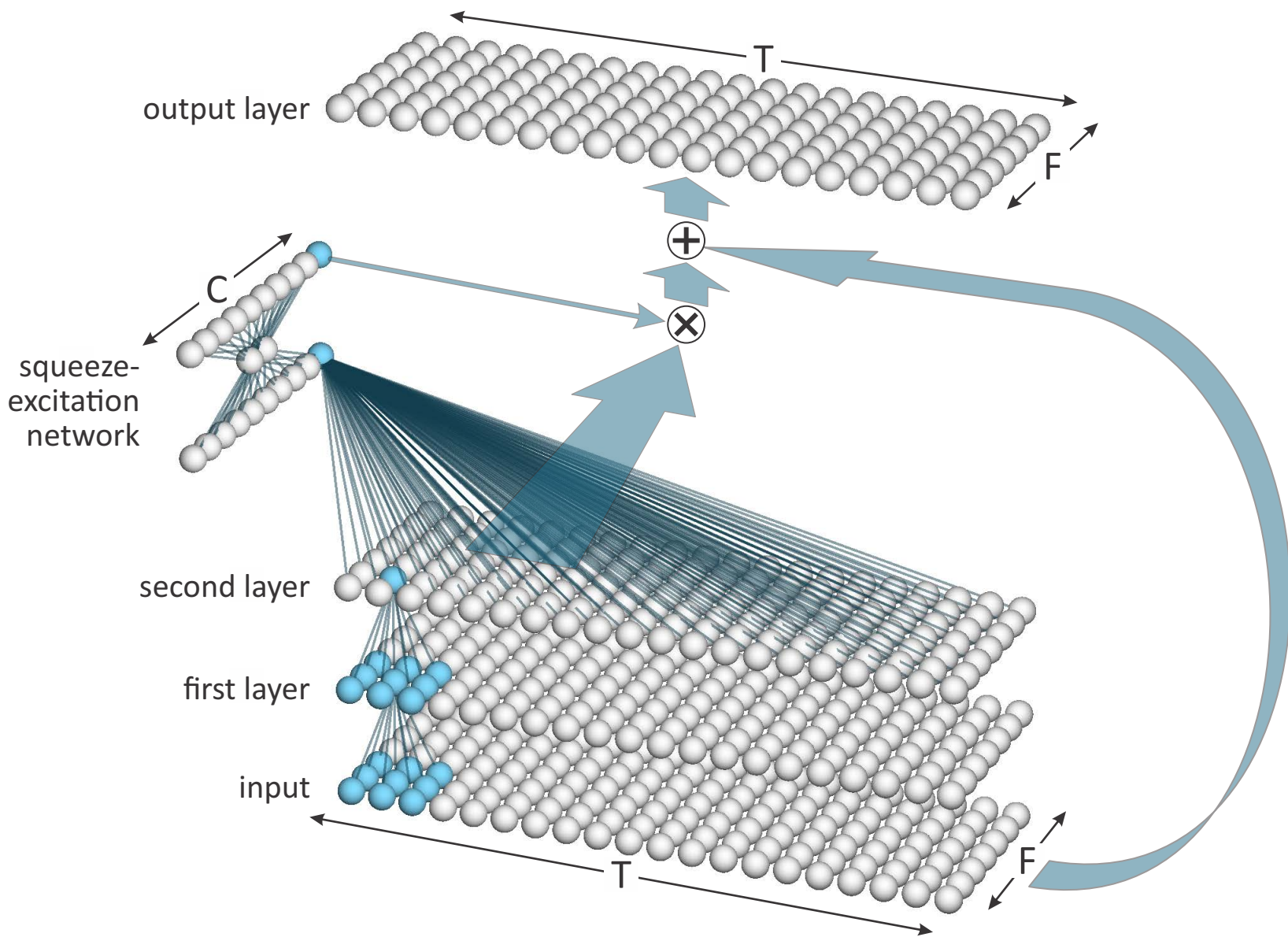


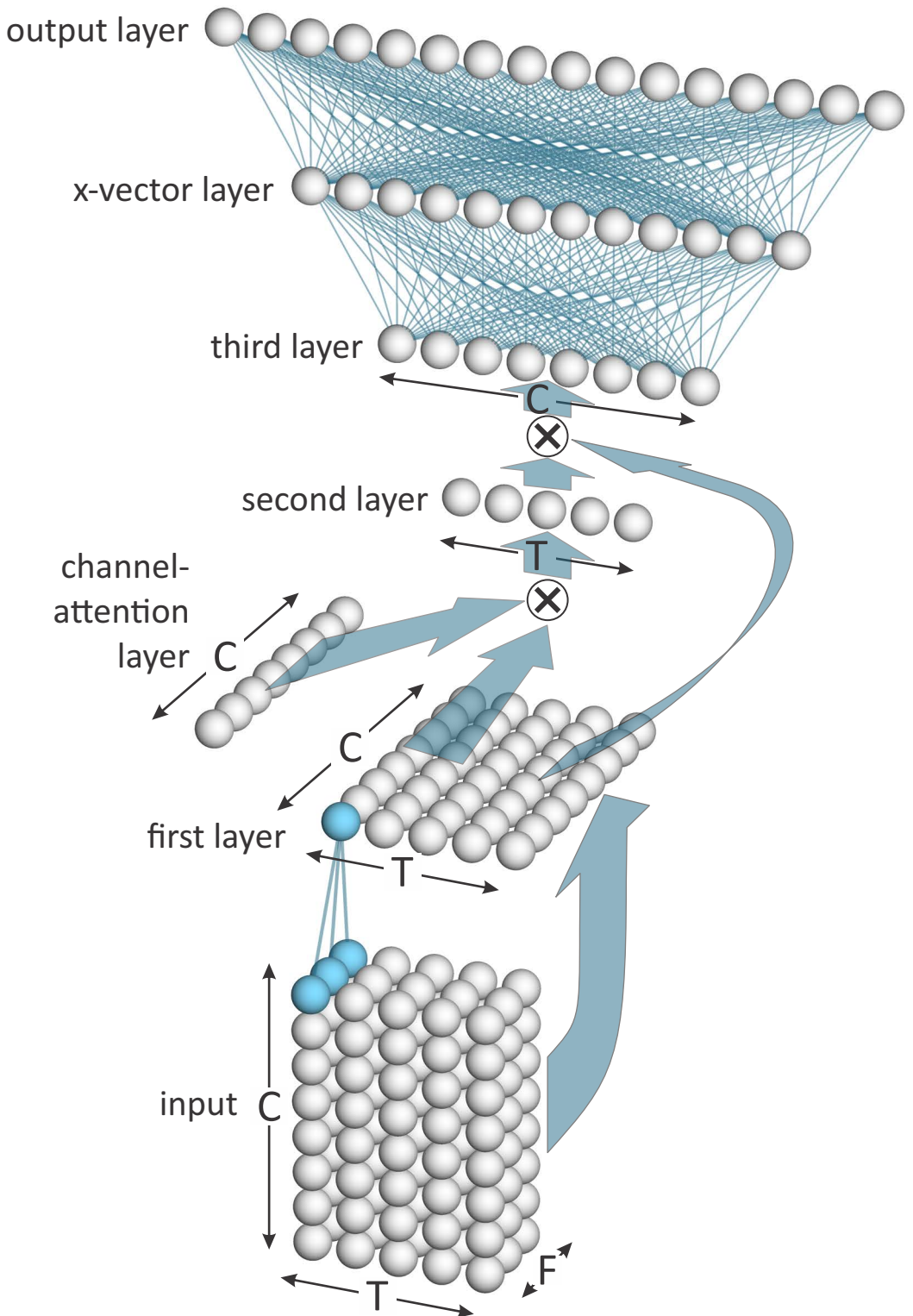


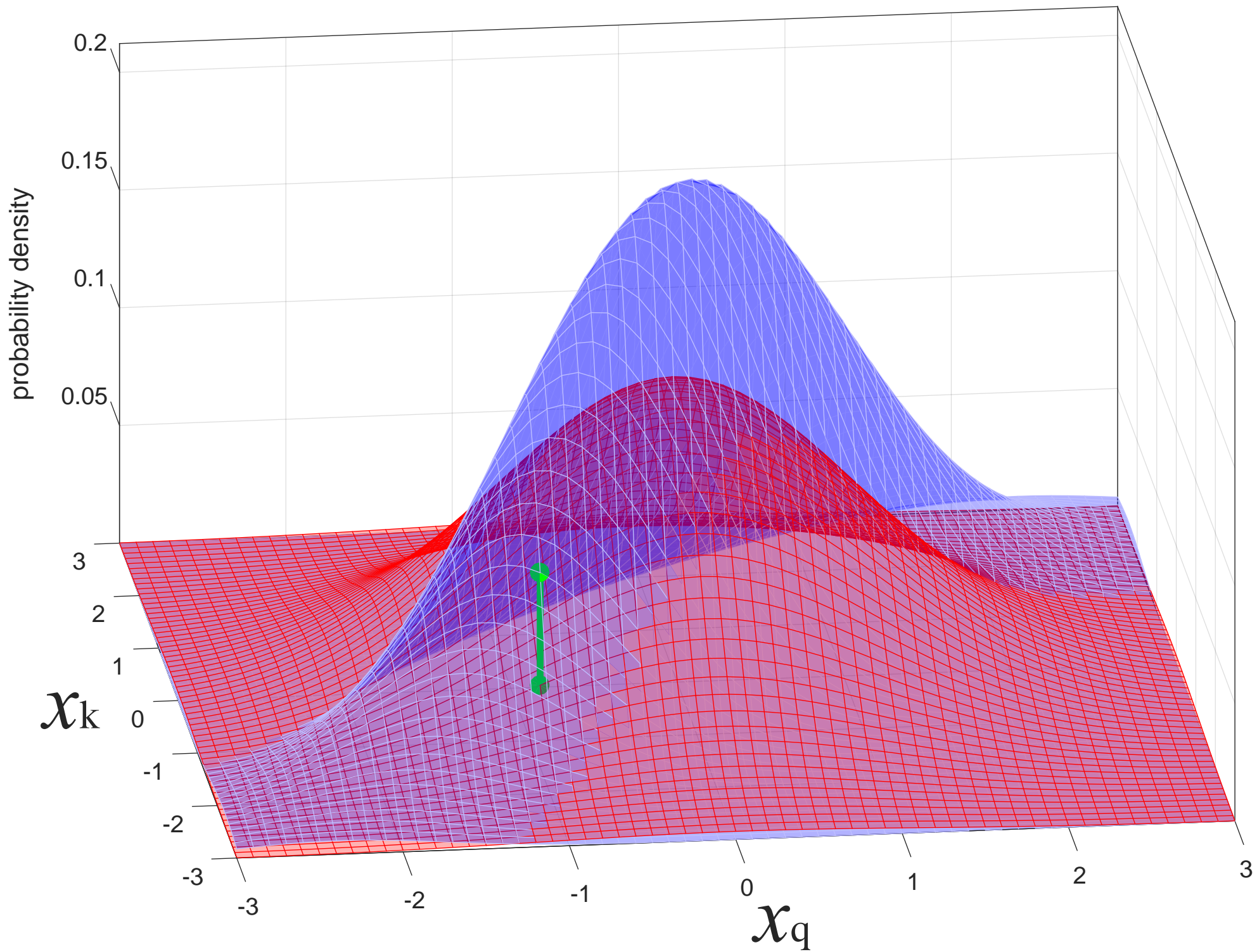


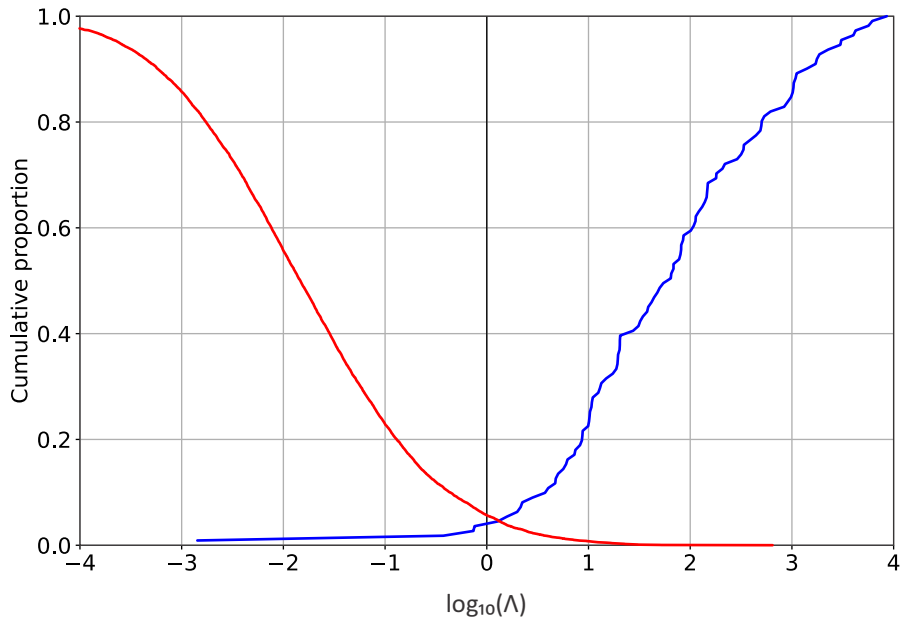












**Table 1.** Dimensions of the components of the ResNet DNN used by the example system for x-vector extraction.

Component	Subcomponent	Dimensions		
		time <i>T</i>	frequency <i>F</i>	channels <i>C</i>
Feature vectors	–	400	40	1
Input layer	–	400	20	16
Group 1	3 blocks	400	20	16
Group 2	4 blocks	200	10	32
Group 3	6 blocks	100	5	64
Group 4	3 blocks	100	5	128
Statistics-pooling block	Layer 1	100	1	128
	Channel-attention layer	1	1	128
	Layer 2	100	1	1
	Layer 3	1	1	128
x-vector layer	–	1	1	512
Output layer	–	1	1	Number of training speakers

**Table 2.**  $C_{lr}$  values from the best-performing version of each system validated in the *Speech Communication* virtual special issue (Morrison & Enzinger, 2019), plus the  $C_{lr}$  result for the example system ( $E^3FS^3\alpha$ ).

<b>System</b>	<b>Type</b>	<b><math>C_{lr}</math></b>
Batvox 3.1	GMM-UBM	0.593
MSR GMM-UBM	GMM-UBM	0.576
MSR GMM i-vector	GMM i-vector	0.449
Batvox 4.1	GMM i-vector	0.365
Phonexia XL3	DNN bottleneck	0.294
Nuance 9.2	GMM i-vector	0.285
VOCALISE 2017B	GMM i-vector	0.267
Nuance 11.1	DNN senone	0.255
VOCALISE 2019A	x-vector	0.246
<b><math>E^3FS^3\alpha</math></b>	x-vector	0.208
Phonexia BETA4	x-vector	0.207