



Feedback to NIST AI 800-2 Draft

Authors: Rokas Gipiškis (AI Standards Lab), Ayrton San Joaquin (AI Standards Lab), Sean McGregor (AI Verification and Evaluation Research Institute), Miles Brundage (AI Verification and Evaluation Research Institute), Malek Ben Salem (AILeap), Aakash Gupta (Think Evolve Labs), Faiza Khan Khattak (Monark Health), Vassil Tashev (Garden Path Research)

Published date: March 10th, 2026

Correspondence: rokas@aistandardslab.org

AI Standards Lab is an independent non-profit dedicated to accelerating the development of AI safety standards. We are structured as a virtual lab, bringing together experts from all over the world. Our team has expertise in computer science, AI research, safety engineering, standards development, and law.

We would like to refer the reader to our peer-reviewed paper "[Deprecating Benchmarks: Criteria and Framework](#)", which proposes a methodology for retiring obsolete benchmarks, and discusses a potential DOI-based versioning system.

Below we list some reasons why an evaluator should consider deprecating some automated benchmarks. These reasons are then followed by bullet points listing recommendations. As some aspects below fall under either Practices 1 or 2, we recommend either incorporating them into their respective sections or creating a new section (Practice 4) entirely.

Benchmark lifecycle and deprecation

- We recommend that Practice 1 and 2 treat benchmarks as active tools requiring ongoing lifecycle assessment, rather than as static tools. Evaluators should actively assess whether a benchmark is still fit for purpose.
- Practice 1.2 could more explicitly incorporate benchmark deprecation criteria (e.g., saturation, contamination, statistical bias, annotation error rates, task obsolescence, invalidated assumptions, and semantic drift) as factors to consider when selecting or continuing to use a benchmark.
- Practice 1.2 could also recommend that evaluators check whether a benchmark has already been deprecated or flagged by its developers or other relevant governance actors before selecting it.

Versioning

- While the draft briefly mentions benchmark versioning as an emerging practice for evaluation code (Practice 2.2), it does not explicitly recommend that benchmark version numbers be prominently cited alongside all reported results, nor that evaluators verify they are using the most current non-deprecated version of a benchmark. We recommend making these points explicit.

Saturation

- Practice 1.2 could include guidance on assessing whether a benchmark is near its performance ceiling before selecting it. A saturated benchmark cannot meaningfully differentiate model capabilities, which directly undermines Practice 1.2's goal of selecting benchmarks that meet evaluation objectives.

Annotation quality

- Practice 1.2.4's quality assessment ("Is the benchmark of sufficient quality?") could include an explicit recommendation to report on the annotator agreement rate for all qualitative coding of inputs and/or outputs, as this is a concrete, measurable quality signal. It could either be listed as a separate point, or more explicitly incorporated in 1.2.4c.

Statistical bias in benchmark construction

- We recommend expounding on *Diversity of test items in the evaluation item* (Practice 1.2.4a). Readers can better gain familiarity of this item with a more thorough discussion of how idiosyncrasies can be exploited by the model, along with specific examples of lack of diversity, such as imbalance or label distribution skew in the datasets. Finally, it would be helpful to discuss common techniques to address this problem, such as resampling or weighing of examples.

Semantic drift

- Benchmarks can become outdated or unrepresentative as language, cultural context, or field conventions evolve over time, which erodes external validity and may run counter to the original evaluation objectives defined in Practice 1.1. This is a particular issue for automated benchmarks, where human oversight may only be triggered by pre-defined narrow events. Practice 1.2 could define events that may trigger Semantic Drift and how these will be incorporated into the monitoring phase.

Popular but flawed benchmarks

- Some widely-used benchmarks can be flawed benchmarks. Practice 1.2.5's claim could include a caveat that, while it may be good for validating the evaluation setup and designing some sanity checks, a benchmark being popular does not validate the benchmark itself.

Good reporting practice

- Practice 3.2 ("Share details of evaluation and evaluation data") should recommend that evaluators document the conditions under which their chosen benchmark becomes invalid. This can help with post-hoc analysis of an evaluation or when improving the evaluated model.

This document is released under [CC0 1.0](#) (Public Domain). You are free to reproduce, adapt, and incorporate this content into your own work without restriction or attribution.