

Paper Writing Best Practices

This is a working document. Check back regularly for further ideas and links.

As a starter, we highly recommend [Bill Freeman's slides](#) on how to write good papers. To state the obvious, one writes a paper for others to read it. The first advice is that the paper should be written so that it finds readers (within our field) who will care about it. A specific type of reader is the reviewer (and the other PC members). In fact, their job is to model the typical reader. Understanding details of how this is done can be helpful, hence it may be good to check out the [reviewer tutorial](#) and the [review form](#).

The following checklist, modeled after the AAI 2021 paper submission checklist, inspired by similar previous checklists (in particular, Joelle Pineau's [checklist](#)), may also be useful. When writing the paper, think of whether your paper satisfies these points. The main point is that we expect ICML papers to advance the knowledge of the field. Write papers that achieve this.

Paper checklist.

1. The paper clearly states what claims are being made, and in particular it clearly describes the problem addressed (claims, problems)
2. The paper clearly explains how the results substantiate the claims (soundness)
3. The paper explicitly identifies limitations or technical assumptions (honesty)
4. If the paper is studying a new problem, the problem is well motivated and interesting for the readers in our community, at the time the paper is written (novelty, significance)
5. The paper is essentially self-contained: An expert reader (reader familiar with the topic) has a good chance to understand the paper
6. The paper discusses prior work and, in particular, how it is related to the contributions made. Future readers will be able to understand why this paper was targeting the specific problems that it described (context)
7. The paper helps readers not familiar with all the background necessary to understand the paper by citing appropriate review articles, tutorials, or books that the readers can check for background
8. Writing, organization:
 1. The abstract is short and gives an objective summary of the contributions
 2. The title expresses the main contribution and is easy to remember
 3. The paper separates the problem description from the contributions
 4. The paper includes a conceptual (human readable) outline and/or pseudocode description of the algorithms and methods if there are any
 5. The paper provides random access for the expert reader (this means, expert readers have a good chance of understanding the contributions by reading the problem definition, skimming the notation, then looking at the claims and their justification)
 6. The paper is proofread for typos and grammatical errors
9. Provided that the paper has theoretical contributions
 1. All assumptions are stated in a clear, formal fashion
 2. The novel claims are formally stated (e.g., in theorem statements)
 3. The theorem statements are self-contained, or the reader will find it

- assumptions if any, but do not expect the reader to check all the text before the statements.
4. Proofs of nontrivial statements are included
 5. A superbly written paper will also give intuitive arguments for why the statements hold.
 6. If the paper improves results compared to state of the art, the paper clearly explains the reason of what made this improvement possible, or where it is coming from: What is improved in the algorithm/analysis?
 7. The reader is told which part of the proof is new and the rest is appropriately and generously attributed (e.g., this proof essentially follows that of ..)
 8. Nonstandard definitions are included
 9. All external results are appropriately cited (do not cite a book or a long paper for a theorem, give the reader precise information where to find the result, e.g., the number of the theorem or the page number)
10. If the paper includes computational experiments
1. The paper explains the design choices made that leads to the specific choice of experiments: What questions are answered by the experiments
 2. The paper separates the presentation of the experiment design, the description of the experiment (including how data is obtained and the properties of data), the results, and the interpretation of the results
 3. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results
 4. The paper formally describes evaluation metrics used and explains the motivation for choosing these metrics
 5. This paper states which algorithms are used to compute each reported result (giving appropriate references). If a result is the obtained by running an algorithm multiple times, the number of runs is included
 6. The analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information, as appropriate. The measure of variation included is justified (e.g., do not report standard deviation if the data is far from normally distributed)
 7. This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments
 8. This paper states the number and range of values tried per (hyper-)parameter during development of the paper, along with the criterion used for selecting the final parameter setting. Beware of overfitting to benchmarks.
 9. All source code required for conducting experiments is included in the supplementary material or the appendix
 10. All source code required for conducting experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes
 11. The paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks
11. If the paper uses one or more data sets, or other benchmarks (e.g., simulation environments in reinforcement learning)
1. All novel benchmarks introduced are included in a data appendix/supplementary
 2. All novel benchmarks introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes
 3. All benchmarks drawn from the existing literature (potentially including authors' own previously published work) are accompanied by

4. All benchmarks drawn from the existing literature (potentially including authors' own previously published work) are publicly available
5. All benchmarks that are not publicly available are described in detail (how the benchmark was designed, the data was obtained, descriptive statistics, etc.)

Additional Resources

Much here is written for writing mathematics. The reason this is relevant is because in our field, even if not describing theoretical results, we use mathematics to express our thoughts in a precise manner.

- Aaditya Ramdas' [stat-ML paper checklist](#)
- Pat Langley's ICML 2002 piece on [Crafting Papers on Machine Learning](#)
- Zach Lipton's [Heuristics for Scientific Writing \(a machine learning perspective\)](#)
- Ian P. Gent and Toby Walsh: [How not to do it](#); a list of things to avoid while running, designing, analyzing and reporting on computer experiments
- D. Mitchell, H. Levesque: [Some Pitfalls for Experimenters with Random SAT](#) While not machine learning, but AI, this is relevant if one chooses to run experiments on randomly generated instances. The lesson is that choosing instances from a distribution does not automatically give useful information about the properties of the algorithms tested. This is illustrated in the context of designing SAT solvers, but applies equally well to, e.g., reinforcement learning methods.
- JMLR's "[Checklist of common JMLR formatting errors](#)"; some good editorial advice (just ignore the JMLR specific parts)
- Kevin P. Lee's [A Guide to Writing Mathematics](#); some good advice here, though this is geared towards project reports, which are less formal. For a paper, it is not OK just list line after line equivalent expressions without any explanation, or words connecting them.
- Oded Goldreich: [How to write a paper?](#)
- Simon Peyton Jones: How to write a great research paper? ([pdf of slides](#)/[talk webpage](#))
- D. Bertsekas: [Ten Simple Rules for Mathematical Writing](#)
- [Rules and Tips for Writing Mathematics](#)
- [Guide to Writing Mathematics](#) (The U. of Hong Kong). Detailed, good advice.
- [Elements of Style](#) by Strunk and White
- [How to write mathematics](#) by Paul R. Halmos
- [How to write a clear math paper](#) by Igor Pak. Long, a bit scary start, but some good advice.
- JS Milne's [Tips for Authors](#) (funny, teaching by showing what not to do)
- Steven G. Krantz: [A Primer of Mathematical Writing](#) (a whole book)