What AI-Music Detectors Actually Detect: Scaling Up, the Round-Trip Test, and Why Two Methods Become One

Part III (final) of a series. Part I documented a single-generator piano detector and its confounds; Part II surveyed alternative, generalisation-oriented approaches and tested simple hand-crafted decoder-artefact features. This part scales the supervised detector to many generators, implements the neural-decoder round-trip as a second layer, and arrives at a unifying conclusion about what these detectors are really doing.

Daniel Bordovský[1]

Abstract

We scaled the supervised approach to a multi-generator, multi-genre detector (a ResNet-18 trained on ~13,500 tracks spanning twelve AI generators and a large real-music corpus, ~210,000 spectrogram segments) and implemented the neural-decoder *round-trip* as a second, generator-agnostic layer. The scaled CNN reached ~96% validation accuracy on unseen tracks of *known* generators and was robust to bitrate, but failed in two revealing ways: it classified an unseen generator (Lyria) as "100% human," and failed on an *unseen version* of a generator it knew (Suno v4), even though it had trained on both v1–v3.5 *and* v5.5, confirming that any version absent from training degrades it. The round-trip layer, trained on a single codec (Encodec), worked perfectly on its own task (99.8% on its own reconstructions) but transferred to nothing else, flagging Suno, Udio, and even MusicGen at ~0% — a consequence of each decoder leaving a *different* fingerprint. A single experiment then unified the picture: a *real* track passed only through Encodec was flagged as AI by both the round-trip (99.8%) and the CNN (97.7%). The supervised CNN is therefore itself, in large part, a neural-decoder-artefact detector. This explains every result, both detectors succeed on generators whose codec is represented in their experience and fail identically on those with proprietary decoders, while both false-positive on real audio rendered through a familiar codec. The two layers are correlated, not complementary; they share a blind spot. We conclude that robust detection requires three things in concert: exhaustive supervised coverage, exhaustive codec coverage, and a fundamentally different, structural analysis of the music itself, and that this is realistically the domain of well-resourced, well-motivated teams.

[1]Independent Researcher. Correspondence to: borrtex@borrtex.com

1. Introduction

Part I built a single-generator piano detector and showed its high accuracy rested partly on confounds; Part II surveyed the routes toward generalisation and found that simple hand-crafted decoder-artefact features top out modestly and are blind to the best commercial generators. Two questions remained: does *scaling* the supervised detector across many generators and genres buy real generalisation, and does the *learned* round-trip (which we did not implement in Part II) succeed where hand-crafted features failed? This part answers both, and the answers converge on a single, clarifying observation about what these systems detect.

2. A multi-generator detector at scale

We trained a ResNet-18 on mel-spectrograms drawn from a large, diverse corpus: the AIME dataset (lossless audio from twelve generators, including Suno, Udio, MusicGen, Stable Audio, Riffusion, AudioLDM, Mustango and others) as the AI class, and a large genre-balanced real-music collection (FMA, MTG-Jamendo) as the human class. After bitrate unification and track-level splitting, the corpus comprised 13,503 tracks (~210,000 three-second segments; 169,332 training / 41,330 validation images), with balanced class weighting.

To keep the detector from simply favouring the majority class, we applied inverse-frequency class weights in the loss — ≈1.39 for AI and ≈0.78 for real — so that each class contributes equally to training regardless of its raw segment count.

Validation accuracy reached ~96% on unseen *tracks of known generators*, and the detector was robust to bitrate — a real track at 128 kbps was correctly classified, this is genuine generalisation across content and genre *within the set of generators it has seen*.

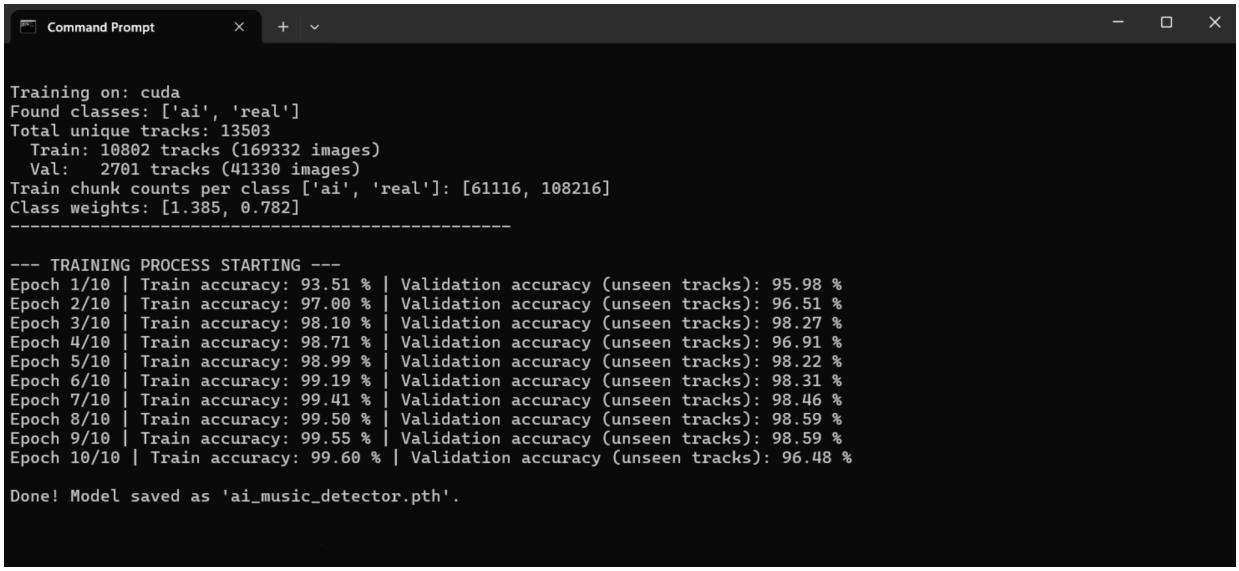

Figure 1. Training the detector after cleaning (13,503 tracks, track-level split). Validation accuracy is high across epochs (95.98%–98.59%).

3. Generalisation under stress

Two out-of-distribution tests exposed the limits. **An unseen generator.** A track from Google's Lyria 3 — entirely absent from training — was classified as 100% human. This is the canonical out-of-distribution failure: confronted with a generator whose characteristics it never learned, the detector defaults to the safe-looking class, producing a confident false negative. This echoes Part I, where the same failure appeared, but there it was easy to write off as a symptom of scale: a single genre, one generator, fewer than a hundred tracks. Here the dataset is more than 100x larger, spans twelve generators, and is genre-diverse, so the open question was whether that breadth would cure the problem. It did not, Lyria 3 still read as 100% human. That is the important point: the unseen-generator failure is *not* an artefact of too little data. It survives an order of magnitude more training material and broad genre coverage, which suggests it is intrinsic to the supervised approach rather than a gap that more data closes.

An unseen *version* of a known generator. The training data contained Suno v1–v3.5 *and* v5.5, both of which the detector recognised correctly. But Suno v4, a version it had never seen, sitting *between* known releases, was misclassified, falling to 76.8% "real." This is the more important result, the model knew v3.5 and v5.5 yet failed on the v4 wedged between them, so it is not *newness* that defeats it but *absence from training*.

Part 3: What AI-Music Detectors Actually Detect

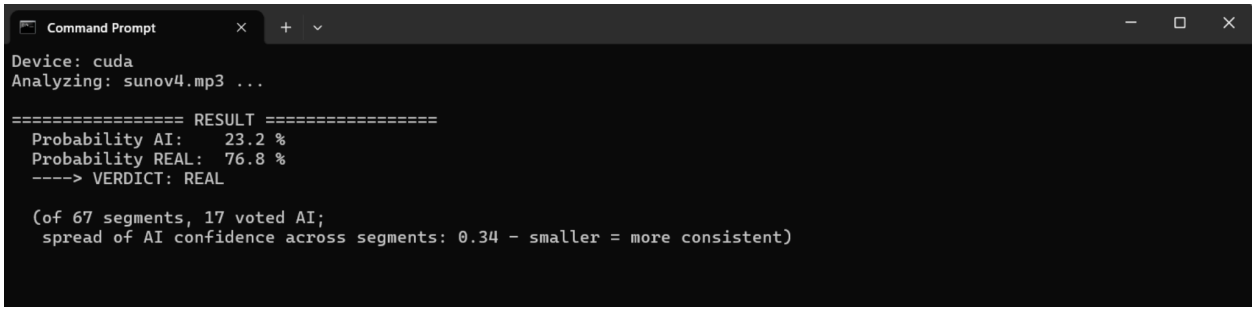

Figure 2. Suno V.4 was miscategorized. The detector learned the fingerprints of specific versions, not an invariant property of the generator; any version it did not see, regardless of release order, reads as unfamiliar.

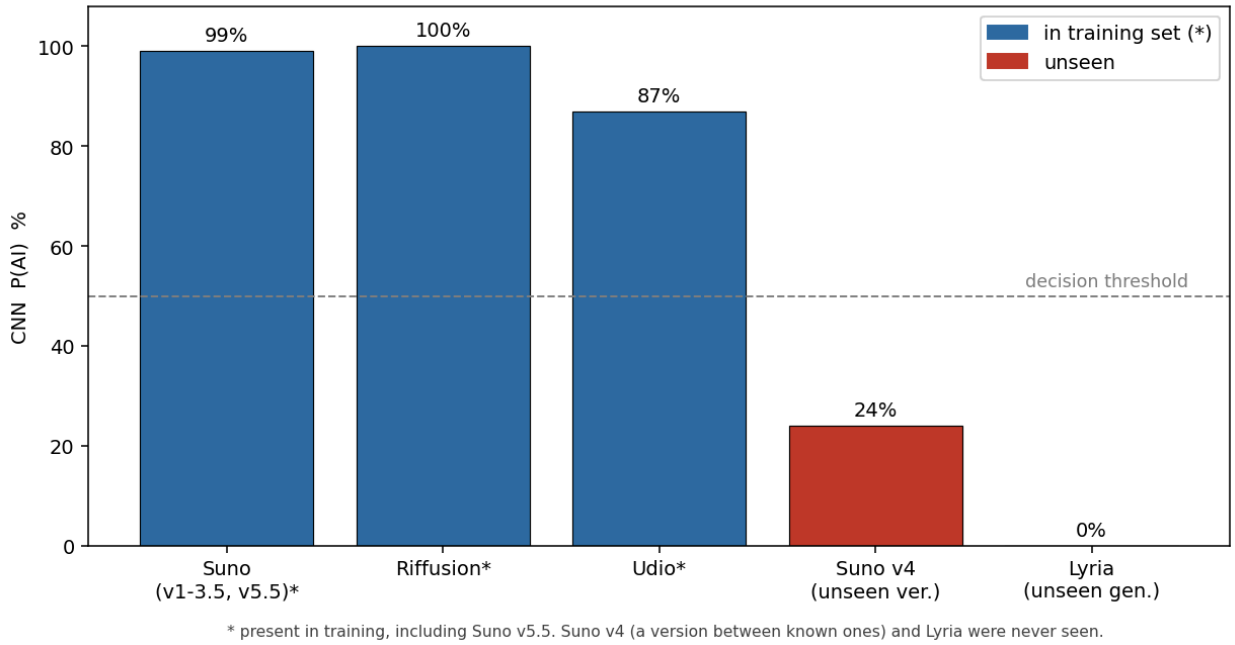

Figure 3. The generalisation cliff; confidence collapses on any version it never trained on.

4. The round-trip layer and its codec-specificity

We implemented round-trip as a second layer: each real track was reconstructed through a neural codec (Encodec, 24 kHz), and a classifier was trained to distinguish original anchors from their reconstructions, a setup that is confound-free by construction, since the pair differs only by the decoder artefact (Part II, §3). Trained this way, the model can in principle flag any audio carrying that artefact, including prompt-generated music.

The model learned its task perfectly, on held-out Encodec reconstructions it scored 99.8%. But it transferred to nothing else. On real AI music; Suno, Udio, and even MusicGen, it reported ~0% artefact. MusicGen was the telling case as it *uses Encodec*, but a *different* variant (the 32 kHz music model), and that difference alone was enough to make its artefact invisible to a detector trained on the 24 kHz codec. The decoder fingerprint is not merely generator-specific; it is *codec-version*-specific. A single-codec round-trip therefore detects essentially one thing, that exact codec, and generalises to no real generator.

5. The unifying experiment: both detectors detect the decoder

A single test then tied the entire series together. We took a real track, passed it through Encodec (nothing else), and ran it through both models. The round-trip, as expected, flagged it at 99.8%. But the supervised CNN also flagged it as AI, at 97.7%, even though it is real music and the CNN never saw this track.

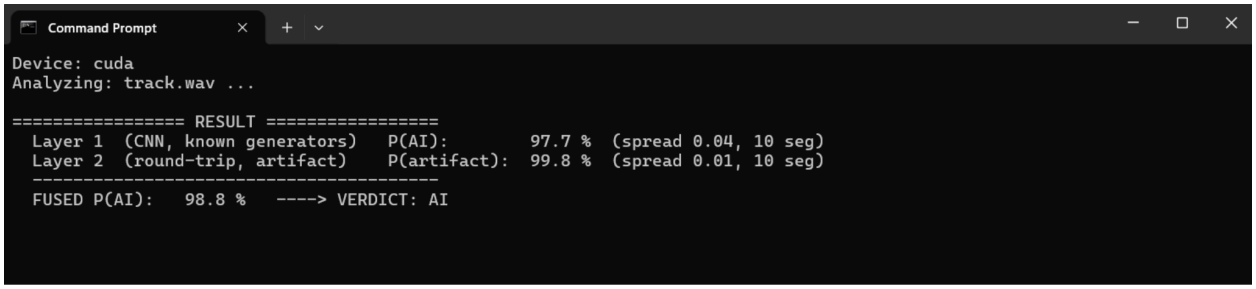

Figure 4. The unifying experiment. A real human track, passed only through the Encodec codec (nothing else changed), is flagged as AI by both models.

The supervised CNN is itself, in large part, a neural-decoder-artefact detector. Because several of its AI training examples were rendered through neural codecs, it learned to associate that fingerprint with "AI."

Part 3: What AI-Music Detectors Actually Detect

This single fact explains the whole of Sections 2–4:

- It succeeds on codec-based generators because their artefact is represented in its training.
- It fails on Suno, Udio, and Lyria 3 because those use proprietary decoders whose artefact it never learned, so they look "clean" and read as human.
- It false-positives on real audio rendered through a familiar codec, because it detects the *rendering*, not the *authorship*.

Two consequences follow. First, the supervised CNN and the round-trip layer are **correlated, not complementary**: they key on the same family of artefacts, so they share the same blind spot (proprietary-decoder generators) and the same false trigger (familiar-codec audio). Ensembling them adds little of the generalisation we hoped for. Second, the false-positive vector is real and worth stating plainly — *any* genuine recording passed through a neural codec, increasingly common in streaming and transmission, can be mislabelled AI. This is the literature's "detects rendering, not authorship" caveat, reproduced here on our own detector with a single round-trip.

Part 3: What AI-Music Detectors Actually Detect

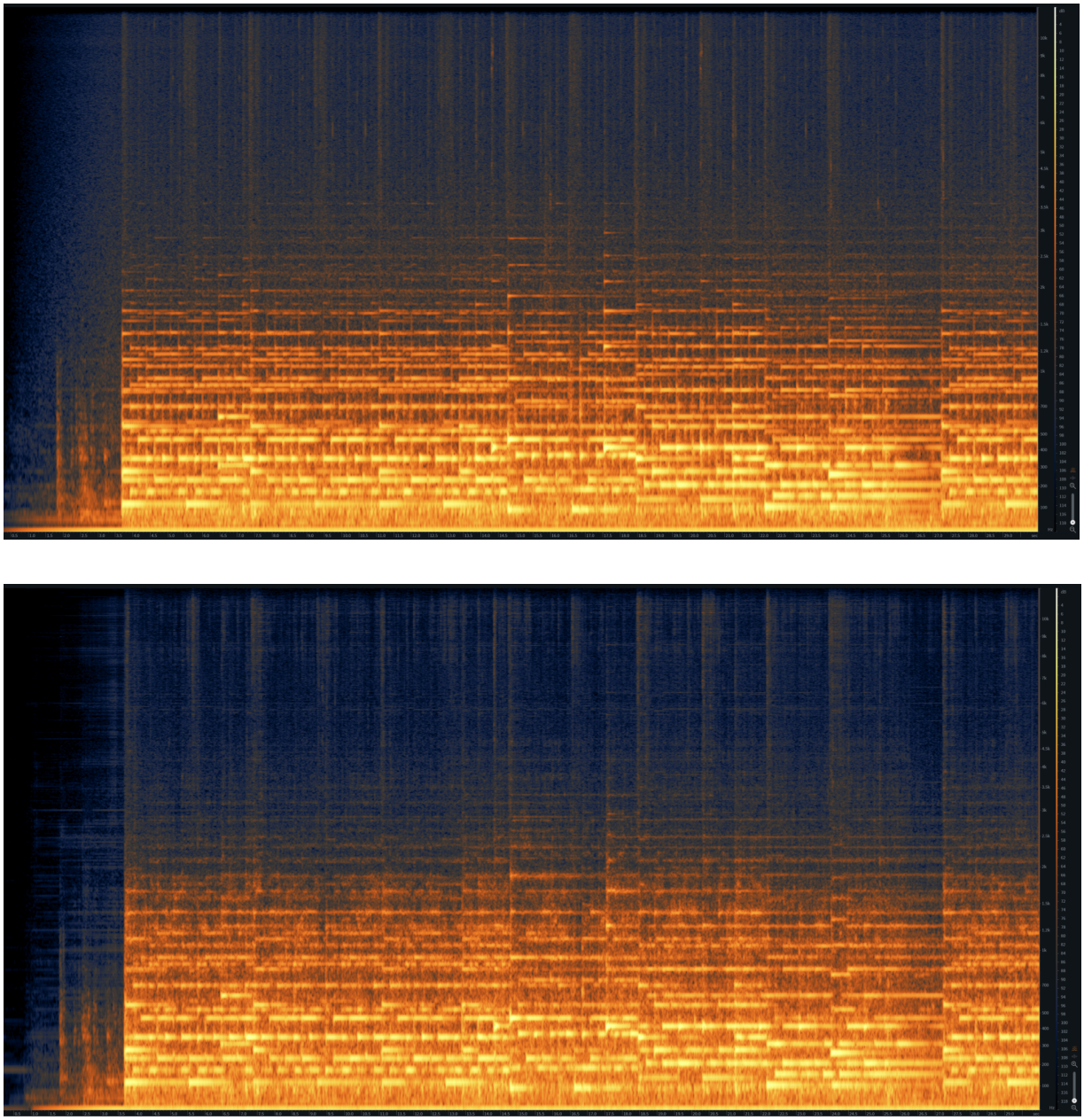

Figure 5. The same real track before (top) and after (bottom) passing through the Encodec codec, that round-trip is the only difference between them. The reconstruction coarsens and smears the fine detail of the upper frequencies, replacing the original's crisp harmonic structure with the grainier, more uniform texture characteristic of a neural decoder. This codec fingerprint is precisely what both detectors react to: it is why a genuine recording, altered in no other way, was flagged as AI by both the round-trip and the CNN (Figure 4).

6. Discussion: two treadmills and a third path

The series converges on a clear account of why robust AI-music detection is hard, and what it would actually require.

Treadmill one — supervised coverage. A supervised detector breaks on anything it has not seen. To be reliable it would need training data covering *every relevant generator*, *every released version* of each (since versions differ, §3), and *broad genre diversity* — and it would need to be retrained on each new release. The detector must, in effect, have already met every kind of AI track that will ever be tested. The moment it meets something new, it defaults to "human."

Treadmill two — codec coverage. The decoder-artefact layer is real and principled (the artefact exists, as Part II and this part confirm), but each decoder leaves a different fingerprint, down to the codec version (§4). To be useful it would need real music reconstructed through *every codec at every sample rate* a generator might use, and it breaks the moment a vendor adopts a new decoder. It trades the supervised treadmill for a codec treadmill running alongside it.

A third, more durable path. Both treadmills chase a moving target at the level of the rendered *signal*. The more robust direction is to analyse the music *as music*, its structure rather than its surface: the logic and evolution of chord progressions, melodic and harmonic coherence over time, the velocity and micro-timing of individual notes and whether their variation looks naturalistic, the interaction and balance of instruments across an arrangement. These are properties a generator must get right to be *musically* convincing, and they are far harder to fake than a noise floor or to erase with a better decoder. This is the hardest path to build and the least mature, but it is the one that does not reset every time a vendor ships an update.

A genuinely reliable real-world detector is therefore not one method but an ensemble of all three, continuously maintained: broad supervised coverage for known generators, multi-codec artefact detection for some unseen ones, and structural analysis as the generalising backstop, together with honest uncertainty when an input matches none of them. That standing maintenance, breadth of data, and engineering are realistically the province of well-resourced teams with both the means and the motivation to keep a detector current — and the mandate to disclose, clearly and accountably, whether a track is AI, which is the point of building one at all.

6.1. Foundation features (MERT)

After the main experiments we ran one further test, on an approach that Part II had only surveyed: replacing the from-scratch CNN with a frozen music foundation model (MERT) as the feature extractor, training only a small classifier on its 768-dimensional embeddings.

No spectrograms are involved, MERT reads the waveform directly, and because the features are pretrained, the classifier needs far less data; we used 10-second excerpts from 6,000 tracks per class. The outcome was instructive precisely because it was mixed. In general the MERT detector was *less confident* than the CNN and *biased toward the "human" verdict*. a known-AI track scored only 40% AI (a miss), while real tracks were consistently classified as human, although with varying confidence, sometimes as low as 67% human.

Yet on the cases that had defeated the CNN, the richer features helped. MERT flagged Udio at 90% AI — confidently correct, where the CNN had been shaky (Part I), and called the unseen Suno v4 at 83.7% AI, landing it on the right side of the decision, where the CNN had failed outright (24% AI / 76% human). So the foundation features did generalise somewhat better across generators and versions, exactly where one would hope.

The wall, however, held. The genuinely unseen generator, **Lyria 3, still read as 79.4% human**, the same failure as the CNN. Better features migh sharpen the supervised approach at the margins and can rescue individual hard cases, but they do not escape the fundamental limit. A generator with a wholly unfamiliar character is invisible to MERT just as it is to the CNN. The generalisation wall is feature-agnostic, it holds even for the strongest music foundation model available, which strengthens rather than weakens the central claim thet the unseen-generator problem is intrinsic to the supervised approach, not a deficiency of any particular front end.

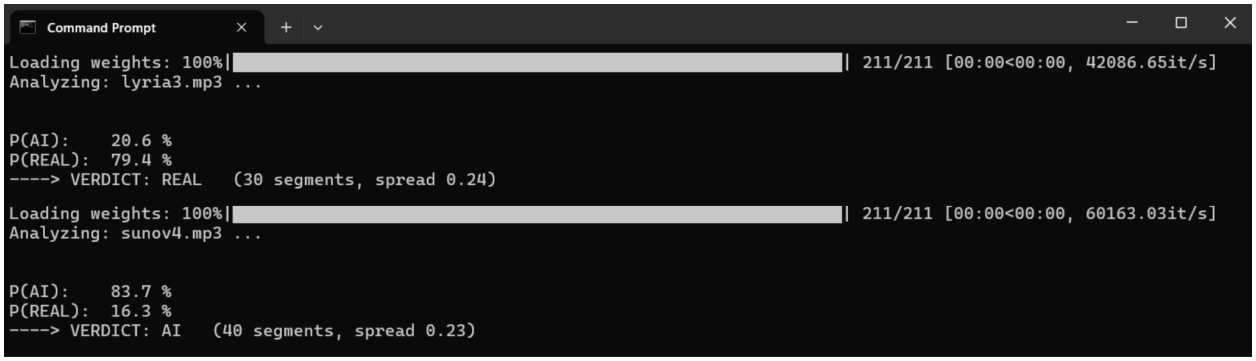

Figure 6. MERT also failed to correctly detect a completely unseen track from an unseen Lyria 3 generator by Google, but was better than CNN in detecting an unseen version of Suno.

6.2. Beyond the arms race

One caveat tempers even this hope, and it is worth stating plainly. These generators are trained on millions of real recordings, so there is no principled reason they cannot learn musical *structure* as well as they have learned timbre — convincing harmonic motion, naturalistic micro-timing, coherent development are, after all, exactly what the training objective rewards. Structural analysis is more durable than signal-level forensics because today's models are still weakest there, but it is not obviously *permanent*: as systems scale, the very properties we would use to catch them are the ones they are learning to reproduce. Whether "analyse the music as music" stays a detectable boundary or eventually closes like the others is, at this point, closer to a philosophical question than an empirical one — we may simply be naming the next surface to be mastered.

There is a pointed asymmetry in all of this. These generators were built on enormous corpora the companies are not obliged to — and largely do not — disclose, precisely because the provenance of that training data is legally fraught. The result is systems powerful enough to be nearly indistinguishable from human work, trained on material whose origins remain opaque, while the entire burden of transparency falls on the *output* side, in the form of detectors and disclosure labels built by everyone else. That a detector has to be reverse-engineered at all is, in part, a consequence of how little is disclosed at the source.

---

7. Limitations

These remain exploratory, single-operator experiments. The generalisation tests on Lyria and on Suno v4/v5.5 are individual tracks — illustrative, not statistical. The round-trip used a single codec; multi-codec training would broaden it (though, given even MusicGen escaped, likely not to Suno/Udio). The unifying experiment used one real-through-Encodec example; the effect is clear but its prevalence across content was not quantified. As throughout the series, the "human" corpus includes sample-library productions, so the contrast is generative-AI synthesis versus real-acoustic source. No claim here is a benchmark; the contribution is the mapping of where each method works, where it fails, and why.

8. Conclusion

Scaling the supervised detector across twelve generators bought real generalisation across content and genre, but not across generators or even versions: it called an unseen generator "100% human" and lost confidence on newer releases of a generator it knew. The learned round-trip worked flawlessly on its own task and transferred to nothing, defeated by codec specificity so fine that a different Encodec variant sufficed to hide the artefact. And a single experiment revealed that both methods are, at bottom, the same thing — neural-decoder-artefact detectors that succeed on familiar codecs, fail identically on proprietary ones, and mistake any codec-rendered real audio for AI. The honest end-point of three parts of experimentation is that signal-level detection is a perpetual arms race with two treadmills, and that the durable escape — analysing music as music — is also the hardest to build. A trustworthy detector is an ensemble kept perpetually current, which is why it belongs to those with the resources and the resolve to maintain it. The most useful thing a small experiment can contribute is not a detector but a clear map of the terrain, which generators are caught, which slip through, and exactly why — so that the larger effort knows what it is up against.

Notes. The scaled detector used a ResNet-18 over mel-spectrograms (PyTorch/torchvision), trained on the AIME dataset and real-music collections (FMA, MTG-Jamendo). The round-trip used Hugging Face Encodec (24 kHz) via the transformers library. Generalisation and unifying results come from small single-run tests and are reported as approximate. The neural-decoder artefact theory and round-trip method are due to Afchar, Meseguer-Brocal and Hennequin (Deezer Research).

References

[1] Afchar, D., Meseguer-Brocal, G., Akesbi, K., & Hennequin, R. (2025). A Fourier Explanation of AI-music Artifacts. Proceedings of the 26th International Society for Music Information Retrieval Conference (ISMIR), Daejeon, South Korea. https://arxiv.org/abs/2506.19108

[2] Afchar, D., Meseguer-Brocal, G., & Hennequin, R. (2025). AI-Generated Music Detection and its Challenges. IEEE ICASSP 2025. https://arxiv.org/abs/2501.10111