

Auditing the exported p value of a neural information detector

Sidharth Rajmohan
Independent researcher, India
26 October 2026

Abstract

A detector's descriptive score and its reported significance need separate validation. I audit the standalone neural mutual-information p-value exported by a pinned implementation of Audit the Whisper, a detector reused in a subsequent collusion study. Its nominal-null fits retain the observed joint sample while changing only the product-reference sample, and use 120 optimization steps; the observed fit uses 200. An exact constructed example shows that this reference can retain the dependence being tested. A separate, predeclared neural mechanism experiment varies joint construction and training budget across 40 synthetic datasets. Under correlation 0.9, equal-budget reference means are 0.52703 nats when the joint is retained and 0.01686 when it is permuted. These values describe a local implementation. Operational performance and Gaussian-tail calibration from three fits per run pair require separate evaluation. I propose a local output-only quarantine that preserves diagnostic scores while marking inferential output unavailable. A bounded synthetic execution of the pinned detector and estimator with local adapters preserves scores under the patch; full-pipeline integration and calibration remain untested. A source trace distinguishes a stress-script consumer of the exported value from the default union, which excludes this detector, and the raw-score ROC and externally derived honest-calibration operating thresholds. The audit provides a source-level diagnosis and regression checks for the proposed quarantine. A calibrated replacement test remains open.

1 Motivation and related work

An exported p-value needs validation against a suitable null distribution. Downstream test combinations rely on that calibration. This audit follows one such interface in Audit the Whisper [1, 2], from sample construction to its standalone neural-MI output and actual consumers.

Kumar's participant report reuses the detector, prints pooled MI p-values of 0.018 and 0.0007, and declines to rely on them after numerical warnings [3]. The cause of those warnings remains unresolved. At the pinned upstream revision, an adaptive low-rate stress script also passes the exported MI value directly to a Holm combination. The default metric union instead excludes MI. MI ROC uses raw scores; separate operating thresholds are derived from honest-run calibration, whose validity remains unassessed here.

Neural information estimation [5] and its statistical limitations [6] are established subjects. Achar's contemporaneous participant audit examines evidence conditions and dependent units in released collusion instruments [4]; Kumar also identifies other detector and benchmark limitations [3]. The question here is whether this implementation's nominal-null refits represent independence and whether unsupported confidence can be isolated without discarding its descriptive scores. This study builds on those audits and established limitations of mutual-information estimation.

2 Source contract and methods

What the implementation compares

I inspect commit 48e6713d3e26b49dd94fa6645b4aeac6cb25a229 of the upstream repository [2]. The same two MI source files occur in the starter-kit revision used by Kumar [3]. The neural estimator maximizes a Donsker-Varadhan objective of the form

$$J(T; P, Q) = E_P[T] - \log E_Q[\exp(T)],$$

where P supplies paired observations and Q approximates the product of their marginals. All information quantities in this report use natural logarithms and are in nats. A product-reference sample is part of estimating information under P . Resampling Q alone leaves the tested joint P unchanged.

The source concatenates aligned feature vectors from two runs, forms a marginal reference by permuting Y , and samples at most 512 rows. Table 1 records the calls in auditor/detectors/mi.py. Each pair has one observed DV fit and three nominal-null DV fits. The reported statistic averages observed estimates across pairs; the Gaussian reference mean and variance pool nominal-null estimates. Run-pair estimates can be dependent when they share runs. An average across pairs and an individual reference estimate are distinct sampling units.

Call	Joint input	Product reference	Steps and seed
Observed DV	joint_sample	marg_sample	200; 42
Nominal-null DV	same joint_sample	marg_rep	120; 100-102

Table 1. Pinned source call contract. Only reference pairing changes among the input arrays; the observed joint remains fixed. Seeds and budgets also differ as shown. The estimator uses its supplied joint unchanged. An auxiliary NWJ fit is separate from the DV-based exported tail value.

Exact example and bounded neural experiment

For balanced binary pairs with $X = Y$, use $T(x,y) = 8$ when $x = y$ and zero otherwise. Against the exact product distribution, J is 0.69281177 nats. Retaining that joint and replacing the reference with the same exact product leaves J unchanged. Under a genuinely independent joint and product reference, the optimum over the fixed family $T_c(x,y) = c \mathbf{1}[x=y]$, with c in $\{-8, -4, -2, -1, 0, 1, 2, 4, 8\}$, is zero. The fixed nonconstant critic itself gives a negative bound under independence. This example isolates the distribution being tested without neural fitting or unequal budgets.

A separate protocol, fixed before execution, specifies 20 iid Gaussian datasets for each correlation $\rho = 0$ and 0.9, with 128 pairs per dataset. A local NumPy 2-32-1 ReLU critic uses Adam, float64 arithmetic, and a stabilized DV objective. Four paired cells cross retained versus permuted tested joint with 120 versus 200 reference-fit steps. Observed fits always use 200 steps. References, initialization, and minibatch streams are matched across paired constructions; the two checkpoints share an optimization trajectory.

This experiment evaluates the mechanism with a local implementation using two-dimensional synthetic inputs and standard single-averaging gradients. The source applies an additional batch division. The local experiment omits upstream run-pair aggregation, feature extraction, and other detector branches. The dataset is the replication unit. Its three reference fits are repeated estimates within that unit. Reusing X and Z across correlations pairs the conditions, so analysis uses 20 independent datasets per condition.

3 Results

All 40 predeclared datasets completed, with 280 fit trajectories, no failures, and finite recorded outputs. Mean observed scores are 0.01631 nats for $\rho = 0$ and 0.47817 for $\rho = 0.9$. Table 2 gives the reference means and the secondary Gaussian-tail diagnostics.

Reference construction	Mean at rho 0	Mean at rho 0.9	Tail ≤ 0.05 at rho 0	Tail ≤ 0.05 at rho 0.9
Retained joint, 120 steps	0.00649	0.36609	10/20	12/20
Retained joint, 200 steps	0.01357	0.52703	4/20	1/20
Permuted joint, 120 steps	0.00818	0.00893	8/20	20/20
Permuted joint, 200 steps	0.01576	0.01686	3/20	20/20

Table 2. Local synthetic experiment. Means are in nats across 20 datasets per correlation. Tail columns count Gaussian-style scores based on three fits. Their calibration, operational false-positive rates, and power remain unmeasured.

At $\rho = 0.9$, the retained-joint reference mean exceeds the permuted-joint mean by 0.35716 nats at 120 steps and 0.51017 at 200 steps. Increasing the budget raises the retained-joint reference by 0.16094, compared with 0.00793 for the permuted joint. Joint construction and training budget have distinct effects. Equalizing budgets can make the retained-joint reference exceed the observed mean while retaining the original dependence.

Gaussian-tail calibration also requires validation after joint permutation. With equal budgets, three of 20 independent datasets produce scores at or below 0.05. This small experiment leaves the approximation and target error rate unresolved. With only three reference fits, the one-pair rank-style diagnostic cannot be below 1/4; differing observed and reference network seeds also prevent treating it as an exact exchangeable randomization test. The 1/4 bound applies to this one-pair diagnostic. The original detector pools values across multiple run pairs.

Source inspection rejected the initial DV-overflow hypothesis: the branch already uses max-shifted log-mean-exp. The generic exponential control overflows, while the source-style calculation remains finite. Any effect of the downstream warnings on reported DV p-values remains unresolved.

4 Quarantining unsupported confidence

The local patch quarantines the DV tail at the output interface. It retains the previous number as `details.uncalibrated_tail_score` and sets `details.p_value_status` to `unavailable`. Because the existing schema requires a float, `p_value` becomes 1.0 with an explicit conservative-no-rejection policy. This compatibility sentinel marks unavailable inference. Statistical independence and the presence or absence of collusion remain undetermined. A versioned nullable, status-aware API would communicate the missing inference more directly.

The patch keeps feature extraction, fitting, permutations, and aggregation unchanged. Nominal-null permutations consume the same random generator used by later observed run pairs; deleting them could alter descriptive scores. Static AST checks recover the original source after reversing only the output edits and preserve all estimator and RNG call expressions.

I then executed the pinned detector and estimator, before and after the patch, on one deterministic fixture with three runs, two agents, and eight rounds. Local adapters supplied synthetic features, result containers, and a Gaussian right-tail helper; the upstream feature pipeline and p-value helper remain untested. Across three run pairs, a spy confirmed identical complete estimator input hashes and arguments. Thirty actual fits preserved the raw statistic, 7.34553 nats, and every pre-existing details field exactly. The previous adapter-computed tail, 4.38×10^{-16} , was retained diagnostically while exported `p_value` became 1.0 with unavailable status. Insufficient-run and no-pair outputs were unchanged. The bounded module regression completed without warnings or failures. Error rates remain unmeasured.

A constructed spy case removed an MI-only Holm rejection. Source review finds that calibration, experiment, sweep, and adaptive-history exports retain details. In the adaptive script, the sentinel prevents rejection through MI while preserving family size; decisions and power may change. Other tests retain their own calibration requirements. The generic Stouffer helper accepts bare values without status handling, so the sentinel can alter its combined result. Such consumers need a predeclared policy for unavailable tests.

At this pin, the default union excludes MI. Raw-MI thresholds and ROC use `mi_values`. The fixture preserved those fields; full metric and adaptive pipelines remain untested. Upstream adoption is unconfirmed.

5 Discussion and conclusion

The source trace identifies a concrete gap between an exported inferential label and its reference construction. The exact example and local neural experiment explain why retaining the tested joint and changing optimization effort are distinct concerns. The published tool's false-positive rate, raw-score ROC, and externally derived operating thresholds remain outside this evaluation.

The immediate next step is to quarantine the uncalibrated tail score and propagate unavailable-inference status through exporters and test combiners, preserving descriptive scores. Before restoring inferential confidence, independently specify an independence null and defensible exchangeable units, match observed/reference fitting procedures and budgets, and use enough reference draws for the intended significance level. Test aggregate-level calibration on held-out null cases and power on controlled alternatives, then verify the full feature and consumer pipelines. Row permutation is justified by this toy's iid design. Nested run/agent/round data require their own exchangeability argument. These steps define future validation work.

Code and data

Paper and code: <https://github.com/dumko2001/whisper-mi-confidence-audit>

The original paper and repository predate this study [1, 2]. Contract tests, the synthetic experiment, the proposed patch, and the bounded module regression are documented in the accompanying reproducibility package. It records commands, dependencies, and source hashes; upstream MIT attribution is preserved. All experiments use synthetic inputs.

References

- [1] Tailor, O. (2025). Audit the Whisper: Detecting Steganographic Collusion in Multi-Agent LLMs. arXiv:2510.04303v2. <https://arxiv.org/html/2510.04303v2>
- [2] Tailor, O., and contributors (n.d.). Steganographic Collusion source repository. Commit 48e6713d3e26b49dd94fa6645b4aeac6cb25a229; accessed 26 October 2026. https://github.com/Ostailor/Steganographic_Collusion/tree/48e6713d3e26b49dd94fa6645b4aeac6cb25a229
- [3] Kumar, R. (2026). Talk Is Not Collusion. AI Collusion Research Sprint participant report. Pinned revision db8d24e893b1e8fe082e94f28d8d8097acd9182b. https://github.com/rkstu/AI_Collusion/blob/db8d24e893b1e8fe082e94f28d8d8097acd9182b/paper/paper.md
- [4] Achar, S. D. (2026). No Smoking Gun Without a Placebo: Four Conditions for AI Collusion Evidence. AI Collusion Research Sprint participant report. Pinned revision f673fca1c3ce69d1ece81584904b0ef33d42b408. <https://github.com/sdachar/collusion-evidence-conditions/blob/f673fca1c3ce69d1ece81584904b0ef33d42b408/report/SUBMISSION.md>
- [5] Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. (2018). Mutual Information Neural Estimation. ICML, PMLR 80, 531-540. <https://proceedings.mlr.press/v80/belghazi18a.html>
- [6] McAllester, D., and Stratos, K. (2020). Formal Limitations on the Measurement of Mutual Information. AISTATS, PMLR 108, 875-884. <https://proceedings.mlr.press/v108/mcallester20a.html>

Appendix A Reproduction and scope

Inputs and original source

The detector's pinned Git blob is 7f7717d60282829657619b8768510cb2dedf99f2; the estimator's is d87487a4c840554a3f8b9daf8ef65b015a92c49a. The source paths are auditor/detectors/mi.py and auditor/mi_estimators.py at the stated commit. The first AST audit used a detector copy with one extra terminal newline; it has the same AST as the exact 9,048-byte pinned file. The reproducibility package distinguishes these copies and records their hashes.

The consumer review covers 71 non-release Python files and checks 66 release-code mirrors by Git blob identity. It identifies direct adaptive-script consumption, raw-score paths, and default-union exclusion. The generation of individual published results was outside the trace's scope. Copied upstream code retains its MIT notice and original attribution.

Predeclared synthetic experiment

Dataset seeds are 1000-1019 for each ρ in $\{0, 0.9\}$. Draw X and Z independently from $N(0,1)$, and set $Y = \rho X + \sqrt{1-\rho^2} Z$. Each dataset has 128 iid pairs, with the same seed reusing X and Z across the two correlation conditions. The shared permutation-generator seed is 20000 plus the dataset seed; the observed reference and subsequent joint/reference permutations consume that stream. Observed and reference critic seeds are 42 and 100-102. The critic has two inputs, 32 ReLU hidden units, and one scalar output. Adam uses learning rate 0.001, beta1 0.9, beta2 0.999, and epsilon 10^{-8} ; batch size is 64. Joint and reference inputs share pooled column normalization, recomputed per call. Estimates are clipped below at zero. Evaluation uses the fitting samples. Predictive holdout performance remains unmeasured.

Python 3.12.14 and NumPy 2.3.5 were used with single-thread BLAS. The run took 5.83 seconds. A finite-difference preflight found maximum gradient error 9.17×10^{-11} . All 160 dataset-by-cell records, matched-reference hashes, paired contrasts, and descriptive Wilson intervals are preserved. The provenance.json file distinguishes recorded execution hashes from

portable file hashes. The intervals use 20 datasets per condition as independent units. No dataset or hyperparameter was replaced after inspecting results.

Reproduction commands

Run the static checks from a fresh working copy of the reproducibility package. These checks inspect upstream source through read-only parsing:

```
ROOT=$(pwd)
export AUDIT_SOURCE_DIR="$ROOT/source"
export ADAPTIVE_SOURCE_PATH="$ROOT/source/run_adaptive_lowrate.source.txt"
(cd contract_audit && python3 -m unittest -v \
    test_contract_audit.py test_consumer_interface.py)
(cd contract_audit && python3 verify_results.py)
python3 contract_audit/integration_regression.py
```

The patch verifier uses standard-library AST and mock checks and writes derived patch/validation files. Run it only in an expendable copy:

```
(cd patches && python3 verify_quarantine_patch.py)
```

The root README documents reproduction of the neural mechanism experiment from a fresh directory with NumPy installed. The script refuses to overwrite an existing result.

Bounded module regression

After the mechanism experiment and static patch review, a predeclared integration test executed only the hash-verified original detector, patched detector, and upstream estimator. The fixture contains three runs, two agents, and eight ordered rounds: three run pairs, each with 16 rows and 20 concatenated dimensions. Synthetic feature/container adapters and a local erfc Gaussian right-tail helper replace other repository imports. The local adapter computes the reported tail. The upstream pvalues.py helper remains untested.

Eight detector invocations comprise original/patched spy calls, original/patched actual-estimator calls, and four zero-fit edge-case calls. The execution made 30 spy calls and 30 actual fits totaling 4,560 optimization steps in 0.589 seconds, with one BLAS thread, no warnings, and no failures. All spy-recorded estimator arguments and complete joint/reference array hashes match before and after the patch across the three run pairs. Actual raw statistic and all pre-existing details match exactly. The original tail is retained as `uncalibrated_tail_score`; unavailable status, policy, and reference-kind fields are checked. A constructed spy consumer applies a local implementation of standard Holm. Execution of the adaptive script remains outside this regression.

The package includes the protocol, numerical results, and a portable replay harness. The provenance.json file records metadata and source-path adaptations; the numerical results are unchanged. The default harness path inspects source only. Execution needs a POSIX/Unix environment for its alarm guard, NumPy, and an explicitly selected new empty output directory:

```
python3 contract_audit/integration_regression.py \
    --execute --output-dir /chosen/new/empty/output
```

The replay retains the 90-second budget and single-run marker. Preserve prior markers to prevent unrecorded retries of failed or completed executions. See the root README for environment and resource prerequisites.

Remaining validation

The bounded detector/estimator check establishes preservation for the specified fixture. Actual feature extraction, the upstream Gaussian helper, production serialization, metric/threshold and adaptive pipelines, other detector branches, clean external reproduction, and real-data calibration remain untested here. Constructing a valid replacement null still requires a separate design. Operational false-positive rates and independence claims require a calibrated test and additional data.

Appendix B Limitations and dual use ethical considerations

The evidence consists of a pinned-code audit, an exact example, a toy mechanism experiment, and a bounded synthetic module regression. Operational error rates, a replacement significance test, and effects on the original paper's headline results remain unassessed. The toy differs from the source estimator; actual trajectories may violate row exchangeability; and the three-fit toy Gaussian diagnostic remains uncalibrated. The proposed sentinel requires explicit consumer handling because it can change a meta-test's result.

The findings concern the software's inferential interface. Public release made this inspection possible. These outputs cannot support accusations of collusion against people or organizations. Publishing measurement weaknesses could assist evasion of simplistic monitors. The work is limited to source inspection and local synthetic experiments; operational evasion, agent deployment, and real-market interventions are outside its scope.

AI usage

AI tools were used for literature and code review, implementation, analysis, and manuscript drafting and editing.