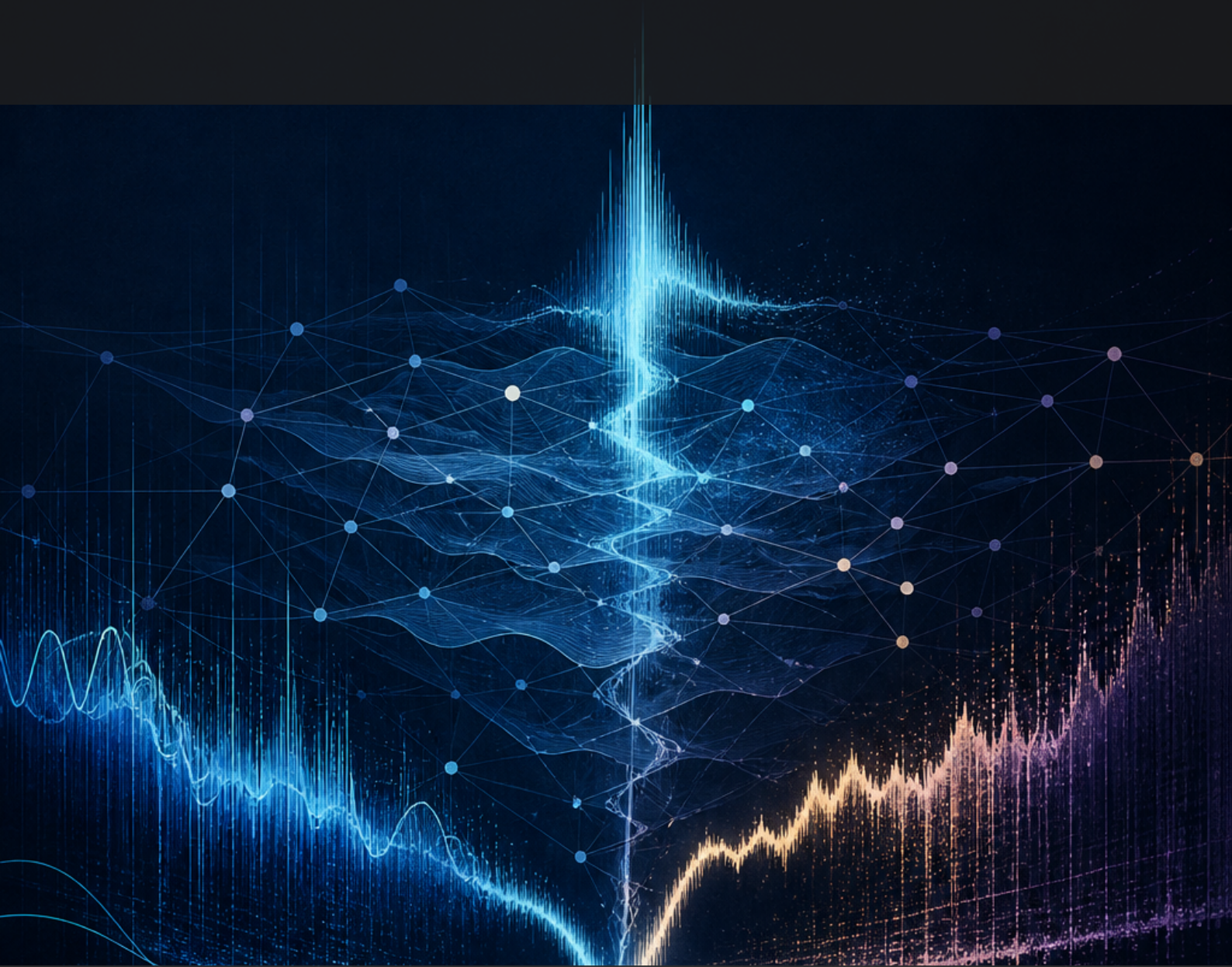


Shortcut learning in speech deepfake detection

Samuel Goldie

to obtain the degree of MSc in Data Science and Artificial Intelligence Technology
Delft University of Technology, The Netherlands
in collaboration with [Sensity AI](#)



Student number: 6125735
Project duration: November 2025 – July 2026
Thesis committee: Prof. dr. Odette Scharenborg (MC),
Dr. Luciano Cavalcante Siebert (II),
External Advisor: Vasileios Papadopoulos

TU Delft, Chair
TU Delft
Sensity AI

Summary

Speech deepfake detectors can perform extremely well when they are trained and tested on one benchmark, yet perform poorly on recordings from other sources. This raises a concern: a detector may learn differences between datasets rather than a general distinction between genuine and deepfake speech. This thesis tests whether three detectors — RawNet2, LCNN, and Wav2Vec-AASIST — use such dataset-related shortcuts.

First, a data audit of four public datasets shows that simple classifiers can identify which dataset a recording came from using several kinds of audio measurements. This establishes that source-related clues are available, but not that the deepfake detectors use them. To test detector reliance directly, the thesis introduces the *mixing ratio protocol*. The protocol changes which of two datasets supplies the genuine and deepfake recordings during training, while the evaluation recordings remain fixed. If a detector uses dataset source as a shortcut, reversing this relationship should cause its decisions to change.

This is what occurs. Across all three detectors, reversing the training composition moves the equal error rate (EER) on the same test recordings from 0.003–0.028 to 0.78–0.87. An EER of 0 is perfect, 0.5 is chance, and a value above 0.5 means that the predictions are systematically inverted. A similar response appears with a second dataset pair. By contrast, the results are much flatter when the same procedure is applied to two groups of speakers from one dataset. This control shows that the mixing procedure does not create the main response by itself.

Further experiments test whether one audio property explains the dependence. Reducing the frequency range, matching the average signal level, and removing low-frequency energy do not remove it. Dataset source can also be predicted from several different kinds of measurements. These findings are consistent with source information being represented through several overlapping or untested properties, but they do not identify which properties each detector used. Overall, dataset identity is available as an exploitable shortcut in speech deepfake detection. Future benchmarks should therefore include diverse recording and generation pipelines in both genuine and deepfake classes.

Contents

Summary	i
1 Introduction	1
1.1 Speech deepfake detection and standard evaluation	1
1.2 The central problem: learning the source instead of the task	1
1.3 What prior evidence shows—and what it does not	2
1.4 Research gap, objective, and research questions	3
1.5 Approach at a glance	3
1.6 Thesis outline	3
2 Technical background	4
2.1 From speech to an audio file	4
2.1.1 Waveforms, sampling, and frequency	4
2.1.2 Containers, codecs, and numerical representation	4
2.1.3 Stored files and decoded waveforms	5
2.2 How genuine and deepfake speech are produced	5
2.2.1 Genuine recordings	5
2.2.2 Text-to-speech and voice conversion	5
2.2.3 Vocoder and neural-codec waveform generation	5
2.3 From an audio file to a detector input	6
2.3.1 Decoding and preprocessing	6
2.3.2 Representation, front end, and classifier	6
2.4 Three detector paths	6
2.4.1 Raw waveform: RawNet2	6
2.4.2 Time–frequency representation: LCNN	7
2.4.3 Pretrained speech representation: Wav2Vec-AASIST	7
3 Related work	8
3.1 Shortcut learning and benchmark interpretation	8
3.2 Generalisation in speech deepfake detection	8
3.2.1 Representation choice and partial SSL robustness	9
3.3 Documented sources of unintended information	9
3.3.1 Dataset-construction cues	9
3.3.2 Generator and vocoder fingerprints	10
3.3.3 Recording, channel, encoding, and storage traces	10
3.4 How shortcut reliance has been tested	10
3.4.1 Availability and association	10
3.4.2 Cue removal and artefact balancing	11
3.4.3 Cross-dataset evaluation and changing the source–label relationship	11
3.5 Research gap and positioning of this thesis	11
4 Research design and methodology	12
4.1 Research logic and method ownership	12
4.2 Data flow and units of analysis	12
4.3 Evidence block 1: availability of source information	13
4.4 Evidence block 2: controlled test of source dependence	13
4.4.1 Formal construction	13
4.4.2 Fixed evaluation configurations	14
4.4.3 Primary fixed-test comparison	14
4.4.4 Intermediate ratios	14
4.4.5 Control, replication, and architectural scope	15

4.5	Evidence block 3: testing possible sources of the shortcut	15
4.5.1	Retraining after selected audio changes	15
4.6	Supporting context and inference boundaries	15
5	Experimental implementation	16
5.1	Implementation roadmap	16
5.2	Datasets and selection rationale	16
5.3	Preprocessing, data units, and fixed test conditions	17
5.3.1	From selector entry to model input	17
5.3.2	File-level selection balance	18
5.3.3	Fixed test configurations	18
5.4	Detector configurations and training	18
5.5	Evaluation measures	19
5.6	Evidence block 1: availability of source information	20
5.7	Reference context: within-dataset performance and gap decomposition	20
5.8	Evidence block 2: controlled test of source dependence	20
5.8.1	Same-corpus speaker baseline and control	20
5.8.2	Primary source-label intervention	21
5.8.3	Second-pair and seeded replication	21
5.9	Evidence block 3: characterising the dependence	21
5.9.1	Prediction pattern	21
5.9.2	Processing changes used to test possible sources of the shortcut	21
6	Results and general discussion	23
6.1	How to read the results	23
6.2	Evidence block 1: availability of source information	23
6.2.1	Visualising source and class structure	23
6.2.2	Predicting the source dataset	24
6.2.3	Supporting feature associations	25
6.3	Reference performance within and across datasets	25
6.4	Evidence block 2: controlled test of source dependence	25
6.4.1	Same-corpus speaker baseline and control	25
6.4.2	Primary source-label intervention	26
6.4.3	Replication with SpeechFake and ASVspoof 5	26
6.5	Evidence block 3: characterising the dependence	28
6.5.1	What the trained models predict	28
6.5.2	Testing possible sources of the shortcut	28
6.6	General discussion	30
6.6.1	Answer to RQ1: to what extent do the detectors rely on shortcuts?	30
6.6.2	Answer to RQ2: what is the nature of the shortcuts?	30
6.6.3	Implications for benchmark interpretation	30
6.6.4	Implications for detector architectures	31
6.6.5	Methodological contribution and researcher reflection	31
6.6.6	Societal and field context	31
6.6.7	Limitations	31
6.6.8	Future work	31
7	Conclusion	33
7.1	Research problem and approach	33
7.2	Answers to the research questions	33
7.3	Main contribution and implications	34
7.4	Reproducibility	34
7.5	Scope and next steps	34
	References	35
A	Supplementary analyses and full results	38
A.1	Supplementary source-availability results	38
A.2	Reference baselines and gap decomposition	40

A.3	Exact seeded-replication results	40
A.4	Possible shortcut sources	41
A.4.1	Transformation designs	41
A.4.2	High-pass result grids	41
A.4.3	Bandwidth-limitation grids	43
A.4.4	Matched RMS-processing grid	44
B	Use of generative AI	45

Introduction

Is this recording genuine human speech, or was it generated or altered by a computer? Speech deepfake detection addresses this forensic question. As systems that generate and modify speech have improved, distinguishing genuine recordings from deepfakes has become an important research problem [1].

The difficulty is not only whether a detector gives the correct answer. It is also *why* the detector gives that answer. A model can perform well because it has learned evidence of speech generation or manipulation. It can also perform well because it has found an accidental difference between the genuine and deepfake recordings in its benchmark. Both models may obtain the same test score, although only the first has learned the intended task. This thesis investigates how such unintended dependence can be exposed.

1.1. Speech deepfake detection and standard evaluation

Speech deepfake detection is commonly formulated as a binary classification task. A detector receives an audio recording and produces a score indicating whether the speech is *genuine* or *deepfake*. In this thesis, *genuine speech* means speech produced by a human speaker, without synthetic generation or manipulation. *Deepfake speech* means speech that was generated or manipulated by a computer. These terms are used consistently throughout the thesis.

The intended task is broader than recognising the examples used to develop a detector. A useful detector should respond to evidence associated with generation or manipulation and should remain useful across speakers, recording conditions, and generation systems. This is challenging because different generation systems can leave different traces and because every audio file contains information about both the speech and the process through which the file was created [1].

Most detectors are developed using supervised learning. A labelled benchmark is divided into training, validation, and test sets. The training set is used to fit the model, the validation set supports development decisions, and the test set is held back for final evaluation. When all three sets come from the same benchmark, this is called *within-dataset evaluation*. Fixed benchmarks and held-out recordings allow researchers to compare systems under controlled conditions without testing on the exact examples used for training. The ASVspoof challenge series is a prominent example of this procedure [2, 3, 4].

The thesis reports performance mainly using the *equal error rate* (EER). A lower EER indicates better separation between genuine and deepfake recordings. Chapter 5 defines this measure in detail.

A held-out score therefore shows how well a detector separates the two classes under the benchmark conditions. It does not reveal which information the detector used. This matters because the training, validation, and test recordings from one benchmark can share the same speakers, generation systems, recording procedures, and audio-processing choices. A detector can use any shared property that also predicts the class.

1.2. The central problem: learning the source instead of the task

Genuine and deepfake recordings have different creation histories. Within a benchmark, the two classes may also come from different speakers, recording conditions, generation systems, or data-processing

pipelines. If these differences remain stable across the benchmark splits, they can help predict the class even when they are not a general sign of deepfake speech.

Consider two datasets, A and B, each containing genuine and deepfake speech. Suppose that the training and validation data contain mostly genuine recordings from A and mostly deepfakes from B. The dataset source now helps predict the class. This connection between dataset source and the assigned genuine/deepfake label is called the *source-label relationship*. A detector can improve its benchmark score by learning differences between A and B, even when those differences are not a dependable distinction between genuine and deepfake speech. If the source-label relationship later changes, this rule may fail.

This is an example of *shortcut learning*: a model uses a pattern that predicts the label in its development data but is not a dependable rule for the intended task [5]. Here, *dataset source* means where the recordings came from and the properties associated with that origin; it is not a literal dataset name stored in the audio. Some source-related clues may be meaningful evidence of a particular generator, while others may be accidental products of data collection or processing. A benchmark score alone cannot reveal which clues the detector used.

Figure 1.1 illustrates the relationship examined in this thesis. In the aligned composition, source and label point in the same direction: source A contributes the genuine recordings and source B contributes the deepfakes. In the reversed composition, the assignment is exchanged. The central question is whether changing this relationship changes what a detector learns.

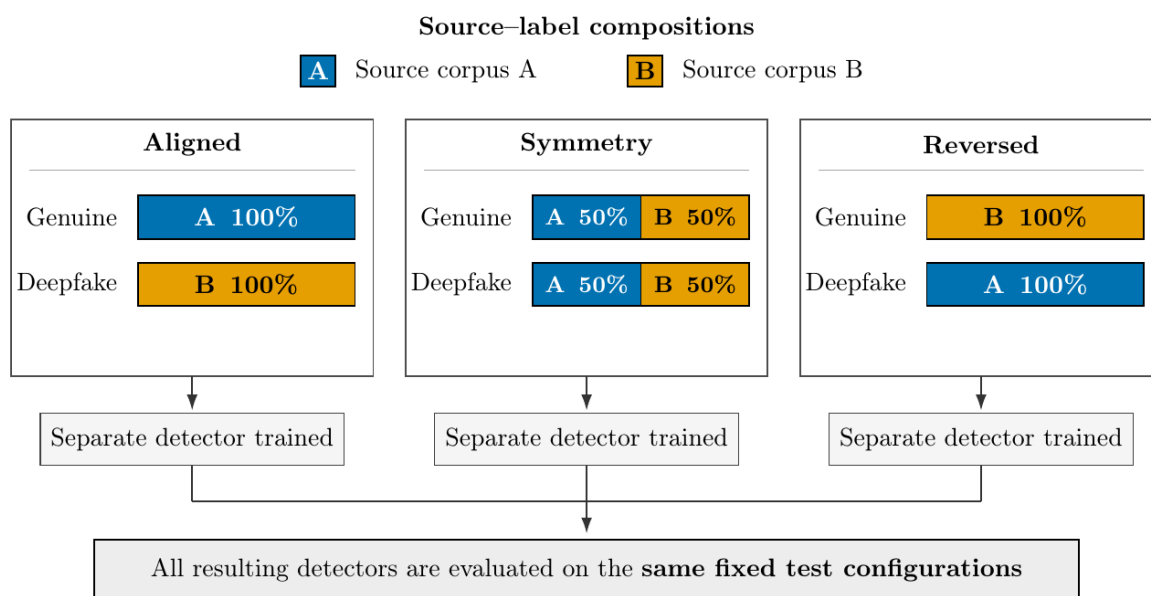


Figure 1.1: Simplified source-label compositions. A separate detector is developed under each composition, and all detectors are evaluated on the same recordings. In the middle panel, sources A and B contribute equal proportions within both classes.

Changing the source-label relationship changes many source-related properties at once. The experiment can therefore show that a detector depends on source-related information, but it cannot identify the particular property responsible for that dependence. The thesis treats these as two separate questions.

1.3. What prior evidence shows—and what it does not

Previous studies show that speech deepfake datasets can contain unintended clues. Differences in silence, telephone tones, corrupted audio, and background noise have predicted class labels or affected detector performance [6, 7, 8]. Generation systems can also leave recognisable patterns, and a detector that specialises in the systems seen during training may fail on unfamiliar ones [9, 10, 11]. These findings show that narrow dataset- or generator-specific rules are possible.

Cross-dataset evaluation provides a related warning: detectors that work well on one dataset often perform worse on another [12, 13, 11]. Prior analyses suggest that, in several tested settings, much of this decrease occurs because the training and test datasets differ, rather than simply because the test dataset is intrinsically harder [14, 15]. However, the decrease does not reveal what the detector learned, because many properties change at once. Likewise, showing that a clue is recognisable in the data does not show that a neural detector uses it. This motivates a controlled test of detector dependence.

1.4. Research gap, objective, and research questions

Previous work has identified individual shortcuts and has shown that detector performance often decreases across datasets. What remains unclear is whether detector development depends on the broader source–label relationship. Ordinary cross-dataset evaluation trains a detector on one dataset and evaluates it on another. It does not deliberately change how two sources predict the label in the training and validation data while keeping the evaluation recordings fixed.

The objective of this thesis is therefore to test whether speech deepfake detectors change their decisions when the source–label relationship is changed, and then to investigate which kinds of source-related information may contribute to that dependence. It addresses two research questions:

- **RQ1:** To what extent do speech deepfake detectors rely on shortcuts?
- **RQ2:** What is the nature of those shortcuts?

The questions are intentionally limited to the experiments in this thesis. RQ1 asks how strongly the tested detectors respond when the source–label relationship is changed. RQ2 examines selected properties that might explain that response. Because dataset source includes many linked properties, the experiment can establish dependence without identifying one property that is always responsible.

1.5. Approach at a glance

The study proceeds in three steps. First, visualisations and simple classifiers are used to ask whether dataset source can be recognised from properties of the recordings. This shows whether source information is available to learn, but not whether the speech deepfake detectors use it.

Second, the relationship between source and class is gradually changed from aligned to reversed in the training and validation data. A separate detector is trained under each version of that relationship, while evaluation always uses the same recordings. A systematic change across these conditions can therefore be attributed to differences in how the detectors were trained and selected, rather than to a changing test set. As a control, the procedure is repeated using two groups of speakers from the same dataset; this checks whether the mixing procedure creates such a pattern by itself. A second dataset pair and repeated runs test whether the finding recurs.

Third, selected properties of the recordings are altered to see whether any tested change removes the response. These tests examine possible explanations involving frequency range (bandwidth), signal level, and low-frequency energy. They can narrow the explanation, but they cannot examine every possible difference between the source datasets.

The comparison uses RawNet2, LCNN, and Wav2Vec-AASIST, which process audio in different ways. A shared response is therefore less likely to be peculiar to one model design. Chapters 2 and 5 explain how the models operate and how they were implemented.

Together, these steps ask three distinct questions: is source information available, do the detectors use information associated with the source–label relationship, and can any tested property explain that dependence?

1.6. Thesis outline

Chapter 2 explains how speech recordings are created, stored, and represented by the three detector families. Chapter 3 reviews shortcut learning, cross-dataset performance, unintended speech cues, and methods for testing detector reliance. Chapter 4 presents the research design and the inferences supported by each evidence block. Chapter 5 gives the datasets, preprocessing, models, and experimental details. Chapter 6 presents and discusses the results in the same evidence order. Finally, Chapter 7 answers the research questions and states the contribution, limitations, and conclusion.

2

Technical background

Chapter 1 defined speech deepfake detection as deciding whether speech in a recording is genuine or has been generated or manipulated by software. This chapter supplies the technical background needed to follow how those recordings are created, stored, prepared, and represented by a detector. It does not describe the study’s datasets or argue for the experimental design; those study-specific choices are introduced later.

2.1. From speech to an audio file

2.1.1. Waveforms, sampling, and frequency

Sound consists of changes in air pressure over time. A microphone converts those changes into an electrical signal. To store the signal digitally, its value is measured at regular moments. The resulting sequence of numbers is a *waveform*, and each number is a *sample* representing the signal amplitude at one moment.

The *sample rate* states how many samples are stored per second. For example, audio sampled at 16 kHz contains 16,000 samples for each second of recording. The sample rate also limits the frequencies that can be preserved. Under the Nyquist sampling principle, a 16 kHz recording cannot preserve frequencies above 8 kHz [16]. Converting it to a higher sample rate later creates more samples but cannot restore frequency content that was never recorded.

A waveform shows how amplitude changes over time. A *spectrum* instead shows how signal energy is distributed across frequencies. Because speech changes over time, spectra are often calculated over short, overlapping windows. Placing these short spectra next to one another produces a *spectrogram*: a two-dimensional representation in which one axis denotes time, the other frequency, and colour or intensity denotes energy.

The *bandwidth* of a recording is the frequency range in which it contains meaningful signal content. Sample rate places an upper limit on bandwidth, but the two terms are not identical. A high-sample-rate file may still contain little high-frequency energy because of the microphone, transmission channel, filtering, or earlier processing.

2.1.2. Containers, codecs, and numerical representation

A stored audio file contains encoded sound and the information needed to interpret it. Four terms distinguish parts of this representation [17]:

Container. The container specifies how information is arranged in the file. WAV is one example.

Codec. The codec specifies how the audio is encoded and decoded. PCM stores waveform samples directly. FLAC compresses them without changing the waveform reconstructed by the decoder and is therefore lossless. MP3, AAC, and Opus reduce file size by discarding some information and are therefore lossy.

Sample format and bit depth. These specify how individual waveform values are stored. Common choices include 16-bit integers and floating-point values. More bits allow amplitude to be described

using finer numerical steps.

Bitrate. Bitrate is the number of encoded bits used for one second of audio. For uncompressed PCM it follows from sample rate, bit depth, and channel count. For compressed audio it also depends on the codec and its settings.

The word *format* is sometimes used for several of these properties at once. They should be distinguished: WAV names a container, while PCM describes one way of encoding the waveform stored inside it.

2.1.3. Stored files and decoded waveforms

Before a neural detector can process an audio file, software reads the container and codec and reconstructs a sequence of waveform samples [17]. This operation is called *decoding*, and its output is the *decoded waveform*. A waveform-based detector normally receives this sequence rather than the filename, container header, or original compressed bytes.

Decoding does not erase every effect of recording and storage. Compression, filtering, resampling, and other operations can change the waveform itself, so traces of those operations may remain after the header and codec name are removed. Audio forensics studies how recording and processing histories can leave such traces [18, 19]. The existence of a trace and its use by a detector are separate questions.

Several signal properties recur later in this thesis. *Signal level* describes the overall strength of a waveform. One common measure is root-mean-square (RMS) amplitude. *Silence* denotes regions with little or no speech energy. *Noise* may come from the room, microphone, transmission channel, or later processing. *Duration* describes the length of a file or speech region. *Frequency content* describes how energy is distributed across lower and higher frequencies.

An *artefact* is a detectable pattern introduced by generation, recording, processing, or storage rather than by the intended linguistic content. An artefact is not automatically evidence of a deepfake because genuine recordings also pass through microphones, codecs, and editing systems.

2.2. How genuine and deepfake speech are produced

2.2.1. Genuine recordings

A genuine recording begins with a person speaking. The captured signal is influenced by the speaker, room, background noise, microphone, and recording device. It may then be filtered, resampled, compressed, transmitted, or edited before it is stored. A genuine recording therefore contains information about both the speech and its recording and processing history.

2.2.2. Text-to-speech and voice conversion

Text-to-speech (TTS) generates an utterance from written text. A common neural TTS system first predicts an intermediate description of the speech, including information such as pronunciation, timing, pitch, and energy. A waveform-generation component then turns that description into audible speech [20].

Voice conversion (VC) instead begins with an existing speech recording. It modifies speaker-related characteristics while attempting to preserve what was said [21]. For example, a system may separate a representation of the words from a representation of the speaker, replace the latter, and reconstruct the utterance. TTS and VC therefore start from different inputs, but both ultimately produce a waveform.

2.2.3. Vocoder and neural-codec waveform generation

A *vocoder* is a component that converts an intermediate speech representation into a waveform. The intermediate representation contains far fewer values per second than a waveform, so the vocoder must turn each step into many waveform samples. This process is called *upsampling*.

Upsampling is commonly performed in stages with filtering between them. An imperfect combination can create repeated or tonal patterns in the frequency spectrum. These *spectral replicas* arise from waveform generation rather than from the intended speech [9]. They are possible traces of a particular synthesis design, not universal markers of every deepfake.

A *neural audio codec* is a learned system for compressing and reconstructing audio. Its encoder turns a waveform into a compact representation, often a sequence of discrete values called tokens, and its decoder reconstructs a waveform from those tokens [22]. A generative model can create speech

by predicting audio tokens and passing them to the codec decoder [23]. This is called neural audio codec-based synthesis [24].

TTS and VC describe what a system is asked to do, whereas a vocoder or neural codec describes one mechanism that may be used to create the waveform. Whatever the route, the detector ultimately receives the final processed audio rather than a record of every generation step.

2.3. From an audio file to a detector input

2.3.1. Decoding and preprocessing

A stored file is rarely passed unchanged to a neural network. A typical input pipeline:

1. decodes the file into waveform samples;
2. converts multi-channel audio to one channel when required;
3. resamples the waveform to the rate expected by the detector;
4. identifies regions likely to contain speech;
5. divides the selected speech into model-input segments; and
6. prepares each segment in the form required by the model.

Resampling changes the number of waveform samples per second. Cropping removes part of a long input. Each preparation step can change the audio and information that reach the detector.

Voice activity detection (VAD) estimates where speech is present. It does not decide whether that speech is genuine or deepfake. Depending on the recording and VAD settings, one file may yield no usable speech, one speech region, or several regions.

It is therefore important to distinguish a *source file* from a *model-input segment*. The source file is the stored recording supplied to preprocessing. A segment is one portion of its decoded waveform that is passed to a detector. Several segments derived from one file share its speaker, recording conditions, and processing history. They are several model inputs, but not independent source recordings.

2.3.2. Representation, front end, and classifier

A detector can be described in terms of two main parts. Its *front end* converts the waveform into a representation that later components can process. Its *classifier*, sometimes called the back end or classification head, combines information in that representation and produces a genuine/deepfake score.

During training, the model sees input segments together with their labels, and its parameters are adjusted to separate genuine from deepfake examples. During evaluation, the labels are not provided to the model, and the trained model produces scores for new segments. The information available to the classifier depends partly on the representation produced by the front end.

2.4. Three detector paths

The three detector families used in this thesis begin with the same kind of decoded waveform but represent it differently. Figure 2.1 provides the organising comparison.

2.4.1. Raw waveform: RawNet2

A raw-waveform detector receives decoded audio samples directly. Here, *raw* means that no fixed spectrogram or other hand-designed audio representation is calculated first. It does not mean that the model receives the original encoded file, header, or filename.

RawNet2 is a representative model in this family [25]. The implementation used in this thesis begins with a learnable Sinc-based filterbank. This is a set of band-pass filters that each retain one frequency range while suppressing others; the model learns the lower and upper boundary of each range. Later layers combine the resulting patterns over time and produce a score. Its initial representation is therefore calculated from decoded samples rather than supplied as a designed time–frequency map.

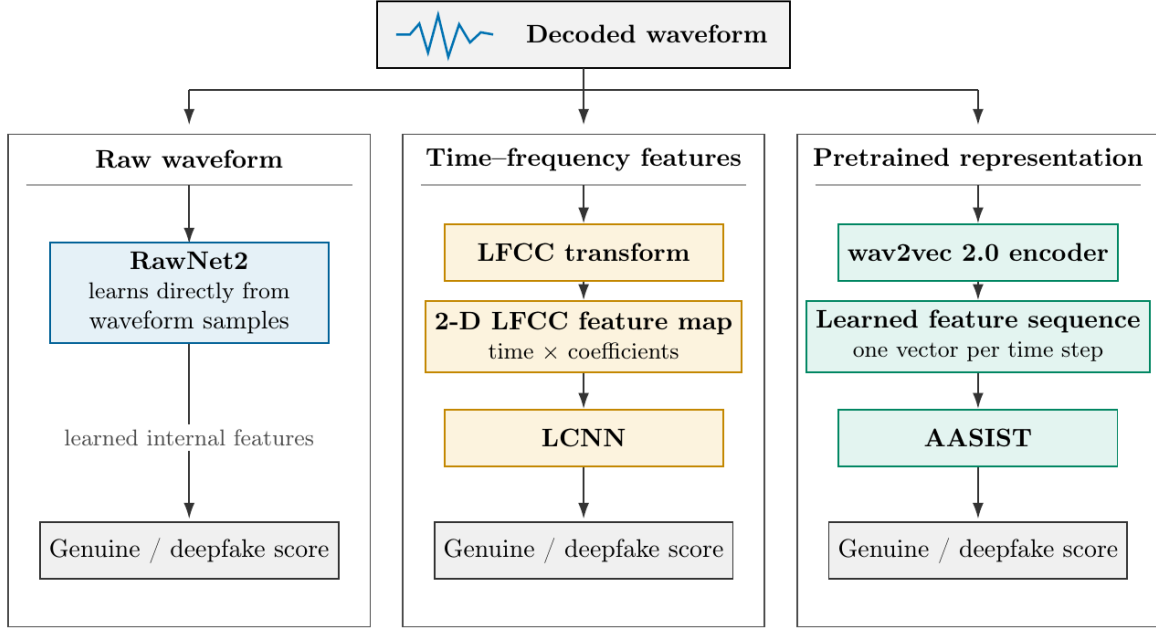


Figure 2.1: Three paths from a decoded waveform to a detector score. RawNet2 develops features from waveform samples, LCNN processes a designed time-frequency representation, and Wav2Vec-AASIST processes representations produced by a pretrained speech encoder.

2.4.2. Time-frequency representation: LCNN

A spectral detector first divides the waveform into short windows and summarises the frequency content of every window. Placing these summaries in temporal order produces a two-dimensional time-frequency feature map.

The LCNN used in this thesis receives *linear-frequency cepstral coefficients* (LFCCs) [26]. LFCC extraction measures signal energy using linearly spaced frequency bands and converts these measurements into a compact grid of numerical features. The grid resembles an image, but its axes and values still describe audio. The LCNN uses convolutional layers, which examine small neighbouring regions of the grid, to learn recurring patterns and combine them into a detection score.

2.4.3. Pretrained speech representation: Wav2Vec-AASIST

A third path begins with a speech encoder that has already learned a representation from a separate collection of unlabelled audio. This is called *self-supervised learning*: the learning target is created from the audio itself rather than from manually assigned genuine/deepfake labels. For example, wav2vec 2.0 learns by hiding parts of an encoded speech sequence and using the surrounding audio to identify the correct representation for each hidden part [27].

After pretraining, the encoder converts a waveform into a sequence of feature vectors. Each vector is a learned list of numbers describing part of the speech signal. A classifier can then use those vectors for deepfake detection. If the encoder is *frozen*, its parameters remain unchanged while the classifier is trained. If it is *fine-tuned*, some or all encoder parameters are also updated for the new task.

Wav2Vec-AASIST combines a wav2vec 2.0 encoder with the AASIST classification architecture [28, 29]. The encoder produces learned speech representations, and AASIST analyses relationships within their sequence and combines them into a detection score.

These paths are alternatives, not successive stages or a ranking. They expose different representations of the same decoded signal to the classifier. A pattern may therefore be easier for one detector family to use than another. Access to that pattern, however, does not prove that a trained detector relies on it.

3

Related work

A speech deepfake detector should distinguish genuine speech from speech that was generated or manipulated. However, a detector can obtain a good benchmark score by learning patterns created when that benchmark was collected or prepared. Such patterns may not help on recordings from another source. This chapter reviews evidence for this problem, the unintended information found in speech datasets, and the methods used to test whether detectors rely on it. It ends by identifying the question addressed by this thesis.

3.1. Shortcut learning and benchmark interpretation

Shortcut learning occurs when a model uses an unintended pattern that predicts the correct label in a benchmark [5]. In speech deepfake detection, a shortcut could be a recording or processing pattern that happens to separate genuine and deepfake recordings in one dataset. If the same pattern occurs in that dataset’s training, validation, and test splits, a detector can achieve a low EER without learning a general rule for detecting generated speech.

Training does not force every model to use the same rule. *Underspecification* means that several models can obtain similar test scores while basing their predictions on different patterns; their behaviour may then differ when the conditions change [30]. A benchmark score therefore shows that a detector separates the two classes under the tested conditions. It does not reveal whether the detector learned evidence of generated speech, a dataset-specific pattern, or both.

The “Name That Dataset” study by Torralba and Efros [31] provides a useful precedent. Classifiers could identify which image dataset an example came from, showing that datasets contained recognisable collection-specific properties beyond their intended object labels. The implication for speech is not that image and audio shortcuts are identical. It is that dataset origin can itself be predictable. When origin also correlates with the task label, it offers an alternative route to a correct benchmark prediction.

These general studies explain how a strong benchmark score can coexist with an unintended rule. They do not show that a particular speech detector uses a dataset-source shortcut. Evidence from speech deepfake detection is therefore needed.

3.2. Generalisation in speech deepfake detection

Cross-dataset evaluation trains a detector with one dataset and tests it with another. Müller et al. [13] evaluated twelve detection architectures on independently collected genuine and deepfake recordings. Systems that had performed strongly on established benchmarks produced substantially higher EERs on the new data. Kawa et al. [11] similarly kept the generation methods used for evaluation out of the development data and found that performance varied considerably across the unseen methods. These studies show that success on a familiar benchmark does not reliably transfer to every new dataset or generation method.

Higher cross-dataset EER can have two broad causes. First, the second dataset may be harder for detectors even when they are trained on it. Second, training and test data may differ in ways that make a rule learned from the first dataset less useful on the second. Lu et al. [15] introduced a method for

separating these two parts of a performance gap, which Müller et al. [14] adapted to speech deepfake detection. In their comparisons, the difference between the training and test datasets explained most of the gaps, although one evaluation set also had a remaining difficulty contribution. This shows that the choice of development data matters. It does not identify which dataset difference caused the additional errors.

Some studies test a more specific change. Zhang et al. [32] applied channel transformations to otherwise matched data and reproduced a performance decrease. This result points to information affected by microphones, codecs, or transmission conditions. Wu et al. [24] found that detectors trained on conventional deepfake datasets transferred poorly to speech produced with neural audio codecs, while adding codec-generated speech to the training data improved performance on that synthesis family. These experiments show that the generation and processing conditions represented during training can affect transfer. They do not show that these factors explain every cross-dataset gap.

3.2.1. Representation choice and partial SSL robustness

Representation choice can change the severity of transfer failure. Pascu et al. [33] compared pretrained speech representations across eight audio deepfake datasets using a deliberately simple logistic-regression classifier. Their strongest frozen self-supervised representation transferred substantially better than RawNet2 in that benchmark, suggesting that the representation itself contained more portable information. Wang and Yamagishi [34] likewise found better transfer for self-supervised front ends than for several hand-designed acoustic features, and showed that the compared systems depended on different frequency regions.

These findings motivate including a self-supervised detector alongside waveform- and spectral-input detectors. They do not show that self-supervised pretraining removes shortcut learning. A complete architecture may avoid one cue while using another, and differences between systems cannot be attributed to the front end alone when their classifiers and training procedures also differ.

Overall, cross-dataset studies reveal a real transfer problem and show that differences in datasets, generators, channels, and model inputs can matter. However, many properties usually change at the same time when the dataset changes. A higher EER therefore shows that performance changed, but not which rule the detector learned.

3.3. Documented sources of unintended information

Previous work has identified several ways in which recordings can reveal how a dataset was constructed. The studies below are grouped by where this information enters the recording. Throughout the section, two claims must remain separate. A measurement may contain enough information to predict a label or dataset source, but a trained deepfake detector may not use that information.

3.3.1. Dataset-construction cues

Chettri et al. [7] audited ASVspoof 2017 and found class-related differences in silence, dual-tone telephone sounds, corrupted files, and noise. They also showed that speech-attack detection models could benefit from these properties. Although the benchmark concerned replayed rather than generated speech, the study demonstrates the relevant mechanism: if the two classes pass through different construction procedures, those procedures can create an easier predictive rule.

Müller et al. [6] examined one construction cue more directly. Genuine and deepfake recordings in ASVspoof 2019 and 2021 contained systematically different leading and trailing silence. Silence duration alone predicted the class, establishing that the cue was available. RawNet2 performed worse after silence trimming, providing evidence that the detector had used information changed by the intervention.

Smeu et al. [8] reported a related leading-silence pattern in FakeAVCeleb and AV-Deepfake1M. A simple classifier separated the classes using that property, and supervised detectors performed worse after it was removed. Together, these studies show that class-dependent silence can contribute to detector performance. They do not show that silence is the only cue present, because trimming can change several boundary and duration properties while leaving other construction differences intact.

Table 3.1: What different shortcut-evaluation designs can and cannot establish by themselves.

Method	What it shows	What it does not show on its own
Visualisation or cue classifier	Whether a selected view contains predictive information	Whether the deepfake detector uses that information
Association or explanation applied after training	Which measured properties or examples align with model outputs	Whether changing the highlighted property would change the output
Cue removal or balancing	Whether behaviour changes after the operation	Whether only the named property changed or whether it alone explains the behaviour
Cross-dataset evaluation	Whether performance transfers to another dataset	Which changed property caused success or failure
Changing the source-label relationship	Whether learned behaviour changes when this relationship is deliberately varied	Which single source property carries the effect when several properties change together

3.3.2. Generator and vocoder fingerprints

The system that produces a deepfake waveform can leave recognisable patterns. Pons et al. [9] showed that upsampling operations used in neural audio synthesis can introduce tonal artefacts and spectral replicas. Yan et al. [10] then held the acoustic model fixed and showed that outputs from eight vocoders could be distinguished from their signal features. Wu et al. [24] extended the concern to neural audio-codec synthesis by showing poor transfer from conventional deepfake training data to that generation family.

A generator fingerprint is not automatically an unwanted cue: a pattern caused by synthesis can be valid forensic evidence. The limitation is breadth. A detector that equates the fingerprints of generators represented in its development data with “deepfake” may fail when a different generator leaves another pattern. Generator attribution and general deepfake detection are therefore related but not identical tasks.

3.3.3. Recording, channel, encoding, and storage traces

Genuine and deepfake recordings also pass through recording, transmission, and storage processes. Audio-forensics research has long examined the traces left by these processes [18]. Hisşönmez et al. [35] showed that codec and transcoding history could be identified from encoded audio files. Such a file contains stored bytes and header information. Decoding turns it into a waveform and discards the header and original compressed bytes. The result therefore does not show that a detector given only the decoded waveform can use the same evidence.

Some processing history remains measurable in the waveform after decoding. Hennequin et al. [19] detected previous lossy compression from patterns in the decoded audio, including frequency cuts and holes. For speech-attack detection, Zhang et al. [32] showed that controlled channel changes could degrade detector performance. These findings explain why “format” should not be treated as one variable. A file header is different from a codec trace that remains in the decoded waveform.

Across all three groups, the same distinction applies. Predicting dataset source from selected measurements shows that source information is available in those measurements. Showing that a deepfake detector relies on source information requires an additional test of the detector itself.

3.4. How shortcut reliance has been tested

Prior studies use several designs to move from suspected cues to evidence about model behaviour. Table 3.1 orders these designs by the question they answer rather than treating them as interchangeable.

3.4.1. Availability and association

A visualisation or cue classifier examines selected measurements, such as silence duration or file properties. Successful source or class prediction shows that these measurements contain a possible shortcut. It does not show that the main detector receives or uses the same information.

Associations with detector outputs and explanation methods applied after training move closer to the trained model. Such methods can generate useful hypotheses about the information associated with a model's predictions. Their conclusions depend on the explanation method and analysed examples, so a highlighted pattern is not equivalent to showing that changing the pattern would change the prediction.

3.4.2. Cue removal and artefact balancing

Removal experiments provide behavioural evidence by changing a suspected cue and measuring the detector again. The silence studies follow this logic: performance changed after trimming, so the trained systems depended on information affected by that operation [6, 8]. Shih et al. [36] used a balancing strategy instead. Their BEAR protocol created evaluation conditions in which comparable artificial patterns or noise occurred in both classes, and detector performance decreased when the original asymmetry was removed.

These interventions implicate the information they change, but their conclusions remain operation-specific. Trimming or balancing may alter more than the named cue, and changing one cue does not remove other source-related differences. This matters when dataset origin includes generators, speakers, content, recording conditions, and processing choices at the same time.

3.4.3. Cross-dataset evaluation and changing the source-label relationship

Cross-dataset evaluation is a broad stress test: it asks whether the learned rule survives when the dataset changes. Its weakness for attribution is that the changed properties are not controlled individually. A performance decrease can identify neither the information used during development nor the property responsible for failure.

Another type of experiment deliberately varies how strongly a suspected shortcut predicts the label. Colored MNIST is a simple precedent: Arjovsky et al. [37] made colour predict the digit label during development and changed that relationship across conditions. A classifier that relied on colour then failed when the relationship was reversed. The relevant precedent for this thesis is this test of reliance, not the proposed Invariant Risk Minimisation training algorithm.

Natural speech sources make the interpretation less specific. Colour can be added as a relatively isolated image property, whereas changing the source of speech recordings changes several naturally co-occurring properties. A detector response can therefore establish dependence on the manipulated source-label relationship without identifying codec, silence, generator family, or another component as the single responsible property.

3.5. Research gap and positioning of this thesis

Previous research shows that speech deepfake detectors often transfer poorly, that individual construction and processing cues can predict class or dataset source, and that some detectors use information changed by cue-removal or balancing interventions. These findings establish a practical transfer problem and several possible explanations.

One question remains unresolved: what happens when the relationship between dataset source and the genuine/deepfake label is deliberately changed during training while the same recordings are used for evaluation? Cross-dataset evaluation does not manipulate this relationship. A source classifier shows that source information is available, not that the detector uses it. Removing one suspected cue tests only the information changed by that operation, not all differences associated with dataset source.

This thesis therefore combines established diagnostic tools with a controlled source-label experiment developed for this study. Visualisations, held-out probes, association measures, cross-dataset comparisons, and targeted audio transformations are adopted or adapted from existing practice. The source-label mixing protocol changes how two datasets supply genuine and deepfake recordings during training, then compares the trained detectors on unchanged evaluation recordings. The protocol follows the test logic of designs such as Colored MNIST, but changes natural dataset source rather than adding one synthetic cue. Consequently, several source properties change together.

This evidence chain keeps three questions separate: whether source information is available, whether detectors use the source-label relationship (RQ1), and which source properties may contribute to that reliance (RQ2). Chapter 4 explains how each question is tested and what the corresponding evidence can support.

4

Research design and methodology

Previous work shows that recordings can contain information about their dataset source and that detector performance can fall when training and evaluation data come from different datasets. It does not directly test whether detectors use a relationship between dataset source and the genuine/deepfake label during development. This chapter explains how the study tests that question and what can be concluded from each comparison. Chapter 5 provides the datasets, settings, counts, and other implementation details.

4.1. Research logic and method ownership

The study follows three questions. First, can dataset source be identified from selected measurements of the recordings? This tests whether source information is available. Second, do detectors use the relationship between dataset source and the genuine/deepfake label? To test this, the study deliberately changes that relationship during development and evaluates every resulting detector on the same recordings. This is the primary test for RQ1. Third, which possible source properties might contribute? The study repeats the main experiment after changing selected audio properties. This addresses RQ2 without assuming that one property is the sole cause.

Table 4.1 gives the complete roadmap. The order matters because each question needs different evidence. Showing that source information exists does not show that a detector uses it. Showing that a detector depends on source-related information does not identify the responsible property. Finally, changing one audio property cannot make every other difference between the datasets disappear.

The individual tools in this chain are not all new. The study adopts t-SNE, small classifier probes, feature-association measures, cross-dataset evaluation, and the hardness–difference comparison from earlier work [38, 39, 15, 14]. Here, a probe is a small classifier trained to predict dataset source or class from selected measurements. The study-specific contribution is the *source–label mixing protocol*: two source datasets are used during development, their relationship with the two classes is deliberately varied, and the resulting detectors are compared on unchanged evaluation recordings.

4.2. Data flow and units of analysis

The source composition is determined at the source-file level, before audio preprocessing. The relevant flow is

$$\text{source file} \longrightarrow \text{VAD segment} \longrightarrow \text{model input} \longrightarrow \text{prediction}.$$

Voice activity detection (VAD) can find no usable speech segment, one segment, or several segments in a source file. Each retained segment becomes a separate model input and later produces a separate prediction.

Segments produced from one source file share speaker, content, and recording history. They are therefore kept together in probe folds and treated as one cluster in resampling-based uncertainty analyses, rather than counted as independent source recordings.

Table 4.1: Evidence chain used to answer the research questions.

Analysis	Purpose	Question
Source-information availability	RQ2 context	Can selected measurements identify dataset source?
Same-corpus speaker baseline	RQ1 baseline	Does the ratio procedure create the same pattern when the two sources are speaker groups from one corpus?
Controlled source dependence	RQ1 primary	Does reversing the source-label relationship during development change results on fixed recordings?
Second-pair replication	RQ1 scope	Does the response recur with another dataset pair and several training seeds?
Possible sources of the shortcut	RQ2	Is the dependence reduced by selected audio changes?
Cross-dataset comparison	Context	Is poorer transfer related to a harder target dataset or to a mismatch between training and test data?

4.3. Evidence block 1: availability of source information

The first block asks whether selected measurements contain information about dataset source. It uses three complementary analyses. First, t-SNE provides an exploratory two-dimensional view of whether recordings group more clearly by source than by genuine/deepfake class [38]. Second, small classifier probes test whether the source of held-out files can be predicted from format-and-level measurements, spectral measurements, or both. Third, feature associations identify measurements that are more closely related to source than to class and therefore suggest properties for later experiments [39].

These analyses test whether source information is available and help select possible explanations. They do not show that the neural detectors use this information. Chapter 5 gives the sampling, feature, and validation settings.

4.4. Evidence block 2: controlled test of source dependence

RQ1 requires evidence about trained detectors, not only about the data. The central comparison therefore changes the source-label relationship during model development. In one condition, source A mainly supplies genuine recordings and source B mainly supplies deepfakes. In the reversed condition, source B supplies genuine recordings and source A supplies deepfakes. Separate detectors are developed for these and intermediate compositions. Every detector is then tested on exactly the same recordings.

Keeping the test recordings fixed is crucial. If results change across training ratios, the change cannot be caused by using an easier or harder test set at one ratio. It instead shows that changing the source-label relationship during development changed what the resulting detector learned. This fixed-test comparison is the primary answer to RQ1.

4.4.1. Formal construction

Let D_A and D_B be two source datasets containing both genuine and deepfake recordings. The mixing ratio r determines which source supplies each class during development, as shown in Figure 4.1.

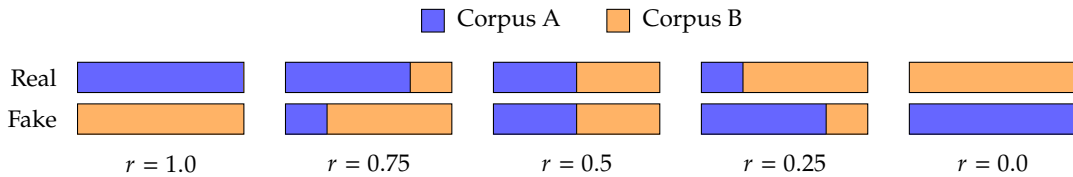


Figure 4.1: File-selection composition under the source-label mixing protocol. At $r = 1$, source A supplies the genuine class and source B the deepfake class. At $r = 0$, the assignment is reversed. At $r = 0.5$, both classes receive the same 50:50 source distribution among selected entries.

At $r = 1$, source D_A supplies the genuine class and source D_B supplies the deepfake class. At $r = 0$, this assignment is reversed. At $r = 0.5$, both classes receive the same 50:50 source distribution among

the selected files. The midpoint is therefore a principled *file-selection balance point* rather than an arbitrary value. Intermediate ratios weaken the source–label relationship in steps.

The same number of file entries is selected for each class at every ratio, so training-set size cannot explain the response across training ratios. The same ratio is used for training and validation, and validation EER controls early stopping and checkpoint selection. The intervention therefore changes the complete model-development process.

4.4.2. Fixed evaluation configurations

Every trained checkpoint is evaluated on four configurations assembled from held-out source–label pools:

1. genuine from D_A and deepfake from D_B
2. genuine from D_B and deepfake from D_A
3. genuine and deepfake both from D_A
4. genuine and deepfake both from D_B

Each configuration is fixed across all ratios. The first two are mirrored cross-source alignments; the other two evaluate both classes within one source. They are four fixed, overlapping configurations rather than four independent datasets: a source–label pool contributes to more than one configuration, and one source file can contribute several segments.

The first two conditions show what happens when the source relationship is reversed. For example, suppose a detector has learned that recordings from D_A are usually genuine. That rule helps when the test set also uses genuine recordings from D_A , but hurts when D_A supplies the deepfakes instead. Under the score convention used here, EER above 0.5 means that genuine and deepfake scores are more often ordered in the wrong direction than the correct one. This is consistent with strong use of reversed source-related information, but does not mean that the detector uses source alone.

4.4.3. Primary fixed-test comparison

For a fixed test configuration c , let $\text{EER}(r, c)$ be the EER of the checkpoint developed at ratio r . The primary endpoint magnitude is

$$\Delta_c = |\text{EER}(0, c) - \text{EER}(1, c)|. \quad (4.1)$$

Because the test recordings in c are unchanged, Equation 4.1 measures how much EER changes when development switches from one source–label assignment to the reverse assignment. The signed difference and mirrored test conditions describe the direction of that change. The intervention supports a causal claim about dependence on the changed development regime under the tested conditions. It does not isolate one technical property. Speakers, language, content, generation systems, recording conditions, and technical processing remain bundled in the source comparison.

The logic is related to controlled-correlation diagnostics such as Colored MNIST [37]. The important difference is that the manipulated attribute here is a natural, compound dataset source rather than one synthetically added cue. The estimand also does not state how common source-related shortcut reliance is across all benchmarks or deployed detectors.

4.4.4. Intermediate ratios

Intermediate ratios show how detector behaviour changes between the two endpoints. A relatively flat response across training ratios is consistent with limited sensitivity to the manipulated relationship, but cannot rule out shortcuts that remain unchanged across r . A systematic change is consistent with source-related dependence, and reversed score ordering near the opposite endpoint is especially informative.

The curve shape is descriptive. A gradual change may reflect sensitivity to progressively changing source proportions, while an abrupt change may reflect a learning or model-selection threshold. Optimisation and seed variation can also differ across ratios. The magnitude of the change between the two endpoints on fixed evaluation recordings therefore remains the primary causal comparison for RQ1.

4.4.5. Control, replication, and architectural scope

The same-corpus comparison provides the baseline for interpreting the cross-dataset experiments. It applies the same mixing procedure to two speaker-disjoint groups drawn from one corpus. Because both groups share the corpus-level collection and processing pipeline, a comparatively flat response would show that arbitrary grouping and mixing do not automatically create the cross-dataset pattern. Speaker-group differences can still remain, and one partition and seed cannot establish how every possible same-corpus split would behave.

The experiment is then repeated with a second dataset pair and, where available, several training seeds. This checks whether the response is limited to the deliberately contrastive primary pair or to one training run. The two comparisons share one corpus, so recurrence strengthens the evidence but does not make the tested pairs representative of all speech datasets.

Finally, the protocol is applied to three architectures that receive different audio representations. Agreement across architectures broadens the architectural scope of the result. Quantitative differences cannot be assigned to one component because the complete systems differ. Uncertainty observed in a multi-seed comparison is not transferred to single-seed cells elsewhere.

4.5. Evidence block 3: testing possible sources of the shortcut

The controlled experiment tests whether detectors depend on source-related information, but it does not identify which source property carries that information. Predictions are grouped by source and class to show where errors occur. This describes the pattern of failure but does not identify what information the detectors used.

4.5.1. Retraining after selected audio changes

The primary datasets differ in their original frequency range, signal level, and low-frequency energy. Each property is tested by changing it consistently during training, validation, and evaluation and then developing a new detector at every ratio. This asks whether the response across training ratios remains after the targeted source difference has been reduced. Chapter 5 gives the exact processing settings.

If the endpoint change becomes much smaller, information affected by the processing change may have contributed to the original dependence. If the endpoint change remains, the implemented change was not sufficient to remove the source dependence. Neither result identifies one responsible property: each operation can affect several parts of the signal, and other differences between the datasets remain.

4.6. Supporting context and inference boundaries

As supporting context, the study also compares within-dataset and cross-dataset performance using the hardness–difference framework [15, 14]. It asks whether poorer cross-dataset performance occurs because the evaluation dataset is harder even for a detector developed on it, or because the development and evaluation datasets differ. This is an observational comparison. It does not identify what differs between the datasets or show that a detector used source information. The full definition and calculation are provided in Appendix A.2.

The main intervention also has clear limits. Dataset source changes several properties together, so the endpoint comparison cannot identify one responsible property.

Within these limits, the method separates three questions: whether source information is available, whether changing the source–label relationship changes detector behaviour, and whether selected audio changes reduce that dependence.

Experimental implementation

Chapter 4 explained why the evidence can answer the research questions. This chapter describes how the study was implemented: the datasets and models, the conversion from files to model inputs, the comparisons, and the evaluation measures. Standard within- and cross-dataset runs provide reference values.

The reported training inventory contains 201 detector runs. Most comparisons use one training seed, so their overall patterns are more informative than small differences between individual values. The raw and RMS-matched SpeechFake replication grids use three seeds. Analyses of existing predictions or audio measurements do not add detector-training runs to this total. Two auxiliary label-shuffle runs exist in the archive but are excluded because they do not contribute to a reported analysis.

5.1. Implementation roadmap

Table 5.1 maps the three evidence blocks to their datasets, models, comparisons, and primary outputs. The ASVspooF 5 speaker baseline, primary pair, and second-pair replication are three connected comparisons within one controlled study.

Table 5.1: Implementation roadmap. “Runs” counts detector-training runs; feature and prediction analyses reuse existing data or checkpoints.

Block	Datasets	Models	Comparison	Primary output	Runs
1: availability	Four-corpus feature audit	Logistic regression probes	Source and class prediction from three feature views	t-SNE views, grouped accuracy, Cramér’s V	—
Reference	ASVspooF 5, FakeAVCeleb	All three detectors	Within- and cross-dataset evaluation	EER and gap decomposition	6
2a: baseline	ASVspooF 5 speaker groups	All three detectors	Five development ratios on fixed test recordings	Same-corpus EER curves	15
2b: primary dependence	ASVspooF 5, FakeAVCeleb	All three detectors	Five development ratios on fixed test recordings	EER curves and endpoint changes	15
2c: replication	SpeechFake, ASVspooF 5	All three detectors	Five development ratios, second pair, three reported seeds	Mean EER curves and endpoint changes	60
3: characterisation	Primary and replication pairs	All three detectors	Prediction crosstabs and three processed ratio grids	Cell accuracy and response persistence	105
Total detector-training runs					201

5.2. Datasets and selection rationale

Three corpora form the development and fixed-test conditions used in the main study. Table 5.2 gives the same descriptive fields for each one. The primary pair was deliberately chosen to be strongly contrastive. This makes it suitable for testing whether detectors use a source–label relationship when one is available. The replication pair shares the same original 16 kHz sample rate and 16-bit depth, but changes the second corpus. VoiceWukong is used only as the fourth corpus in the source-availability audit; it is not used in detector training or a mixing-ratio experiment.

Table 5.2: Datasets used in the study and the reason for including each one. Stored-audio properties describe the archived source files before the common preprocessing pipeline.

Dataset	Genuine and deepfake material	Stored audio	Role in the study
ASVspoof 5 [4]	Crowdsourced genuine speech and deepfakes spanning multiple generation and attack systems	FLAC, 16 kHz, 16-bit, mono	Anchor corpus in both source pairs; also divided internally for the same-corpus control
FakeAVCeleb [40]	YouTube-sourced celebrity speech and cloned-voice deepfakes	WAV, 22.05 kHz, 32-bit floating-point PCM	Technically and procedurally contrastive partner in the primary pair
SpeechFake BD [41]	English and Chinese genuine speech and deepfakes produced by many generation systems	WAV, 16 kHz, 16-bit integer PCM	Second partner for replication; shares the 16 kHz sample rate with ASVspoof 5

Table 5.3: Available source-file selector entries. Counts are shown as genuine/deepfake. SpeechFake values describe the full bilingual selectors; the reported no-LibriTTS training selector contains 55,962/45,000 entries.

Dataset	Training	Validation	Test
ASVspoof 5	18,797 / 163,560	31,334 / 109,616	138,688 / 542,086
FakeAVCeleb	25,595 / 23,027	2,803 / 2,090	7,313 / 6,579
SpeechFake BD	75,708 / 45,000	12,618 / 7,500	37,248 / 309,065

Table 5.3 reports the available source-file entries from which the training, validation, and test selections are drawn.

Primary pair. ASVspoof 5 and FakeAVCeleb differ in several conspicuous ways. Their archived files use different original sample rates (16 and 22.05 kHz), containers and file types (FLAC and WAV), and sample representations (16-bit integer and 32-bit floating-point PCM). ASVspoof 5 is based on crowdsourced speech, whereas FakeAVCeleb uses YouTube-sourced celebrity speech. Their recording and generation pipelines also differ. Both corpora nevertheless contain genuine and deepfake speech. The experiment can therefore change which corpus supplies each class during development. After decoding, the detectors receive waveform samples or features derived from them; they do not receive the filename extension or container header directly. The intervention changes the broader set of differences associated with corpus source, not one format property in isolation.

Replication pair. SpeechFake’s bilingual subset contains genuine and deepfake speech in both English and Chinese. The replication removes SpeechFake entries derived from LibriTTS. This reduces potential source-material or speaker overlap with the LibriVox-derived material underlying ASVspoof 5, although the anonymised ASVspoof 5 identifiers do not permit exact speaker matching. A with-LibriTTS variant was also trained as a sensitivity check.

5.3. Preprocessing, data units, and fixed test conditions

5.3.1. From selector entry to model input

All audio is loaded as mono, resampled to 16 kHz, and processed using previously computed VAD intervals. Every detected interval becomes a separate dataset item. An interval longer than four seconds is randomly cropped during training and deterministically cropped from its beginning during evaluation.

The analysis therefore uses several nested units:

Selector pool. The full list of paths available for one corpus, split, and class.

Table 5.4: Fixed test configurations for the ASVspoof 5–FakeAVCeleb pair. Configuration names are retained because they also identify the prediction files used in the Results chapter.

Configuration	Genuine recordings	Deepfake recordings
cross_ASV5_real	ASVspoof 5	FakeAVCeleb
cross_FAV_real	FakeAVCeleb	ASVspoof 5
ASV5_standard	ASVspoof 5	ASVspoof 5
FAV_standard	FakeAVCeleb	FakeAVCeleb

Selected entry. One path sampled from that list for a particular ratio. Sampling is without replacement over entries.

Source file. One unique stored recording. This is the appropriate cluster when several derived segments must be kept together.

VAD segment. One speech interval found inside a selected file. Each interval receives the file’s genuine/deepfake label and becomes an independently weighted optimisation item.

Model input. The waveform produced from one VAD segment and supplied to the detector.

Test observation. One scored VAD segment in a prediction file. The primary EER is calculated from these segment scores.

One source file can yield many observations; the audited test pools contain as many as 37 segments from one file. Segments from the same file share speaker, recording, and processing characteristics. The source-file-clustered bootstrap used for Figures 6.2 and 6.3 therefore carries them together rather than treating them as independent recordings.

5.3.2. File-level selection balance

The primary ASVspoof 5–FakeAVCeleb grid uses 18,797 training and 2,087 validation file entries per class. The SpeechFake–ASVspoof 5 grids use 18,797 training entries per class and the 5,000-entry validation cap. The same-corpus control uses 18,616 training entries and the 5,000-entry validation cap. At $r = 0.5$, both classes receive the same proportions of entries from the two sources.

The same ratio is applied to training and validation; validation EER controls early stopping and checkpoint selection. The endpoint comparison therefore estimates the total effect of changing the selected-file source composition within the complete model-development and checkpoint-selection procedure. It does not separate those components.

5.3.3. Fixed test configurations

Every checkpoint in a source-pair grid is evaluated on four fixed configurations assembled from held-out source–label pools. Table 5.4 illustrates the primary pair. The two cross-corpus configurations reverse which corpus supplies the genuine class; the two within-corpus configurations contain both classes from one corpus. None of the test pools changes when the development ratio changes.

The primary configurations contain between 15,968 and 43,563 segment rows. The SpeechFake replication uses analogous configurations containing between 22,189 and 43,563 rows. These are *overlapping* configurations rather than independent test sets: each source–label pool appears in one cross-corpus and one within-corpus configuration.

5.4. Detector configurations and training

The study compares three detector families introduced in Section 2.4. Their different input representations make the comparison informative: RawNet2 begins with waveform samples, LCNN begins with a hand-designed LFCC representation, and Wav2Vec-AASIST begins with features produced by a pretrained speech encoder. Table 5.5 reports the exact as-run parameter counts and optimisation settings.

RawNet2. The RawNet2 implementation [25] receives 64,000 waveform samples. It uses an initial Sinc-based filter layer with 128 filters, six residual blocks, a three-layer gated recurrent unit with hidden size 1,024, and a two-layer classification head.

Table 5.5: Detector configurations. Parameter counts were verified by loading representative as-run checkpoints. Batch size varies by experiment and is reported separately below.

	RawNet2	LCNN	Wav2Vec-AASIST
Input	Waveform	LFCC map	wav2vec 2.0 features
Learning rate	10^{-4}	10^{-3}	10^{-5}
Weight decay	10^{-5}	10^{-4}	10^{-5}
Total parameters	25,433,602	542,498	94,786,186
Trainable parameters	25,433,602	542,498	414,474
Frozen parameters	0	0	94,371,712

LCNN. The LCNN implementation [26] computes 60 LFCCs using 128 linear filters, a 512-point Fourier transform, a 400-sample window, and a 160-sample hop. Nine convolutional layers with Max-Feature-Map activations process the resulting map before the classification head.

Wav2Vec-AASIST. The SSL-based detector combines the facebook/wav2vec2-base encoder [27] with a projection layer and AASIST back end [29, 28]. The as-run encoder is the base model pretrained on 960 hours of LibriSpeech. Its 94,371,712 encoder parameters remain frozen; only the 414,474 projection and AASIST parameters are updated.

All detectors use cross-entropy loss, AdamW, and cosine learning-rate annealing over at most 20 epochs, with the final learning rate set to 1% of the initial value. Early stopping has patience five, and the checkpoint with the lowest validation EER is retained. The primary and SpeechFake raw grids use batch size 128. The within-dataset baselines, bandwidth, primary-pair RMS, and high-pass grids use batch size 64; the same-corpus control uses batch size 32. The SpeechFake RMS grids use batch size 128. Comparisons between grids with different batch sizes are descriptive. The three-seed raw-versus-RMS SpeechFake comparison is batch-matched.

Seed 42 is used unless stated otherwise. The primary, control, and most processed grids therefore provide one trained checkpoint for each model-ratio cell. The reported SpeechFake no-LibriTTS raw and RMS grids use seeds 41, 42, and 43. Their results are summarised using the mean and sample standard deviation across those three independently trained checkpoints.

5.5. Evaluation measures

Equal error rate. Equal Error Rate (EER) is the primary outcome. It is the threshold at which the false-positive and false-negative rates are equal [3]. Lower values indicate better separation. If the score ordering is reversed, EER can exceed 0.5. EER is calculated from the scored speech segments. The threshold is found from the evaluation labels; it is not a fixed deployment threshold or a measure of score calibration.

Test-pool uncertainty in the single-seed fixed-test figures. Figures 6.2 and 6.3 show pointwise 95% confidence intervals obtained by resampling the source files 5,000 times (seed 20260812). Every VAD segment from a sampled file stays with that file, so segments from one recording are not treated as independent. The same file draws are used across models and ratios. These intervals describe uncertainty from the fixed test recordings for each existing single-seed checkpoint. They do not describe variation from retraining or selecting different corpora. The bars mark intervals at the five tested ratios; the shading only connects those pointwise intervals visually.

Primary endpoint comparison. The primary controlled comparison is the absolute change in EER between $r = 1$ and $r = 0$ on the *same fixed test configuration*:

$$\Delta_{\text{endpoint}} = |\text{EER}_{r=1} - \text{EER}_{r=0}|. \quad (5.1)$$

The mirrored cross configuration records whether the direction of failure also reverses. Intermediate ratios describe the response across training ratios. The absolute difference between the two cross alignments at one ratio is reported as a descriptive *cross-alignment asymmetry*.

Supporting measures. Balanced accuracy gives equal weight to the two classes and is used with per-source, per-class accuracy to describe prediction patterns at a fixed threshold [42]. Source-probe accuracy measures whether the handcrafted feature views contain information about dataset origin. Cramér’s V describes association between binned continuous features and either dataset origin or class [39]. These supporting quantities answer different questions and are not substituted for the fixed-condition endpoint comparison.

The primary, same-corpus, and most processed grids are single-seed. The three-seed SpeechFake results are summarised with the mean and sample standard deviation for those matched cells only.

5.6. Evidence block 1: availability of source information

Hypothesis reminder. If the selected measurements contain source information, held-out examples should be classifiable by dataset origin. This block implements the availability test defined in Chapter 4.

Implementation. The exploratory t-SNE analysis samples 2,500 segments from each corpus (10,000 in total). Numerical features are standardised, and categorical variables are one-hot encoded. The two-dimensional projection uses PCA initialisation, perplexity 40, and seed 42. Separate views use format/container and level-related measurements, spectral measurements excluding constant and silence fields, and their combination. Each projection is coloured once by source dataset and once by the genuine/deepfake label.

The visualisation is followed by two quantitative probes. First, a four-corpus probe samples 20,000 segments per corpus, balanced between genuine and deepfake speech. A standardised logistic-regression classifier is evaluated using five-fold grouped cross-validation, with all segments from one source file assigned to the same fold. It predicts the four-way dataset origin from three feature views: format-and-level measurements, spectral features, and their combination. Separate within-corpus probes predict the genuine/deepfake class.

A complementary continuous-feature audit computes Cramér’s V between each measurement and, separately, dataset source and the genuine/deepfake class. Fundamental-frequency mean and standard deviation were unavailable, leaving 16 measurements. Two more variables could not be divided into five distinct groups, leaving 14 reported associations. The t-SNE projections are treated as qualitative, while the grouped probes provide the held-out quantitative output.

5.7. Reference context: within-dataset performance and gap decomposition

Each of the three architectures is trained once on ASVspoof 5 and once on FakeAVCeleb, producing six runs. Each run uses genuine and deepfake entries from the same corpus. Only $r = 0.5$ is run because both nominal sources point to the same dataset; changing the ratio would reproduce the same source pool. The within-corpus checkpoints are evaluated on both corpora to supply the four directional EERs used by the hardness–difference decomposition of Lu et al. and Müller et al. [15, 14]. The main text reports only the concise reference comparison; the full decomposition is provided in Appendix A.2.

5.8. Evidence block 2: controlled test of source dependence

This block contains three connected comparisons. The same-corpus speaker-group baseline is presented first to show what the ratio procedure produces when the two nominal sources are speaker-disjoint groups from one corpus. The primary cross-corpus grid then tests dependence on a dataset source–label relationship, and the seeded second-pair grid tests whether the response across training ratios recurs.

5.8.1. Same-corpus speaker baseline and control

Hypothesis reminder. If arbitrary grouping and the ratio procedure alone create a large response, changing the assignment of two speaker-disjoint groups from one corpus should produce one. A comparatively flat response instead provides a baseline for the cross-corpus comparisons.

Implementation. ASVspoof 5 speakers are divided into groups A and B using protocol speaker identifiers. The training split contains 200 speakers in each group; validation contains 392 and 393.

The groups are directly verified as speaker-disjoint within each split, with no path overlap, and together reconstruct the original selectors. The training groups contain 9,489/82,560 and 9,308/81,000 genuine/deepfake entries, respectively. The same five ratios and three architectures produce 15 runs, using batch size 32.

5.8.2. Primary source–label intervention

Hypothesis reminder. Dependence on the development source–label relationship should appear as a large EER change on an unchanged test configuration when that relationship is reversed.

Implementation. ASVspoof 5 is designated source *A* and FakeAVCeleb source *B*. Five separate detectors of each architecture are trained at $r \in \{1, 0.75, 0.5, 0.25, 0\}$, producing 15 runs. At $r = 1$, all selected genuine entries come from ASVspoof 5 and all selected deepfake entries from FakeAVCeleb. At $r = 0$, this mapping is reversed. At $r = 0.5$, both labels receive equal proportions of selected file entries from both corpora. Intermediate ratios gradually change the file-selection association.

The same file-selection ratio is applied to training and validation, and validation EER selects the checkpoint. The treatment is therefore the complete learning-and-selection pipeline described in Section 5.3.2. All four test configurations in Table 5.4 remain fixed across ratios.

5.8.3. Second-pair and seeded replication

Hypothesis reminder. If the primary response is not unique to FakeAVCeleb or seed 42, the response across training ratios should recur after replacing FakeAVCeleb with SpeechFake and across the repeated seeds.

Implementation. SpeechFake is source *A* and ASVspoof 5 source *B*. The five-ratio, three-model grid is first trained at seed 42 in two variants: one using the complete SpeechFake bilingual selectors and one excluding LibriTTS-derived entries. These variants account for 30 runs. The no-LibriTTS raw grid is then repeated with seeds 41 and 43, adding 30 runs. Results are reported primarily for the no-LibriTTS variant.

The four fixed test configurations mirror the primary design: `cross_SpeechFake_real`, `cross_ASV5_real`, `SpeechFake_standard`, and `ASV5_standard`. The stored prediction files contain 33,643, 32,109, 22,189, and 43,563 rows, respectively.

A matched RMS-processed version of the no-LibriTTS grid is also trained at seeds 41, 42, and 43. The seed-42 RMS grid is counted among the processing interventions; seeds 41 and 43 contribute the remaining 30 runs. Together, the raw and RMS conditions supply three matched checkpoints per model, ratio, and processing condition.

5.9. Evidence block 3: characterising the dependence

This block implements two follow-up analyses: a cell-level audit of the aligned models and repetitions of the ratio grid after three audio-processing changes.

5.9.1. Prediction pattern

Hypothesis reminder. If the fully aligned models rely mainly on the development source–label relationship, their errors should concentrate in test cells that contradict that relationship.

Implementation. The audit uses the seed-42 $r = 1$ primary checkpoints and their stored predictions. Accuracy is tabulated separately for ASVspoof 5 genuine, ASVspoof 5 deepfake, FakeAVCeleb genuine, and FakeAVCeleb deepfake segments at threshold 0.5, together with balanced accuracy. This shows where the errors occur; it does not identify which source property caused them.

5.9.2. Processing changes used to test possible sources of the shortcut

Hypothesis reminder. One simple possibility is that a conspicuous source property carries most of the effect. If a tested property is necessary for the response, changing it consistently during training, validation, and evaluation and then retraining the models should make the response across training ratios substantially flatter.

All three analyses retrain the three architectures across all five ratios and apply the same processing change during training, validation, and evaluation. They repeat the controlled comparison from Evidence Block 2; they are not tests performed only on an already trained model.

Table 5.6: Processing changes used to test possible sources of the shortcut. Every change is applied consistently during training, validation, and evaluation.

Processing change	Why it was tested	Training runs
Reduce bandwidth to 8 kHz	The primary corpora differ in original sample rate; this removes usable content above 4 kHz	15; seed 42
Match RMS signal level	Different recording pipelines may produce a consistent difference in waveform strength	60; three matched seeds on the replication pair
Remove low-frequency energy	The source audit identified very-low-frequency energy as a possible source difference	30; seed 42

Exact processing changes, run configurations, and the reported result grids are given in Appendix A.4.

6

Results and general discussion

This chapter reports the results and brings the evidence together to answer the two research questions. Supporting numerical results are reported in Appendix A.

6.1. How to read the results

The chapter follows three questions. First, can dataset source be identified from the available measurements? Conventional within- and cross-dataset results then provide a reference point. Second, does changing the relationship between dataset source and class change detector behaviour? The same-corpus speaker baseline, main experiment, and seeded replication answer RQ1. Third, what do the error patterns and tests of possible cues reveal about the information involved? This addresses RQ2. Each block states both its conclusion and what that conclusion does not establish. EER is reported as a fraction: lower is better, and 0.5 is chance performance.

6.2. Evidence block 1: availability of source information

Question. Before asking whether a detector uses a shortcut, it is necessary to ask whether the datasets contain information from which their source can be recognised. The analyses in this section test availability only. They do not yet test whether any detector uses that information.

6.2.1. Visualising source and class structure

Figure 6.1 provides the exploratory starting point. Each column shows one feature view projected to two dimensions. The upper row colours the 10,000 samples by their source corpus; the lower row colours the same projections by the genuine/deepfake label.

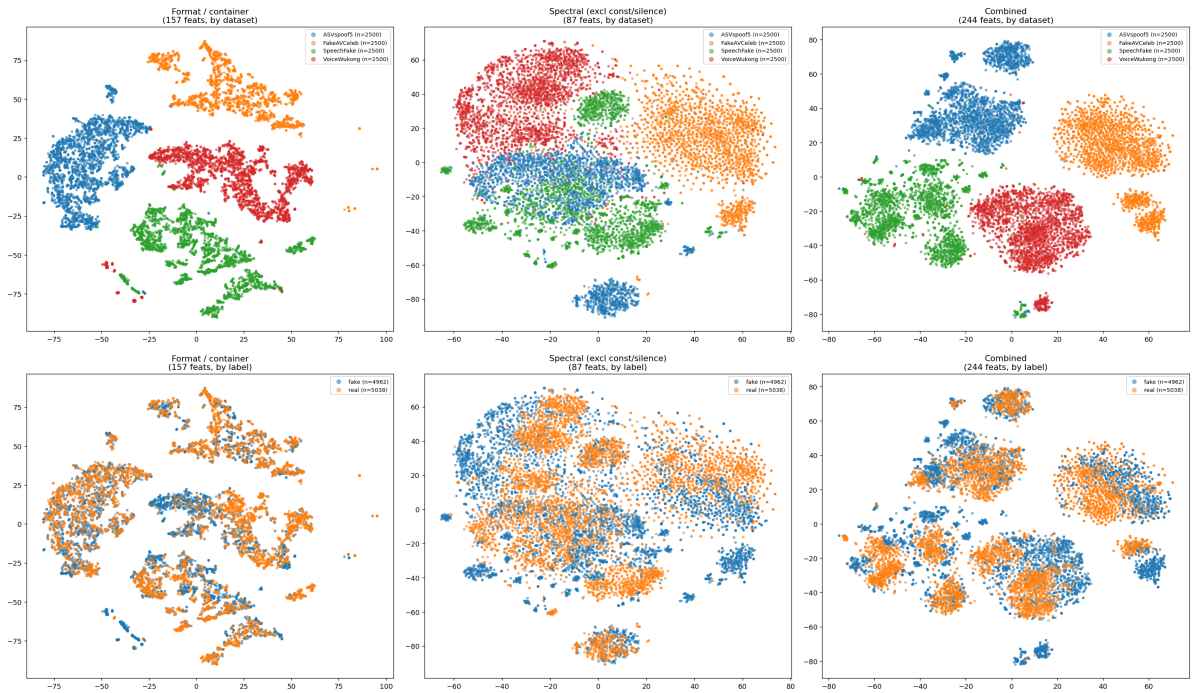


Figure 6.1: Exploratory t-SNE projections of three feature views for 10,000 segments (2,500 per corpus). The upper row is coloured by source dataset and the lower row by genuine/deepfake class. Format/container and combined measurements form visibly source-specific regions, while the spectral view retains partial source structure and more overlap. Genuine and deepfake samples are intermingled within many of the same regions. The projections use standardisation, PCA initialisation, perplexity 40, and seed 42.

When coloured by source, the format/container and combined views form visibly separate corpus regions. The spectral view is less cleanly divided, but it still contains substantial source-related structure. Colouring the same projections by class produces considerably more local overlap. This visual contrast makes source information a plausible shortcut opportunity in the measured data.

The figure is qualitative: t-SNE can emphasise local neighbourhoods and cannot establish how well source will be predicted for held-out recordings. The next analysis therefore tests the visual pattern with grouped classifiers.

6.2.2. Predicting the source dataset

Table 6.1 reports how accurately small classifiers identify which of four datasets supplied an example from a source file that was not used to train that classifier. Chance accuracy is 25%. The main comparison is between the format-and-level measurements and the spectral measurements.

Table 6.1: Held-out source-dataset prediction accuracy (four classes; chance 25%). Grouped five-fold cross-validation keeps segments from the same source file in the same fold. The mixed format-and-level view combines file properties with waveform-derived level, dynamics, and silence measurements.

Feature view	All examples	Genuine only	Deepfake only
Format and level	100.0%	100.0%	100.0%
Spectral	94.7%	97.6%	96.0%
All features	100.0%	100.0%	100.0%

Table 6.1 shows that a simple classifier can identify the source dataset from more than one type of measurement. Even when file-header properties are excluded, the spectral measurements identify the source with high accuracy. In plain language, the corpora differ enough in their sound or processing for their origin to be recognised in several ways. This shows that possible source cues are present in the data; it does not show that the three deepfake detectors use them.

6.2.3. Supporting feature associations

The same feature views can also predict the genuine/deepfake label, although source prediction is easier. Within the primary pair, 12 of 14 audited scalar measurements are more strongly associated with source identity than with class. Peak level, estimated noise floor, digital silence, and spectral roughness are among the strongest source markers. These observations motivate the later tests of possible cues, but they show only that potential cues are available. They do not show that a trained detector uses them. The complete label-probe and association results are reported in Appendix A.1.

Block takeaway and limit. **Dataset source is recognisable from several measured views.** This establishes that source-related information is available as a possible shortcut. It does not show that RawNet2, LCNN, or Wav2Vec-AASIST actually uses that information; Evidence Block 2 tests use directly.

6.3. Reference performance within and across datasets

Question. How well do the three architectures perform under standard within-dataset evaluation, and what does ordinary cross-dataset evaluation say before the controlled intervention is applied?

All three architectures obtain much lower within-dataset EER on FakeAVCeleb (0.002–0.007) than on ASVspoof 5 (0.310–0.383). A low within-dataset EER shows that a model separates held-out examples under that dataset’s construction conditions; it does not identify the rule used. This distinction matters because FakeAVCeleb contains a documented leading-silence cue [8].

Ordinary cross-dataset evaluation is strongly directional. The additional EER associated with dataset mismatch is 0.002–0.081 when FakeAVCeleb-trained models are evaluated on ASVspoof 5, but 0.231–0.398 in the reverse direction. This is useful context, but it compares separately trained systems and different test distributions. It shows that transfer is difficult; it cannot establish which training correlation caused the difficulty. The complete baseline and gap tables are reported in Appendix A.2. The next block therefore begins with a same-corpus speaker baseline and then changes one specified relationship across datasets while holding each test condition fixed.

6.4. Evidence block 2: controlled test of source dependence

This block answers RQ1 in three steps. The same-corpus speaker baseline checks whether the ratio procedure alone creates a large response. The primary pair tests whether reversing the relationship between dataset and class changes detector behaviour. The seeded replication checks whether the result occurs again with another corpus pair and several random training runs.

6.4.1. Same-corpus speaker baseline and control

Question. What response does the ratio procedure produce when its two nominal sources are speaker-disjoint groups from ASVspoof 5 rather than two different datasets?

Figure 6.2 shows the complete response across the five training ratios. The checkpoints are trained with the same ratio procedure as the later interventions, but group A and group B both come from ASVspoof 5. The two fixed test compositions are retained from the primary experiment so that the control and cross-corpus curves are directly comparable; their inclusion does not make the development intervention cross-corpus.

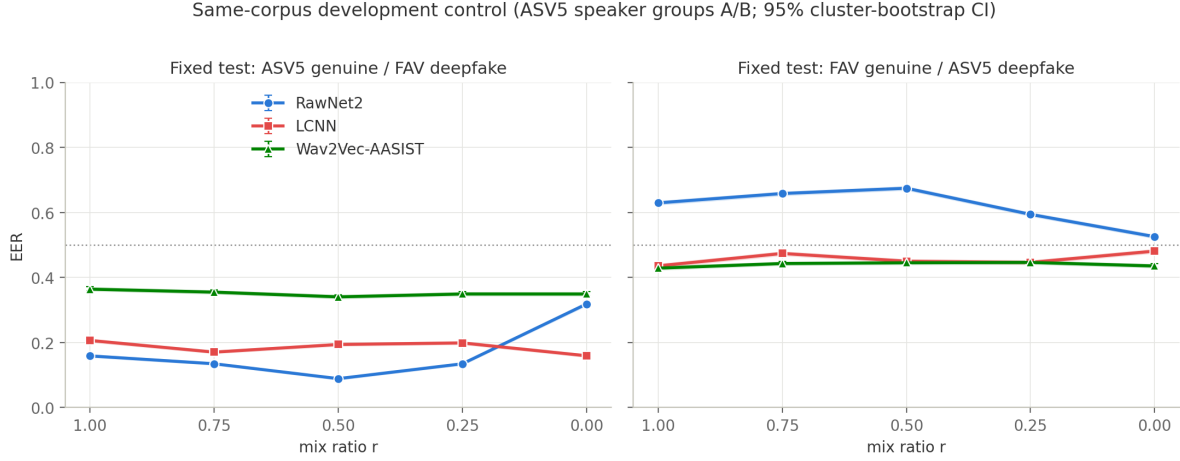


Figure 6.2: Same-corpus speaker baseline. Training and validation range from $r = 1$ to $r = 0$ by changing which of two speaker-disjoint ASVspooF 5 groups supplies each class. The panels reuse the primary experiment’s two fixed cross-corpus test compositions; ASV5 denotes ASVspooF 5 and FAV denotes FakeAVCeleb. Each panel holds its test recordings fixed. Lines show segment-level EER for the fixed, single-seed checkpoint developed at each ratio. Capped bars and shading show pointwise 95% percentile intervals from 5,000 source-file-clustered bootstrap resamples, stratified by source and class; all VAD segments from a sampled file are retained. These intervals quantify test-pool sampling uncertainty conditional on each checkpoint, not variation from retraining. The dotted line marks chance EER.

The baseline does not reproduce the later cross-corpus collapse. RawNet2 varies more than the other models, so the result is not perfectly flat. However, neither fixed test composition shows the coordinated, architecture-wide inversion found when dataset source is aligned with class. The result argues against arbitrary partitioning and recomposition as a sufficient explanation. It does not show that ASVspooF 5 is free of shortcuts, nor does it provide a universal uncertainty band for the other single-seed experiments.

6.4.2. Primary source–label intervention

Question. Do the detectors’ predictions depend on which dataset usually provides the genuine examples and which dataset usually provides the deepfakes during model development?

Figure 6.3 shows EER for the primary ASVspooF 5 $\times$ FakeAVCeleb pair. Each line represents separately trained models across the five mixing ratios. The left panel uses a fixed test set in which genuine examples come from ASVspooF 5 and deepfakes come from FakeAVCeleb. The right panel reverses that test composition. The reader should first compare the two endpoints of each line, $r = 1$ and $r = 0$, and then check whether the direction reverses between the two panels.

All three models change greatly when the source–label assignment used during development is reversed. When that assignment agrees with the fixed test composition, the models perform well. When it conflicts with the test composition, their score ordering is often reversed. Because the evaluated recordings do not change, this provides strong evidence that model development relied on information associated with the source–label relationship.

The experiment changes several connected parts of development: which files are selected for training and validation, the source composition available to the model, and which checkpoint is selected. It therefore does not identify container, codec, speaker, content, generation method, or recording chain as the individual cause.

The intermediate points describe the response shape, but the primary causal contrast is the fixed-test $r = 1 \leftrightarrow 0$ endpoint. In the primary pair the shape is threshold-like rather than perfectly monotonic. At $r = 0.5$, both classes receive the same source proportions among selected file entries. This is the file-selection balance point; the endpoint contrast remains the primary causal comparison.

The within-corpus test conditions remain informative: they retain discrimination beyond a purely source-only rule. They do not reveal whether that residual discrimination is based on forensic evidence or on another within-corpus cue.

6.4.3. Replication with SpeechFake and ASVspooF 5

Question. Does the main result recur with a second, more technically similar corpus pair?

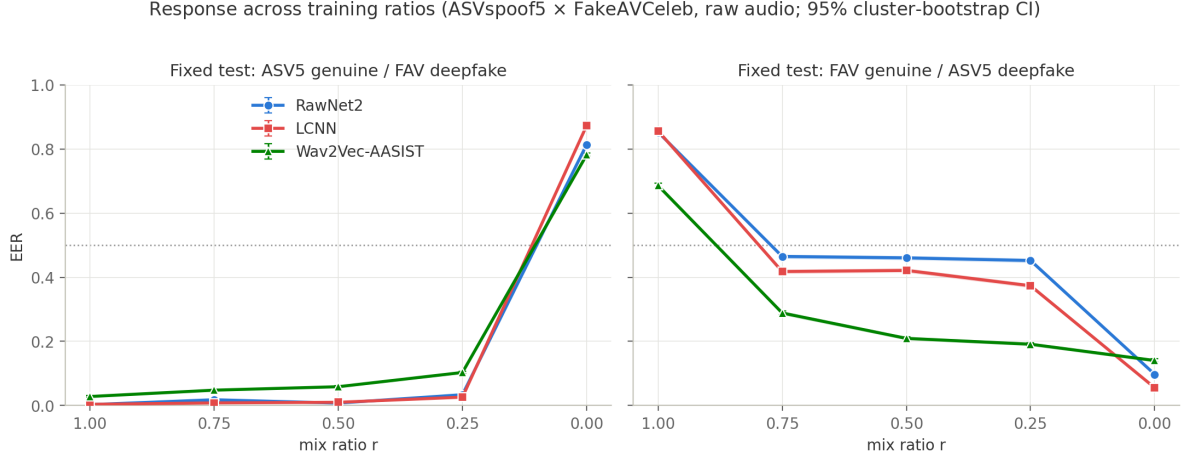


Figure 6.3: Primary source-label intervention. Training and validation range from the aligned composition ($r = 1$), through the file-selection balance point ($r = 0.5$), to the reversed composition ($r = 0$). Each panel holds its test examples fixed; ASV5 denotes ASVspooF5 and FAV denotes FakeAVCeleb. Lines show segment-level EER for the fixed, single-seed checkpoint developed at each ratio. Capped bars and shading show pointwise 95% percentile intervals from 5,000 source-file-clustered bootstrap resamples, stratified by source and class; all VAD segments from a sampled file are retained. These intervals quantify test-pool sampling uncertainty conditional on each checkpoint, not variation from retraining. The dotted line marks chance EER.

Figure 6.4 shows the replicated response for the SpeechFake-BD × ASVspooF5 pair after excluding SpeechFake’s LibriTTS-derived genuine subset. Lines are means over three random seeds and the bands are sample standard deviations. The left panel keeps genuine SpeechFake and deepfake ASVspooF5 examples fixed; the right panel uses the mirror composition. The exact values are reported once in Appendix Table A.5.

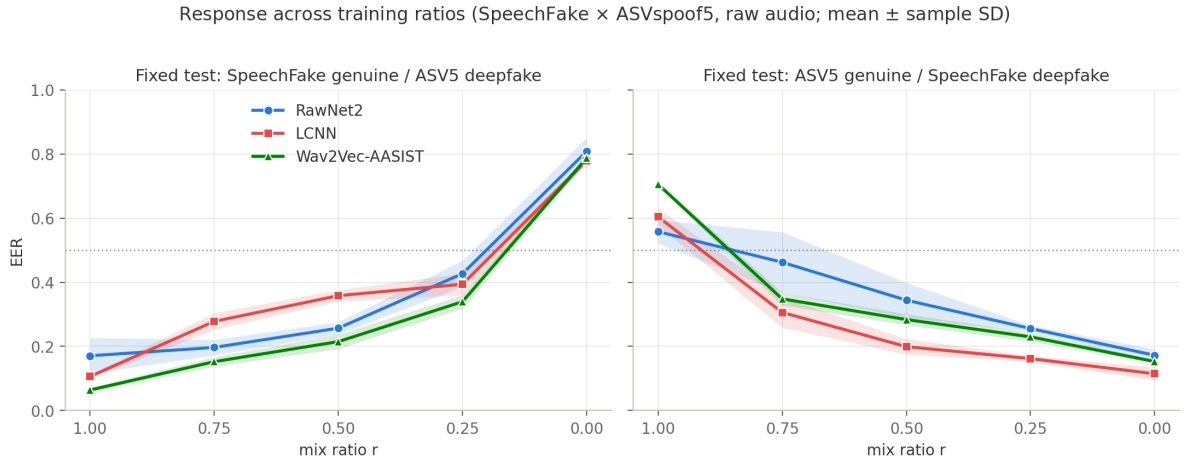


Figure 6.4: Replication on SpeechFake-BD (LibriTTS-excluded) × ASVspooF5. Lines show mean EER and shaded bands show sample standard deviation over seeds 41, 42, and 43. Each panel keeps its evaluation recordings fixed while the source-label relationship used during development changes across the five ratios; ASV5 in the panel titles denotes ASVspooF5. The dotted line marks chance EER.

All three architectures again change in the predicted direction, and the mirrored test composition changes in the opposite direction. Unlike the primary pair, the response is visibly graded across the intermediate ratios. The effect is therefore not unique to FakeAVCeleb or to one random seed. However, both tested pairs include ASVspooF5, so the replication does not establish the same magnitude for arbitrary pairs of corpora.

RQ1 block takeaway and limit. Under both tested multi-source settings, all three detector families depend strongly on how dataset source predicts the genuine/deepfake label during development.

Reversing that relationship changes predictions on the same test recordings; an arbitrary same-corpus split does not reproduce the coordinated effect, and the response recurs for a second pair and three seeds. This is strong evidence of source-related shortcut dependence under the tested conditions. The intervention changes the complete development pipeline and a bundle of source-related properties, so it does not identify one individual cue or estimate prevalence across all benchmarks.

6.5. Evidence block 3: characterising the dependence

Evidence Block 2 shows that the detectors use source-correlated information. It does not reveal which examples fail or which source difference carries the effect. This block addresses those two questions by examining prediction patterns and repeating the ratio experiment after altering possible cues.

6.5.1. What the trained models predict

Question. Do the fully aligned models predict class only from dataset source, or can they still distinguish genuine and deepfake speech within the same dataset?

Table 6.2 reports accuracy separately for each combination of dataset and true class. A literal rule that treats all ASVspoof 5 recordings as genuine and all FakeAVCeleb recordings as deepfake would fail on every ASVspoof 5 deepfake and every genuine FakeAVCeleb recording.

Table 6.2: Accuracy of the fully aligned ($r = 1$) models at threshold 0.5, separated by source and class.

Model	ASV5 genuine	ASV5 deepfake	FAV genuine	FAV deepfake
RawNet2	0.998	0.008	0.666	0.996
LCNN	0.998	0.006	0.606	0.994
Wav2Vec-AASIST	0.950	0.209	0.437	0.986

The decisions are source-dominated and strongly asymmetric. The most striking failure is that ASVspoof 5 deepfakes are usually labelled genuine, while the cells that match the development composition are classified accurately. Nevertheless, the non-zero FakeAVCeleb genuine performance shows residual class- and corpus-specific discrimination. The full decision function should therefore not be described as a literal dataset-identity classifier.

6.5.2. Testing possible sources of the shortcut

The primary pair differs in several obvious ways, including original frequency range, signal level, and low-frequency energy. Each was tested briefly as a possible explanation: bandwidth reduction asks whether the difference above 4 kHz is necessary; RMS matching asks whether overall signal level contributes; and high-pass filtering asks whether very-low-frequency information carries the effect. Each change is applied during training, validation, and evaluation, and new models are developed at every ratio. If one changed property were necessary for the original response, the response across training ratios should become substantially flatter. If it remains steep, the implemented processing change was insufficient; this does not prove that the model ignored that information.

Table 6.3: Summary of the tests of possible shortcut sources. “Flattened” asks whether the fixed-test response across training ratios became substantially flat after the processing change.

Intervention	Main result	Flattened?
Reduce bandwidth to 8 kHz	The endpoint reversal remains for all three architectures.	No
Match RMS signal level	RawNet2 changes materially in some conditions; LCNN and Wav2Vec-AASIST change less, but every architecture retains a response across training ratios.	No
Remove low-frequency energy	Neither the 80 Hz nor the 300 Hz filter removes the endpoint reversal.	No

Table 6.3 gives the conclusion of the three tests. Reducing bandwidth and removing low-frequency

energy do not remove the endpoint reversal. Their exact endpoint values and single-seed grids are reported in Appendix A.4.

The matched RMS analysis is shown in more detail because it is the only such test with a three-seed, batch-matched raw comparison on the replication pair. Figure 6.5 compares raw and RMS-normalised pipelines on the replication pair. It reports means and sample-standard-deviation bands over three seeds. The reader should first compare the two processing policies within each model, ratio, and fixed test composition, and then check whether either processing change produces a flat curve. Exact values and paired differences are given once in Appendix Table A.13.

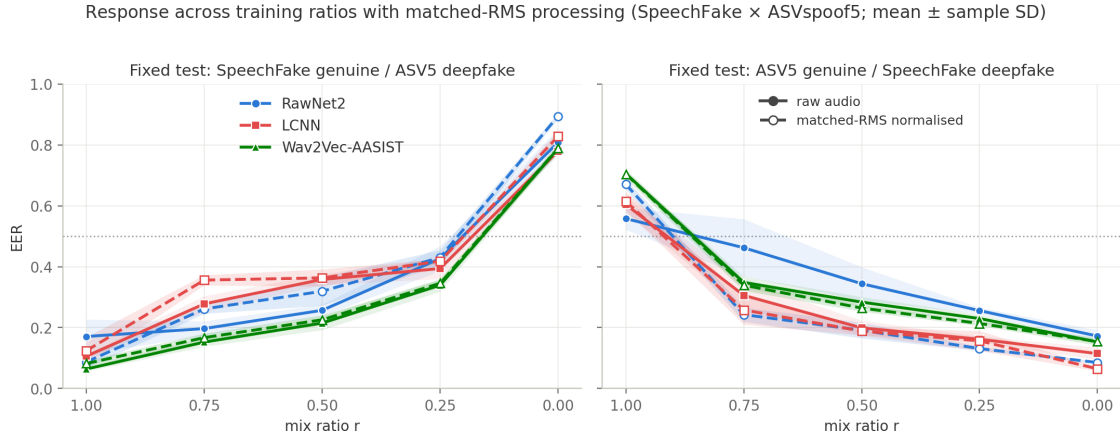


Figure 6.5: Raw and matched RMS-normalised pipelines on the replication pair (mean $\pm$ sample standard deviation over three seeds). Each panel overlays all three architectures: colour and marker identify the architecture, while solid and dashed lines identify the raw and matched-RMS pipelines, respectively. ASV5 in the panel titles denotes ASVspoof 5. RMS scaling toward the target and the peak ceiling are applied to each VAD interval throughout development and evaluation.

Across the 30 model–ratio–test-composition rows, the maximum absolute mean paired EER change is 0.2196 for RawNet2, compared with 0.0781 for LCNN and 0.0196 for Wav2Vec-AASIST.

RMS matching affects RawNet2 most strongly and has smaller, mixed effects for LCNN and Wav2Vec-AASIST. No architecture’s response across training ratios becomes flat. The matched comparison therefore provides strong causal evidence that RawNet2’s learned solution depends on a corpus-correlated level cue in particular conditions. It does not establish scalar RMS level as the unique mediator: the intervention retrains the model and changes the waveform pipeline, and its effect varies by ratio and test composition.

None of the tested processing changes was sufficient to remove the source dependence. This is consistent with source information being represented through several overlapping properties, or through relevant properties not covered by the completed interventions. It does not show that each detector jointly used a particular combination of cues.

RQ2 block takeaway and limit. The observed behaviour is source-dominated and asymmetric, but not source-only, and no tested processing change removes the dependence. Matched level processing changes RawNet2 materially in some conditions, while the other tests effects are smaller or inconsistent and every response across training ratios remains steep. These tests narrow the possible explanations without identifying one responsible property.

6.6. General discussion

The three evidence blocks answer a connected sequence of questions. The first shows that source information is available. The second establishes that detector development depends on the manipulated source relationship and uses its control and replication to define the scope of that result. The third describes the resulting errors and asks whether selected information families account for the dependence. Keeping these levels separate prevents cue availability from being mistaken for cue use or a possible explanation from being presented as the complete mechanism.

6.6.1. Answer to RQ1: to what extent do the detectors rely on shortcuts?

All three tested architectures rely strongly on information that is correlated with dataset source in the two studied multi-source settings. The main evidence is the large fixed-test reversal caused by changing source-label composition during model development. The same direction of effect appears in a second corpus pair and across three seeds, while the same-corpus speaker-split control does not reproduce it. The source-by-class results for the fully aligned models further show that the failures are concentrated in cells that contradict the development composition.

This is stronger than showing ordinary cross-dataset degradation. Earlier work has shown that speech deepfake detectors often transfer poorly between datasets [13, 14]. The present intervention shows one reason why such failure can arise: when source and label are correlated during development, changing that relationship can change or invert predictions on the same test examples. The finding is consistent with the broader shortcut-learning literature, in which models use easy predictive regularities that are valid in the training environment but unstable elsewhere [5, 30].

The conclusion is deliberately bounded. The comparison measures the total effect of changing the complete development procedure under the tested conditions. It does not mean that each model bases every decision on source identity, and it does not measure how common shortcut reliance is across all detectors or benchmarks.

6.6.2. Answer to RQ2: what is the nature of the shortcuts?

The detectors relied on information associated with dataset source, but the individual source of that information was not identified. Dataset source could be predicted from several different kinds of measurements. The largest model errors occurred in the source-label groups that contradicted the relationship seen during development. However, reducing bandwidth, matching signal level, and removing low-frequency energy did not remove the overall dependence. The level-processing experiment changed RawNet2 in some conditions, but it did not explain the pattern across all models and test compositions.

This conclusion agrees with earlier evidence that detectors can exploit dataset-construction cues such as leading or trailing silence [6, 8], and with research showing that synthesis and transmission systems leave detectable traces [9, 10, 32]. The present study extends that concern from one known cue to the wider relationship between dataset source and label. The fact that source can be recognised from several kinds of measurements does not show that the trained detectors use all of them or switch from one cue to another after retraining.

6.6.3. Implications for benchmark interpretation

The results do not make within-dataset EER meaningless. Such an EER remains a valid description of discrimination under the sampled dataset conditions. What it cannot establish on its own is that the model learned a general detection rule that transfers to new recordings. This distinction matters when genuine and deepfake examples have different origins, recording chains, or processing histories. The long-standing dataset-bias problem in computer vision provides a close analogy: strong performance can depend on properties of how a benchmark was collected rather than on the intended object concept [31].

For speech deepfake benchmarks, a stronger evaluation should therefore make the source-label relationship explicit and challenge it. Useful checks include reporting what proportion of genuine and deepfake files came from each source, testing both the original and reversed source-label assignments, and using held-out datasets whose source conditions do not reproduce the development relationship. The claim is not that published results trained on one dataset are automatically untrustworthy. It is that a single in-domain score is insufficient evidence of general detection ability when dataset origin and

class label may be closely linked.

6.6.4. Implications for detector architectures

RawNet2, LCNN, and Wav2Vec-AASIST process speech in different ways, yet all respond strongly to the source intervention. RawNet2 and LCNN show the same source dependence even though one processes raw waveforms and the other processes LFCC summaries. This shows that removing sample-level waveform detail does not by itself prevent the effect. The source-prediction analysis also shows that spectral measurements contain source information. Wav2Vec-AASIST has the smallest change on the primary pair but the largest mean change in the replication, so the results do not support ranking the three architectures by robustness. The comparison also cannot separate the effect of self-supervised pretraining from the frozen encoder, AASIST back end, training procedure, or corpus pair. Prior work finds that self-supervised representations can improve cross-dataset performance [34, 33]; the present results show that a self-supervised front end does not guarantee resistance to source-label shortcuts.

6.6.5. Methodological contribution and researcher reflection

The main methodological contribution is an evaluation method rather than a new detector. It builds on established cross-dataset tests, simple cue classifiers, and experiments that vary an unwanted relationship [37, 5]. The study-specific step is to change how often each corpus supplies genuine and deepfake examples during model development, then compare every trained model on the same recordings. The mirrored test conditions and the separation between available cues, learned dependence, and possible explanations are also parts of this design.

Three choices strengthened the final argument. The mirrored fixed tests make reversal of the learned relationship visible; the same-corpus control tests whether arbitrary grouping is sufficient; and the second corpus pair shows that the effect is not confined to FakeAVCeleb. One choice should be improved in a follow-up: the primary pair differed in too many ways to isolate one explanation. A cleaner study would define the exact comparison before running the analyses and use more tightly matched corpus pairs.

6.6.6. Societal and field context

Speech deepfake detectors may support decisions about fraud, authenticity, or platform safety. A benchmark result can therefore be misleadingly reassuring if the detector recognises properties of the benchmark's construction rather than evidence that transfers to new recordings. The consequence is not only lower accuracy: errors may concentrate on recording or processing conditions that were poorly represented during development, while a headline score hides that dependence.

The appropriate response is not to dismiss benchmark evaluation, but to make its scope transparent. Dataset documentation should report the origins and processing histories of both classes; evaluation should include conditions that break known source-label relationships; and deployment claims should remain bounded to the evidence tested. These practices would reduce the risk of presenting dataset-specific discrimination as general forensic capability.

6.6.7. Limitations

1. **Limited corpus pairs.** Two pairs were tested, and both contain ASVspoof 5. The direction of the effect replicates, but its magnitude should not be generalised to arbitrary benchmarks or deployment settings.
2. **Several properties change together.** Changing source-label composition changes speakers, content, generators, recording conditions, and file histories together. The experiment identifies dependence on this set of differences, not one responsible property.

6.6.8. Future work

Future work should test additional pairs that do not share ASVspoof 5; report the selected-file composition explicitly; enforce speaker- and source-file-disjoint evaluation; and report multi-seed uncertainty with source-file-clustered resampling. Studies of possible cues should then vary one factor at a time using matched recording or re-encoding pipelines, so that codec history, level, silence, bandwidth, and generator family can be separated rather than bundled.

At the benchmark level, the same logic suggests source-aware dataset design. Genuine and deepfake examples should be matched as closely as possible in collection and post-processing, while evaluation conditions should deliberately break any remaining relationship between source and label. Such work would move from demonstrating that a detector can separate one benchmark toward testing whether it recognises evidence that remains meaningful when dataset origin changes.

Conclusion

7.1. Research problem and approach

Speech deepfake detection asks whether a recording contains genuine human speech or speech that was generated or manipulated by a computer. Detectors are commonly evaluated on held-out recordings from a benchmark. This is useful for comparing systems under shared conditions, but a good score does not reveal which rule a detector has learned. A detector may respond to evidence of generation, or it may use an easier pattern created by the way the benchmark was assembled. If that pattern changes outside the benchmark, performance can fail even though the original test score was strong.

This thesis examined one form of this problem: a relationship between dataset origin and the genuine/deepfake label. When genuine and deepfake recordings are drawn from different sources, the sources may differ in speakers, generation systems, recording conditions, silence, signal level, or audio processing. Any of these differences can help predict the label. In this thesis, a *shortcut* therefore means dependence on source-related information that is useful under the development conditions but is not a dependable general rule for detecting deepfakes.

The research followed three steps. First, data analyses tested whether dataset source could be identified from technical, signal, and spectral measurements. Second, the relationship between source and class was deliberately changed during model development, while every detector was evaluated on the same recordings. The same-corpus control checked whether the ratio procedure itself created the effect, and a second corpus pair tested whether the result repeated. Third, prediction errors and selected audio-processing changes were examined to investigate which information might contribute. Together, these steps separate three stages of evidence: is source information available, do detectors use it, and can one responsible property be identified?

7.2. Answers to the research questions

RQ1: to what extent do speech deepfake detectors rely on shortcuts? To answer this question, the relationship between dataset source and class label was deliberately changed during model development. Separate detectors were trained under aligned, mixed, and reversed source-label compositions, while every detector was evaluated on the same fixed recordings. The comparison covered three detector architectures, a second corpus pair, and a same-corpus speaker control.

The answer is that **under the tested multi-source conditions, information associated with dataset source formed an exploitable shortcut for all three detector architectures**. Reversing the source-label relationship during development caused large changes in detector scores and EER on the same evaluation recordings. The effect appeared with both corpus pairs and across all three architectures, whereas the same-corpus control did not show a comparable response. The detectors had therefore not learned only a rule that worked independently of dataset source; their behaviour also depended strongly on the source-label relationship available during development.

This conclusion is specific to the implemented experiments. It does not mean that every prediction was based on source information, that every benchmark contains the same problem, or that the observed effect size applies to all detectors. Dataset source combines speakers, content, generation systems,

recording conditions, and processing history. The intervention therefore establishes dependence on a set of source differences that change together, not on one isolated file or signal property.

RQ2: what is the nature of those shortcuts? To answer this question, the study examined whether source information was present in several feature views, where the trained detectors made errors, and whether the main response remained after retraining with selected audio-processing changes.

The answer is that **the shortcut was associated with dataset source, but this study could not reduce it to one technical or signal property**. Dataset source could be predicted from several different kinds of measurements. This shows that many kinds of source information were available in the data; it does not show that every detector used all of them. The largest detector errors followed the source–label relationship used during development. However, reducing bandwidth, matching signal level, and removing low-frequency energy did not remove the overall source dependence. Level matching affected RawNet2 in some conditions, but did not explain the pattern across all architectures and test compositions.

The experiments therefore do not establish that codec, container, sample rate, loudness, silence, bandwidth, or another individual property is responsible for the shortcut. The completed interventions cover only a small part of the much larger space of possible source differences and interactions. Identifying one responsible property would require a substantially larger programme of carefully matched experiments that vary one factor at a time. Not identifying one responsible property does not weaken the answer to RQ1: showing that detectors use source-related information and identifying exactly which property carries it are separate questions.

Taken together, the two research questions support the central conclusion of this thesis: **under the tested conditions, all three detector architectures exploited information associated with dataset source, although the specific information carrying this dependence could not be isolated**.

7.3. Main contribution and implications

Previous research has shown that speech deepfake detectors can exploit unintended silence patterns and other dataset-construction artefacts [6, 8], and that performance often decreases when detectors are evaluated on a different dataset [13, 14]. Those studies establish that specific cues can matter and that transfer is difficult. The present work adds a controlled link between these observations: when dataset source was correlated with class during development, deliberately changing that relationship changed the learned detector’s behaviour on fixed recordings.

This does not invalidate within-dataset benchmarks or the studies that use them. A benchmark score remains a valid description of performance under that benchmark’s sampled conditions. The result instead changes how such a score should be interpreted. On its own, it is not evidence that the detector learned a rule that will remain useful when speakers, generators, recording chains, and processing histories change. This is an instance of the wider shortcut-learning problem, in which a model can solve the available benchmark through a predictive but unstable rule [5, 30].

The methodological lesson is equally direct. Ordinary cross-dataset evaluation shows whether transfer succeeds, but it changes the evaluation data and many of their properties at once. The controlled comparison developed in this thesis instead changes the relationship available during model development and keeps the evaluation recordings fixed. It can therefore test whether detector development depends on that relationship, even when the exact property carrying the effect remains unknown.

7.4. Reproducibility

Reproducibility and availability. All code used to prepare the data selections, train the models, and reproduce the reported analyses is available through the [TU Delft Research Repository](#).

7.5. Scope and next steps

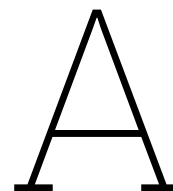
The study tested two corpus pairs, both involving ASVspoof 5, and several experiment grids used one training seed. It therefore establishes shortcut dependence under the tested conditions, not how common it is across all benchmarks. The completed tests of possible cues also cover only a small part of the technical, acoustic, speaker, content, and generation differences between natural datasets. Future work could use explainability methods to identify candidate cues more systematically, followed by matched retraining experiments that alter each candidate during training, validation, and evaluation. Explainability would guide the search, while the controlled experiments would test whether the highlighted information actually contributes to the shortcut.

References

- [1] B Zhang et al. “Audio Deepfake Detection: What Has Been Achieved and What Lies Ahead”. In: *Sensors* 25.7 (2025), p. 1989.
- [2] Zhizheng Wu et al. “ASVspoof 2015: The First Automatic Speaker Verification Spoofing and Countermeasures Challenge”. In: *Interspeech 2015*. 2015, pp. 2037–2041.
- [3] Massimiliano Todisco et al. “ASVspoof 2019: Future horizons in spoofed and fake audio detection”. In: *Interspeech 2019*. 2019, pp. 1008–1012.
- [4] Xin Wang et al. “ASVspoof 5: Crowdsourced speech data, deepfakes, and adversarial attacks at scale”. In: *ASVspoof Workshop 2024*. 2024, pp. 1–8. doi: [10.21437/ASVspoof.2024-1](https://doi.org/10.21437/ASVspoof.2024-1).
- [5] Robert Geirhos et al. “Shortcut learning in deep neural networks”. In: *Nature Machine Intelligence* 2.11 (2020), pp. 665–673.
- [6] Nicolas M Müller et al. “Speech is silver, silence is golden: What do ASVspoof-trained models really learn?” In: *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*. 2021, pp. 55–60. doi: [10.21437/ASVSPPOOF.2021-9](https://doi.org/10.21437/ASVSPPOOF.2021-9).
- [7] Bhusan Chettri, Emmanouil Benetos, and Bob L T Sturm. “Dataset artefacts in anti-spoofing systems: A case study on the ASVspoof 2017 benchmark”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), pp. 3018–3028.
- [8] Stefan Smeu et al. “Circumventing Shortcuts in Audio-Visual Deepfake Detection Datasets with Unsupervised Learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2025, pp. 18815–18825.
- [9] Jordi Pons et al. “Upsampling artifacts in neural audio synthesis”. In: *ICASSP 2021*. 2021, pp. 3005–3009.
- [10] Xinrui Yan et al. “An Initial Investigation for Detecting Vocoder Fingerprints of Fake Audio”. In: *Proceedings of the 1st International Workshop on Deepfake Detection for Audio Multimedia*. 2022, pp. 61–68.
- [11] Piotr Kawa, Marcin Plata, and Piotr Syga. “Attack agnostic dataset: Towards generalization and stabilization of audio deepfake detection”. In: *Interspeech 2022*. 2022, pp. 4023–4027. doi: [10.21437/Interspeech.2022-10078](https://doi.org/10.21437/Interspeech.2022-10078).
- [12] Tianxiang Chen et al. “Generalization of audio deepfake detection”. In: *Odyssey 2020 The Speaker and Language Recognition Workshop*. 2020, pp. 132–137.
- [13] Nicolas M Müller et al. “Does Audio Deepfake Detection Generalize?” In: *Interspeech*. 2022, pp. 2783–2787. doi: [10.21437/Interspeech.2022-108](https://doi.org/10.21437/Interspeech.2022-108).
- [14] Nicolas M Müller et al. “Harder or Different? Understanding Generalization of Audio Deepfake Detection”. In: *Interspeech 2024*. 2024, pp. 2705–2709. doi: [10.21437/Interspeech.2024-247](https://doi.org/10.21437/Interspeech.2024-247).
- [15] Shangyun Lu et al. “Harder or Different? A Closer Look at Distribution Shift in Dataset Reproduction”. In: *ICML 2020 Workshop on Uncertainty and Robustness in Deep Learning*. 2020.
- [16] Alan V. Oppenheim and Ronald W. Schaffer. *Discrete-Time Signal Processing*. 3rd ed. Pearson, 2009. ISBN: 9780131988422.
- [17] Marina Bosi and Richard E. Goldberg. *Introduction to Digital Audio Coding and Standards*. Springer, 2003. doi: [10.1007/978-1-4615-0327-9](https://doi.org/10.1007/978-1-4615-0327-9).
- [18] Robert C Maher. “Audio Forensic Examination: Authenticity, Enhancement, and Interpretation”. In: *IEEE Signal Processing Magazine* 26.2 (2009), pp. 84–94.
- [19] Romain Hennequin, Jimena Royo-Letelier, and Manuel Moussallam. “Codec independent lossy audio compression detection”. In: *ICASSP 2017*. 2017, pp. 726–730.

- [20] Xu Tan et al. "A Survey on Neural Speech Synthesis". In: *arXiv preprint arXiv:2106.15561* (2021). doi: [10.48550/arXiv.2106.15561](https://doi.org/10.48550/arXiv.2106.15561).
- [21] Berrak Sisman et al. "An Overview of Voice Conversion and Its Challenges: From Statistical Modeling to Deep Learning". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021), pp. 132–157. doi: [10.1109/TASLP.2020.3038524](https://doi.org/10.1109/TASLP.2020.3038524).
- [22] Neil Zeghidour et al. "SoundStream: An End-to-End Neural Audio Codec". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 30 (2022), pp. 495–507. doi: [10.1109/TASLP.2021.3129994](https://doi.org/10.1109/TASLP.2021.3129994).
- [23] Zalán Borsos et al. "AudioLM: A Language Modeling Approach to Audio Generation". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 31 (2023), pp. 2523–2533. doi: [10.1109/TASLP.2023.3288409](https://doi.org/10.1109/TASLP.2023.3288409).
- [24] Haibin Wu, Yuan Tseng, and Hung-yi Lee. "CodecFake: Enhancing Anti-Spoofing Models Against Deepfake Audios from Codec-Based Speech Synthesis Systems". In: *Interspeech 2024*. 2024, pp. 1770–1774. doi: [10.21437/Interspeech.2024-2093](https://doi.org/10.21437/Interspeech.2024-2093).
- [25] Hemlata Tak et al. "End-to-end anti-spoofing with RawNet2". In: *ICASSP 2021*. 2021, pp. 6369–6373.
- [26] Galina Lavrentyeva et al. "Audio replay attack detection with deep learning frameworks". In: *Interspeech 2017*. 2017, pp. 82–86.
- [27] Alexei Baevski et al. "wav2vec 2.0: A framework for self-supervised learning of speech representations". In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 12449–12460.
- [28] Hemlata Tak et al. "Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation". In: *The Speaker and Language Recognition Workshop (Odyssey 2022)*. 2022, pp. 112–119. doi: [10.21437/Odyssey.2022-16](https://doi.org/10.21437/Odyssey.2022-16).
- [29] Jee-weon Jung et al. "AASIST: Audio Anti-Spoofing Using Integrated Spectro-Temporal Graph Attention Networks". In: *ICASSP 2022*. 2022, pp. 6367–6371.
- [30] Alexander D'Amour et al. "Underspecification presents challenges for credibility in modern machine learning". In: *Journal of Machine Learning Research* 23.226 (2022), pp. 1–61.
- [31] Antonio Torralba and Alexei A Efros. "Unbiased look at dataset bias". In: *CVPR 2011*. 2011, pp. 1521–1528.
- [32] You Zhang et al. "An Empirical Study on Channel Effects for Synthetic Voice Spoofing Countermeasure Systems". In: *Interspeech 2021*. 2021, pp. 4309–4313. doi: [10.21437/Interspeech.2021-1820](https://doi.org/10.21437/Interspeech.2021-1820).
- [33] Octavian Pascu et al. "Towards Generalisable and Calibrated Audio Deepfake Detection with Self-Supervised Representations". In: *Interspeech 2024*. 2024, pp. 4828–4832. doi: [10.21437/Interspeech.2024-1302](https://doi.org/10.21437/Interspeech.2024-1302).
- [34] Xin Wang and Junichi Yamagishi. "Investigating self-supervised front ends for speech spoofing countermeasures". In: *The Speaker and Language Recognition Workshop (Odyssey 2022)*. 2022, pp. 100–106. doi: [10.21437/Odyssey.2022-14](https://doi.org/10.21437/Odyssey.2022-14).
- [35] Sevinç Hiçsönmez, Hüsrev T Sencar, and İsmail Avcıbaşı. "Identification of Digital Audio Coding and Transcoding Procedures". In: *Digital Signal Processing* 23.5 (2013), pp. 1720–1730.
- [36] Tsu-Hsien Shih, Chin-Yuan Yeh, and Ming-Syan Chen. "Does Audio Deepfake Detection Rely on Artifacts?" In: *ICASSP 2024*. 2024. doi: [10.1109/ICASSP48485.2024.10446558](https://doi.org/10.1109/ICASSP48485.2024.10446558).
- [37] Martin Arjovsky et al. "Invariant risk minimization". In: *arXiv preprint arXiv:1907.02893* (2019).
- [38] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605.
- [39] Harald Cramér. *Mathematical Methods of Statistics*. Vol. 9. Princeton Mathematical Series. Princeton, NJ: Princeton University Press, 1946.
- [40] Hasam Khalid et al. "FakeAVCeleb: A novel audio-video multimodal deepfake dataset". In: *NeurIPS 2021 Datasets and Benchmarks Track (Round 2)*. 2021.

- [41] Wen Huang et al. “SpeechFake: A Large-Scale Multilingual Speech Deepfake Dataset Incorporating Cutting-Edge Generation Methods”. In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. arXiv:2507.21463. 2025, pp. 9985–9998. doi: [10.18653/v1/2025.acl-long.493](https://doi.org/10.18653/v1/2025.acl-long.493).
- [42] Kay Henning Brodersen et al. “The Balanced Accuracy and Its Posterior Distribution”. In: *2010 20th International Conference on Pattern Recognition*. IEEE, 2010, pp. 3121–3124. doi: [10.1109/ICPR.2010.764](https://doi.org/10.1109/ICPR.2010.764).



Supplementary analyses and full results

This appendix retains analyses and numerical detail that support, but are not required to follow, the main argument in Chapter 6. The material is organised by its role in the three evidence blocks.

A.1. Supplementary source-availability results

Table A.1 applies the same feature views used for source prediction to the genuine/deepfake task. Source prediction is easier than label prediction from these broad views, although spectral measurements still separate the two classes well within some individual corpora.

Table A.1: Held-out genuine/deepfake prediction by feature group (EER; chance 0.5).

Evaluation scope	Format and level	Spectral	All features
All four datasets pooled	0.372	0.213	0.200
ASVspooF 5	0.418	0.147	0.147
FakeAVCeleb	0.336	0.052	0.054
SpeechFake	0.293	0.180	0.168
VoiceWukong	0.201	0.105	0.098

Table A.2 reports all usable scalar-feature associations in the primary pair. For 12 of the 14 measurements, association with source identity is stronger than association with class. This is evidence that multiple source-related differences are available, not evidence that a trained detector uses each measurement.

Table A.2: Cramér’s V of all 14 usable continuous features with source identity and the genuine/deepfake label. Continuous features were divided into quintiles; two further features were excluded because their binning was degenerate.

Feature	Source identity	Genuine/deepfake label
peak_signal	0.848	0.071
noise_floor_db	0.778	0.099
max_zero_run_ms	0.732	0.244
dead_silence_pct	0.714	0.217
spectral_roughness	0.700	0.117
mfcc_std	0.533	0.181
spectral_flatness	0.480	0.157
zero_crossing_rate	0.439	0.152
nyquist_energy	0.435	0.146
phase_continuity	0.418	0.119
dynamic_range_db	0.350	0.047
mfcc_mean	0.162	0.080
spectral_rolloff	0.114	0.199
leading_zeros_ms	0.104	0.146

A.2. Reference baselines and gap decomposition

Let $E(A \rightarrow B)$ denote the EER of a detector developed on dataset A and evaluated on dataset B . The hardness–difference framework separates the cross-dataset performance gap as follows [15, 14]:

$$\underbrace{E(D \rightarrow D') - E(D \rightarrow D)}_{\text{cross-dataset performance gap}} = \underbrace{E(D' \rightarrow D') - E(D \rightarrow D)}_{\text{hardness component}} + \underbrace{E(D \rightarrow D') - E(D' \rightarrow D')}_{\text{difference component}}. \quad (\text{A.1})$$

The hardness component asks whether D' is harder even for a detector developed on D' . The difference component asks how much additional EER comes from developing on D rather than on D' . In the tables below, this second quantity is called the mismatch component.

Table A.3: Within-dataset reference results for the three architectures (EER).

Model	ASVspoof 5	FakeAVCeleb
RawNet2	0.378	0.003
LCNN	0.383	0.002
Wav2Vec-AASIST	0.310	0.007

Table A.4: Cross-dataset EER and the corresponding mismatch component. Each “mismatch” column subtracts the within-dataset EER of the evaluation corpus reported in Table A.3.

Model	FAV→ASV5	Mismatch	ASV5→FAV	Mismatch
RawNet2	0.456	0.078	0.348	0.345
LCNN	0.385	0.002	0.233	0.231
Wav2Vec-AASIST	0.391	0.081	0.405	0.398

The gap is strongly directional. This is an observational comparison between separately trained models and changing test distributions; it motivates the controlled source–label intervention but does not identify a cause.

A.3. Exact seeded-replication results

Table A.5: Replication on SpeechFake-BD (LibriTTS-excluded) × ASVspoof 5: EER, mean ± sample standard deviation over seeds 41, 42, and 43.

r	Genuine SpeechFake / deepfake ASV5			Genuine ASV5 / deepfake SpeechFake		
	RawNet2	LCNN	Wav2Vec	RawNet2	LCNN	Wav2Vec
1.00	0.171 ± 0.056	0.106 ± 0.006	0.064 ± 0.007	0.558 ± 0.036	0.604 ± 0.026	0.706 ± 0.007
0.75	0.197 ± 0.024	0.278 ± 0.026	0.152 ± 0.015	0.462 ± 0.094	0.306 ± 0.047	0.348 ± 0.019
0.50	0.257 ± 0.019	0.358 ± 0.018	0.215 ± 0.023	0.344 ± 0.054	0.199 ± 0.024	0.284 ± 0.017
0.25	0.427 ± 0.041	0.394 ± 0.034	0.339 ± 0.021	0.256 ± 0.010	0.162 ± 0.009	0.230 ± 0.014
0.00	0.808 ± 0.042	0.779 ± 0.008	0.787 ± 0.009	0.172 ± 0.019	0.115 ± 0.020	0.153 ± 0.005

A.4. Possible shortcut sources

A.4.1. Transformation designs

Bandwidth limitation. Each VAD interval is resampled from 16 kHz to 8 kHz and back to 16 kHz, removing usable content above 4 kHz. The processing change is applied during training, validation, and evaluation for 15 seed-42 runs. Because this grid and the primary grid use different batch sizes, exact cross-grid differences are descriptive; the within-grid response is the supported comparison.

Matched RMS processing. Every VAD interval is scaled toward an RMS target of -23 dBFS and limited to a -1 dBFS peak ceiling. The primary-pair grid contains 15 seed-42 runs; the SpeechFake no-LibriTTS grid contains 45 runs at seeds 41, 42, and 43, with a raw, batch-matched counterpart.

High-pass filtering. Two policies apply cascaded biquad filters with cut-offs of 80 and 300 Hz. The 80 Hz condition targets very-low-frequency channel effects; the 300 Hz condition also suppresses substantial pitch-related speech energy. Both are applied during training, validation, and evaluation. Together they contribute 30 seed-42 runs.

In the result grids below, ASV5 denotes ASVspoof 5, FAV denotes FakeAVCeleb, and SF denotes SpeechFake-BD. Fixed test compositions are written as “genuine source / deepfake source” so that the source of each class is explicit.

Table A.6: Endpoint EER on the fixed ASV5 genuine / FAV deepfake test composition after bandwidth limitation. Each row compares separately trained $r = 1$ and $r = 0$ models within the transformed pipeline.

Intervention	Model	EER at $r = 1$	EER at $r = 0$
8 kHz round trip	RawNet2	0.004	0.793
	LCNN	0.004	0.869
	Wav2Vec-AASIST	0.015	0.823

A.4.2. High-pass result grids

Table A.7: RawNet2 EER across four fixed test compositions: baseline and high-pass filtering at 80 Hz and 300 Hz; single seed.

Fixed test composition	Variant	Mixing ratio r				
		1.00	0.75	0.50	0.25	0.00
ASV5 genuine / FAV deepfake	Base	0.003	0.018	0.008	0.034	0.814
	80 Hz	0.003	0.004	0.011	0.016	0.819
	300 Hz	0.004	0.007	0.013	0.018	0.809
FAV genuine / ASV5 deepfake	Base	0.853	0.465	0.461	0.452	0.097
	80 Hz	0.822	0.576	0.445	0.306	0.129
	300 Hz	0.819	0.511	0.361	0.319	0.113
ASV5 genuine / ASV5 deepfake	Base	0.412	0.385	0.379	0.400	0.509
	80 Hz	0.390	0.346	0.361	0.385	0.532
	300 Hz	0.420	0.346	0.333	0.358	0.518
FAV genuine / FAV deepfake	Base	0.063	0.018	0.006	0.023	0.310
	80 Hz	0.068	0.007	0.010	0.012	0.360
	300 Hz	0.066	0.008	0.013	0.020	0.352

Table A.8: LCNN EER across four fixed test compositions: baseline and high-pass filtering at 80 Hz and 300 Hz; single seed.

Fixed test composition	Variant	Mixing ratio r				
		1.00	0.75	0.50	0.25	0.00
ASV5 genuine / FAV deepfake	Base	0.004	0.008	0.010	0.026	0.873
	80 Hz	0.006	0.041	0.027	0.051	0.862
	300 Hz	0.004	0.025	0.032	0.111	0.864
FAV genuine / ASV5 deepfake	Base	0.857	0.418	0.421	0.374	0.055
	80 Hz	0.848	0.391	0.411	0.361	0.061
	300 Hz	0.878	0.359	0.417	0.342	0.066
ASV5 genuine / ASV5 deepfake	Base	0.460	0.361	0.384	0.385	0.526
	80 Hz	0.463	0.405	0.397	0.413	0.518
	300 Hz	0.457	0.389	0.403	0.406	0.528
FAV genuine / FAV deepfake	Base	0.082	0.005	0.005	0.010	0.325
	80 Hz	0.054	0.028	0.021	0.014	0.267
	300 Hz	0.050	0.018	0.018	0.037	0.266

Table A.9: Wav2Vec-AASIST EER across four fixed test compositions: baseline and high-pass filtering at 80 Hz and 300 Hz; single seed.

Fixed test composition	Variant	Mixing ratio r				
		1.00	0.75	0.50	0.25	0.00
ASV5 genuine / FAV deepfake	Base	0.028	0.048	0.059	0.103	0.781
	80 Hz	0.025	0.038	0.062	0.088	0.800
	300 Hz	0.030	0.042	0.057	0.084	0.776
FAV genuine / ASV5 deepfake	Base	0.686	0.289	0.209	0.191	0.140
	80 Hz	0.772	0.286	0.228	0.193	0.127
	300 Hz	0.733	0.273	0.200	0.188	0.125
ASV5 genuine / ASV5 deepfake	Base	0.335	0.298	0.280	0.303	0.562
	80 Hz	0.373	0.295	0.301	0.298	0.544
	300 Hz	0.382	0.295	0.282	0.302	0.539
FAV genuine / FAV deepfake	Base	0.191	0.035	0.033	0.050	0.311
	80 Hz	0.210	0.037	0.038	0.042	0.295
	300 Hz	0.193	0.034	0.030	0.037	0.298

A.4.3. Bandwidth-limitation grids

Table A.10: RawNet2 EER across four fixed test compositions: baseline and bandwidth limitation (BW); single seed with matched training/evaluation transforms within each grid.

Fixed test composition	Variant	$r=1.00$	0.75	0.50	0.25	0.00
ASV5 genuine / FAV deepfake	Base	0.003	0.018	0.008	0.034	0.814
	BW	0.004	0.008	0.009	0.018	0.793
FAV genuine / ASV5 deepfake	Base	0.853	0.465	0.461	0.452	0.097
	BW	0.844	0.428	0.379	0.287	0.094
ASV5 genuine / ASV5 deepfake	Base	0.412	0.385	0.379	0.400	0.509
	BW	0.410	0.385	0.359	0.397	0.534
FAV genuine / FAV deepfake	Base	0.063	0.018	0.006	0.023	0.310
	BW	0.057	0.009	0.005	0.012	0.265

Table A.11: LCNN EER across four fixed test compositions: baseline and bandwidth limitation (BW); single seed with matched training/evaluation transforms within each grid.

Fixed test composition	Variant	$r=1.00$	0.75	0.50	0.25	0.00
ASV5 genuine / FAV deepfake	Base	0.004	0.008	0.010	0.026	0.873
	BW	0.004	0.011	0.024	0.038	0.869
FAV genuine / ASV5 deepfake	Base	0.857	0.418	0.421	0.374	0.055
	BW	0.889	0.394	0.431	0.338	0.064
ASV5 genuine / ASV5 deepfake	Base	0.460	0.361	0.384	0.385	0.526
	BW	0.443	0.371	0.372	0.369	0.516
FAV genuine / FAV deepfake	Base	0.082	0.005	0.005	0.010	0.325
	BW	0.130	0.009	0.022	0.015	0.266

Table A.12: Wav2Vec-AASIST EER across four fixed test compositions: baseline and bandwidth limitation (BW); single seed with matched training/evaluation transforms within each grid.

Fixed test composition	Variant	$r=1.00$	0.75	0.50	0.25	0.00
ASV5 genuine / FAV deepfake	Base	0.028	0.048	0.059	0.103	0.781
	BW	0.015	0.050	0.042	0.072	0.823
FAV genuine / ASV5 deepfake	Base	0.686	0.289	0.209	0.191	0.140
	BW	0.749	0.334	0.228	0.210	0.117
ASV5 genuine / ASV5 deepfake	Base	0.335	0.298	0.280	0.303	0.562
	BW	0.342	0.306	0.282	0.299	0.561
FAV genuine / FAV deepfake	Base	0.191	0.035	0.033	0.050	0.311
	BW	0.199	0.056	0.026	0.035	0.278

A.4.4. Matched RMS-processing grid

Table A.13: Complete matched loudness-normalisation comparison on SpeechFake-BD (LibriTTS-excluded) $\times$ ASVspoof 5. Values are EER mean $\pm$ sample sd over seeds 41/42/43. Raw and RMS-normalised columns use separate checkpoints trained, validated, and evaluated under their respective pipelines. Paired Δ is RMS-normalised minus raw, computed within seed before aggregation. SF denotes SpeechFake-BD, ASV5 denotes ASVspoof 5, and each fixed test composition lists the genuine source before the deepfake source.

Model	Fixed test composition	r	Raw	RMS-normalised	Paired Δ
RawNet2	SF genuine / ASV5 deepfake	1.00	0.1708 $\pm$ 0.0560	0.0871 $\pm$ 0.0202	-0.0837 $\pm$ 0.0548
RawNet2	SF genuine / ASV5 deepfake	0.75	0.1967 $\pm$ 0.0241	0.2613 $\pm$ 0.0154	+0.0646 $\pm$ 0.0157
RawNet2	SF genuine / ASV5 deepfake	0.50	0.2568 $\pm$ 0.0192	0.3193 $\pm$ 0.0466	+0.0624 $\pm$ 0.0538
RawNet2	SF genuine / ASV5 deepfake	0.25	0.4267 $\pm$ 0.0406	0.4303 $\pm$ 0.0244	+0.0036 $\pm$ 0.0507
RawNet2	SF genuine / ASV5 deepfake	0.00	0.8079 $\pm$ 0.0418	0.8942 $\pm$ 0.0094	+0.0863 $\pm$ 0.0445
RawNet2	ASV5 genuine / SF deepfake	1.00	0.5578 $\pm$ 0.0363	0.6706 $\pm$ 0.0172	+0.1128 $\pm$ 0.0232
RawNet2	ASV5 genuine / SF deepfake	0.75	0.4623 $\pm$ 0.0942	0.2427 $\pm$ 0.0233	-0.2196 $\pm$ 0.0841
RawNet2	ASV5 genuine / SF deepfake	0.50	0.3441 $\pm$ 0.0545	0.1913 $\pm$ 0.0260	-0.1528 $\pm$ 0.0795
RawNet2	ASV5 genuine / SF deepfake	0.25	0.2557 $\pm$ 0.0099	0.1311 $\pm$ 0.0063	-0.1247 $\pm$ 0.0116
RawNet2	ASV5 genuine / SF deepfake	0.00	0.1722 $\pm$ 0.0192	0.0852 $\pm$ 0.0034	-0.0870 $\pm$ 0.0176
LCNN	SF genuine / ASV5 deepfake	1.00	0.1060 $\pm$ 0.0062	0.1241 $\pm$ 0.0444	+0.0181 $\pm$ 0.0489
LCNN	SF genuine / ASV5 deepfake	0.75	0.2779 $\pm$ 0.0258	0.3560 $\pm$ 0.0166	+0.0781 $\pm$ 0.0278
LCNN	SF genuine / ASV5 deepfake	0.50	0.3581 $\pm$ 0.0176	0.3634 $\pm$ 0.0280	+0.0053 $\pm$ 0.0143
LCNN	SF genuine / ASV5 deepfake	0.25	0.3939 $\pm$ 0.0338	0.4182 $\pm$ 0.0078	+0.0243 $\pm$ 0.0350
LCNN	SF genuine / ASV5 deepfake	0.00	0.7794 $\pm$ 0.0080	0.8277 $\pm$ 0.0179	+0.0483 $\pm$ 0.0185
LCNN	ASV5 genuine / SF deepfake	1.00	0.6041 $\pm$ 0.0263	0.6147 $\pm$ 0.0308	+0.0105 $\pm$ 0.0474
LCNN	ASV5 genuine / SF deepfake	0.75	0.3060 $\pm$ 0.0473	0.2580 $\pm$ 0.0471	-0.0480 $\pm$ 0.0866
LCNN	ASV5 genuine / SF deepfake	0.50	0.1991 $\pm$ 0.0239	0.1886 $\pm$ 0.0140	-0.0105 $\pm$ 0.0223
LCNN	ASV5 genuine / SF deepfake	0.25	0.1619 $\pm$ 0.0093	0.1564 $\pm$ 0.0324	-0.0054 $\pm$ 0.0417
LCNN	ASV5 genuine / SF deepfake	0.00	0.1146 $\pm$ 0.0202	0.0647 $\pm$ 0.0075	-0.0499 $\pm$ 0.0140
Wav2Vec-AASIST	SF genuine / ASV5 deepfake	1.00	0.0639 $\pm$ 0.0070	0.0820 $\pm$ 0.0102	+0.0181 $\pm$ 0.0065
Wav2Vec-AASIST	SF genuine / ASV5 deepfake	0.75	0.1523 $\pm$ 0.0147	0.1670 $\pm$ 0.0138	+0.0147 $\pm$ 0.0145
Wav2Vec-AASIST	SF genuine / ASV5 deepfake	0.50	0.2148 $\pm$ 0.0226	0.2256 $\pm$ 0.0154	+0.0108 $\pm$ 0.0169
Wav2Vec-AASIST	SF genuine / ASV5 deepfake	0.25	0.3393 $\pm$ 0.0209	0.3454 $\pm$ 0.0124	+0.0061 $\pm$ 0.0092
Wav2Vec-AASIST	SF genuine / ASV5 deepfake	0.00	0.7874 $\pm$ 0.0092	0.7886 $\pm$ 0.0150	+0.0013 $\pm$ 0.0066
Wav2Vec-AASIST	ASV5 genuine / SF deepfake	1.00	0.7056 $\pm$ 0.0068	0.7041 $\pm$ 0.0149	-0.0015 $\pm$ 0.0122
Wav2Vec-AASIST	ASV5 genuine / SF deepfake	0.75	0.3482 $\pm$ 0.0194	0.3394 $\pm$ 0.0180	-0.0088 $\pm$ 0.0057
Wav2Vec-AASIST	ASV5 genuine / SF deepfake	0.50	0.2837 $\pm$ 0.0172	0.2641 $\pm$ 0.0111	-0.0196 $\pm$ 0.0169
Wav2Vec-AASIST	ASV5 genuine / SF deepfake	0.25	0.2297 $\pm$ 0.0136	0.2136 $\pm$ 0.0076	-0.0161 $\pm$ 0.0114
Wav2Vec-AASIST	ASV5 genuine / SF deepfake	0.00	0.1528 $\pm$ 0.0053	0.1541 $\pm$ 0.0124	+0.0013 $\pm$ 0.0071

B

Use of generative AI

Generative AI tools, including ChatGPT, OpenAI Codex, and Claude Code, were used as assistants during the research and preparation of this thesis. They supported programming, code inspection and debugging, the formatting and refinement of draft text, consistency checks, and the preparation or refinement of selected figures. They were not treated as evidence or as substitutes for the underlying research outputs and cited sources.

The research questions, experimental design, analytical decisions, interpretation, and conclusions are the author's own. All AI-assisted text, code, and visual material was reviewed, revised where necessary, and checked against the relevant code, stored outputs, and sources. The author retains full intellectual and academic responsibility for the thesis.