Response to NIST RFI: Best Practices for Automated Benchmark Evaluations (NIST AI 800-2)

About EvalEval

The EvalEval Coalition is an interdisciplinary, entirely voluntary research community focused on evaluating evaluations, hosted by Hugging Face, the University of Edinburgh, and EleutherAI, with contributors from Stanford, ETH Zurich, MIT, IBM Research, the Weizenbaum Institute, and other institutions. Launched at a NeurIPS 2024 workshop, the coalition has since grown into a sustained collaborative effort organized around three working groups: Research (advancing evaluation science, including work on construct validity, benchmark saturation, and evaluation cards), Infrastructure (building tooling and standards for evaluation reporting, including the Every Eval Ever metadata schema and converters for Inspect AI, HELM, and lm-eval-harness), and Organization (coordinating community engagement, outreach, and events such as ACL 2026 workshop on "Evaluating AI in Practice"). The coalition's north star is to raise the floor for evaluation quality across the AI ecosystem, making it easier for all stakeholders to conduct, interpret, and compare evaluations, and harder to publish misleading or unreproducible claims.

The EvalEval Coalition appreciates the opportunity to provide input on the initial public draft of NIST AI 800-2, Practices for Automated Benchmark Evaluations of Language Models.

Individual Contributors: Ichhya Pant, David Manheim, Chad Atalla, Yixiong Hao, Subho Majumdar, Usman Gohar, Avijit Ghosh

Executive Summary

There is much to commend in NIST AI 800-2. The three-stage framework is well-structured; the emphasis on reproducibility and statistical rigor aligns with established measurement science; the honest labeling of emerging practices provides appropriate calibration; and the treatment of agentic evaluation challenges is timely. At the same time, we identify several opportunities for improvement, which we summarize here and develop in detail below.

Result selection bias and hill-climbing. The document does not adequately address the prevalent practice of running evaluations multiple times and reporting only the most

favorable result. We recommend explicit disclosure requirements for evaluation run counts, selection criteria, and protocol iterations.

Evaluation accessibility and availability. The document omits a critical practical consideration: whether evaluations themselves are publicly available, openly licensed, or accessible through established channels for third-party evaluation. We recommend adding this as a factor in benchmark selection guidance.

Actionable reporting standards. Section 3 catalogs what should be reported but offers limited guidance on how to report it in a structured, machine-readable, and interoperable manner. We recommend that NIST endorse or reference specific evaluation reporting schemas, such as the EvalEval Coalition's Every Eval Ever, to accelerate convergence.

Broader impact and social evaluation dimensions. The document would benefit from more explicit acknowledgment of what automated benchmarks cannot measure beyond Table I.1. This would include distributional harms, long-term societal effects, and impacts on marginalized communities, and would similarly benefit from guidance on when complementary evaluation methods are needed.

Temporal validity and benchmark lifecycle. The document does not substantively address benchmark saturation, the phenomenon in which benchmarks lose discriminative power as model performance approaches ceilings. We recommend guidance on assessing saturation and on lifecycle management of benchmarks.

Risks of adoption. The document's comprehensiveness presents a risk of compliance theater, asymmetric capability favoring large AI developers, and paradigm lock-in. We recommend attention to capacity-building and adaptive standardization.

General Comments

We welcome NIST CAISI's pre-standardization effort and are encouraged to see NIST taking a leading role in codifying best practices that have, until now, existed primarily as informal norms among evaluation practitioners. Given our general alignment with the purposes and structure of the draft, we offer this feedback in a constructive spirit. Where we identify gaps or risks, we pair them with concrete recommendations grounded in our coalition's ongoing work.

Scope and Definitions

Working Definition and Why Scope Matters

NIST AI 800-2 defines its scope as "automated benchmark evaluations," that is, evaluations that, once set up, can be run without human-in-the-loop input, focused on "using these evaluations to measure model capabilities." We generally agree with this scoping decision as a practical matter for a first publication, but we note several important implications.

This definition effectively excludes human-in-the-loop evaluations such as red teaming, user studies, and annotation-based assessments, which remain critical for evaluating properties like safety, fairness, and alignment that are difficult to capture through fully automated pipelines. It also excludes broader impact evaluations that assess downstream social, economic, or cultural effects of model deployment. Such evaluations may not be reduced to a set of benchmark items with programmatically verifiable answers. Finally, it excludes evaluation of evaluation validity itself, that is, the meta-level question of whether a given benchmark actually measures what it claims to measure, which requires both human judgment and empirical validation.

We urge NIST to clearly communicate, in the final publication, that **the scope limitation to automated benchmark evaluations is a deliberate first step and does not imply that excluded evaluation types are less important.** Table I.1 in the draft gestures at this distinction, but the framing could be strengthened to avoid an implied hierarchy of evaluation methods.

Constraints of Adopting Such a Definition

The automated benchmark paradigm carries inherent constraints that the document should acknowledge more explicitly.

First, there is the matter of Goodhart's Law dynamics. When benchmark scores become targets (for instance, for procurement decisions or regulatory compliance), the pressure to optimize for scores rather than genuine capability increases. The document's treatment of contamination (Practice 1.2, item 4c) is a start, but the broader incentive dynamics warrant more attention.

Second, there is the problem of construct underrepresentation. Automated benchmarks, by their nature, tend to measure what is easy to automate rather than what is most important to measure. The document's detailing of construct validity guidance (Practice 1.2, items 1 and 2) understates the urgency of this challenge.

Third, there is the issue of cultural and linguistic narrowness. The overwhelming majority of existing benchmarks are English-centric and underperform or have disparate performance in other languages and contexts in which systems may be deployed. This is not merely a "coverage" gap (Practice 1.2, item 2a.i) but a systematic bias in what the field considers evaluable, or worthy of evaluation. In order to export the American tech stack, better holistic evaluations are needed.

Relevant Distinctions for Consideration

There are several distinctions that the document could draw more sharply.

The first is **emphasizing benchmark creation as a future component.** The document explicitly defers guidance on benchmark creation (p. 1: "Future work may address benchmark development"). While appropriate for scoping, this leaves a significant upstream gap: the quality of evaluation is bounded by the quality of available benchmarks.

The second is **clearly defining and differentiating guidelines for measurement and evaluation.** While the process of running an evaluation involves measurements that generate evidence, the purpose of an evaluation is to choose what evidence to generate and to interpret that evidence to make a judgment or claim. This also relates to the difference between benchmarks and evaluations more broadly: most automated evaluations are benchmarks, which can measure model characteristics and behavior; the latter can also evaluate downstream impacts and systemic behaviors, and this is harder but not impossible to evaluate in automated ways.

The third is **delineating different best practices for developer self-evaluation and third-party evaluation.** The document treats these scenarios in a unified manner, but the incentive structures, access constraints, and trust dynamics differ substantially. Developer self-evaluations are subject to selection pressures and access advantages that third-party evaluations may not share, and vice versa. This has implications, including about what should be reported and how others will want to verify results, so distinct guidance for each evaluator type will improve practical utility.

The fourth is **between capability evaluation and safety evaluation**. The document focuses on capability measurement and notes that "many practices also apply to evaluating other behavioral properties of models (e.g., robustness)." Safety evaluations, however, often require fundamentally different methodological choices, such as emphasis on worst-case rather than average-case behavior, different baseline constructs, concerns about evaluation awareness and sandbagging, and sensitivity to other adversarial conditions. A brief discussion of where the capability-oriented practices may need adaptation for safety contexts would strengthen the document.

The fifth is **between measurement and evaluation proper.** A measurement computes a score; an evaluation interprets that score in light of purpose and decision context. NIST AI 800-2 covers how to run benchmarks and compute scores (measurement), but its practices stop short of guiding the interpretive step. For instance, how to weigh conflicting results across benchmarks, what score thresholds imply fitness for a given deployment, or how to integrate benchmark evidence with other forms of assessment. Yet the document frames itself throughout as providing "evaluation" guidance, which risks giving practitioners the impression that computing and reporting scores constitutes a

complete evaluation. The final publication should either explicitly scope itself as measurement guidance or extend its coverage to the interpretive stage.

Overall Assessment of NIST AI 800-2

Strengths

The draft publication reflects a high degree of technical sophistication and practical grounding. The set of practices and recommendations is well constructed and near-exhaustive. We highlight several particular strengths.

The three-stage structure (defining objectives, then implementing and running, then analyzing and reporting) mirrors the actual workflow of evaluation practitioners and provides a natural checklist for ensuring completeness. This is no small achievement given the heterogeneity of evaluation workflows across organizations and domains.

The design principles in Section 2.1.1 (comparability, external validity, cost control, and performance optimization) are well-chosen and provide useful heuristics for practitioners who may not have formal measurement science training. The worked examples in Table 2.1 and throughout the document are particularly valuable for translating abstract principles into concrete implementation guidance.

The document's attention to agentic evaluation challenges, including agent budgets, scaffolding configurations, execution environments, and submission attempts, is timely given the rapid deployment of AI agent systems. The distinction between scaffolding settings and task settings (Table 2.2) is a useful contribution to the emerging vocabulary of agent evaluation.

Practice 3.1's emphasis on uncertainty quantification, appropriate aggregate statistics, and the decomposition of sources of variation is consistent with NIST's broader mandate for measurement excellence.

The labeling of certain practices as "emerging" (for instance, evaluation awareness, benchmark versioning, canary strings, and evaluation cheating detection) honestly communicates the current state of the field and avoids premature prescription. This calibration of maturity is important in a rapidly evolving domain.

Section 2.4's catalog of common bugs (degraded serving, tool calling errors, test item solvability, refusals, evaluation cheating, evaluation awareness) and quality assurance techniques is highly practical and reflects hard-won experience from CAISI's own evaluation campaigns. The inclusion of CAISI's own debugging logs in Table 2.1 sets a commendable standard for transparency.

Gaps

While the document is strong within its scope, we identify several gaps that, if addressed, would substantially strengthen the final publication.

Gap 1: Result Selection Bias and Hill-Climbing: Disclose number of evaluation runs and from which run evaluations are reported. The document does not adequately address the problem of result selection bias, that is, the practice of running an evaluation multiple times and reporting only the most favorable result. Without model developers reporting whether an evaluation was run on earlier version(s) of a model, it is unclear if the presented results are hill-climbing. This should be addressed by disclosure of not just the final run of an evaluation, but also of how many times the evaluation was run before the final results were selected for reporting, ideally via some form of preregistration. If evaluation protocols were iteratively modified (e.g., prompt engineering or scaffold adjustments) prior to the reported run, the nature and extent of such modifications should be documented. These details should be documented within the evaluation protocol prior to evaluation implementation, and iteratively updated and documented as needed.

Gap 2: Evaluation Accessibility and Availability. Under Practice 1.2, item 5 ("What other practical considerations may affect benchmark usage?"), the document addresses ease of use and results reported by others, but omits a critical practical consideration: the availability and accessibility of the evaluation itself. The adoption of evaluations partly depends on the availability of the evaluation. This can be addressed by open-source licensing, public APIs, or explicit channels to have non-public evaluations run by the developer.

We recommend an addition to Practice 1.2, item 5: evaluators should consider whether the benchmark is publicly available, whether its use requires licensing agreements, and whether there are established channels for accessing non-public benchmarks (for instance, through developer-mediated evaluation programs or trusted third-party evaluation services). Open-source benchmarks with permissive licensing generally facilitate broader adoption, independent verification, and reproducibility.

Gap 3: Actionable Reporting Standards. In Section 3 (Analyzing and Reporting Results), the recommendation to use "an interoperable format or schema to share these details" (Practice 3.2, item 1) is a step in the right direction, but without pointing to specific schemas or infrastructure, this practice is unlikely to drive convergence.

The evaluation ecosystem currently suffers from extreme fragmentation in how results are reported. The same benchmark (for instance, MMLU) evaluated through different harnesses (lm-eval-harness, HELM, original Berkeley implementation) can produce meaningfully different scores due to differences in prompt formatting, answer extraction, and few-shot ordering, yet all are reported simply as "MMLU scores." The EvalEval Coalition's Every Eval Ever project has demonstrated that a standardized metadata schema capturing source information, model access details, generation configuration, and scoring semantics can substantially address this fragmentation.

We recommend that NIST actively support the adoption and further development of standardized evaluation reporting schemas, such as Every Eval Ever or similar community-developed standards, and consider endorsing or referencing specific schemas in the final publication.

Gap 4: Broader Impact and Social Evaluation Dimensions. The document could benefit from a more prominent discussion of what automated benchmarks cannot measure (for instance, distributional harms, long-term societal effects, impacts on marginalized communities). Automated benchmarks can measure (and extrapolate claims from) properties of model outputs, but the impacts that stakeholders are motivated by (e.g., benefit or harm) are mediated by users and society. There's potential for dangerous conflation: automated benchmarks measure properties of model outputs, never impacts directly. We recommend providing further guidance on when and how to complement automated benchmarks with other evaluation methods, acknowledgement that benchmarks targeted at evaluating harm are built upon assumptions of how outputs will lead to impacts, and acknowledgment that benchmark selection itself is a values-laden decision. The absence of benchmarks targeted toward certain harm types reflects structural biases in the evaluation ecosystem.

One concrete mechanism for addressing this gap is structured AI failure repositories, such as the AI Vulnerability Database (AVID), modeled on NIST's own CVE/NVD infrastructure. When Practice 1.2 recommends documenting what a benchmark measures, it should also recommend documentation best practices for what known risk classes the benchmark fails to capture, transforming benchmark selection into an exercise that systematically identifies coverage gaps.

Gap 5: Temporal Validity and Benchmark Lifecycle. The document discusses contamination risks (Practice 1.2, item 4c) and benchmark versioning (Practice 2.2, item 3), but does not substantively address benchmark saturation, that is, the phenomenon by which benchmarks lose discriminative power as models improve over time and scores cluster near ceilings. The final publication should include guidance on assessing whether a benchmark has saturated and on the lifecycle management of benchmarks.

Gap 6: Considering elicitation when interpreting benchmark score. Practice 2.1.1 recommends iterative protocol optimization to find performance bounds, and Practice 3.3 requires qualifying claims. However, neither is connected to an explicit interpretive principle: that capability benchmark scores reflect what a model demonstrably did, not what it can do. Elicitation effort is a ceiling on observed performance, not a property of the model.

We recommend adding to Practice 3.3 a requirement that capability claims distinguish between observed performance and capability ceiling, with disclosure of elicitation effort expended. Additionally, extend Practice 2.1.1's performance optimization guidance to

include reporting how much optimization was applied. This will enable readers to calibrate how conservative the reported score is.

Gap 7: Failures disappear from the record. Practice 2.4.1 catalogs failure types (degraded serving, refusals, cheating, evaluation awareness) and Practice 2.4.2 prescribes detection techniques. Both are framed as debugging activities internal to evaluation QA. There is no requirement to document or report observed failure patterns as a formal evaluation output so failures currently inform corrections but disappear from the record.

We recommend adding a sub-practice to Practice 3.2 suggesting the reporting of failure patterns observed during execution, including type, frequency, and disposition. This will provide readers with important context, especially on the LLM system's 'default' propensities.

Gap 8: Cross-model comparison and trend standards. As evaluators increasingly report capability trajectories across time, benchmark contamination accumulates, benchmarks saturate, and protocol versions drift, all of which can confound trend interpretation. We recommend an addition to Practice 3.3 that trend claims disclose benchmark version, and contamination risk.

Gap 9: Consistent narrative on statistical rigor. Practice 3.1 item 5 recommends paired difference tests but says nothing about multiplicity correction when comparing k models across m benchmarks. The combinatorial explosion of pairwise comparisons promotes false discoveries without appropriate adjustment. It also conflates the fixed-effects question (do models differ on this benchmark?) with the random-effects question (would they differ on similar items?). Such benchmark vs. generalized accuracy distinctions are formalized in the follow-on report NIST AI 800-03, which 800-02 should cross-reference.

Risks

Structural and Technical Challenges

The document's comprehensiveness, while a strength, also presents a risk of compliance theater: organizations may adopt the document's practices superficially (for instance, checking boxes on a reporting template) without meaningfully engaging with the underlying measurement science. NIST should consider how to incentivize substantive adoption and more comprehensive reporting rather than form-filling.

The practices described require significant technical expertise, computational resources, and institutional infrastructure. Without attention to capacity-building, particularly for smaller organizations, civil society evaluators, and academic researchers in under-resourced settings, the practices may primarily serve large AI developers who already have well-resourced evaluation teams. This asymmetry of capability is worth

addressing directly. In the future, a recommended progression of practices may help under-resourced actors.

By codifying current practices, there is also a risk of lock-in to the current paradigm in ways that may prove inadequate for future AI systems, such as multimodal models, multi-agent systems, or AI systems with persistent memory. The "emerging practice" labels help mitigate this, but the final publication should include a more explicit discussion of known limitations of the current paradigm.

Usefulness, Practicality, and Rigor of Proposed Practices

Section 2.4 recommends debugging evaluations to identify and fix errors, which is appropriate. However, debugging a benchmark on a specific model that will be run on biases the outcome towards that model. If an evaluator iteratively adjusts prompts, parsing logic, or scaffolding configurations to get a particular model to perform well (or even just to "work"), those adjustments may inadvertently disadvantage other models. The debugging section should explicitly flag this risk and recommend debugging across multiple diverse models when possible, and documenting changes made to the benchmark.

The document acknowledges the growing role of LLM judges in evaluation but labels this as an "emerging practice." Given how widely LLM-as-a-judge is already deployed in production evaluations (for instance, Arena-Hard, MT-Bench, AlpacaEval), the document should provide stronger guidance on known failure modes of LLM judges. Specific recommendations could include:

a) Documenting the judge LLM’s orchestration with the same rigor as the LLM system being evaluated.
b) Spot-checking judge LLM outputs.
c) Defending against known failures of LLM judges, including but not limited to: position bias, self-preference, verbosity bias, and rubric sensitivity.

Recommendations for NIST and the Broader Evaluation Community

Immediate

The most impactful short-term improvement would be to help standardize eval reporting by pointing to schemas for reporting and infrastructure for implementing and providing the reports, not just listing items to report. This could help move from providing “an evaluation design checklist” (page 13, line 7) towards encouraging evaluation transparency and reporting. The EvalEval Coalition's Every Eval Ever schema, developed with input from NIST CAISI and other stakeholders, captures source metadata, who ran

the evaluation, on what model, with what settings, what these scores actually mean, and instance-level scores. Endorsing or referencing such schemas would dramatically accelerate the interoperability that Section 3 calls for.

NIST should also add result selection bias guidance, including explicit recommendations for disclosing evaluation run counts, selection criteria, and protocol iterations in Practice 3.2.

A further near-term improvement would be to expand evaluation accessibility guidance by adding a sub-practice under Practice 1.2, item 5 addressing the availability, licensing, and accessibility of benchmarks as a factor in benchmark selection and adoption.

Finally, NIST should strengthen the LLM-as-a-judge section by upgrading it from an "emerging practice" notation to a full sub-practice with concrete guidance on validation, inter-rater agreement, known biases, and transparency requirements.

Medium-Term

NIST should continue pre-standardization efforts and, given the rapid pace of change, push for adaptive and updatable standardization methods around reporting. Rather than static document publications alone, NIST might consider maintaining a living resource (for instance, a versioned online companion) that can incorporate new evaluation science findings, emerging benchmark types, and community feedback on implementation experience.

For safety benchmarking specifically, NIST should encourage the creation of useful and high-quality benchmarks focused on edge cases, such as harms contextual to low-resource languages, code-switching, and Global South cultures. Current safety evaluations overwhelmingly reflect English-language and Western cultural contexts, creating both significant blind spots in the evaluation of globally deployed systems, and easier routes for non-western users to exploit the resulting gaps in safety.

NIST should also support meta-evaluation infrastructure by investing in the development and validation of tools for evaluating evaluation quality itself, including measurement validity assessments, benchmark saturation analyses, and contamination detection. Clarifying validation practices will equip evaluators to make better decisions when choosing benchmarks. Making these validation practices more explicit and broadly known can also aid in shifting the burden of validation from the evaluator to the original creator of a benchmark, amortizing the high cost of rigorous validation. Similarly, developing tools to assist with reporting can also reduce the relative costs for smaller and non-commercial evaluators to comply with otherwise burdensome practices.

Long-Term

The most consequential long-term investment would be in the science of creation and dissemination of high-quality benchmarks. The current document focuses on evaluating with existing benchmarks but explicitly defers guidance on benchmark development. The quality of the evaluation ecosystem is bound by the quality of available benchmarks. NIST should invest in research on principled benchmark design, including coverage analysis, difficulty calibration, cultural and linguistic diversity, and resistance to gaming.

NIST should also develop capacity-building resources to ensure that the practices in NIST AI 800-2 are accessible beyond large AI labs. This includes educational materials, workshops, and tooling that lower the barrier to conducting high-quality evaluations, as well as support for open-source evaluation frameworks, computational resources for under-resourced evaluators, and simplified guidance documents for non-specialist audiences.

Additionally, labor uplift from AI usage could soon become load-bearing for key economic and safety decisions. NIST CAISI should prepare to provide guidance on practices for randomized controlled trials in the near future, to make these stronger forms of evidence more accessible to practitioners and consumers of uplift data.

Finally, NIST should establish a feedback loop with the evaluation community, creating a structured mechanism for ongoing community input beyond periodic public comment periods, to ensure that the guidelines remain current and practically useful.

Conclusion

NIST AI 800-2 represents an important and well-executed contribution to the nascent infrastructure of Automated AI Benchmark evaluation standardization. The three-stage framework is sound, the practice recommendations are detailed and grounded in real evaluation experience, and the honest labeling of emerging practices provides appropriate calibration of maturity.

Our key recommendations center on four themes. First, make recommended reporting actionable by pointing to implementable schemas and infrastructure, not just lists of items to report. Second, help close the result selection bias gap by requiring disclosure of evaluation run counts and selection criteria, and encourage prepregistration. Third, expand scope awareness by more explicitly framing what automated benchmarks do not or cannot measure and when complementary evaluation methods are needed. Fourth, continue to advance the foundations of evaluation science, benchmark creation, and evaluation accessibility by encouraging relevant NIST experts to contribute to academic and other research projects, to build the capacity of projects outside of government to advance the long-term health of the ecosystem.

We look forward to continued engagement with NIST CAISI on these issues and welcome the opportunity to contribute our coalition's technical expertise and research outputs to the development of final guidelines.

Appendix

Mapping of EvalEval Coalition Work to NIST AI 800-2 Practices

NIST AI 800-2 Practice	Relevant EvalEval Coalition and Other Work
Practice 1.1 (Define evaluation objectives)	Evaluation Cards: structured templates for documenting evaluation design decisions and objectives
Practice 1.2 (Select benchmarks)	Benchmark Saturation: methods for assessing whether benchmarks have lost discriminative power. (Akhtar et al 2026,)
Practice 2.1 (Design evaluation protocol)	Evaluation Science: formal frameworks for evaluation validity and protocol rigor
Practice 2.2 (Write evaluation code)	Various frameworks including Inspect AI, HELM, lm-eval-harness. Note that Every Eval Ever has converters for storing outputs of each.
Practice 3.1 (Statistical analysis)	Evaluation Science: alignment with measurement theory and statistical validity
Practice 3.2 (Share evaluation details)	Every Eval Ever: standardized metadata schema for machine-readable evaluation reporting
Practice 3.3 (Report qualified claims)	AI Evaluation Chart Crisis blog: analysis of misleading evaluation visualization and reporting practices