Detector public research

Evaluation protocol for AI-music detectors

Generator-disjoint testing, human calibration, and reproducible reporting

Author Sonny Fullerton
Version 1.0
Source Public repository at commit `4143b39`
Date July 2026
Status Public technical protocol

A model-agnostic procedure for evaluating cross-generator transfer at operationally meaningful false-positive rates. The protocol is intended for detector authors, auditors, and reviewers who need comparable evidence without requiring public model weights.

Protocol in one page

1. Define the score interface before viewing test labels.
2. Group generated audio by generator family and content by song identity.
3. Hold one entire generator family out of training per fold.
4. Keep all related clips from a song in one partition.
5. Reserve a verified-human calibration set, disjoint from training and test.
6. Select the threshold for the declared target human FPR on calibration data only.
7. Freeze the threshold and score the held-out generator plus human test set.
8. Repeat for every generator family; report each fold, mean, and worst case.
9. Repeat with transformations applied symmetrically to both classes.
10. Publish manifests, counts, hashes, configuration, and a machine-readable result record.

Disclosure boundary. This publication documents the evaluation framework and aggregate reference results. It does not disclose proprietary features, model topology, training recipe, weights, source code, or private data lineage.

1 Scope and threat model

The target is binary review triage: estimating whether an audio item is consistent with AI generation under a stated benchmark. The protocol evaluates *cross-generator generalisation*, not provider attribution. It assumes no active adversary in the core run; adversarial edits belong in a separately labelled stress suite.

The primary failure being controlled is a false positive on verified human music. Secondary failures include memorising a generator family, leaking song identity across splits, learning source-corpus differences, and exploiting transformations applied to only one class.

2 Required inputs

2.1 Scorer contract

The submitted detector must implement a deterministic callable that maps a waveform and sample rate to one scalar:

$$f(\text{waveform}, \text{sample rate}) \to s \in [0, 1],$$

where larger values indicate stronger AI-generated evidence. Any internal segmentation and aggregation must be fixed in advance and described. The evaluator must not require access to features, weights, or architecture.

2.2 Manifest fields

Each item should carry, at minimum:

- stable item and song identifiers;
- class label and generator family where applicable;

- artist or creator identifier when available;
- source corpus, acquisition path, and license status;
- audio duration, format, sample rate, and transformation condition;
- cryptographic hash or equivalent integrity identifier.

No item may enter a public bundle if its redistribution rights are unclear. A manifest may point to data without redistributing it.

3 Split construction

3.1 Generator-disjoint folds

For generator families $G = \{g_1, \ldots, g_K\}$, construct fold k so that all outputs from g_k are excluded from model training and development. The fold test set contains outputs from g_k and a held-out human test set. Repeat until every family has served as the unseen family once.

Hyperparameters may be fixed globally before the LOGO run. They must not be tuned on the held-out family's labels or scores. If retraining for every fold is infeasible, that limitation must be declared and the result must not be called LOGO.

3.2 Content and creator separation

All excerpts, encodes, covers, or transformations derived from one song belong to one split. Where artist identity is available, use artist-disjoint human partitions. Content-matched human originals and AI covers are preferred because they reduce the chance that genre, composition, or mastering differences stand in for the target label.

3.3 Optional temporal holdout

A temporal holdout may be added to estimate drift, but cannot replace LOGO. Training and calibration timestamps must precede the temporal test window, and generator-version changes should be recorded where known.

4 Calibration

Let $H_{cal} = \{h_1, \ldots, h_n\}$ be a verified-human calibration set and let $s_i = f(h_i)$. For a target FPR α, choose a conservative upper-tail empirical quantile of the human scores. One finite-sample form uses the order statistic

$$t_\alpha = s_{(\lceil (n+1)(1-\alpha) \rceil)},$$

with scores sorted in ascending order and the index clipped to the available range. The precise quantile convention must be recorded in code and metadata.

Calibration data must be disjoint from training, hyperparameter selection, and final testing. Reusing the test set to choose t_α invalidates the operating-point claim.

4.1 Sample-size warning

At $\alpha = 0.01$, the decision is governed by the extreme human-score tail. With only a few hundred calibration examples, one or two observations can move the threshold materially. Report n, the selected threshold, and a binomial confidence interval for realized test FPR. Do not describe a nominal 1% target as an observed or guaranteed 1% error rate.

4.2 Conditional calibration

If human score distributions differ across prespecified groups such as genre, estimate a threshold $t_{\alpha,g}$ within group g. Groups must be defined without test-label optimization. Always report group calibration and test counts, realized FPR, and the recall trade-off. If a group lacks enough data, fall back to the global threshold and disclose the limitation.

5 Primary metrics

For every held-out generator, report:

- ROC AUC;
- recall at target human FPRs, minimally 1% and 5%;
- generated and human sample counts;
- realized human test FPR at the frozen threshold;
- confidence intervals or bootstrap intervals where defensible.

Across folds report the unweighted mean and worst case. A sample-weighted mean may be added but may not replace them. Accuracy and F1 may appear only as secondary metrics because both depend on class prevalence and an operating threshold that may not match deployment.

6 Transformation stress tests

Every transformation condition must be *matched*: the same operator and parameters are applied to human and generated items. Recommended ordinary-use conditions include lossy codec export, sample-rate conversion, gain change, short excerpting, and common mastering operations. Each condition receives its own calibrated threshold unless the experiment explicitly tests threshold transfer.

Report clean and transformed results separately. Do not average them into one number. Preserve the content split when creating transformed derivatives, and cache transformation parameters in the manifest.

7 Optional extensions

7.1 Calibration-tail audit

Inspect the highest-scoring verified-human calibration items after the model and threshold are frozen. The audit may identify mislabeled data or a vulnerable subgroup, but any exclusion rule created from the audit must be rerun on a fresh calibration set. Post-hoc cleaning without re-evaluation is not valid evidence.

7.2 Multiple-excerpt decisions

When a track produces several segment scores, declare the aggregation rule before evaluation. Report both segment-level and track-level performance, the number and duration of excerpts, dependence assumptions, and the failure behavior when only one excerpt is available.

8 Submission record

A reproducible result package should contain:

- a unique system name and version;
- repository commit and environment lockfile;
- immutable manifest hashes and split seed;
- calibration policy and target FPRs;
- transformation definitions;
- per-fold machine-readable metrics;
- a plain-language limitations statement;
- disclosure of unavailable code, weights, audio, or licenses.

Reject a submission if it omits sample counts, reports only accuracy or F1, overlaps calibration and test data, uses cross-source classes without matching or controls, or publishes a mean without the worst-performing generator.

9 Minimal result schema

Field	Meaning
`system_id`	Immutable detector/version identifier
`source_commit`	Evaluator and configuration commit
`fold.generator`	Entire generator family excluded from training
`n_human`, `n_generated`	Test sample counts
`calibration.n`, `target_fpr`	Human calibration count and declared target
`threshold`	Frozen score threshold
`auc`, `recall_at_fpr`	Fold metrics
`condition`	Clean or exact matched transformation
`manifest_hash`	Integrity link to evaluated items

References

S. Fullerton. *AI-Music-Detection: public benchmark and evaluation framework*. GitHub repository, commit `4143b39`, 2026. https://github.com/sonnyfully/AI-Music-Detection

A. N. Angelopoulos and S. Bates. “A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification.” *arXiv:2107.07511*, 2021.

S. Bates, E. Candes, L. Lei, Y. Romano, and M. Sesia. “Distribution-Free, Risk-Controlling Prediction Sets.” *arXiv:2101.02703*, 2021.

All benchmark values in this document are reproduced from the public repository at commit `4143b39`.