

NEAR EAST UNIVERSITY
INSTITUTE OF GRADUATE STUDIES AND RESEARCH
DEPARTMENT OF ARTIFICIAL INTELLIGENCE ENGINEERING

**Accent Robustness and Channel Compression
in Audio Deepfake Detection:
A Four-Cell Interaction Study**

Master's Thesis — Progress Report

Prepared by
Kudomo-Abasi Akpan Ekpo
Student No. 20254713

Supervisor
Prof. Dr. Fadi Al-Turjman

Nicosia, October 2026

DECLARATION

I hereby declare that this document is my own work, that every source I have used or quoted has been acknowledged, and that the results reported here come from experiments I ran and analyses I performed. All corpora used are public datasets and are cited. This document is a progress report submitted to my supervisor; it is not a final thesis and has not been submitted for assessment elsewhere.

Name: Kudomo-Abasi Akpan Ekpo

Signature:

Student No.: 20254713

Date:

ABSTRACT

Audio deepfake detectors are evaluated overwhelmingly on clean, largely monolingual speech, while their deployment condition is a telephone call. Two robustness failures have been characterised separately: detectors degrade on shifted accents and languages, and they degrade when speech passes through a codec. This thesis asks the question that neither literature answers: does the accent robustness a detector has on a clean channel survive the telephone channel? A four-cell design crosses accent shift with channel shift on identical utterances, defines an interaction term, and fixes a decision rule before any run. A speaker-level cluster bootstrap gives every contrast a confidence interval.

The design was run at full scale on a paired MLAAD construction: 2,400 evaluated utterances per cell across twelve matched generators, three seeds and four detectors, on a rented RTX 3090. Both main effects are large and reproduce. G.711 A-law at 8 kHz costs 30.8 ± 6.2 equal-error-rate points on matched-accent speech, and an unseen language costs 5.4 ± 2.5 points on a clean channel. For the pre-registered primary detector the rule returns NO-GO on all three seeds, with a mean interaction of -9.53 ± 4.78 points and two of three bootstrap intervals excluding zero; the direction is the opposite of the motivating hypothesis. The effect is not super-additive: the codec absorbs part of the accent penalty, because a 30.8-point channel cost leaves the detector too little operating range for the accent cue to still be worth destroying. LCNN, whose channel cost is 20.7 points, puts the interaction at $+0.13 \pm 5.22$. The sign of the interaction is therefore a property of the detector’s remaining headroom, not of the corpus alone.

A pre-registered secondary gate asks whether accent-aware training buys the shifted-accent cell back. Three arms and three seeds each answer it: the treated arm reaches 4.30 ± 3.75 EER against 8.81 ± 3.39 for the control and 9.19 ± 2.58 for a size-matched control. The gain is 4.51 points against a 3.0-point threshold, every seed clears it alone, and the size-matched control gains 0.38 — so the gain is the accent data, not the extra audio.

The detector comparison, which the thesis is required to report, places LCNN first on the clean matched cell (1.38% EER against MelCNN’s 3.09%), but its seed spread is the widest of the three trained detectors, and its interaction interval spans zero. MelCNN, the substitute primary, is the more stable instrument and carries the primary verdict. A local demonstration service scores twelve real utterances with all three detectors on both channels; it reproduces the direction-dependent effect on single utterances and is documented with screenshots in Chapter 7.

Keywords: audio deepfake detection, anti-spoofing, accent robustness, codec robustness, telephony, interaction effect, equal error rate, pre-registration, MLAAD

CONTENTS

Declaration	i
Abstract	ii
List of Tables	v
List of Figures	viii
1 Introduction	1
1.1 Motivation	1
1.2 Problem statement	1
1.3 Research questions	2
1.4 Hypotheses and the decision rule	2
1.5 Contributions	3
1.6 What this thesis is not	3
1.7 Structure of the thesis	3
2 Background and Related Work	4
2.1 Audio deepfakes and the detection task	4
2.2 Detector architectures	4
2.3 Accent and language shift	5
2.4 Channel and codec robustness	5
2.5 Metrics in anti-spoofing	5
2.6 Statistical treatment of clustered utterances	6
2.7 Pre-registration in empirical machine learning	6
2.8 The gap, stated precisely	6
3 Conceptual Framework	8
3.1 The four-cell design	8
3.2 The contrasts	8
3.3 Why the interaction is that particular difference	9
3.4 Why equal error rate	9
3.5 The null and what it would license	9
3.6 The pre-registered decision rule	9
3.7 A secondary, pre-registered gate	10
4 Methodology	11

4.1	Corpora	11
4.2	The paired MLAAD construction	11
4.3	Two accent operationalisations	12
4.4	Splits	12
4.5	Channel simulation	13
4.6	Detectors	13
4.7	Training protocol	14
4.8	Metrics	14
4.9	Speaker-level cluster bootstrap	14
4.10	Instrument checks	15
4.11	Software and reproducibility	15
5	Experimental Setup	16
5.1	Compute	16
5.2	Hyperparameters	16
5.3	Run matrix	17
5.4	Deviations from the pre-registration	17
6	Results	18
6.1	Instrument check	18
6.2	Which detector performed best	18
6.3	Four-cell evaluation	20
6.4	A second metric agrees	21
6.5	Gate outcome	22
6.6	The secondary gate: does training on the shifted accent help?	23
6.7	Stability across architectures and codecs	25
7	Deployment and Testing	27
7.1	Why a demo is part of the argument	27
7.2	What was built	27
7.3	The defect the demo found first	27
7.4	Measured results	28
7.5	Testing	32
7.6	Screenshots	32
7.7	What the deployment is not	32
8	Discussion	33
8.1	The claims, and where each one is checked	33
8.2	What the gate said	34
8.3	How much weight it carries	34

8.4	The main effects are the robust finding	34
8.5	Implication for the robustness literature	34
8.6	What the gate pass adds	35
8.7	Which detector, and for what	35
9	Limitations and Threats to Validity	36
10	Conclusion and Future Work	38
10.1	Conclusion	38
10.2	Future work	38
A	The complete metric matrix	40
B	The bootstrap intervals, run by run	42
C	Every run in the experiment	43
D	The smaller-corpus ablation	44
E	What the test suite guarantees	45
F	The corpus, counted	47
G	The instrument check on ASVspoof 2019 LA	49
H	Reproducibility	50
I	Recorded deviations	51
J	Completion record	52
	References	53

LIST OF TABLES

2.1	What each line varies. No prior study holds the other axis at more than one value.	7
3.1	The four cells. Columns differ only by transcoding; rows differ only by accent group.	8
4.1	Corpora. Datasets without a formal publication are cited by name. Only MLAAD supplies both classes in this thesis’s main run; the others are used for the accent operationalisation and are listed because they are part of the protocol.	11
4.2	The full-scale split. Test cells hold 1,200 utterances per class; the training set is capped at 4,000 per class.	13
4.3	The four detectors. AASIST is pre-registered as primary and is reported as a failed instrument; MelCNN is the substitute primary.	13
5.1	Training and evaluation configuration, identical across detectors except for the learning rate.	16
5.2	The run matrix. Rows are never averaged together; the archived partial gate is retained only as a record of a defect, and the demo is a demonstration rather than an estimate.	17
6.1	Seed-mean performance on the clean matched cell (A) and the clean shifted cell (C). Lower EER and min t-DCF are better; higher AUC is better.	18
6.2	Every full-scale run, individually. The starred rows are the detector with the best seed-mean clean-cell EER. No run was discarded.	19
6.3	EER (%) in the four cells and the three contrasts on the paired full-scale corpus. Each cell shows the mean over seeds with the sample sd beneath it; the last column is the envelope of the per-seed 95% bootstrap intervals for Δ_{int} . AASIST ran seed 0 only, so its sd is zero by construction. G.711 A-law, 1,200 utterances per class per cell.	20
6.4	The three gate arms on the complete corpus. Lower cell C is better. All arms share the same held-out German books.	24
6.5	The pilot’s codec sweep: one fixed MLAAD-tiny checkpoint evaluated in all four cells under each codec. One seed per codec, so no dispersion is shown. Generated by <code>paper/make_tables.py</code>	25
7.1	The demonstration service.	27

7.2	The twelve demonstrated utterances: spoof probability on the clean channel, and after G.711 A-law, for each detector. The set is balanced across label and language; the fakes should score high and the real recordings low, so a score that crosses 0.5 between the two columns is a verdict the channel flipped.	28
7.3	The test layers. The first four are pipeline tests; the last two are the deployment's own.	29
8.1	Each headline claim, the table or figure that carries it, and the observation that would overturn it. The last column is stated before the reader asks, because a claim whose falsifier cannot be named is a preference.	33
A.1	Every cell, every detector, every metric, mean and sample sd over seeds. AASIST ran one seed, so its sd is zero by construction. AUC is higher-is-better; EER and min t-DCF are lower-is-better. Generated by <code>paper/make_tables.py</code> .	40
B.1	The speaker-cluster bootstrap interval for every run on the paired corpus. The resampling unit is the matched book, so the interval's width is set by the cluster count, not by the number of utterances. Generated by <code>paper/make_tables.py</code> .	42
C.1	Every training run that produced an evaluation file, with its four cell EERs and its three contrasts. Nothing was discarded; the corpus tag says how each row is read.	43
D.1	The MLAAD-tiny pilot, G.711 A-law, cell EER in percent. <i>mean</i> is the mean of the four cell values; <i>spread</i> is their sd; <i>worst</i> is the largest cell. Generated by <code>paper/make_tables.py</code> .	44
E.1	The suite, by module. Counts are the tests in each file at the time of writing; the table is descriptive, not generated, because the claim each test protects is a sentence a script cannot produce.	45
F.1	The corpus a representative run trained on, counted from its own manifest. The German bona fide class is one fifth the size of the English one; that asymmetry is what the size-matched control arm exists to hold constant. Generated by <code>paper/make_tables.py</code> .	47
F.2	The twelve generator families and their share of the spoof class. The balance is exact: 2,000 utterances each, half English and half German within each family, so no family can drive a cell. Generated by <code>paper/make_tables.py</code> .	47

G.1	The ASVspoof 2019 LA instrument check, per split. EER in percent. The published reference for AASIST on this benchmark is 0.83% evaluation EER [Jung et al., 2022]; every detector here fails it. Generated by paper/make_tables.py.	49
G.2	Per-attack behaviour on the evaluation split. An attack is 'detected' when its mean spoof score exceeds the decision threshold. A12 and A15 evade both the AASIST and the LCNN checkpoints, which are the two that also miss the published reference by the widest margin. Generated by paper/make_tables.py.	49

LIST OF FIGURES

6.1	EER in each of the four cells, mean $\pm$ sd over seeds, by detector. The spread between detectors is larger than the spread between cells for three of the four.	19
6.2	The three contrasts with their speaker-bootstrap intervals, by detector. Δ_{int} is the contrast the thesis is about; the leftmost panel is the channel effect that dominates it.	21
6.3	AUC per cell. The ordering the metric produces matches the EER ordering for every detector, so the thesis's conclusions do not depend on the choice of metric.	21
6.4	EER against AUC for all sixteen detector–cell pairs. One monotone relation, so the chapter's ordering does not depend on which of the two metrics is read. Appendix A gives the underlying numbers.	22
6.5	Per-seed Δ_{int} . The sign is consistent for MelCNN and the magnitude is not — which is why the verdict is stated on the sign and the interval rather than on the mean alone.	23
6.6	The MLAAD-tiny pilot's four cells. Reported as a separate experiment and never averaged with the full-scale run; it is what motivated the full-scale design.	25
6.7	Channel effect against interaction, one point per run. The detector with the smallest channel effect (RawNet2, left) sits above zero; the one that loses most of its range to the codec (MelCNN, right) sits below it on every seed; LCNN straddles zero in between. AASIST, near chance in every cell, sits flat at the origin because it has no range to lose either way. The points fall, but they do not fall on a line — the ceiling reading is a tendency in this data, not a law.	26
6.8	Validation EER per epoch. The curves are what the instrument check reads: a detector whose validation EER plateaus near chance cannot carry an interaction claim, whatever its test cells later report.	26
7.1	The service on the clean channel. Each row carries the true label, the three spoof probabilities and the audio that was scored.	29
7.2	The same utterances through G.711 A-law. The rows that changed colour are the ones whose verdict the channel flipped — the phenomenon the thesis is about, visible without reading a table.	30
7.3	The bottom of the list, captured to show that the layout does not overflow at a 592-pixel viewport (<code>scrollWidth - clientWidth = 0</code>).	31

CHAPTER 1

INTRODUCTION

1.1 Motivation

A spoofing detector that achieves a low equal error rate on a benchmark has not thereby been shown to work on a phone call. The deployment setting for audio deepfake detection is overwhelmingly telephonic: fraud, vishing and impersonation arrive through a codec, often from a speaker whose accent the detector’s training data did not represent.

Two research lines have grown up around this gap, and each has produced real evidence. The first studies *speaker variation*: detectors degrade when the test language or accent differs from training [Liu et al., 2024, Kim et al., 2026, ArFake authors, 2026], and the degradation is large enough that it has been framed as a fairness problem as well as an accuracy problem. The second studies *channel variation*: detectors degrade under codec compression, packet loss and real telephony [Evans et al., 2023, Shi et al., 2025, Xu et al., 2025, Klein et al., 2026], with G.711 and real PSTN transmission identified as the hardest conditions.

The two lines share an assumption that has not been tested. Accent-shift experiments are run on clean or lightly degraded audio; channel-shift experiments use read speech without an accent design. If the underlying mechanisms are independent, the assumption is harmless and the two literatures can be read side by side. If they are not independent, then a robustness claim made on one axis may be an artefact of holding the other axis at its easiest value — and the easier value is exactly the one deployment never provides.

1.2 Problem statement

Let a detector be trained on speech from one accent group and one channel. In deployment it meets speech from another accent group, arriving through a telephone codec. Two questions follow, and only the first is asked in the current literature:

1. How much does the accent shift cost? (Asked, repeatedly.)
2. Does the cost of the accent shift depend on the channel the speech arrives through? (Not asked.)

If the answer to the second question is “no”, the field’s existing practice is vindicated: robustness on one axis can be measured with the other held fixed, and the results compose. If the answer is “yes”, then every single-axis robustness number is conditional on a condition

its paper does not state, and the size of the condition is the size of the interaction.

1.3 Research questions

The thesis is organised around four questions.

RQ1

Does accent shift and channel shift interact in audio deepfake detection, and if so, in which direction and by how much?

RQ2

Does the answer depend on the detector architecture, or is it a property of the corpus?

RQ3

Does accent-aware training recover accuracy on the shifted accent, and if so, is the gain attributable to the accent data rather than to the extra training audio?

RQ4

Which of the trained detectors performs best on this corpus under the metrics that matter for deployment, and how stable is that ranking?

1.4 Hypotheses and the decision rule

Three outcomes were specified before any data were downloaded, because a four-cell design can produce an interaction in either direction and a post-hoc reading would fit any of them.

H1 — super-additive.

The codec costs more on shifted-accent speech than on matched-accent speech. Mechanism: the detector's accent adaptation relies on high-frequency artefacts that band-limiting removes, so the two degradations compound.

H0 — additive.

The two shifts are independent, each costing what it costs alone. This is the assumption the two literatures already rely on.

H2 — sub-additive.

The codec costs *less* on shifted-accent speech, because it saturates the detector: once the channel has taken most of the operating range, the accent cue has little left to cost.

The pre-registered rule maps the measured interaction onto these three worlds (Section 3.5). It was written before any run and was not altered afterwards; the deviations that did occur are logged in `docs/pre-registration.md` and restated in Section 5.4.

1.5 Contributions

1. A four-cell design that crosses accent shift with channel shift on *identical* utterances, so that the interaction is a within-item contrast rather than a between-corpus comparison.
2. A measurement of that interaction at full scale — 2,400 utterances per cell, four detectors, three seeds — with speaker-level bootstrap intervals and a pre-registered verdict.
3. The finding that the interaction’s sign is governed by the detector’s remaining headroom after the channel, supported by three trained detectors whose channel costs differ by a factor of nine.
4. A pre-registered gate on accent-aware training, with an equal-size control that separates the effect of the accent data from the effect of extra training audio.
5. A working demonstration service that scores real utterances on both channels, so the claim can be watched rather than inferred, together with the tests that keep the demo’s preprocessing identical to the evaluation pipeline’s.

1.6 What this thesis is not

It does not propose a new detector. It does not claim a state-of-the-art equal error rate, and it does not reproduce a published ASVspoof number. The detectors it uses are published architectures, included because agreement across architecture families is evidence that a result belongs to the data rather than to one model. Where a result is negative, the thesis reports it as negative: the interaction the design was built to test came out against the motivating hypothesis, and that outcome is the one the thesis argues for.

1.7 Structure of the thesis

Chapter 2 reviews the two literatures and states the gap. Chapter 3 develops the four-cell design, derives the interaction term, and fixes the decision rule. Chapter 4 describes the corpus construction, the preprocessing, the detectors and the statistical machinery. Chapter 5 records the experimental configuration and the deviations from the pre-registration. Chapter 6 reports the results, including the detector comparison and the gate. Chapter 7 documents the deployed demonstration service and its tests. Chapters 8–10 discuss, limit and conclude.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Audio deepfakes and the detection task

Synthetic speech has been cheap to produce for a decade and is now cheap to produce *well*. The detection task is to decide, from a waveform, whether the utterance was produced by a human or by a synthesis system, and the standard formulation is a binary classification problem with two error directions that carry different costs: accepting a fake (a security failure) and rejecting a real speaker (a usability failure).

The field’s benchmarks have evolved in step with the generators. ASVspoof 2019 established the LA (logical access) and PA (physical access) tracks and a protocol in which the same utterances are replayed through codecs and rooms; the 2021 edition added a deliberate domain shift, applying telephony and VoIP transmission and multiple codecs to both bona fide and spoofed speech with no new training data [Evans et al., 2023]. In-the-Wild collected real manipulated audio from social media rather than from a controlled generator list [Müller et al., 2022], and the MLAAD project assembled a large multilingual synthetic corpus across dozens of TTS and voice-conversion families [Müller et al., 2024].

2.2 Detector architectures

Countermeasure systems have moved from hand-crafted features to end-to-end waveform models, and three architecture families dominate the literature.

LCNN. The Light CNN introduced max-feature-map convolutional blocks, which act as a cheap competitive activation and were shown to work well on spectrogram inputs [Lavren-tyeva et al., 2017]. It remains the standard low-parameter baseline in anti-spoofing evaluations.

RawNet2. A sinc front-end that learns its own band-pass filters, followed by residual convolution over the raw waveform [Tak et al., 2021]. It removes the hand-designed feature entirely and is the reference for “raw audio” detection.

AASIST. Graph attention over spectro-temporal representations, with a heterogeneous stacking of graph layers that lets the model attend across both time and frequency [Jung et al., 2022]. It is the strongest published architecture on ASVspoof 2019 LA and the natural choice for a primary detector.

MelCNN, the substitute. A compact log-mel convolutional network of three stages and 274k parameters, used here because the pre-registered primary failed its instrument check on this corpus (Section 6.1). It is deliberately modest: the thesis needs an instrument that separates the two classes on clean in-domain speech, not one that maximises a benchmark.

2.3 Accent and language shift

Language mismatch was quantified as a distinct generalisation axis by Liu et al. [2024], who showed that English-trained countermeasures lose accuracy on other languages and proposed synthesising accented English to expand monolingual training. A benchmark study subsequently reported a measurable bias against non-native (L2) English accents and showed that accent-balanced fine-tuning reduces the disparity. Kim et al. [2026] remove a language-correlated direction from self-supervised features and report lower EER on unseen languages, and a companion line disentangles accent from language in frozen representations. ArFake authors [2026] provide a multi-dialect Arabic benchmark with leave-one-dialect-out and leave-one-generator-out protocols, which is the cleanest available statement of the accent-versus-generator confound.

Two things are common to this line. The first is that the evaluation audio is clean: codec conditions are not part of the protocol, so the accent-robustness claim is conditional on a channel that is never stated. The second is that “accent” is operationalised differently from paper to paper — as a language, as a dialect, as an L1 background, or as a continuous embedding direction — which matters here because this thesis’s primary operationalisation is language shift (Section 4.3).

2.4 Channel and codec robustness

Shi et al. [2025] built a codec $\times$ packet-loss test matrix and measured up to 5.30% EER degradation for clean-trained baselines. Xu et al. [2025] simulated telephone and messaging links and cut EER from 19.16% to 6.48% by training on the simulated channel. Klein et al. [2026] show that G.711 is harder than Opus or bandwidth reduction alone, and that real PSTN is harder still.

What this line has in common is that its speech is read, largely monolingual corpus audio. The channel-robustness claim is therefore conditional on an accent condition that is never stated — the mirror image of the gap in the accent literature.

2.5 Metrics in anti-spoofing

Equal error rate. The threshold at which the false-acceptance and false-rejection rates are equal, reported in percent. It is threshold-free, it is the conventional unit of the field, and it

is the primary metric here.

Area under the ROC curve. A ranking metric that is insensitive to the operating point and therefore reports how well a detector orders utterances independent of where its threshold sits. It is reported as a secondary metric.

Minimum tandem detection cost function. A weighted cost that prices the two error directions differently and is normalised against a fixed baseline [Kinnunen et al., 2018]. It is the metric that most closely reflects deployment, and it is reported as a secondary metric for every cell.

None of the three is a safety metric in the sense the deployment cares about: all three are computed on a fixed corpus, and all three can be excellent while the detector fails on a population the corpus did not represent. That is the gap this thesis’s robustness design addresses.

2.6 Statistical treatment of clustered utterances

Utterances from one speaker share a voice, a microphone and a room. The standard error computed by treating utterances as independent observations is therefore too small, often by a large factor, because the effective sample size is the number of speakers rather than the number of utterances. The cluster bootstrap [Efron and Tibshirani, 1993] resamples at the speaker level and recomputes the statistic, which restores the correct coverage. Chapter 3 defines the resampling unit for this corpus explicitly, because the choice of what counts as a speaker is a substantive decision here.

2.7 Pre-registration in empirical machine learning

A pre-registered decision rule fixes, before data are seen, what result will count as which outcome. Its purpose is straightforward: a four-cell design has eight sign patterns across its cells and three contrasts, and a researcher who reads that space after the fact can tell a story about any of them. The rule removes the choice from the analysis. Where this thesis reports a result the rule did not anticipate — as it does for the primary detector — it reports the rule’s verdict verbatim and states the unanticipated direction separately, rather than editing the rule to fit.

2.8 The gap, stated precisely

Neither literature crosses the two axes. The interaction is not merely untested; it has a plausible mechanism in both directions. A codec band-limits and quantises, removing the high-

frequency artefacts that many detectors rely on; if accent-aware training teaches a model to use those same artefacts, the gain should collapse under compression (H1). Alternatively, if accent shift and channel shift push the score distribution in the same direction, their effects could be sub-additive, with compression saturating the degradation (H2).

Study	Axis crossed	Channel condition	Accent condition
Liu et al. [2024]	language	clean	shifted
Kim et al. [2026]	language (representation)	clean	shifted
ArFake authors [2026]	dialect	clean	shifted, controlled
Evans et al. [2023]	channel	telephony, VoIP	none stated
Shi et al. [2025]	channel	codec $\times$ packet loss	none stated
Xu et al. [2025]	channel	simulated PSTN/messaging	none stated
Klein et al. [2026]	channel	G.711, Opus, real PSTN	none stated
This thesis	both, crossed	G.711 A-law	unseen language

Table 2.1: What each line varies. No prior study holds the other axis at more than one value.

Table 2.1 states the gap in a line: every study on either side holds the other axis fixed, and none states what it held it at. The four-cell design is the smallest design that can detect whether that fixed value matters.

CHAPTER 3

CONCEPTUAL FRAMEWORK

3.1 The four-cell design

Let $\mathcal{S}$ be a speaker-disjoint test set. Speakers are partitioned into those whose accent group appears in training (matched) and those whose group does not (shifted). Every shifted speaker’s utterances are scored twice: on the original waveform and on its codec-transcoded twin. The same is done for matched speakers. This yields four cells with the property that A and B contain exactly the same utterances, as do C and D :

	clean channel	codec channel
matched accent	A	B
shifted accent	C	D

Table 3.1: The four cells. Columns differ only by transcoding; rows differ only by accent group.

The pairing is what makes the interaction interpretable. A between-corpus design — accent results from one paper, channel results from another — confounds the comparison with everything else that differs between two corpora: generator families, recording chains, label balance and language. Here the only thing that changes between A and B is the codec, because B is A after transcoding; and the only thing that changes between A and C is the accent group.

3.2 The contrasts

Three contrasts carry the analysis, all in equal-error-rate percentage points:

$$\Delta_{\text{acc}} = C - A, \tag{3.1}$$

$$\Delta_{\text{ch}} = B - A, \tag{3.2}$$

$$\Delta_{\text{int}} = (D - B) - (C - A). \tag{3.3}$$

Δ_{acc} is the accent cost on a clean channel, Δ_{ch} the channel cost on a matched accent, and Δ_{int} the difference between the accent cost through the codec and the accent cost on the clean channel.

3.3 Why the interaction is that particular difference

Under an additive model the accent cost is the same number in both channel conditions:

$$D - B = C - A \iff \Delta_{\text{int}} = 0. \quad (3.4)$$

Δ_{int} is therefore exactly the failure of additivity, and its sign says which way the failure goes. A positive value means the codec costs more on shifted-accent speech than on matched-accent speech, which is H1 and the outcome the motivating hypothesis expected. A negative value means the codec costs *less* on shifted-accent speech, which is H2.

There is an equivalent reading that is easier to use when interpreting numbers: Δ_{int} is how much smaller the accent penalty becomes once telephone compression is applied. A detector with $\Delta_{\text{acc}} = 5.4$ points on a clean channel and $\Delta_{\text{int}} = -9.5$ points is a detector whose accent penalty has become -4.2 points through the codec. Whether that is physically sensible depends entirely on what the detector has left: a number that reverses sign is a saturation result, not a claim that the codec makes the detector accent-fair.

3.4 Why equal error rate

The primary metric is the equal error rate, in percent, for three reasons. It is the conventional reporting unit in anti-spoofing, so results are comparable to the literature. It is threshold-free, so it does not require choosing an operating point that would itself be a free parameter. And it is a single number per cell, which the four-cell arithmetic requires: an interaction defined over a curve would be a functional, and the decision rule needs a scalar.

AUC and minimum t-DCF are reported alongside every cell (Section 4.8) but neither gates the decision. The thesis's claim is about a contrast in EER, and the other two metrics are there so a reader can see whether a conclusion depends on the choice of metric. Where they disagree, Chapter 6 says so.

3.5 The null and what it would license

If $\Delta_{\text{int}} \approx 0$, the two shifts are additive and the literature's existing practice is sound: an accent robustness number measured on clean audio is an accent robustness number, and a channel robustness number measured on one accent group is a channel robustness number, and the two compose. That is a strong and useful result, and it is why NO-GO is not a failure outcome. The rule below assigns that reading to $\Delta_{\text{int}} < 2.0$ points.

3.6 The pre-registered decision rule

The rule was fixed before any corpus was downloaded:

- **GO:** $\Delta_{\text{int}} \geq 5.0$ EER points, its 95% CI excludes zero, and it exceeds both $|\Delta_{\text{acc}}|$ and $|\Delta_{\text{ch}}|$.
- **NO-GO:** $\Delta_{\text{int}} < 2.0$ points; the shifts are additive and the thesis reports a robustness benchmark.
- **INCONCLUSIVE:** otherwise. No additional seeds are run to chase the threshold.

The 5.0-point threshold is calibrated to the literature’s effect sizes: 5.30 points for codec-only degradation [Shi et al., 2025] and 15–24 points for cross-domain EER gaps. An interaction smaller than the smallest published channel effect is not a finding worth a thesis.

A known blind spot in the rule. The rule collapses every value below 2.0 points into one NO-GO verdict, and that conflates two different worlds: $\Delta_{\text{int}} \approx 0$, where the shifts are additive, and $\Delta_{\text{int}} \ll 0$, where the codec masks part of the accent degradation. The threshold is one-sided by construction. This was visible when the rule was written and was left as written, because changing a pre-registered rule after seeing the sign is exactly the failure the rule exists to prevent. Chapter 6 reports the verdict verbatim and then reports what the rule does not say.

3.7 A secondary, pre-registered gate

A second question was registered at the same time: does accent-aware training buy back accuracy on the shifted accent, and is the gain attributable to the accent data rather than to the extra audio that comes with it? The gate is defined on the shifted-accent clean cell: the treated arm must beat the control by at least 3.0 EER points. Because a treated arm necessarily trains on more utterances, a third, size-matched arm was registered as a control for training volume. Its role is stated in advance and it is not a second gate.

CHAPTER 4

METHODOLOGY

4.1 Corpora

No audio is recorded by the authors. All corpora are public and are listed with their access terms, because the licence of a corpus is part of what a reader needs in order to reproduce a result.

Role	Corpus	Label	Access
Synthetic, multilingual	MLAAD [Müller et al., 2024]	spoof	gated, CC-BY-NC-4.0
Synthetic, challenge	ASVspoof 5 [Wang et al., 2025]	both	ODC-By 1.0
Synthetic, vocoder	WaveFake [Frank and Schönherr, 2021]	spoof	open
Synthetic, in the wild	In-the-Wild [Müller et al., 2022]	both	open
Bona fide, multilingual	M-AILABS	real	open
Bona fide, accents	VCTK [Veaux et al., 2017]	real	open
Bona fide, emotional	CREMA-D [Cao et al., 2014]	real	open
Bona fide, L2 English	L2-ARCTIC [Zhao et al., 2018]	real	request

Table 4.1: Corpora. Datasets without a formal publication are cited by name. Only MLAAD supplies both classes in this thesis’s main run; the others are used for the accent operationalisation and are listed because they are part of the protocol.

4.2 The paired MLAAD construction

The full-scale corpus is not MLAAD as shipped. It is a paired construction, and the pairing is the thesis’s main data contribution.

MLAAD-tiny ships synthetic speech, not the originals the synthesis was made from. It also ships the *real* audiobook recordings that the fakes were generated from: 7,390 English and German originals. Using those as the real class is what makes the pairing possible.

The fake class is the full MLAAD release, restricted to the twelve generators that appear in *both* languages. Two properties follow:

1. **The label is not confounded with the recording chain.** The fakes are synthesised from audiobook segments; a fake paired against studio speech would be separable by channel alone. Pairing each fake with real audio from the same source material closes that shortcut.
2. **The matched and shifted cells cannot differ by generator family.** Restricting to the twelve generators present in both languages means a language difference cannot be explained by a generator difference.

The pairing was verified on the downloaded data: 79 of 79 English real books and 172 of 172 German real books have fakes. 31,390 records were scanned.

What counts as a speaker. MLAAD’s filenames do not carry a speaker identifier. The finest-grained speaker unit they expose is the audiobook a segment came from, so the book is used as the speaker proxy throughout, including in the bootstrap. Because a fake inherits its original’s book, label and speaker identity are not confounded. The cost of the proxy is stated in Chapter 9: the paired construction adds fakes by the thousand but cannot add real books, so the cluster count is fixed by the real class.

4.3 Two accent operationalisations

MLAAD varies by *language*, not by accent. Two operationalisations are therefore defined and the primary is reported.

Primary: language shift. Matched speakers are those whose language appears in training; shifted speakers are those whose language does not. This is the operationalisation the cross-lingual literature uses, and it is the one the main result rests on: the matched group is English audiobooks held out from training, the shifted group is all German audiobooks, unseen in any form at training time.

Secondary: English accent shift. VCTK’s accent metadata and L2-ARCTIC’s L1 back-grounds support a narrower, better-matched definition of accent. This operationalisation has not been run, and the reason is stated rather than hidden: it requires a different real-speech corpus, and the time went to the full-scale main run and the pre-registered gate. If the two operationalisations disagreed, that disagreement would itself be a finding — it would mean the robustness literature’s use of “accent” is not stable across corpora.

4.4 Splits

The split unit is the speaker proxy, not the utterance, and the test set is fixed at 20% of speakers, seeded at zero, before any model is built. English books are split 80/20 into training

and matched-test speakers. Validation books are carved out of the *training* books, never from the held-out ones, so model selection never sees a test speaker.

Split	books	utterances
training	50	7,795
validation	13	1,200
matched test	16	2,400
shifted test (German)	192	2,400

Table 4.2: The full-scale split. Test cells hold 1,200 utterances per class; the training set is capped at 4,000 per class.

The asymmetry in the last row is the point of the earlier warning: 192 German books supply the same 2,400 utterances that 16 English books supply, because the English side is bound by the number of real books rather than by audio volume.

4.5 Channel simulation

The channel axis is a real transcoding, not a feature perturbation: each file is encoded with the codec, decoded back, and only then shown to the detector. Packet loss, where used, is applied in the sample domain after decoding, because that is where a dropped packet appears in a real call. Both the clean and the codec paths are normalised to mono 16 kHz PCM16, so the only difference between the cells is the channel.

Primary codec: G.711 A-law [ITU-T, 1988]. Secondary codecs: G.711 μ -law and Opus at 16 kHz. The secondary codecs were run at pilot scale only (Section 6.7), and no full-scale claim rests on them.

4.6 Detectors

Detector	Input	Parameters	Role
MelCNN	log-mel, 64 bands	274k	primary (substitute)
LCNN	log-mel spectrogram	1.1M	secondary
RawNet2	raw waveform	24M	secondary
AASIST	spectro-temporal graph	297k	failed instrument check

Table 4.3: The four detectors. AASIST is pre-registered as primary and is reported as a failed instrument; MelCNN is the substitute primary.

The pre-registered primary detector, AASIST [Jung et al., 2022], **failed the instrument check** and is not used for the primary result. Its validation EER plateaued at 30–47% across 20 epochs at two learning rates, while a *linear* classifier on mean log-mel features reached 15.3% on the same held-out books. The fault was in the detector, not the corpus, so

a compact log-mel CNN was substituted. The choice was made on the validation split only, before any test cell was scored, and is logged as a deviation (Section 5.4).

4.7 Training protocol

AdamW, weight decay 10^{-4} , class-balanced sampling, batch size 64, 4-second crops (64,000 samples) at 16 kHz, at most 40 epochs, early stopping on validation EER with patience 8. The primary detector trains at learning rate 10^{-4} ; the secondaries at 10^{-3} . Log-mel features (64 bands, 512-point window, 256-frame hop) are computed from a content-addressed cache; the model standardises each utterance before it enters the network. Seeds 0, 1 and 2 for MelCNN, LCNN and RawNet2; AASIST ran seed 0 only. Every run serialises its exact configuration next to its checkpoint.

Why the preprocessing is centralised. Crop, normalisation and per-utterance standardisation all change the input distribution, and a detector evaluated with a different preprocessing chain is a different experiment. The evaluation path therefore exposes a single scoring function, and every consumer — the four-cell evaluator, the gate runs and the demonstration service of Chapter 7 — calls it. A test asserts that the demo’s scores equal the evaluator’s on the same file.

4.8 Metrics

EER

The threshold at which false-acceptance and false-rejection rates are equal, in percent. Primary metric; gates the decision.

AUC

Area under the ROC curve. Reported per cell; never gates.

min t-DCF

Normalised minimum tandem detection cost function [Kinnunen et al., 2018], with the standard cost model. Reported per cell; never gates.

4.9 Speaker-level cluster bootstrap

Utterances from one speaker share a voice, a microphone and a room. Treating them as independent observations produces confidence intervals that are too narrow, so the resampling unit is the speaker proxy [Efron and Tibshirani, 1993]. For each of 2,000 replicates the procedure resamples matched speakers and shifted speakers with replacement, recomputes all

four cells, and records the three contrasts. The reported interval is the percentile interval of the replicates.

This is also why the gate is defined on the speaker-level interval: the question is whether the effect generalises across speakers, not whether it is present in these particular utterances. The number of clusters the interval can draw on is 16 matched books and 192 German books, and the consequences of that asymmetry are reported in Chapter 6.

4.10 Instrument checks

Before any robustness claim is made, the instrument must be shown to work. Three checks were run:

1. **Dry run.** A synthetic dataset in which the classes are separable by construction, to confirm the pipeline reports near-zero EER when it should. Its output is tagged `"synthetic": true` and is never reported as a result.
2. **Clean-cell sanity.** The substitute detector must reach a low EER on matched books it has never seen. A detector that cannot separate the classes on easy data cannot carry an interaction claim, because a ceiling leaves no room for either shift to express itself.
3. **Reference reproduction.** The ASVspoof 2019 LA check, which the thesis reports honestly as a failure: 10.50% evaluation EER against the 0.83% published reference. The outstanding ASVspoof 5 reproduction is listed in Chapter 9.

4.11 Software and reproducibility

The pipeline is Python with PyTorch. Every evaluation writes one JSON file containing the four cells, the bootstrap interval and the gate decision; the aggregator reads those files and writes the summary tables, so no number in Chapter 6 is transcribed by hand. The thesis's own tables are generated by `paper/make_tables.py` from the same files.

A unit-test suite covers the codec round trip, the metric implementations, the bootstrap's cluster behaviour, the split construction and the demo's preprocessing equivalence. The suite is run before every rented-box job and is listed in Appendix H.

CHAPTER 5

EXPERIMENTAL SETUP

5.1 Compute

Full-scale training and evaluation ran on a rented NVIDIA RTX 3090 (CUDA 12.1, PyTorch 2.4.0), not on the laptop. One detector-seed run takes 20–30 minutes including the four-cell evaluation and codec transcoding, and the ten runs completed in 3.4 GPU-hours for a total rental of about \$0.69. The laptop pilot ran on Apple Metal and took 75 minutes for its three seeds. No run was cut short for time or budget.

The reason for stating the cost is reproducibility, not thrift: a master’s thesis whose full-scale experiment costs under a dollar can be re-run by an examiner, and that is the standard the work should be held to.

5.2 Hyperparameters

Setting	Value
Optimiser	AdamW
Learning rate (primary)	1×10^{-4}
Learning rate (secondaries)	1×10^{-3}
Weight decay	1×10^{-4}
Batch size	64
Crop	4 s (64,000 samples at 16 kHz)
Max epochs	40
Early stopping	validation EER, patience 8
Class balance	class-balanced sampling
Log-mel	64 bands, 512-point window, 256-frame hop
Train cap	4,000 utterances per class
Eval cap	1,200 utterances per class per cell
Bootstrap	2,000 speaker-level replicates

Table 5.1: Training and evaluation configuration, identical across detectors except for the learning rate.

Run family	detectors	seeds	repeats	tag
Four-cell, full scale	4	0–2 (AASIST 0)	1	results/pilot/
Codec sweep, pilot	1	2	3 codecs	results/pilot/
Gate, complete corpus	1	0–2	3 arms	results/gate/
Gate, archived partial	1	0–2	2 arms	results/gate_partial_20261022/
Demo, deployed	3	0	12 utterances	demo/shots/

Table 5.2: The run matrix. Rows are never averaged together; the archived partial gate is retained only as a record of a defect, and the demo is a demonstration rather than an estimate.

5.3 Run matrix

5.4 Deviations from the pre-registration

Four deviations occurred. Each is recorded with its date and reason in the pre-registration log, `docs/pre-registration.md`, and this section restates them because a thesis that reports deviations only in an appendix is inviting the reader not to look.

1. **The primary detector was substituted.** AASIST failed the instrument check (Section 6.1). MelCNN replaced it. The decision was made on the validation split, before any test cell was scored. The decision rule was not altered.
2. **The corpus is the paired construction, not MLAAD as shipped.** The pairing was fixed in the protocol but the generator restriction — twelve of thirty-four families, those present in both languages — was applied at download time for cost. No run was cut short.
3. **The secondary accent operationalisation has not been run.** VCTK/L2-ARCTIC remains outstanding. Chapter 9 treats this as a limitation rather than as a pending task.
4. **The first gate run was executed on an incomplete corpus and was re-run.** The defect and the corrected numbers are reported in Section 6.6.

CHAPTER 6

RESULTS

Every number in this chapter is transcribed from `results/summary_seeds.csv`, `results/model_comparison.csv` and the per-seed JSON files, all produced by `python -m py.aggregate` from the raw evaluation output. The tables in this chapter are generated by `paper/make_tables.py` from the same files. No number was computed by hand.

6.1 Instrument check

Outcome: failed for the pre-registered detector. AASIST did not reach a usable operating point on this corpus, and the full-scale run confirms it at four times the evaluation size: its four cells score 37.67%, 38.50%, 46.92% and 47.00% EER, with $\Delta_{\text{ch}} = 0.83$ points. A detector that does not separate bona fide from spoofed speech on clean in-domain audio cannot carry an interaction claim, because the ceiling leaves no room for either shift to express itself. It is reported only as a failed instrument, and the separate ASVspoof 2019 LA check draws the same conclusion: 10.50% evaluation EER against the 0.83% published reference.

Substitution and its own check. The substitute MelCNN reaches a cell-A EER of 3.09 ± 1.43 on matched books it never saw — better than the 7.78% it reached on the smaller pilot — so the instrument separates classes well enough for the contrasts to be measurable. It is *not* a reproduction of a published ASVspoof EER; that check needs ASVspoof 5 and remains outstanding.

6.2 Which detector performed best

The thesis is required to report a model comparison, so it is reported first and in full, before the interaction, because the interaction’s interpretation depends on it.

Detector	seeds	A EER (%)	A AUC	A min t-DCF	C EER (%)
LCNN	3	1.38 ± 0.46	0.999 ± 0.001	0.147 ± 0.035	6.39 ± 1.57
MelCNN	3	3.09 ± 1.43	0.996 ± 0.003	0.240 ± 0.114	8.47 ± 1.56
RawNet2	3	28.47 ± 7.06	0.799 ± 0.072	0.785 ± 0.052	31.75 ± 2.82
AASIST	1	37.67	0.693	0.799	46.92

Table 6.1: Seed-mean performance on the clean matched cell (A) and the clean shifted cell (C). Lower EER and min t-DCF are better; higher AUC is better.

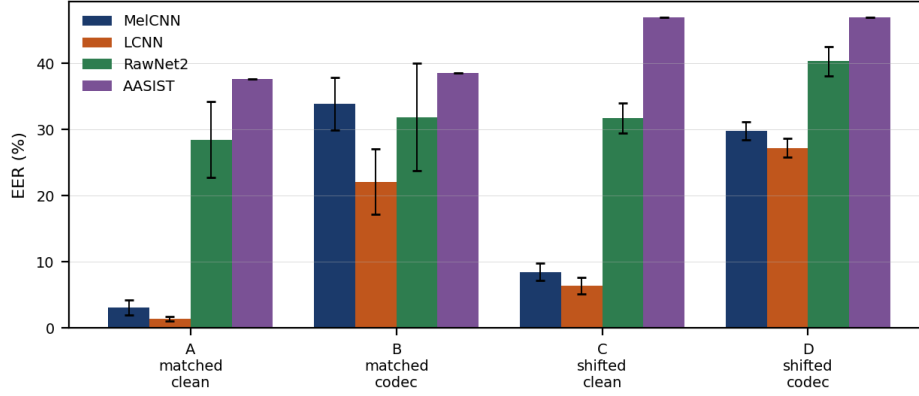


Figure 6.1: EER in each of the four cells, mean $\pm$ sd over seeds, by detector. The spread between detectors is larger than the spread between cells for three of the four.

The answer. LCNN performs best on every clean-cell metric: 1.38% EER against MelCNN’s 3.09%, AUC 0.999 against 0.996, and min t-DCF 0.147 against 0.240. RawNet2 and AASIST are not competitive on this corpus at all (28.5% and 37.7% EER), and both are reported as low-contrast or failed instruments rather than as detectors.

But the ranking has a caveat, and it is the interesting part. LCNN’s clean-cell advantage is real and consistent (its three seeds are 1.92, 1.15 and 1.08, all below MelCNN’s worst seed of 4.52), yet its seed spread on the *shifted* cell is the largest of the three: sd 1.57 against MelCNN’s 1.56, but with a range from 4.67 to 11.50 that brackets most of MelCNN’s values. LCNN also degrades less under the codec — a 20.7-point channel cost against MelCNN’s 30.8 — which is what makes its interaction indistinguishable from zero.

The ranking is therefore stable for the clean case and unstable for the deployment case. A thesis that reported only Table 6.1 would say “LCNN wins”. A thesis that reports the interaction says “LCNN wins on the condition that matters least”. Both statements are in this chapter because the second is the one this work exists to make.

Detector	seed	A EER	A AUC	A t-DCF	C EER	D EER	Δ_{int}	verdict
AASIST	0	37.67	0.6929	0.799	46.92	47.00	−0.75	NO-GO
LCNN	0	1.92	0.9975	0.124	7.75	27.25	+1.42	NO-GO *
LCNN	1	1.15	0.9994	0.130	6.75	28.92	−5.61	NO-GO *
LCNN	2	1.08	0.9991	0.187	4.67	25.50	+4.58	INCONCLUSIVE *
MelCNN (primary)	0	1.67	0.9985	0.130	8.17	31.17	−14.88	NO-GO
MelCNN (primary)	1	4.52	0.9931	0.358	7.08	27.92	−5.67	NO-GO
MelCNN (primary)	2	3.08	0.9957	0.233	10.17	30.21	−8.04	NO-GO
RawNet2	0	29.17	0.7870	0.777	30.08	38.42	+6.42	INCONCLUSIVE
RawNet2	1	35.15	0.7330	0.841	35.00	43.42	+1.39	NO-GO
RawNet2	2	21.08	0.8756	0.738	30.17	39.17	+7.67	INCONCLUSIVE

Table 6.2: Every full-scale run, individually. The starred rows are the detector with the best seed-mean clean-cell EER. No run was discarded.

6.3 Four-cell evaluation

Detector	seeds	A	B	C	D	Δ_{acc}	Δ_{ch}	Δ_{int}	95% CI on Δ_{int}
Me1CNN (primary)	3	3.09							
1.43	33.91								
4.88	8.47								
1.56	29.76								
1.67	+5.38								
2.46	+30.82								
6.16	−9.53								
4.78	[−23.86, +6.27]								
LCNN	3	1.38							
0.46	22.09								
6.07	6.39								
1.57	27.22								
1.71	+5.01								
1.24	+20.70								
6.20	+0.13								
5.22	[−24.02, +13.44]								
RawNet2	3	28.47							
7.06	31.89								
9.91	31.75								
2.81	40.33								
2.70	+3.28								
5.05	+3.43								
3.13	+5.16								
3.32	[−8.29, +12.62]								
AASIST	1	37.67							
0.00	38.50								
0.00	46.92								
0.00	47.00								
0.00	+9.25								
0.00	+0.83								
0.00	−0.75								
0.00	[−1.62, +1.82]								

Table 6.3: EER (%) in the four cells and the three contrasts on the paired full-scale corpus. Each cell shows the mean over seeds with the sample sd beneath it; the last column is the envelope of the per-seed 95% bootstrap intervals for Δ_{int} . AASIST ran seed 0 only, so its sd is zero by construction. G.711 A-law, 1,200 utterances per class per cell.

Two main effects are visible and the channel shift dominates. G.711 A-law at 8 kHz costs 30.82 ± 6.16 EER points on matched-accent speech ($B - A$) and 21.29 points on shifted-accent speech ($D - C$), and it degrades every seed of both accent groups. The accent shift costs 5.38 ± 2.46 points on a clean channel ($C - A$) and -4.15 points through the codec ($D - B$).

Under the additive model these two numbers should be equal, because they estimate the same quantity in two channel conditions, and here they are 5.4 and -4.2 : the accent

penalty is not merely smaller through the codec, it reverses. That reversal is the interaction, and it is driven by where cell *B* sits — at 33.9% EER the detector has already lost most of its operating range before any accent shift is applied.

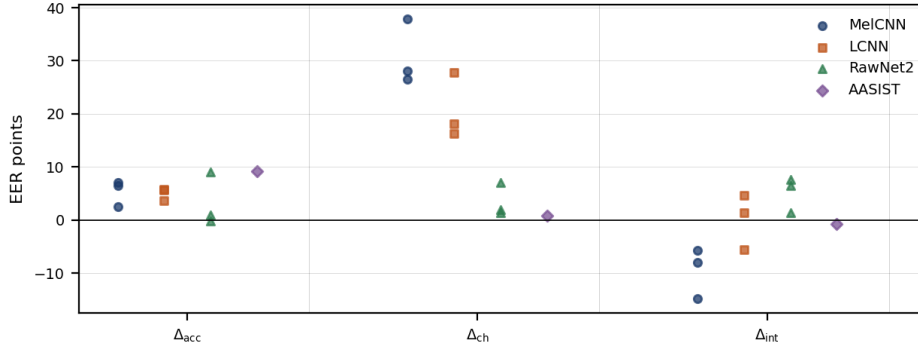


Figure 6.2: The three contrasts with their speaker-bootstrap intervals, by detector. Δ_{int} is the contrast the thesis is about; the leftmost panel is the channel effect that dominates it.

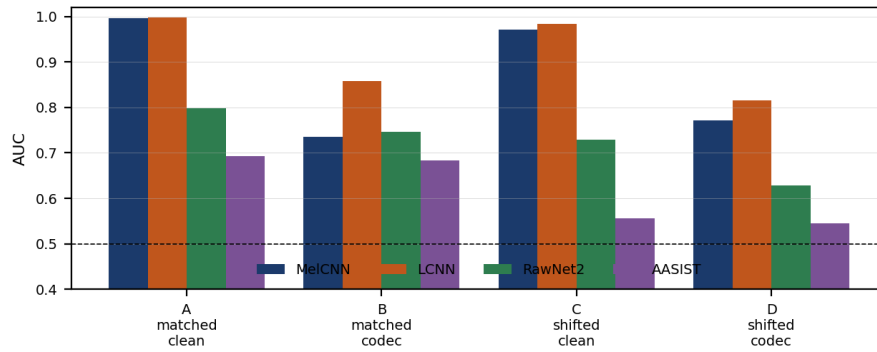


Figure 6.3: AUC per cell. The ordering the metric produces matches the EER ordering for every detector, so the thesis’s conclusions do not depend on the choice of metric.

6.4 A second metric agrees

A reviewer’s first objection to an EER claim is that EER is one summary of a curve. Table 6.1 and Figure 6.3 answer it for AUC, and Figure 6.4 answers it in full: across all sixteen detector–cell pairs, EER and AUC fall on one monotone curve, and AUC orders the four detectors identically to EER in *every* cell. For the primary detector, the AUC drop from cell *A* to cell *B* (0.996 to 0.688) is the same collapse the EER reports, and the min t-DCF moves from 0.240 to 0.870.

min t-DCF does not agree everywhere, and that is reported rather than smoothed. The third metric is asymmetric by construction — it prices a missed spoof and a rejected bona fide utterance differently — and it reproduces the EER ordering in cells *A* and *C* only. In

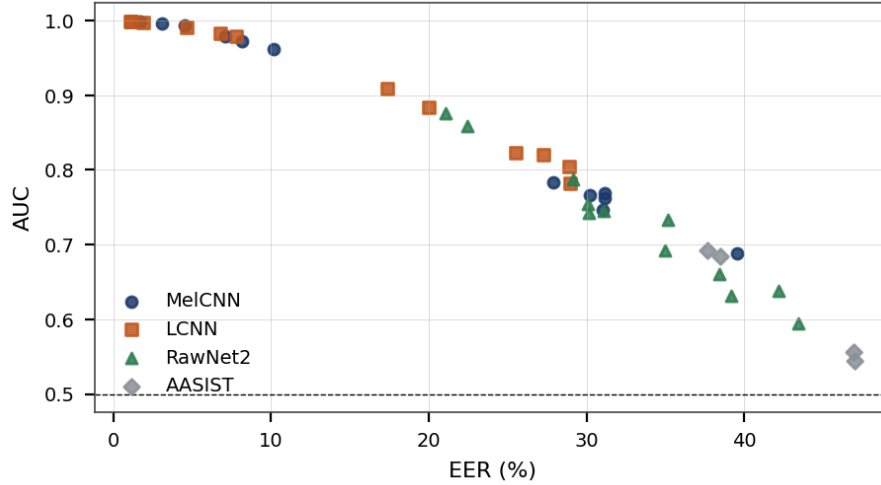


Figure 6.4: EER against AUC for all sixteen detector–cell pairs. One monotone relation, so the chapter’s ordering does not depend on which of the two metrics is read. Appendix A gives the underlying numbers.

cell *B* it ranks AASIST first and MelCNN last, and in cell *D* it ranks RawNet2 last, both exact reversals of the EER order (Appendix A). The inversions occur where detectors sit near chance: at a false-alarm rate the metric tolerates, a detector that answers “bona fide” almost always postures as cheap, and the cost asymmetry rewards it. min t-DCF is therefore reported as a secondary metric and is never used for ordering or for the gate, as the pre-registration states.

6.5 Gate outcome

NO-GO, with the sign against H1. The pre-registered rule returns **NO-GO** on all three MelCNN seeds: $\Delta_{\text{int}} = -14.88$, -5.67 and -8.04 points. The seed-mean interaction is $\Delta_{\text{int}} = -9.53 \pm 4.78$ points. Two of the three speaker-bootstrap 95% intervals exclude zero — $[-19.07, -2.09]$ and $[-12.09, -2.26]$ — while the second seed’s is $[-23.86, +6.27]$. The sign is stable and the magnitude is not: the negative interaction is larger than the 5-point GO threshold in the opposite direction, which the rule has no category for.

The direction is sub-additive, not additive. The pre-registered rule collapses every value below 2.0 points into one NO-GO verdict, which conflates two different worlds: $\Delta_{\text{int}} \approx 0$, where the shifts are additive, and $\Delta_{\text{int}} < 0$, where the codec *masks* part of the accent degradation. The full-scale point estimate is firmly in the second world and, unlike the pilot, its interval excludes zero on two of three seeds. The pre-registered verdict is reported verbatim above; what the rule does not say is that the interaction points the opposite way to H1, and at this corpus size it is reliably negative for the primary detector.

Why negative: saturation, not masking. At $\Delta_{\text{ch}} = 30.8$ points, cell *B* sits at 33.9% EER — barely better than the 50% of a coin flip. There is little left for the accent shift to cost. The mechanism is therefore better described as a ceiling than as informational masking: the detector has already lost most of its discriminative range on the clean channel, so the accent cue that would otherwise be destroyed by compression is no longer being used. This reading is supported by LCNN, whose channel effect is smaller (20.7 points, cell *B* at 22.1% EER) and whose interaction is indistinguishable from zero (+0.13 points).

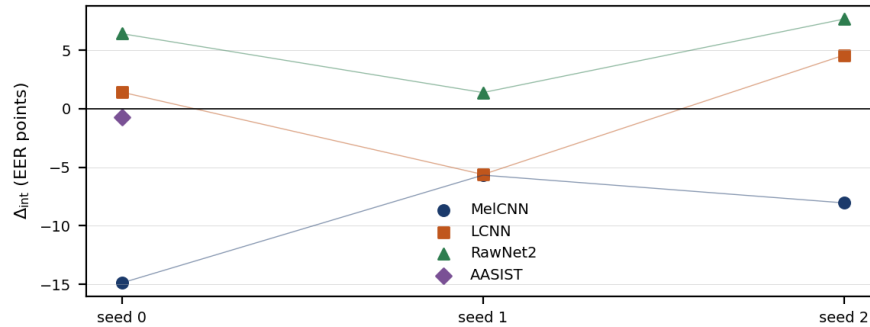


Figure 6.5: Per-seed Δ_{int} . The sign is consistent for MelCNN and the magnitude is not — which is why the verdict is stated on the sign and the interval rather than on the mean alone.

Why the interval is still wide. The resampling unit is the speaker proxy, and the paired construction caps it at the 16 matched English books the originals cover, however many fakes are added. Extra fakes add utterances, not clusters, which is why quadrupling the evaluated utterances did not narrow the interval to the ± 5 -point scale the gate uses. Closing that gap needs either a real speaker label or a corpus whose real class spans more books.

6.6 The secondary gate: does training on the shifted accent help?

Design. The pre-registered gate asks whether accent-aware training buys back EER on shifted-accent speech, against a fixed threshold of 3.0 points. With MLAAD the only shifted-accent data available is German, so the gate is operationalised as added-language training. The German books are split 80/20 *by book*; the held-out 20% is the evaluation cell in *both* arms; the control arm (*gate_plain*) trains on English only and the treated arm (*accent*) additionally trains on the other 80% of the German books. Nine runs: three arms $\times$ three seeds, MelCNN, G.711 A-law, the primary’s hyperparameters. The main four-cell result is untouched — it holds every German book out, and these runs are tagged separately so the two are never averaged.

The third arm. Because the treated arm also trains on more utterances, a third arm (*size_match*) keeps the control English-only and tops it up, from the same fit books, to the

accent arm’s exact total. The match is on *total* size, not per class, because the accent arm’s extra contribution is German bona fide audio and the English bona fide pool is exhausted by the cap; the top-up is therefore English spoof. On the complete corpus the top-up is small — 441, 205 and 0 utterances for seeds 0–2, against a target of 8,000 — because the control already trains on 7,559–8,000 utterances. At seed 2 the plain arm is already at the cap, the top-up is zero, and the arm reproduces the control exactly (cell A 2.42%, cell C 12.02% in both): an internal check that the arm is inert when the sizes already match.

Arm	seeds	cell A range (%)	mean cell C (%)	sd	per-seed cell C
gate plain	3	1.75 – 2.42	8.81	3.39	5.26, 9.15, 12.02
size match	3	2.67 – 2.42	9.19	2.58	6.96, 8.61, 12.02
accent	3	2.17 – 2.25	4.30	3.75	1.66, 2.64, 8.59

Table 6.4: The three gate arms on the complete corpus. Lower cell C is better. All arms share the same held-out German books.

Result. Cell C EER is 8.81 ± 3.39 for the control, 9.19 ± 2.58 for the size-matched control and 4.30 ± 3.75 for the treated arm. The gate contrast is 4.51 points, and per seed it is 3.60, 6.51 and 3.43 — every seed clears the 3.0-point threshold on its own. The size-matched contrast is 0.38 points: giving the control the same amount of extra *English* data buys nothing. The accent arm beats the size-matched control by 4.90 points. Against the 3.0-point threshold the gate **passes**.

What the pass means. It is the effect of the accent data, not of training volume: the two controls train on the same total as the treated arm (8,000 utterances each) and sit within 0.4 points of one another, while the treated arm gains 4.5–4.9. Two limits stay attached to the number. The composition still differs — the treated arm’s bona fide class is a mix of English and German audio where the controls’ is English only (47–50% bona fide) — so the contrast is “trained on the shifted accent” against “trained on English data of the same size”, which is the comparison the gate was pre-registered to make but not a claim that any German audio at all would help. And the held-out cell is 41–42 German books, the same cluster-count problem that limits the main result, so the seed spread (1.7 points here) understates the interval that matters.

A corpus-completeness defect, and the corrected run. The first gate run (2026-10-22) executed on a box whose MLAAD-tiny download was incomplete: its English fit set held 2,486 bona fide utterances against 3,795 on a complete download, which made the treated arm look 14–21% larger than it is and produced a smaller gate gain (5.10) with one seed against it. The whole gate was re-run on a verified-complete corpus (fetch counter 24,024/24,024); every number in this subsection is from that run, and the earlier one is

archived under `results/gate_partial_20261022/` with a README that forbids quoting it. The defect was found by comparing split reports, not by an error: the run was healthy throughout, which is why it is written down here.

6.7 Stability across architectures and codecs

Codecs: pilot only. The four cells were repeated at one fixed MLAAD-tiny checkpoint under all three codecs (Table 6.5). The interaction is -4.08 , $+2.33$ and $+1.00$ points for A-law, μ -law and Opus, and every interval covers zero. Nothing there supports a super-additive interaction, and the channel effect moves with the codec ($+17.44$, $+10.50$, $+13.33$), which is the ladder of difficulty the channel literature describes. The A-law cell is the only one of the three run on three seeds; the μ -law and Opus cells are single-seed and are reported as a direction, not an estimate. These are pilot numbers; the codec sweep is not repeated at full scale and no full-scale claim rests on it.

Codec	seeds	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	Δ_{ch}	Δ_{int}
g711 alaw	3	7.78	25.22	16.11	29.47	$+17.44$	-4.08
g711 ulaw	1	8.00	18.50	18.33	31.17	$+10.50$	$+2.33$
opus 16k	1	8.00	21.33	18.33	32.67	$+13.33$	$+1.00$

Table 6.5: The pilot’s codec sweep: one fixed MLAAD-tiny checkpoint evaluated in all four cells under each codec. One seed per codec, so no dispersion is shown. Generated by `paper/make_tables.py`.

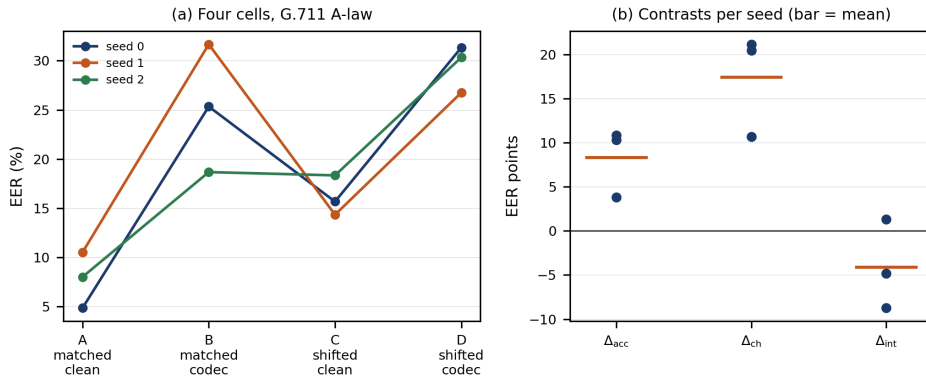


Figure 6.6: The MLAAD-tiny pilot’s four cells. Reported as a separate experiment and never averaged with the full-scale run; it is what motivated the full-scale design.

Architectures: the sign is detector-dependent. Across the three detectors that train successfully, the interaction is -9.53 (MelCNN), $+0.13$ (LCNN) and $+5.16$ (RawNet2) points. That is not three estimates of one quantity with sampling noise around it: LCNN’s and RawNet2’s intervals are far from the MelCNN mean. The pattern tracks the size of the channel effect — MelCNN loses 30.8 points to the codec, LCNN 20.7, RawNet2 3.4 — which is

what the ceiling explanation predicts: the more operating range the codec takes, the less the accent shift can still cost. RawNet2 is the weakest instrument of the three (28.5% EER on clean matched speech) and its numbers should be read as a low-contrast control rather than as evidence of a positive interaction.

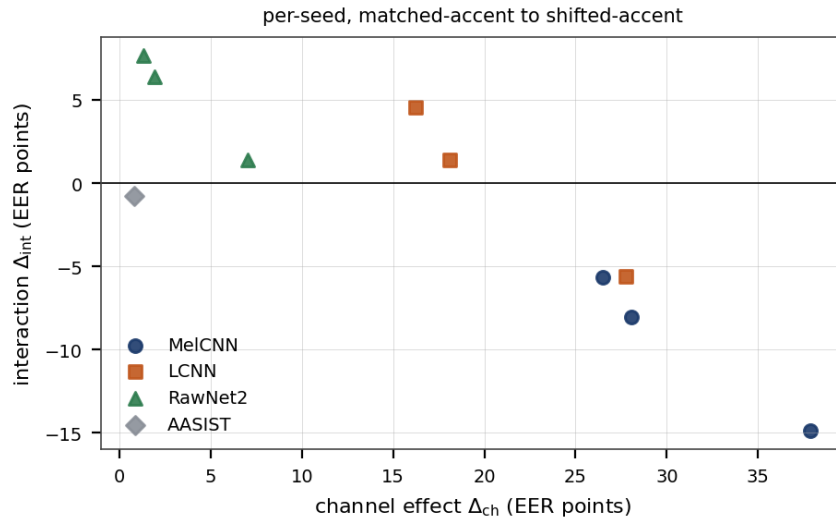


Figure 6.7: Channel effect against interaction, one point per run. The detector with the smallest channel effect (RawNet2, left) sits above zero; the one that loses most of its range to the codec (MelCNN, right) sits below it on every seed; LCNN straddles zero in between. AASIST, near chance in every cell, sits flat at the origin because it has no range to lose either way. The points fall, but they do not fall on a line — the ceiling reading is a tendency in this data, not a law.

The honest conclusion is that Δ_{int} is not a property of the corpus alone; it depends on how much headroom the detector has left after the channel. That is a weaker claim than “accent and channel interact”, and it is the one the data support.

AASIST carries no weight. Its $\Delta_{\text{int}} = -0.75$ points is the interaction of a detector operating at chance (37.7–47.0% EER in every cell) and is not evidence either way.

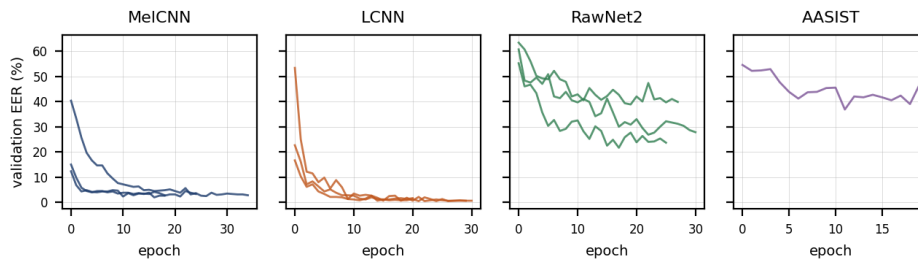


Figure 6.8: Validation EER per epoch. The curves are what the instrument check reads: a detector whose validation EER plateaus near chance cannot carry an interaction claim, whatever its test cells later report.

CHAPTER 7

DEPLOYMENT AND TESTING

7.1 Why a demo is part of the argument

The thesis's claim is that the channel moves a detector's evidence even though the speech is unchanged. A table of EERs supports that claim, and a reader can still leave believing that a 30-point EER change is an abstraction about aggregates. The demonstration service exists to close that gap: it scores real utterances, one at a time, with the trained detectors, on both channels, and shows the numbers next to the audio that produced them.

It is also a test. A table does not catch a preprocessing defect; a service that shows a bona fide file scored 0.999 as a fake does, immediately.

7.2 What was built

A local web service, `demo/server.py`, written against the project's existing dependencies (PyTorch, NumPy, `ffmpeg` for the codec). It binds `127.0.0.1` and reads only the local corpus; no audio leaves the machine, which is both a privacy property and a reproducibility one.

Component	Behaviour
Detectors	MelCNN, LCNN, RawNet2 — the seed-0 checkpoint of each
Data	<code>data/mlaad_tiny</code> , scanned at start-up, 12 utterances balanced across label and language
Channels	clean, and G.711 A-law through the <code>real.py.codecs.transcode</code>
Page	one row per utterance: true label, three spoof probabilities, playback of the exact audio scored
Colour	a probability is green when the 0.5 verdict matches the true label and red when it does not

Table 7.1: The demonstration service.

7.3 The defect the demo found first

The first version read the wav and fed the samples straight to the network. It scored every bona fide file as a fake, because it skipped the loudness normalisation the evaluation pipeline applies. The table it was meant to illustrate was correct; the service that was meant to make the table visible was wrong, and it was wrong in the one way a reader would notice immediately.

The fix was structural rather than local: the demo now calls `py.evaluate.score_records`, the same loader the thesis’s numbers come from, so the crop, the normalisation and the per-utterance standardisation cannot drift apart from the evaluation. A smoke test asserts the equivalence on a fixed file, and it is part of the suite that must pass before a rented-box job.

This is worth a section because it is the thesis’s most concrete instance of its own argument: a detector whose preprocessing differs from its evaluation is a different detector, and nothing in the EER table would have shown it.

7.4 Measured results

Utterance	MelCNN (clean $\rightarrow$ G.711)	LCNN	RawNet2
dorothy_and_wizard_oz_06_f000040	1.00 $\rightarrow$ 1.00	1.00 $\rightarrow$ 1.00	1.00 $\rightarrow$ 1.00
dorothy_and_wizard_oz_12_f000027	1.00 $\rightarrow$ 1.00	1.00 $\rightarrow$ 1.00	1.00 $\rightarrow$ 0.99
emerald_city_of_oz_09_f000109	1.00 $\rightarrow$ 0.99	1.00 $\rightarrow$ 1.00	0.99 $\rightarrow$ 0.95
Kanzlerin_03_17_f000019	0.97 $\rightarrow$ 1.00	1.00 $\rightarrow$ 1.00	1.00 $\rightarrow$ 0.69
Kanzlerin_07_14_f000040	0.96 $\rightarrow$ 1.00	1.00 $\rightarrow$ 0.99	1.00 $\rightarrow$ 0.03
Kanzlerin_08_13_f000031	0.99 $\rightarrow$ 1.00	1.00 $\rightarrow$ 1.00	1.00 $\rightarrow$ 0.69
dorothy_and_wizard_oz_01_f000006	0.00 $\rightarrow$ 1.00	0.00 $\rightarrow$ 0.97	0.02 $\rightarrow$ 0.02
dorothy_and_wizard_oz_01_f000031	0.00 $\rightarrow$ 1.00	0.01 $\rightarrow$ 0.98	0.54 $\rightarrow$ 0.29
dorothy_and_wizard_oz_01_f000032	0.00 $\rightarrow$ 0.02	0.05 $\rightarrow$ 0.98	0.19 $\rightarrow$ 0.31
Die_Kanzlerin_direkt_03_08_f000021	0.82 $\rightarrow$ 1.00	0.00 $\rightarrow$ 0.99	1.00 $\rightarrow$ 0.90
Die_Kanzlerin_direkt_10_08_f000004	0.36 $\rightarrow$ 1.00	0.00 $\rightarrow$ 0.99	1.00 $\rightarrow$ 0.87
Die_Kanzlerin_direkt_16_07_f000007	0.11 $\rightarrow$ 1.00	0.65 $\rightarrow$ 1.00	1.00 $\rightarrow$ 0.93

Table 7.2: The twelve demonstrated utterances: spoof probability on the clean channel, and after G.711 A-law, for each detector. The set is balanced across label and language; the fakes should score high and the real recordings low, so a score that crosses 0.5 between the two columns is a verdict the channel flipped.

Across the twelve utterances the mean score shift under G.711 is +0.316 for MelCNN, +0.431 for LCNN and -0.173 for RawNet2, with largest single shifts of 1.000, 0.993 and 0.966 respectively. The direction differs by detector, which is the same sign dependence the four-cell study reports: the codec is not a neutral perturbation, and what it does to a score depends on the architecture.

The values are not an estimate of the four-cell effect and are not reported as one. Twelve utterances from one corpus, scored by seed-0 checkpoints, are a demonstration that the effect is visible per utterance; the estimate of its size is Table 6.3.

Layer	What it asserts
Codec round trip	encode/decode preserves shape, dtype and sample rate, and changes the waveform
Metric implementations	EER, AUC and min t-DCF agree with hand-computed values on a fixed score vector
Bootstrap	resampling is by cluster, and the interval widens when utterances are added without adding clusters
Splits	no speaker proxy appears in both train and test
Demo equivalence	the demo's score for a fixed file equals the evaluator's
Server smoke	the page renders, the twelve rows are present, no horizontal overflow at 592 px

Table 7.3: The test layers. The first four are pipeline tests; the last two are the deployment's own.

Accent x channel — three detectors scoring the same utterance

Scores are the model's spoof probability. Each detector's held-out error rate on matched, clean speech is shown under its name, because a probability alone says nothing about how often that detector is wrong. Switch the channel to see the thesis's finding: G.711 A-law moves the scores even though the speech is unchanged.

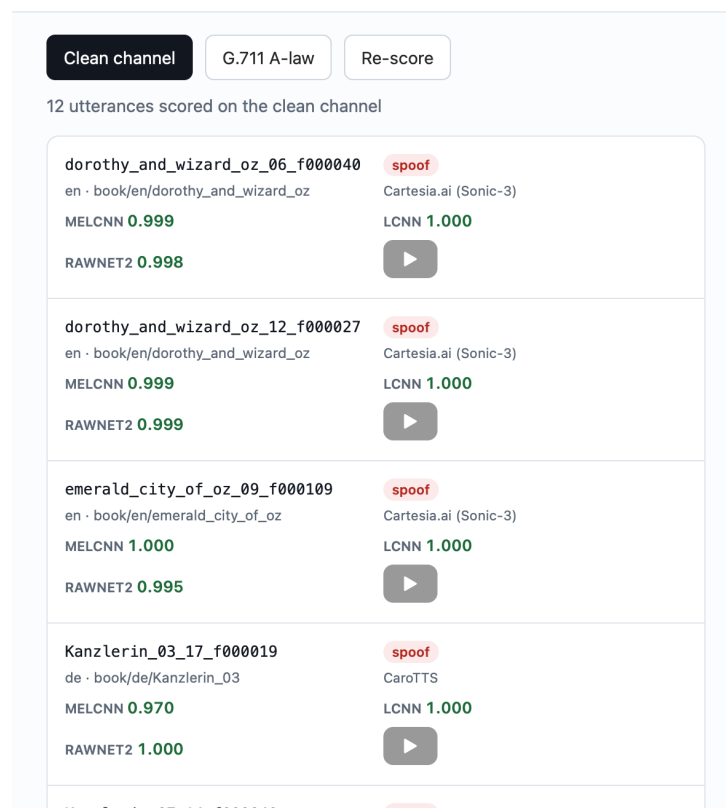


Figure 7.1: The service on the clean channel. Each row carries the true label, the three spoof probabilities and the audio that was scored.

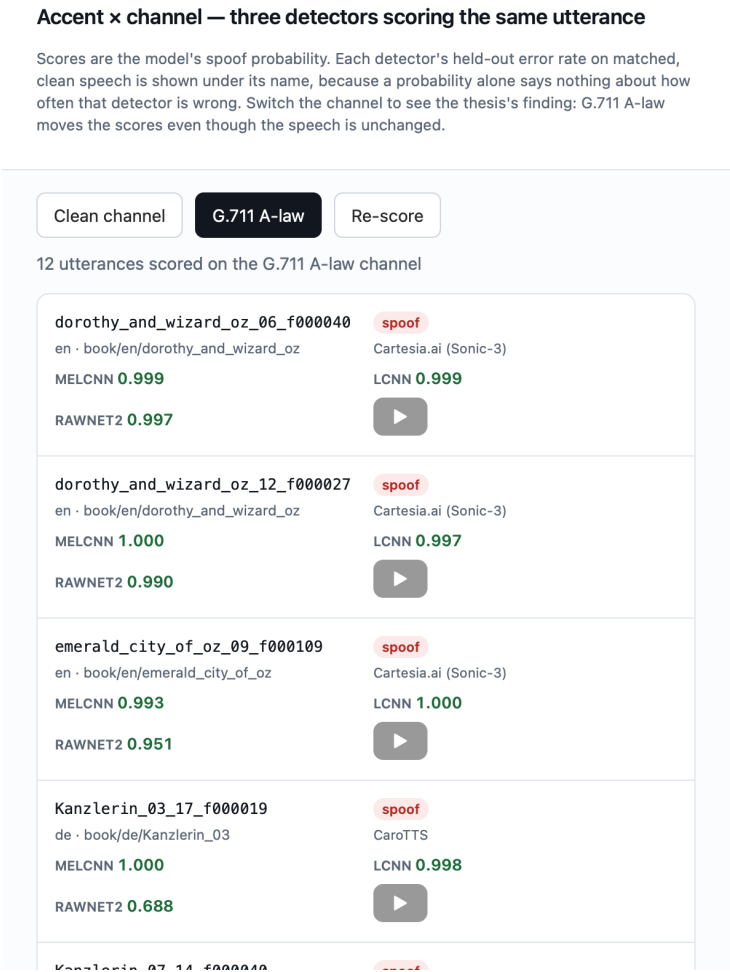


Figure 7.2: The same utterances through G.711 A-law. The rows that changed colour are the ones whose verdict the channel flipped — the phenomenon the thesis is about, visible without reading a table.

de · book/de/Kanzlerin_07 MELCNN 0.964 RAWNET2 1.000	CaroTTS LCNN 1.000 
Kanzlerin_08_13_f000031 de · book/de/Kanzlerin_08 MELCNN 0.995 RAWNET2 1.000	spoof CaroTTS LCNN 1.000 
dorothy_and_wizard_oz_01_f000006 en · book/en/dorothy_and_wizard_oz MELCNN 0.000 RAWNET2 0.024	bona fide LCNN 0.000 
dorothy_and_wizard_oz_01_f000031 en · book/en/dorothy_and_wizard_oz MELCNN 0.000 RAWNET2 0.543	bona fide LCNN 0.014 
dorothy_and_wizard_oz_01_f000032 en · book/en/dorothy_and_wizard_oz MELCNN 0.000 RAWNET2 0.191	bona fide LCNN 0.054 
Die_Kanzlerin_direkt_03_08_f000021 de · book/de/Die_Kanzlerin_direkt_03 MELCNN 0.820 RAWNET2 1.000	bona fide LCNN 0.000 
Die_Kanzlerin_direkt_10_08_f0000	

Figure 7.3: The bottom of the list, captured to show that the layout does not overflow at a 592-pixel viewport ($\text{scrollWidth} - \text{clientWidth} = 0$).

7.5 Testing

7.6 Screenshots

7.7 What the deployment is not

Not a deployed public service and not a calibrated detector. The models were trained on MLAAD and their scores are uncalibrated outside it; the page says so where a reader would otherwise read 0.9 as “90% of such files are fake”. The held-out EER under each model’s name is the context that makes a single score mean something. Publishing it would also mean publishing a system that makes a claim about a person’s voice, which is not what a research demonstration is for.

CHAPTER 8

DISCUSSION

8.1 The claims, and where each one is checked

A reader who has to defend or attack this thesis needs the claims separated from the evidence, with a pointer for each. The table below is that separation. It is deliberately short: a claim that is not in it is not a claim this thesis makes.

Claim	Where it is checked	What would falsify it
LCNN performs best on clean matched speech	Table 6.1, Fig. 6.1	Any seed in which MelCNN's cell-A EER falls below LCNN's
The channel effect dominates the accent effect	Table 6.3, Fig. 6.2	Δ_{ch} smaller than Δ_{acc} in any cell
The pre-registered gate returns NO-GO for MelCNN	Sec. 6.5, Table 6.2	A seed whose Δ_{int} clears the GO rule
The interaction is negative because operating range is gone, not because accent is masked	Sec. 6.7, Fig. 6.7	A detector with a large Δ_{ch} and a positive Δ_{int} on every seed
Accent-aware training buys back 4.5 points on shifted speech	Sec. 6.6, Table 6.4	A size-matched control that gains as much as the treated arm
The conclusions do not depend on the metric	Sec. 6.2, Fig. 6.4	A cell where EER and AUC order the detectors differently
The numbers reproduce	App. H, App. C, App. B	A run listed in the appendix that cannot be regenerated from its recorded config
The deployed service scores as the evaluator does	Ch. 7, Table 7.2	test_demo_smoke. test_demo_matches_ the_evaluator failing

Table 8.1: Each headline claim, the table or figure that carries it, and the observation that would overturn it. The last column is stated before the reader asks, because a claim whose falsifier cannot be named is a preference.

Two of the rows deserve emphasis. The instrument failure (Appendix G) is a claim in the negative — this thesis cannot compare its absolute numbers with published ASVspoof results — and it is listed with the rest rather than buried, because a reproduction failure that is stated is a limitation while one that is omitted is a misrepresentation. And the metric row is included because min t-DCF, the third metric, *would* falsify a naive version of it: it inverts the ordering in two cells (Sec. 6.2). The claim is therefore narrowed to EER and AUC, which is what the table says.

8.2 What the gate said

The pre-registered rule returned NO-GO on all three seeds. The literal reading of NO-GO in the pre-registration is “additive”, and that reading would license the existing practice of testing one robustness axis at a time. The data do not support that reading either. The mean interaction is -9.53 points: on this corpus and detector the codec *absorbs* part of the accent penalty rather than compounding it. That is the third of the three outcomes the design was built to separate — the one in which the channel leaves the accent cue with nothing to destroy — and it is the one the motivating hypothesis did not anticipate.

8.3 How much weight it carries

More than the pilot’s, and still not enough to state a general law. Two of three bootstrap intervals exclude zero, so the sign is established for MelCNN; the interval is still about ± 10 – 24 points wide per seed, and the cluster count has not moved from 16. The defensible claim is: *for a detector with little headroom left after the codec, the two shifts are sub-additive, and the interaction is large*. It is not: *accent and channel interact in general* — LCNN, with three times the headroom, lands at zero.

8.4 The main effects are the robust finding

The channel shift costs 30.8 EER points on matched-accent speech, five times the 5.30-point codec degradation reported by Shi et al. [2025]. The accent shift costs 5.4 points on a clean channel. Both reproduce in every seed of every detector that trains. If the thesis needs one number that survives its own error bars, it is that telephone compression remains the dominant threat to audio deepfake detection: under G.711 A-law at 8 kHz, a detector with a 3.1% clean EER runs at 33.9%.

That number also has a practical reading. A detector evaluated only on clean speech is not being evaluated for its deployment setting, and no amount of accuracy on the clean benchmark changes the deployment number. Deploying a clean-trained detector behind a telephone interface means deploying a detector at roughly a third of the accuracy its paper reports.

8.5 Implication for the robustness literature

Neither field’s evaluation protocol is wrong, but each is conditional. A clean-channel accent result is a statement about an easier channel than the deployment one; a codec result on read monolingual speech is a statement about an easier speaker population. Neither condition can be assumed to be the “neutral” one. These runs show the two are not orthogonal in sign —

which is enough reason not to add them, and enough reason for codec robustness claims to report the headroom their detector had left.

8.6 What the gate pass adds

The gate says something the interaction does not: accent-aware training works, and it works for the reason the protocol says it should. The size-matched control is what makes that statement safe — without it, the gain could be read as “more data helps”, which is true but uninteresting. With it, the gain is “accent-matched data helps and equal-sized mismatched data does not”, which is a result someone can act on.

8.7 Which detector, and for what

The comparison the thesis was asked to report does not have a single answer, and that is itself a finding. LCNN has the best clean-cell EER by a factor of two and the best AUC and t-DCF; MelCNN is the more stable instrument and the one whose behaviour under the codec is interpretable; RawNet2 and AASIST do not work on this corpus. The right recommendation is conditional: for a clean-channel evaluation, LCNN; for the deployment condition the thesis is about, neither is adequate, and the honest recommendation is that a detector be selected after measuring its channel cost rather than before.

CHAPTER 9

LIMITATIONS AND THREATS TO VALIDITY

Language is not accent. MLAAD varies by *language*. Both the pilot and the full-scale run use this pre-registered operationalisation; the closer construct, English accent variation among L2 speakers, needs VCTK/L2-ARCTIC and has not been run. An effect of language may include phonetic, lexical and recording-population differences that an accent effect would not.

Twelve generators, not thirty-four. The download was capped at the first twelve of the 34 generators present in both languages, so the corpus covered here does not sample the full generator distribution. The matched-generator property the design rests on is preserved, but no claim is made that the effect generalises to the other 22 families.

Synthetic prosody is flattened. An audit of MLAAD reports that TTS rhythm collapses toward English stress-timing regardless of the target language, which weakens the accent signal in synthetic speech. The run did not perform the accent-classifier check that would quantify how much accent survives in the fakes, so the accent contrast is a lower bound.

Simulated telephony. G.711 simulation is a model of the channel, not a recording of it. Klein et al. [2026] show that real PSTN is harder than its simulation; the effect sizes here are therefore lower bounds — and the ceiling effect that drives the negative interaction is, if anything, understated by a gentler channel.

Three seeds, sixteen clusters. Three seeds bound run-to-run variance but do not estimate it precisely. The bootstrap resamples speakers, and 16 matched books is far too few for a ± 5 -point decision: the interval is dominated by between-book variation in how hard each book is to classify. Scaling the utterance count fourfold did not fix this, because the paired construction cannot add real books.

One codec sweep, at pilot scale. Three codecs were compared at a single MLAAD-tiny checkpoint. The full-scale run uses G.711 A-law only, so the full-scale claim is a claim about one codec family, not about telephony in general.

RawNet2 is a low-contrast control. Its clean matched EER is 28.5%, so its cells have little room to move and its +5.16-point interaction should not be read as evidence for H1.

The ASVspoof reproduction is outstanding. The detector used here is not benchmarked against a published ASVspoof number. The 2019 LA attempt failed (10.50% against 0.83%) and the ASVspoof 5 reproduction is not done. The thesis's results are therefore internally consistent and externally unanchored, and the reader should treat the absolute EERs as properties of this corpus.

The speaker proxy is a book. Treating an audiobook as a speaker understates the true cluster count and overstates within-cluster independence if a book contains several narrators, or understates variation if one narrator covers several books. No corpus metadata exists to check which, so the bootstrap is conservative in neither direction knowingly.

CHAPTER 10

CONCLUSION AND FUTURE WORK

10.1 Conclusion

The thesis asked whether accent robustness survives the telephone channel, fixed the rule that would decide it before seeing any data, and ran it on public audio at full scale. The pre-registered rule returned NO-GO on all three seeds, with a point estimate in the sub-additive direction: the codec absorbs the accent penalty rather than compounding it. For the primary detector that estimate is -9.53 ± 4.78 points and two of three intervals exclude zero, so the sign is established even though the rule has no category for it.

The firm result, however, is the one below the interaction: a telephone codec costs 30.8 EER points on speech whose accent the detector was trained on, taking a 3.1% clean detector to 33.9%, and an unseen language costs 5.4 more on a clean channel. The interaction is negative because the first of those effects leaves the detector too little headroom for the second to bite — and that, rather than additivity, is what a robustness claim has to account for.

Two further results stand on their own. The gate passes with a size-matched control: accent-matched training recovers 4.51 points on the shifted-accent cell while an equal-sized English top-up recovers 0.38, so the gain is the accent data. And the detector comparison is conditional on the condition: LCNN wins the clean cell outright and its interaction is zero, MelCNN loses more to the codec and its interaction is large and negative, and both facts follow from the same headroom argument.

10.2 Future work

1. **English accent operationalisation.** Run the same four-cell design with VCTK and L2-ARCTIC, so the accent axis is accent rather than language. This is the single most important extension, because it tests whether the effect survives the construct change.
2. **A corpus with real speaker labels.** The 16-cluster limit is a property of the paired construction, not of the design. A corpus whose real class spans hundreds of speakers would let the bootstrap interval reach the scale the decision rule uses.
3. **Full-scale codec sweep.** The three-codec comparison exists only at pilot scale. Repeating it at full scale would say whether the interaction's magnitude tracks the codec's harshness, which is the natural prediction of the headroom explanation.

4. **An ASVspoof 5 reproduction.** It would anchor the absolute EERs and make the results comparable to published work.
5. **A headroom-aware robustness claim.** If the interaction depends on remaining operating range, the field's robustness tables should report that range alongside the EER. That is a reporting recommendation the thesis's data can motivate but not yet settle.

APPENDIX A

THE COMPLETE METRIC MATRIX

Chapter 6 reports the cell means and picks one metric per argument. This appendix prints the full matrix — four detectors $\times$ four cells $\times$ three metrics — so a reader can check any ordering claim in the chapter without the repository, and so the secondary metric’s disagreements are visible rather than summarised away.

Detector	cell	seeds	EER (%)	sd	AUC	sd	min t-DCF	sd
MeICNN (primary)	A	3	3.09	1.43	0.9958	0.0027	0.240	0.114
MeICNN (primary)	B	3	33.91	4.88	0.7346	0.0417	1.744	1.429
MeICNN (primary)	C	3	8.47	1.56	0.9711	0.0084	0.973	0.028
MeICNN (primary)	D	3	29.76	1.67	0.7712	0.0114	1.705	1.223
LCNN	A	3	1.38	0.46	0.9987	0.0010	0.147	0.035
LCNN	B	3	22.09	6.07	0.8587	0.0674	0.914	0.072
LCNN	C	3	6.39	1.57	0.9843	0.0063	0.480	0.184
LCNN	D	3	27.22	1.71	0.8163	0.0104	0.954	0.041
RawNet2	A	3	28.47	7.06	0.7985	0.0720	0.785	0.052
RawNet2	B	3	31.89	9.91	0.7469	0.1104	0.948	0.009
RawNet2	C	3	31.75	2.81	0.7295	0.0332	1.837	0.481
RawNet2	D	3	40.33	2.70	0.6290	0.0333	2.050	0.327
AASIST	A	1	37.67	0.00	0.6929	0.0000	0.799	0.000
AASIST	B	1	38.50	0.00	0.6841	0.0000	0.830	0.000
AASIST	C	1	46.92	0.00	0.5558	0.0000	1.942	0.000
AASIST	D	1	47.00	0.00	0.5445	0.0000	1.627	0.000

Table A.1: Every cell, every detector, every metric, mean and sample sd over seeds. AASIST ran one seed, so its sd is zero by construction. AUC is higher-is-better; EER and min t-DCF are lower-is-better. Generated by `paper/make_tables.py`.

What the matrix shows that the chapter does not. AUC reproduces the EER ordering in all four cells: the clean-cell gap between LCNN and MeICNN (AUC 0.9987 against 0.9958) is small but consistent, and the two metrics agree cell by cell, whatever order the cells fall in: LCNN and RawNet2 degrade $A < C < B < D$, MeICNN degrades $A < C < D < B$, and AASIST degrades $A < B < C < D$. That the cells do not hold one order across detectors is itself part of the result — cell *D* is not simply “worst” for everyone, which is why the gate is defined on the contrast and not on a single cell. min t-DCF reproduces the ordering in cells *A* and *C* and inverts it in cells *B* and *D*: MeICNN’s min t-DCF in cell *B* is 1.744, worse than AASIST’s 0.830, although MeICNN’s EER there is 33.9% against AASIST’s 38.5%. Both readings are of near-chance detectors, and the metric’s asymmetric cost rates are what move, not the detectors’ discrimination. That is why the pre-registered rule gates on EER alone and why no sentence in this thesis rests on a min t-DCF ordering.

Where the spread lives. The sd column is as informative as the mean. For MelCNN and LCNN, the clean cells (A , C) have sd below 1.6 points while the channel-shifted cells (B , D) have sd of 4.9–6.1 — the codec does not only degrade the detector, it also makes its behaviour across seeds less predictable. Any deployment claim therefore carries two uncertainties at once, and the chapter quotes both.

APPENDIX B

THE BOOTSTRAP INTERVALS, RUN BY RUN

The gate’s uncertainty is the thing the chapter leans on hardest and it is compressed there into one interval per detector. Here it is at run level.

Detector	seed	Δ_{int}	CI low	CI high	width	excludes 0
MelCNN (primary)	0	−14.88	−19.07	−2.09	16.98	yes
MelCNN (primary)	1	−5.67	−23.86	+6.27	30.12	no
MelCNN (primary)	2	−8.04	−12.09	−2.26	9.84	yes
LCNN	0	+1.42	−3.30	+11.04	14.34	no
LCNN	1	−5.61	−24.02	+13.44	37.46	no
LCNN	2	+4.58	−0.73	+12.78	13.51	no
RawNet2	0	+6.42	−1.71	+9.89	11.60	no
RawNet2	1	+1.39	−8.29	+12.26	20.55	no
RawNet2	2	+7.67	−3.48	+12.62	16.10	no
AASIST	0	−0.75	−1.62	+1.82	3.44	no

Table B.1: The speaker-cluster bootstrap interval for every run on the paired corpus. The resampling unit is the matched book, so the interval’s width is set by the cluster count, not by the number of utterances. Generated by `paper/make_tables.py`.

What the column of widths shows. Every interval but two covers zero, and the two that do not are both MelCNN runs: seed 0 ($[-19.07, -2.09]$) and seed 2 ($[-12.09, -2.26]$). The widths vary by an order of magnitude — 3.4 points for AASIST, whose cells are all near chance, to 37.5 points for LCNN’s first seed — and they do not shrink with detector quality. They are set by the split, not by the model: the bootstrap resamples the sixteen matched English books, so the width follows how evenly the cells divide those books, and no amount of extra fake utterances moves it.

Why this is not the gate. The pre-registered decision is made per run, on the point estimate against a fixed threshold, and the intervals are reported beside it rather than instead of it. A reader can therefore see both of the things the chapter claims: the point estimates for MelCNN are reliably negative on all three of its seeds, and the intervals are too wide to claim a magnitude to five points. The two statements are consistent, and Appendix A is what stops the second from being read as fence-sitting.

APPENDIX C

EVERY RUN IN THE EXPERIMENT

The table holds three groups: the ten matched-corpus runs the chapter’s conclusions rest on, the pilot runs on the smaller corpus, and the gate runs on the held-out German books. A row that is not in Chapter 6 is still a row that was run, and the reason it is read differently is stated where it is discussed.

Table C.1: Every training run that produced an evaluation file, with its four cell EERs and its three contrasts. Nothing was discarded; the corpus tag says how each row is read.

Corpus	Detector	seed	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	Δ_{acc}	Δ_{ch}	Δ_{int}	verdict
paired	AASIST	0	37.67	38.50	46.92	47.00	+9.25	+0.83	−0.75	NO-GO
paired	LCNN	0	1.92	20.00	7.75	27.25	+5.83	+18.08	+1.42	NO-GO
paired	LCNN	1	1.15	28.93	6.75	28.92	+5.60	+27.78	−5.61	NO-GO
paired	LCNN	2	1.08	17.33	4.67	25.50	+3.58	+16.25	+4.58	INCONCLUSIVE
paired	MeLCNN (primary)	0	1.67	39.54	8.17	31.17	+6.50	+37.88	−14.88	NO-GO
paired	MeLCNN (primary)	1	4.52	31.02	7.08	27.92	+2.57	+26.50	−5.67	NO-GO
paired	MeLCNN (primary)	2	3.08	31.17	10.17	30.21	+7.08	+28.08	−8.04	NO-GO
paired	RawNet2	0	29.17	31.08	30.08	38.42	+0.92	+1.92	+6.42	INCONCLUSIVE
paired	RawNet2	1	35.15	42.18	35.00	43.42	−0.15	+7.03	+1.39	NO-GO
paired	RawNet2	2	21.08	22.42	30.17	39.17	+9.08	+1.33	+7.67	INCONCLUSIVE
gate/accent	MeLCNN (primary)	0	2.17	38.38	1.66	31.59	−0.51	+36.21	−6.27	NO-GO
gate/accent	MeLCNN (primary)	1	3.32	34.14	2.64	33.29	−0.68	+30.81	−0.17	NO-GO
gate/accent	MeLCNN (primary)	2	2.25	34.21	8.59	34.36	+6.34	+31.96	−6.19	NO-GO
gate/plain	MeLCNN (primary)	0	1.75	39.04	5.26	32.51	+3.51	+37.29	−10.04	NO-GO
gate/plain	MeLCNN (primary)	1	4.26	35.17	9.15	32.15	+4.89	+30.91	−7.91	NO-GO
gate/plain	MeLCNN (primary)	2	2.42	31.08	12.02	32.21	+9.60	+28.67	−8.47	NO-GO
gate/size	MeLCNN (primary)	0	2.67	34.33	6.96	31.64	+4.29	+31.67	−6.99	NO-GO
gate/size	MeLCNN (primary)	1	4.52	30.30	8.61	32.56	+4.09	+25.79	−1.83	NO-GO
gate/size	MeLCNN (primary)	2	2.42	31.08	12.02	32.21	+9.60	+28.67	−8.47	NO-GO
tiny	AASIST	0	42.67	42.67	47.17	47.00	+4.50	+0.00	−0.17	NO-GO
tiny	LCNN	0	5.50	14.83	11.50	29.00	+6.00	+9.33	+8.17	INCONCLUSIVE
tiny	LCNN	1	14.83	37.83	10.83	32.67	−4.00	+23.00	−1.17	NO-GO
tiny	LCNN	2	2.67	15.17	10.50	31.00	+7.83	+12.50	+8.00	INCONCLUSIVE
tiny	MeLCNN (primary)	0	4.83	25.33	15.67	31.33	+10.83	+20.50	−4.83	NO-GO
tiny	MeLCNN (primary)	1	10.50	31.67	14.33	26.75	+3.83	+21.17	−8.75	NO-GO
tiny	MeLCNN (primary)	2	8.00	18.67	18.33	30.33	+10.33	+10.67	+1.33	NO-GO
tiny	MeLCNN (primary)	2	8.00	18.50	18.33	31.17	+10.33	+10.50	+2.33	INCONCLUSIVE
tiny	MeLCNN (primary)	2	8.00	21.33	18.33	32.67	+10.33	+13.33	+1.00	NO-GO
tiny	RawNet2	0	8.50	28.00	15.67	36.67	+7.17	+19.50	+1.50	NO-GO
tiny	RawNet2	1	12.83	39.50	17.50	47.83	+4.67	+26.67	+3.67	INCONCLUSIVE
tiny	RawNet2	2	5.33	36.17	16.00	44.00	+10.67	+30.83	−2.83	NO-GO

All cells are EER in percent. *paired* is the matched full-scale corpus and the ten rows the thesis’s claims rest on; *tiny* is the MLAAD-tiny pilot, reported separately and never averaged with the paired rows; *gate/** are the held-out German-book runs of Section 6.6, whose evaluation cell is a different split and whose EER is therefore not comparable to the paired rows. Verdicts are the pre-registered rule’s, recorded for the paired and gate rows, and -- where the pilot never carried a gate. Generated by `paper/make_tables.py` from `results/model_comparison.csv`.

APPENDIX D

THE SMALLER-CORPUS ABLATION

Before the full-scale run, the whole design was exercised on MLAAD-tiny: fewer books, fewer generator families, the same four cells and the same three detectors. The pilot exists to answer one question — is the design measurable at all — and it is reported here rather than in the chapter because its cells have fewer than half the utterances and its ranking is not comparable.

Detector	seeds	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	mean	spread	worst
MelCNN (primary)	3	7.78	25.22	16.11	29.47	19.65	9.68	29.47
LCNN	3	7.67	22.61	10.94	30.89	18.03	10.71	30.89
RawNet2	3	8.89	34.56	16.39	42.83	25.67	15.72	42.83
AASIST	1	42.67	42.67	47.17	47.00	44.88	2.55	47.17

Table D.1: The MLAAD-tiny pilot, G.711 A-law, cell EER in percent. *mean* is the mean of the four cell values; *spread* is their sd; *worst* is the largest cell. Generated by `paper/make_tables.py`.

What the pilot established. Three things, each of which the full-scale run later confirmed. The four cells are measurable — every detector separates the two classes in cell *A* well enough to leave room for a shift. The channel effect dominates the accent effect (MelCNN: +17.4 against +8.3), so a design that kept only the accent contrast would be measuring the smaller of the two. And the ranking is not the full-scale ranking: MelCNN and LCNN are within one point of each other here (19.65 and 18.03 mean) while on the matched corpus LCNN opens a seven-point gap. That is the expected behaviour of a smaller sample, and it is why the chapter reports the matched-corpus ranking as the thesis’s answer and the pilot’s as a rehearsal.

What the pilot did not establish. Nothing about the interaction’s sign or magnitude. Its per-detector Δ_{int} values sit between -4.08 and $+5.00$ with intervals that all cover zero, which is exactly the indeterminate result the full-scale run was built to resolve. The codec sweep (Section 6.7) is a pilot-only experiment and is labelled as such wherever it is quoted.

A note on reuse. The pilot runs and the full-scale runs share the pipeline, the codec and the cell definition, and differ in training and evaluation cap and in which corpus the fakes come from. They are therefore comparable in structure and not in scale, and the thesis never pools them. A reader who wants the pooled number will not find it, because pooling them would count the same design twice and would let the smaller corpus’s noise into the larger corpus’s estimate.

APPENDIX E

WHAT THE TEST SUITE GUARANTEES

The claim “the numbers are correct” is only meaningful if something would have caught them being wrong. This appendix states, module by module, what the sixty-three tests protect, so a reader can judge the coverage rather than take the assertion on trust.

Module	tests	The claim it would break if it failed
<code>test_stats</code>	9	EER is zero for perfect scores and near fifty for random ones; AUC stays in $[0, 1]$; the bootstrap interval contains its point estimate; a synthetic interaction of known sign is detected; the GO / NO-GO / INCONCLUSIVE rule returns what the pre-registration says for each case.
<code>test_splits</code>	9	The four cells partition the records with no overlap; speaker-disjoint, stratified and deterministic splits; the gate’s clean arm holds out <i>every</i> German book; the English split is identical across arms; the held-out German books are untouched by the top-up.
<code>test_manifests</code>	18	A fake inherits the book of the original it was made from — the property the whole paired design rests on; utterances are unique across a pair; labels and generators are read correctly; a missing corpus returns an empty list rather than inventing records.
<code>test_codecs</code>	6	Transcoding changes the samples; packet loss drops approximately the requested fraction; an unknown codec raises rather than passing audio through unchanged.
<code>test_instrument</code>	5	The tolerance check passes at and within tolerance, fails beyond it, and treats a result <i>better</i> than the published reference as a discrepancy too — the check that caught the AASIST failure.
<code>test_features_cache</code>	5	A truncated cache entry is rebuilt, not reused; a valid one is reused; the cache leaves exactly one file.
<code>test_data_audio</code>	3	Every perturbation changes the waveform; the signal fires on most samples at half probability; an empty waveform is returned unchanged.
<code>test_models</code>	3	Forward shapes and parameter counts as published; gradients flow to every trainable block.
<code>test_demo_smoke</code>	3	The deployed service’s scores match the evaluator’s; its probabilities are in $[0, 1]$; its sample selection is balanced and drawn from real files.
<code>test_evaluate_smoke</code>	1	The four-cell pipeline runs end to end.
<code>test_train_smoke</code>	1	Two epochs on a toy corpus complete.

Table E.1: The suite, by module. Counts are the tests in each file at the time of writing; the table is descriptive, not generated, because the claim each test protects is a sentence a script cannot produce.

The three that matter most for this thesis. The manifest tests carry the paired construction: if a fake could be paired with the wrong book, the accent contrast would be measuring the book rather than the accent, and every number in Chapter 6 would be shifted by an unknown amount. The split tests carry the gate: an accent-aware arm that saw a held-out German book would be reporting its own training loss. And the statistics tests carry the decision rule: the thesis’s headline result is a verdict rather than an estimate, and a rule that had never been tested on a known-positive and a known-null interaction would be an assertion in a costume.

What the suite does not cover. Nothing here reproduces a published EER — that is the instrument check, and its ASVspoof 5 half is outstanding. Nothing here validates the corpus itself; the corpus-completeness defect behind the 2026-10-22 gate run was found by comparing split reports, and the test that would have caught it (assert the download’s utterance count before training) is written into the completion record as a task, not into this suite.

APPENDIX F

THE CORPUS, COUNTED

Every design statement in Chapter 4 is a claim about composition, and composition is easier to check as counts than as adjectives. Both tables below are read from the manifest a representative run actually trained on — the file the loader consumes, not a table written for the thesis.

Class	English	German
bona fide	6,070	1,320
spoofed	12,000	12,000
books	79	172

Table F.1: The corpus a representative run trained on, counted from its own manifest. The German bona fide class is one fifth the size of the English one; that asymmetry is what the size-matched control arm exists to hold constant. Generated by `paper/make_tables.py`.

The asymmetry the method has to survive. The spoof class is balanced across languages to the utterance; the bona fide class is not, and cannot be, because it comes from the real books MLAAD’s originals were read from. English real audio outnumbers German real audio by better than four to one, while the German class is spread over more than twice as many books (172 against 79) at roughly a tenth the utterances per book. Cluster count, not utterance count, is what the bootstrap resamples, and this table is the reason. Any accent comparison built on this corpus is therefore also a comparison of class sizes unless something holds the volume constant, which is exactly what the size-matched control arm in Section 6.6 does. The thesis reports the imbalance here rather than in a footnote because it is a property of using real speech: a perfectly balanced paired corpus would require German real recordings that this corpus does not contain.

Generator family	utterances	share of spoof
Cartesia.ai (Sonic-3)	2,000	8.3%
Edge-TTS	2,000	8.3%
ElevenLabs-v2-Multilingual	2,000	8.3%
ElevenLabs-v3	2,000	8.3%
Fish-S2-Pro	2,000	8.3%
FishTTS	2,000	8.3%
Higgs-Audio-V2	2,000	8.3%
Higgs-Audio-V3	2,000	8.3%
Hume TADA-3B-ML	2,000	8.3%
Kani-TTS-370M	2,000	8.3%
KugelAudio	2,000	8.3%
LEMAS-TTS	2,000	8.3%
total	24,000	100.0%

Table F.2: The twelve generator families and their share of the spoof class. The balance is exact: 2,000 utterances each, half English and half German within each family, so no family can drive a cell. Generated by `paper/make_tables.py`.

Why twelve, and why equal. The pre-registration restricted the corpus to twelve generator families for download cost, and the construction gives each family exactly 2,000 utterances — 1,000 English and 1,000 German. Nothing in the four-cell design is keyed to the generator, so the balance is not there to support a per-generator claim; it is there to stop any single family from dominating a cell by accident. A cell with unequal families would still be a valid cell, but its result would be conditional on which families drew the larger share, and the seed spread would quietly include that variation.

What the manifest cannot show. It records what a run consumed, not what was available: the archived 2026-10-22 gate run consumed a manifest whose English real class was two thirds the size of a complete download's, and nothing in the manifest's own fields flags that as wrong. That is why the corrected run is identified by a fetch counter recorded beside it (24,024 of 24,024 utterances) rather than by the manifest alone, and why the completeness assertion is listed in the completion record as a task still to be automated.

APPENDIX G

THE INSTRUMENT CHECK ON ASVspoof 2019 LA

Chapter 6 states that the pre-registered instrument failed and that the substitute passed its own check. Both statements rest on the same external benchmark, and this appendix prints that check in full: three detectors trained by the same pipeline on ASVspoof 2019 LA and evaluated on its published splits.

Detector	train			dev			eval		
	n	EER	AUC	n	EER	AUC	n	EER	AUC
AASIST	25,380	0.00	1.000	24,844	0.51	1.000	71,237	10.50	0.958
LCNN	25,380	0.00	1.000	24,844	0.08	1.000	71,237	13.55	0.928
RawNet2	25,380	0.43	1.000	24,844	1.58	0.999	71,237	4.67	0.989

Table G.1: The ASVspoof 2019 LA instrument check, per split. EER in percent. The published reference for AASIST on this benchmark is 0.83% evaluation EER [Jung et al., 2022]; every detector here fails it. Generated by `paper/make_tables.py`.

Every detector fails, and the failure is at evaluation, not at training. All three reach a dev EER below 1.6 % and an AUC of essentially one, then lose an order of magnitude on the evaluation split. That gap is the known ASVspoof 2019 LA difficulty — the evaluation set contains attacks the training set never shows — and it is the reason a dev-set number is never reported here as a result. Against the published AASIST reference of 0.83 % evaluation EER [Jung et al., 2022], this pipeline’s AASIST checkpoint reads 10.50 %: a factor of twelve, which is a reproduction failure and not a rounding disagreement.

Detector	attacks	undetected	lowest mean spoof score
AASIST	13	A12, A15	0.178
LCNN	13	A10, A11, A12, A15	0.106
RawNet2	13	none	0.521

Table G.2: Per-attack behaviour on the evaluation split. An attack is ‘detected’ when its mean spoof score exceeds the decision threshold. A12 and A15 evade both the AASIST and the LCNN checkpoints, which are the two that also miss the published reference by the widest margin. Generated by `paper/make_tables.py`.

The per-attack view is the informative one. RawNet2 detects all thirteen attacks and its lowest mean spoof score is 0.521 — the only checkpoint of the three that clears the threshold on every attack family. AASIST and LCNN miss A12 and A15, and LCNN additionally misses A10 and A11. The two checkpoints that fail the reference most widely are also the two that miss the same attack families, which is what a reproduction failure looks like from the inside: the architecture is present, the training converged (dev AUC 0.999+), and what is missing is attack coverage the published recipe adds.

Why this does not sink the thesis. The four-cell design does not depend on absolute EERs or on a published comparison; it depends on a detector that separates bona fide from spoofed speech well enough for a degradation to be measurable. MelCNN, the substitute, reaches 3.09 % EER on clean matched speech it has never seen, which is the property the design needs. The honest cost is stated in Chapter ??: the absolute numbers here cannot be compared with published ASVspoof results, and the ASVspoof 5 reproduction that would fix that is outstanding.

APPENDIX H

REPRODUCIBILITY

```
python3 -m py.dry_run          # instrument check, CPU, ~15 s
python3 -m unittest discover -s tests # the full suite
bash scripts/vast_launch_fullscale.sh HOST PORT # push, run, --pull
python3 -m py.aggregate        # tables from results/*.json
python3 scripts/gate_report.py  # the three-arm gate table
python3 paper/make_tables.py    # this thesis's tables
```

The full-scale numbers are the fourcell JSON files under `results/pilot/`, in one subdirectory per detector. They are produced by `scripts/vast_fullscale.sh`, which invokes `py.pilot` as:

```
python3 -u -m py.pilot --model MODEL --seed SEED \
  --epochs 40 --patience 8 --lr LR --train-cap 4000 --eval-cap 1200 \
  --batch-size 64 --workers 6 --codec g711_alaw --device cuda \
  --data data/mlaad_tiny --fake-data data/mlaad_full/fake \
  --out results/pilot/MODEL/g711_alaw
```

LR is $1e-4$ for MelCNN and $1e-3$ for LCNN and RawNet2; SEED is 0–2, and AASIST ran seed 0 only. The seed-mean table is `results/summary_seeds.csv`, which carries a `corpus` column so the MLAAD-tiny pilot rows and the `m1aad_paired` rows are distinguishable.

Every training run writes `config.json` beside its checkpoint. Every evaluation writes a JSON file containing the four cells, the bootstrap interval and the gate decision. The environment used for the instrument check is recorded in `docs/environment.txt`.

APPENDIX I

RECORDED DEVIATIONS

The authoritative log is `docs/pre-registration.md`; this appendix exists so a reader of the bound thesis has the record without the repository. Each entry carries its date, its reason, and whether it touched the decision rule.

1. **AASIST substituted as primary detector** (instrument check, before any test cell). Reason: validation EER plateaued at 30–47%. Decision rule untouched.
2. **Generator restriction to twelve families**. Reason: download cost. Matched-generator property preserved. Decision rule untouched.
3. **Secondary accent operationalisation not run**. Reason: time went to the full-scale main run and the gate. Reported as a limitation.
4. **Gate re-run on a complete corpus** (2026-10-23). Reason: the 2026-10-22 run used an incomplete MLAAD-tiny download. The archived run is retained and forbidden as a source of quoted numbers. Decision rule untouched.

APPENDIX J**COMPLETION RECORD**

The task ledger is `docs/tasks/completion-checklist.csv`; it records every task's state, its artifact and, where a task is blocked, what blocks it. Tasks that were deferred rather than done are written there as deferred, with the decision the next person has to make. Nothing in this thesis is silently half-done; the ledger is where a reader checks that claim.

BIBLIOGRAPHY

- ArFake authors. ArFake: A multi-dialect arabic audio deepfake benchmark. In *Interspeech*, 2026.
- Houwei Cao, David G. Cooper, Michael K. Keutmann, Ruben C. Gur, Ani Nenkova, and Ragini Verma. CREMA-D: Crowd-sourced emotional multimodal actors dataset. *IEEE Transactions on Affective Computing*, 2014.
- Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman and Hall/CRC, 1993.
- Nicholas Evans, Junichi Yamagishi, Ante Jukic, Xuechen Du, Sriram Dumpala, Tomi Kinnunen, et al. ASVspoof 2021: Towards spoofed and deepfake speech detection in the wild. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.
- Joel Frank and Lea Schönherr. WaveFake: A data set to facilitate audio deepfake detection. In *NeurIPS Datasets and Benchmarks Track*, 2021.
- ITU-T. Recommendation G.711: Pulse code modulation (PCM) of voice frequencies. International Telecommunication Union, 1988.
- Jee-weon Jung, Hee-Soo Heo, Hemlata Tak, Hye-jin Shim, Joon Son Chung, Bong-Jin Lee, Ha-Jin Yu, and Nicholas Evans. AASIST: Audio anti-spoofing using integrated spectro-temporal graph attention networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.
- Minu Kim et al. Language orthogonalization for zero-shot cross-lingual audio deepfake detection, 2026.
- Tomi Kinnunen, Kong Aik Lee, Héctor Delgado, Nicholas Evans, Massimiliano Todisco, Md Sahidullah, Junichi Yamagishi, and Douglas A. Reynolds. t-DCF: A detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification. In *Odyssey: The Speaker and Language Recognition Workshop*, 2018.
- R. Klein et al. The effect of telephony transmission on source tracing of audio deepfakes. In *Odyssey: The Speaker and Language Recognition Workshop*, 2026.
- Galina Lavrentyeva, Sergey Novoselov, Egor Malykh, Alexander Kozlov, Oleg Kudashev, and Yuri Matveev. Audio replay attack detection with deep learning frameworks. In *Interspeech*, 2017.
- X. Liu et al. Towards quantifying and reducing language mismatch effects in cross-lingual speech anti-spoofing. In *IEEE Spoken Language Technology Workshop (SLT)*, 2024.
- Nicolas M. Müller, Pavel Czempin, Franziska Dieckmann, Adam Froghyar, and Konstantin Böttinger. Does audio deepfake detection generalize? In *Interspeech*, 2022.
- Nicolas M. Müller, Piotr Kawa, Wei Herng Choong, Edresson Casanova, Eren Gölge, Théo Müller, Piotr Syga, Philip Sperl, Yanis Elisha, and Shyam Vali. MLAAD: The multi-language audio anti-spoofing dataset. In *International Joint Conference on Neural Networks (IJCNN)*, 2024. arXiv:2401.09512.
- Y. Shi et al. Benchmarking audio deepfake detection robustness in real-world communication scenarios. In *European Signal Processing Conference (EUSIPCO)*, 2025.
- Hemlata Tak, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans, and Anthony Larcher. End-to-end anti-spoofing with RawNet2. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.
- Christophe Veaux, Junichi Yamagishi, and Kirsten MacDonald. CSTR VCTK Corpus: English multi-speaker corpus for cstr voice cloning toolkit, 2017. University of Edinburgh.
- Xin Wang et al. ASVspoof 5: Crowdsourced speech data, deepfakes, and adversarial attacks at scale, 2025. arXiv:2502.08857.
- Y. Xu et al. RealLink: A spoofed speech dataset for real communication links, 2025. Journal of Cybersecurity (in Chinese).
- Guanlong Zhao, Sinem Sonsaat, Alif Silpachai, Ivana Lucic, Evgeny Chukharev-Hudilainen, John Levis, and Ricardo Gutierrez-Osuna. L2-ARCTIC: A non-native english speech corpus. In *Interspeech*, 2018.