





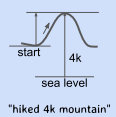






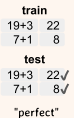

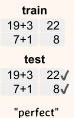

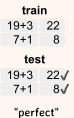

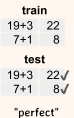

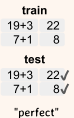

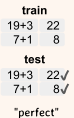


SIGPLAN Empirical Evaluation Checklist

This checklist is meant to **support** informed judgement, not **supplant** it.

Clearly Stated Claims Example Best Practices	 <p>Explicit Claims</p> <p>Claims must be explicit in order for the reader to assess whether the empirical evaluation supports them. Claims should aim to state not just what is achieved but how.</p>	Relevant Metrics Example Best Practices	 <p>Direct or Appropriate Proxy Metric</p> <p>If the most relevant evaluation metric is not (or cannot be) measured directly, the proxy metric used instead must be well justified. For example, a reduction in cache misses is not an appropriate proxy for actual end-to-end performance or energy consumption.</p>
	 <p>Appropriately-Scoped Claims</p> <p>The truth of claims should follow from the evidence provided. Overclaiming is often the consequence of inadequate evidence, e.g., claiming 'works for all Java', but evaluating only a static subset or claiming 'works on real hardware', but evaluating only in (unrealistic) simulation.</p>		 <p>Measures All Important Effects</p> <p>The costs and benefits of a technique may be multifaceted. All facets should be considered, both costs and benefits, and ideally evaluated. For example, compiler optimizations may speed up programs at the cost of drastically increasing compile times.</p>
Suitable Comparison Example Best Practices	 <p>Acknowledges Limitations</p> <p>A paper should acknowledge its limitations to place the scope of results in context. Stating no limitations at all, or only tangential ones while omitting the more relevant ones, may mislead the reader to drawing too-strong conclusions.</p>	Appropriate and Clear Experimental Design Example Best Practices	 <p>Sufficient Information to Repeat</p> <p>Experiments should be described in sufficient detail to be repeatable. All parameters (including default values) should be included, as well as all version numbers of software, and full details of hardware platforms.</p>
	 <p>Appropriate Baseline for Comparison</p> <p>An empirical evaluation of a contribution that improves upon the state-of-the-art should evaluate that contribution against an appropriate baseline, such as the current best-of-breed competitor or a randomized baseline.</p>		 <p>Reasonable Platform</p> <p>The evaluation should be on a platform that can reasonably be said to match the claims. For example, a claim that relates to performance on mobile platforms should not have an evaluation performed exclusively on server.</p>
Principled Benchmark Choice Example Best Practices	 <p>Appropriate Suite</p> <p>Evaluations should be conducted using the appropriate established benchmarks where they exist. Established suites should be used in the designed-for context; for example, it would be wrong to use a single-threaded suite for studying parallel performance.</p>	Appropriate Presentation of Results Example Best Practices	 <p>Explores Key Design Parameters</p> <p>Key parameters should be explored over a range to evaluate sensitivity to their settings. Examples include the size of the heap when evaluating garbage collection and the size of caches when evaluating a locality optimization. All expected system configurations (e.g., from warmup to steady state) should be considered.</p>
	 <p>Non-Standard Suite(s) Justified</p> <p>Sometimes an established benchmark suite does not exist. A rationale should be provided for the selection of homegrown benchmarks or subsetting established benchmark suites.</p>		 <p>Open Loop in Workload Generator</p> <p>Load generators for typical transaction-oriented systems should not be gated by the rate at which the system responds. Rather, the load generator should be 'open loop', generating work independent of the performance of the system under test. See [Schroeder et al, 2006]</p>
Adequate Data Analysis Example Best Practices	 <p>Appropriate Suite</p> <p>Evaluations should be conducted using the appropriate established benchmarks where they exist. Established suites should be used in the designed-for context; for example, it would be wrong to use a single-threaded suite for studying parallel performance.</p>	Appropriate Presentation of Results Example Best Practices	 <p>Cross-Validation Where Needed</p> <p>When a system aims to be general but was developed by training on or close consideration of specific examples, it is essential that the evaluation explicitly perform cross-validation, so that the system is evaluated on data distinct from the training set.</p>
	 <p>Non-Standard Suite(s) Justified</p> <p>Sometimes an established benchmark suite does not exist. A rationale should be provided for the selection of homegrown benchmarks or subsetting established benchmark suites.</p>		 <p>Cross-Validation Where Needed</p> <p>When a system aims to be general but was developed by training on or close consideration of specific examples, it is essential that the evaluation explicitly perform cross-validation, so that the system is evaluated on data distinct from the training set.</p>
Adequate Data Analysis Example Best Practices	 <p>Sufficient Number of Trials</p> <p>In modern systems, which have non-deterministic performance, a small number of trials (e.g., a single time measurement) risks treating noise as signal. Similarly, more trials may be needed to get the system into an intended state (e.g., into a steady state that avoids warm-up effects).</p>	Appropriate Presentation of Results Example Best Practices	 <p>Cross-Validation Where Needed</p> <p>When a system aims to be general but was developed by training on or close consideration of specific examples, it is essential that the evaluation explicitly perform cross-validation, so that the system is evaluated on data distinct from the training set.</p>
	 <p>Appropriate Summary Statistics</p> <p>There are many summary statistics, and each presents an accurate view of a dataset only under appropriate circumstances. For example, the geometric mean should only be used when comparing values with different ranges, and the harmonic mean when comparing rates. When distributions have outliers, a median should be pre-</p>		 <p>Cross-Validation Where Needed</p> <p>When a system aims to be general but was developed by training on or close consideration of specific examples, it is essential that the evaluation explicitly perform cross-validation, so that the system is evaluated on data distinct from the training set.</p>
Adequate Data Analysis Example Best Practices	 <p>Report Data Distribution</p> <p>Reporting just a measure of central tendency (e.g., a mean or median) fails to capture the extent of any non-determinism. A measure of variability (e.g., variance, std deviation, quantiles) and/or confidence intervals help to understand the distribution of the data.</p>	Appropriate Presentation of Results Example Best Practices	 <p>Cross-Validation Where Needed</p> <p>When a system aims to be general but was developed by training on or close consideration of specific examples, it is essential that the evaluation explicitly perform cross-validation, so that the system is evaluated on data distinct from the training set.</p>
	 <p>Report Data Distribution</p> <p>Reporting just a measure of central tendency (e.g., a mean or median) fails to capture the extent of any non-determinism. A measure of variability (e.g., variance, std deviation, quantiles) and/or confidence intervals help to understand the distribution of the data.</p>		 <p>Cross-Validation Where Needed</p> <p>When a system aims to be general but was developed by training on or close consideration of specific examples, it is essential that the evaluation explicitly perform cross-validation, so that the system is evaluated on data distinct from the training set.</p>

Notes

Appropriately-Scoped Claims This includes *implied* generality — implied: ‘works for all Java’, but actually only on a static subset; implied: ‘works on real hardware’, but actually only works in simulation; implied: ‘automatic process’, but in fact required non-trivial human supervision; implied: ‘only improves the systems’ performance’, but actually the approach requires breaking some of the system’s expected behavior.

Acknowledges Limitations One concern we have heard multiple times is that this example, previously titled *Threats to validity*, is not useful. The given reason is that *threats to validity* sections in software engineering papers often mention threats of little significance while ignoring real threats. This is unfortunate, but does not eliminate the need to clearly scope claims, highlighting important limitations. For science to progress, we need to be honest about what we have achieved. Papers often make, or imply, overly strong claims. One way this is done is to ignore important limitations. But doing so discourages or undervalues subsequent work that overcomes those limitations because that progress is not appreciated. Progress comes in steps, rarely in leaps, and we need those steps to be solid and clearly defined.

Appropriate Baseline for Comparison An evaluation of an idea that improves upon the state-of-the-art should evaluate that idea against a baseline. This baseline could be a best-of-breed competitor, but should not be a straw man, e.g., something that once was, but is no longer, the state-of-the-art. The baseline could also be an unsophisticated approach to the same problem, e.g., a fancy testing tool is usefully compared against one that is purely random, in order to see whether it does better.

Fair Comparison For example, the authors were unable to build the state-of-the-art baseline at the -O3 optimization level and used -O0 instead, while using -O3 for their system.

Appropriate Suite This includes misuse of incorrect established suite e.g. use of SPEC CINT2006 when considering parallel workloads.

Non-Standard Suite(s) Justified A concern we heard was that use of standard suites may lead to work that overfits to that benchmark. While this is a problem in theory, and is well known from the machine learning community, our experience is that PL work more often has the opposite problem. Papers we looked at often subset a benchmark, or cherry-picked particular programs. Doing so calls results into question generally, and makes it hard to compare related systems across papers. We make progress more clearly when we can measure it. Good benchmark suites are important, since only with them can we make generalizable progress. Developing them is something that our community should encourage.

Note that ‘benchmark’ in this category includes what is measured and the parameters of that measurement. One example of an oft-unappreciated benchmark parameter is timeout choice.

Appropriate Summary Statistics There are many excellent resources available, including: *Common errors in statistics (and how to avoid them)*. (Phillip I Good and James W Hardin, 2012), *What is a P-value anyway?: 34 stories to help you actually understand statistics*. (Andrew Vickers, 2010), and *Statistical misconceptions*. (Schuyler W Huck, 2009).

Ratios Plotted Correctly For example, if times for a and b are 4 sec and 8 sec respectively for benchmark x and 6 sec and 3 sec for benchmark y, this could be shown as a/b (0.5, 2.0) or b/a (2.0, 0.5), where 1.0 represents parity. Although the results (0.5 & 2.0) are reciprocals, their distance from 1.0 on a linear scale is different by a factor of two (0.5 & 1.0), overstating the speedup. This is why showing ratios (or percentages) greater than 1.0 (100%) and less than 1.0 (100%) on the same linear scale is visually misleading.

FAQ

Why a checklist? Our goal is to help ensure that current, accepted best practices are followed. Per the [Checklist Manifesto](#), checklists help to do exactly this. Our interest is the good practices for carrying out empirical evaluations as part of PL research. While some practices are clearly wrong, many require careful consideration: Not every example under every category in the checklist applies to every evaluation — expert judgment is required. The checklist is meant to assist expert judgment, not substitute for it. ‘[Failure isn’t due to ignorance. According to best-selling author Atul Gawande, it’s because we haven’t properly applied what we already know.](#)’ We’ve kept the list to a single page to make it easier to use and refer back to.

Why now? When best practices are not followed, there is a greater-than-necessary risk that the benefits reported by an empirical evaluation are illusory, which

harms further progress and stunts industry adoption. The members of the committee have observed many recent cases in which practices in the present checklist are not followed. Our hope is that this effort will help focus the community on presenting the most appropriate evidence for a stated claim, where the form of this evidence is based on accepted norms.

Is use of the checklist going to be formally integrated into SIGPLAN conference review processes? There are no plans to do so, but in time, doing so may make sense.

How do you see authors using this checklist? We believe the most important use of the checklist is to assist authors in carrying out a meaningful empirical evaluation.

How do you see reviewers using this checklist? We also view the checklist as a way to remind reviewers of important elements of a good empirical evaluation, which they can take into account when carrying out their assessment. However, we emphasize that proper use of the checklist requires nuance. Just because a paper has every box checked doesn’t mean it should be accepted. Conversely, a paper with one or two boxes unchecked may still merit acceptance. Even whether a box is checked or not may be subject to debate. The point is to organize a reviewer’s thinking about an empirical evaluation to reduce the chances that an important aspect is overlooked. When a paper fails to check a box, it deserves some scrutiny in that category.

How did you determine which items to include? The committee examined a sampling of papers from the last several years of ASPLOS, ICFP, OOPSLA, PLDI, and POPL, and considered those that contained some form of empirical evaluation. We also considered past efforts examining empirical work (Gernot Heiser’s “[Systems Benchmarking Crimes](#)”, the “[Pragmatic Guide to Assessing Empirical Evaluations](#)”, and the “[Evaluate Collaboratory](#)”). Through regular discussions over several months, we identified common patterns and anti-patterns, which we grouped into the present checklist. Note that we explicitly did not intend for the checklist to be exhaustive; rather, it reflects what appears to us to be common in PL empirical evaluations.

Why did you organize the checklist as a series of categories, each with several examples? The larger categories represent the general breadth of evaluations we saw, and the examples are intended to be helpful in being concrete, and common. For less common empirical evaluations, other examples may be relevant, even if not presented in the checklist explicitly. For example, for work studying human factors, the Adequate Data Analysis category might involve examples focusing on the use of statistical tests to relate outcomes in a control group to those in an experimental group. More on this kind of work below.

Why did you use checkboxes instead of something more nuanced, like a score? The boxes next to each item are not intended to require a binary “yes/no” decision. In our own use of the list, we have often marked entries as partially filling a box (e.g., with a dash to indicate a “middle” value) or by coloring it in (e.g., red for egregious violation, green for pass, yellow for something in the middle).

What about human factors or other areas that require empirical evaluation? PL research sometimes involves user studies, and these are different in character than, say, work that evaluates a new compiler optimization or test generation strategy. Because user studies are currently relatively infrequent in the papers we examined, we have not included them among the category examples. It may be that new, different examples are required for such studies, or that the present checklist will evolve to contain examples drawn from user studies. Nonetheless, the seven category items are broadly applicable and should be useful to authors of any empirical evaluation for a SIGPLAN conference.

How does the checklist relate to the artifact evaluation process? Artifact evaluation typically occurs after reviewing a paper, to check that the claims and evidence given in the paper match reality, in the artifact. The checklist is meant to be used by reviewers while judging the paper, and by authors when carrying out their research and writing their paper.

How will this checklist evolve over time? Our manifesto is: Usage should determine content. We welcome feedback from users of the checklist to indicate how frequently they use certain checklist items or how often papers reviewed adhere to them. We also welcome feedback pointing to papers that motivate the inclusion of new items. As the community increasingly adheres to the guidelines present in the checklist, the need for their inclusion may diminish. We also welcome feedback on presentation: please share points of confusion about individual items, so we can improve descriptions or organization.

Feedback via: <http://www.sigplan.org/Resources/EmpiricalEvaluation/>