




Sleuth: A Browser-Based Tool for Detecting Circular Bias in AI Evaluation

Hongping Zhang ¹

¹ Independent Researcher

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: 

Submitted: 16 October 2025

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Summary

Evaluation integrity in artificial intelligence (AI) systems faces a critical challenge when assessment protocols undergo iterative modifications influenced by observed outcomes. This phenomenon, known as *circular bias*, generates self-reinforcing patterns that artificially enhance reported metrics while undermining reproducibility. Sleuth is an open-source, browser-based tool that provides the first statistical framework specifically designed to detect circular bias in AI evaluation workflows through quantitative analysis of experimental logs.

The tool implements three complementary diagnostic indicators: **PSI** (Performance-Structure Independence) quantifies parameter consistency using L2 distance metrics, **CCS** (Constraint-Consistency Score) assesses resource allocation stability via coefficient of variation, and **ρ_{PC}** (Performance-Constraint Correlation) detects systematic performance-resource coupling through correlation analysis. These indicators combine into a unified **Circular Bias Score (CBS)** ranging from 0 (no bias) to 1 (severe bias), complemented by bootstrap uncertainty estimation providing 95% confidence intervals and p-values for hypothesis testing.

Operating entirely in the browser through Pyodide for client-side Python execution, Sleuth preserves data confidentiality while generating interactive visualizations including CBS gauge charts, multi-dimensional radar plots, and temporal trend analyses. Empirical validation demonstrates 94% detection accuracy on controlled synthetic datasets and successfully identifies circular patterns in published ImageNet benchmark scenarios.

Statement of Need

Contemporary AI research workflows commonly employ adaptive evaluation strategies where experimental parameters undergo refinement based on interim performance observations (Bouthillier et al., 2021; Recht et al., 2019). While methodologically legitimate when transparently documented, circular bias emerges when modifications remain undisclosed or retrospectively applied, producing inflated capability claims and diminished reproducibility (Dwork et al., 2015; Kapoor & Narayanan, 2023). This problem pervades competitive leaderboard environments, proprietary development pipelines, and benchmark curation practices (Blodgett et al., 2020; Dehghani et al., 2021).

Existing experiment management platforms (MLflow (Zaharia et al., 2018), Weights & Biases (Biewald, 2020)) provide metadata logging but lack integrated diagnostics for circular evaluation patterns. Reproducibility frameworks rely on author self-attestation without automated validation (Pineau et al., 2021), while algorithmic fairness tools (AIF360 (Bellamy et al., 2019), Fairlearn (Bird et al., 2020)) address model output biases but not evaluation procedure integrity.

Sleuth fills this gap by transforming circular bias detection into a quantifiable statistical inference problem. Its target audience includes:

- **Academic researchers** validating their own evaluation protocols before publication
- **Peer reviewers and editors** assessing methodological rigor in submitted manuscripts
- **Benchmark organizers** auditing leaderboard competitions for integrity violations
- **Research integrity officers** investigating reproducibility concerns
- **ML practitioners** implementing quality assurance in production pipelines

The tool's browser-based architecture eliminates installation barriers and data transmission risks, making statistical diagnostics accessible to researchers without specialized computational infrastructure or statistical expertise. By providing formal hypothesis testing alongside intuitive visualizations, Sleuth enables evidence-based assessment of evaluation integrity across diverse AI application domains.

Key Features

- **Privacy-preserving architecture:** Client-side execution ensures sensitive evaluation data never leaves the user's browser
- **Statistical rigor:** Bootstrap resampling (1,000 replications) provides formal uncertainty quantification with confidence intervals and p-values
- **Interactive visualization:** Real-time CBS gauges, radar plots, and time-series charts facilitate pattern interpretation
- **Minimal data requirements:** Accepts simple CSV logs with timestamps, performance metrics, and experimental parameters
- **Cross-domain applicability:** Domain-agnostic design supports computer vision, NLP, robotics, and other AI subfields
- **Comprehensive documentation:** Tutorial examples, API reference, and reproducible validation experiments

Implementation

Sleuth's frontend combines React for UI components with Chart.js for visualization rendering. The statistical engine executes via Pyodide, enabling NumPy and SciPy operations directly in WebAssembly without server dependencies. Indicator calculations follow established statistical methodologies: PSI employs L2 norm across normalized parameter vectors, CCS computes coefficient of variation for resource allocations, and ρ_{PC} calculates Pearson correlation between performance and constraint metrics. Bootstrap resampling implements stratified sampling to maintain temporal structure while estimating indicator distributions for hypothesis testing (Efron & Tibshirani, 1994). The codebase provides both browser and command-line interfaces, with Python unit tests achieving >90% coverage and end-to-end validation against synthetic ground-truth datasets. Complete source code and documentation are archived at Zenodo (software DOI: 10.5281/zenodo.17201032). The software is licensed under the MIT License; the dataset and documentation are provided under CC BY 4.0.

Acknowledgements

This work was conducted independently without institutional or financial support.

References

- Bellamy, R. K., Dey, K., Hind, M., Hoffman, S. C., Houde, S., Kannan, K., Lohia, P., Martino, J., Mehta, S., Mojsilović, A., & others. (2019). AI fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM Journal of Research and Development*, 63, 4–1. <https://doi.org/10.1147/jrd.2019.2942287>

- 85 Biewald, L. (2020). *Experiment tracking with weights and biases*. <https://www.wandb.com/>.
- 86 Bird, S., Dudík, M., Edgar, R., Horn, B., Lutz, R., Milan, V., Sameki, M., Wallach, H., &
87 Walker, K. (2020). Fairlearn: A toolkit for assessing and improving fairness in AI. *Microsoft*
88 *Research*.
- 89 Blodgett, S. L., Barocas, S., Daumé III, H., & Wallach, H. (2020). Language (technology) is
90 power: A critical survey of "bias" in NLP. *Proceedings of the 58th Annual Meeting of the*
91 *Association for Computational Linguistics*, 5454–5476. [https://doi.org/10.18653/v1/2020.](https://doi.org/10.18653/v1/2020.acl-main.485)
92 [acl-main.485](https://doi.org/10.18653/v1/2020.acl-main.485)
- 93 Bouthillier, X., Delaunay, P., Bronzi, M., Trofimov, A., Nichyporuk, B., Szeto, J., Mohammadi
94 Sepahvand, N., Raff, E., Madan, K., Voleti, V., & others. (2021). Accounting for
95 variance in machine learning benchmarks. *Proceedings of Machine Learning and Systems*,
96 3, 747–769.
- 97 Dehghani, M., Tay, Y., Gritsenko, A. A., Zhao, Z., Houlsby, N., Diaz, F., Metzler, D., &
98 Vinyals, O. (2021). The benchmark lottery. *arXiv Preprint arXiv:2107.07002*.
- 99 Dwork, C., Feldman, V., Hardt, M., Pitassi, T., Reingold, O., & Roth, A. (2015). The reusable
100 holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248), 636–638.
101 <https://doi.org/10.1126/science.aaa9375>
- 102 Efron, B., & Tibshirani, R. J. (1994). *An introduction to the bootstrap*. CRC press.
- 103 Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-
104 based science. *Patterns*, 4(9). <https://doi.org/10.1016/j.patter.2023.100779>
- 105 Pineau, J., Vincent-Lamarre, P., Sinha, K., Larivière, V., Beygelzimer, A., d'Alché-Buc, F.,
106 Fox, E., & Larochelle, H. (2021). Improving reproducibility in machine learning research
107 (a report from the NeurIPS 2019 reproducibility program). *Journal of Machine Learning*
108 *Research*, 22(164), 1–20.
- 109 Recht, B., Roelofs, R., Schmidt, L., & Shankar, V. (2019). Do ImageNet classifiers generalize
110 to ImageNet? *International Conference on Machine Learning*, 5389–5400.
- 111 Zaharia, M., Chen, A., Davidson, A., Ghodsi, A., Hong, S. A., Konwinski, A., Murching, S.,
112 Nykodym, T., Ogilvie, P., Parkhe, M., & others. (2018). Accelerating the machine learning
113 lifecycle with MLflow. *IEEE Data Engineering Bulletin*.