How to Write Technical Papers: UT Austin Swarm Lab

Sandeep Chinchali*
Department of Electrical and Computer Engineering*, The University of Texas at Austin, Austin, TX
E-mail: sandeepc@utexas.edu

***Abstract*—Dummy abstract**

I. Document Organization

Each section is organized into two sub-sections. The first, named "Section Objective", describes advice on what to place in each technical section. The second, named "Guidelines", enumerates good practices and common mistakes that should be avoided.

Before writing full text, make sure to **outline** your paper in bullet-points. Example sentences are given in blue.

A. How to Maximally Benefit from This Guide

Read each section and reference the corresponding section of our prior papers [2, 3, 1]. Try to map each guideline to what you see there. After you outline your paper, write Sections 2-N, one at a time. Focus on clear, short sentences and paragraphs.

Before starting your research or writing, answer the following questions, based on DARPA's Heilmeier Catechism, in 1-2 sentences each. Put the questions and answers at the top of your outline.

1) What is the objective of your research? Minimize technical jargon.
2) If you achieve this goal, what benefit or new capability will it provide?
3) What are key failures with today's state-of-the-art?
4) What is your key technical insight that enables your solution?
5) What are the limitations and assumptions inherent to your approach?

II. Introduction

A. Section Objective

Paragraph 1: Motivation and Potential of the Idea

There are many ways to write this paragraph depending on your style. One way is to describe the interesting potential of a new technology, technical capability, or set of algorithms. For example:

> Learning from rare events, such as traffic disruptions or hazardous weather conditions, can improve the safety and reliability of autonomous vehicles. Today's AVs measure more than 4 TB of rich video and LIDAR sensory data in just a few hours, which can be mined to continually improve computer vision models …

Focus on clear, short, simple sentences and cite survey papers if possible.

Paragraph 2: Problems with State-of-the-Art Solutions

Describe open technical challenges that need to be solved to achieve the goals and capabilities from Paragraph 1. Be precise, but avoid un-necessary technical jargon. Often, we will have a sentence saying "Despite the potential of [new technology], a key open problem is to [succinctly describe problem]."

An example:

> Despite the benefits of continually re-training vision models on large volumes of rich sensory data, we lack algorithms to balance these benefits with systems costs of network bandwidth consumption and cloud computing time.

Paragraph 3: Key Technical Insight for Your Solution Approach

Anyone can describe a grand challenge and open problems. Here, we describe your unique technical insight that allows you to solve the problem. First, describe **your key observation** on why state-of-the-art methods fail today. Then, describe your **key technical insight** that can provide a solution. Then, describe the **the principal contribution** of your paper. Focus on clear, jargon-free sentences with carefully-selected technical terms so an expert can quickly appreciate what your solution and unique contribution will be. Often, reference Figure 1 that shows a clear diagram of your approach, but do not get into deep mathematical details.

1) Related Work: First, describe the 4-5 key sub-topics your work relates to. For each, cite several papers and, crucially, describe how their approach *differs* from yours or is complementary. Do not simply state what each paper does, but emphasize why it differs. Some people prefer to describe one closest competitor paper in detail: "The closest work to ours is [**?**]. The key difference of our approach is we …".

A good example:

> Our work is broadly related to rate distortion theory, autoencoders, and task-driven representation learning in robotics. Several prior works use rate distortion theory to [brief overview and citations]. However, the standard assumption in rate distortion theory is to [describe why it differs], which is in stark contrast to our approach that […].

A bad example:

> Chinchali et. al. do [something]. Blank et. al. [do something else]. Continue with a long list of papers and their descriptions, but nothing to cluster them together based on sub-topic or contrast with your approach.

Key words to use in this paragraph are **contrast, complementary** etc.

2) Contributions : "In light of prior work, our contributions are N-fold." Use a sentence like this to segway between related work and your key contributions. Make an enumerated list of contributions (typically 3-4) with clear, short sentences, links to code or data URLs, and quantitative numbers showing the improvement over benchmarks. Allude to cool hardware demonstrations or simulation platforms here.

3) Paper Organization : Describe the key content of each section and how they relate to the overall flow of the paper.

B. Guidelines

Outline the key contributions, technical insight, and comparison with state-of-the-art in **bullet points**. Then, only once you are done writing the rest of the paper, write the full text.

III. Problem Statement

A. Section Objective

The problem statement is the most important part of your paper and should be written first. First, allude to a diagram that shows the information flow with mathematical notation for your problem, such as Figure 2. Then, systematically introduce key notation that helps you build up to a **formal mathematical optimization problem** with a cost function and constraints (in most problems our lab will focus on).

For a control or networking problem, describe the information flow. Namely, the sensory input, each computation function, and each function's input and output and task. Be formal, but provide a one sentence example for each.

1) The sensory input. Give the variable, an example, and (during the first introduction) describe its dimension. "The robot measures an n-dimensional sensory input $s_t \in \mathbb{R}^n$, such as an image or LIDAR point cloud, at discrete time t."
2) The controller's state space.
3) The action space.
4) The dynamics.
5) The cost function.

1) Formal Problem Statement: This should be in a formal problem statement block, such as in []. The statement should describe the inputs ("givens"), the optimization objective, and constraints. For the inputs and constraints, you likely will reference equations defined earlier.

2) Significance of the Problem: Describe in one short paragraph why the problem is novel compared to state-of-the-art, how it broadly applies to many engineering settings, and what technical parts make it challenging to solve.

B. Guidelines

1) Variables and Equations: The following guidelines govern notation.

1) **Time:** Use a subscript for *discrete* time, such as x_t. A timeseries of variable x from time t_0 to t_f should be given by $x_{t_0:t_f}$.
2) **Use Standard Notation:** In control, the state is generally given by x_t, control by u_t, policy by π, and cost function by J. Strictly follow the same notation as prior papers by Sandeep or key influential textbooks.
3) **Intuitive Variable Choices:** Your reader has a limited attention span, and will forget random notation like ψ, Γ unless it is necessary. Suppose we have K epochs to run an algorithm. Instead of K, write N_{epoch}, which is obvious in case I forget what K is.
4) **Deep Learning Notation:** Parameters of a model are given by θ. If you have different models, call them $\theta_{\text{enc.}}$, θ_{ctrl} for an encoder and controller (for example), rather than use a slew of unintuitive varibles like α, ψ etc. Loss functions should be given by $\mathcal{L}$, datasets by $\mathcal{D}$, etc.
5) Use MATHRM in LaTeX for English words in a LaTeX equation.

IV. Solution Approach

A. Section Objective

We have now formalized a challenging problem. Now, we want to introduce an algorithm or analytical solution. If we are doing pure deep learning, we should at least describe a loss function, model architecture, and training procedure.

1) Illustrative Toy Example: Often, we describe a toy example that clearly demonstrates the idea works in simulation. First, define the toy problem parameters using the *same notation* from the problem statement. Then, show the solution and a figure with the key results. Make sure to describe **the intuition** that the toy problem provides and how we can **generalize** the results to more sophisticated examples that will follow.

2) Algorithm or Derivation: This should artfully use insights from the toy problem to come up with a general algorithm or formal derivation. Tell why you do each step and place minute details for proofs in the Appendix if you have space constraints.

A bad example simply states **what** you did without logical transitions or stating **why**.

3) Benchmarks: This lists benchmarks that you will compare your algorithm with. In the list, put a formal, short name for the benchmark that will **uniformly** appear throughout the plots and rest of the text. Describe each benchmark in 2-3 sentences and argue why it is an appropriate benchmark, such as it is the current state-of-the-art, common practice, or tests a key feature of your algorithm.

B. Guidelines

An algorithm should appear as a stand-alone Figure with a caption. All algorithm lines should be **numbered** with **in-line comments** if space permits. Each line of the Algorithm should be described (and referenced) sequentially with intuition on why it matters. Do not simply state *what* you do, but tell us *why*. Here is an example:

A good example:

> We now describe the key steps in Algorithm 1 to co-design a task-driven encoder and decoder. First,

on line 1, we initialize the parameters θ randomly. Then, our next step is to ....

V. Experimental Results

A. Section Objective

We have now described our problem and what we *claim* is the best solution. We now need to provide hard quantitative evidence that proves this is true on diverse datasets or experimental domains. For a fair comparison, we need to describe:

1) **Evaluation Metrics:** What metrics define a good solution to the optimization problem stated in the problem formulation? Typically, this is a low overall cost or loss, but we also should describe individual terms of the loss function. These metrics **must appear** in a table or clear figure for **all benchmarks**.
2) **Benchmarks:** Recap the benchmarks and specific variants of your algorithm you will test.
3) **Diverse Experimental Domains:** Allude to each dataset you test your algorithm on in order of sophistication. Ideally, the last experiment should be on a real robot or hardware.

Paragraph 1: Describe evaluation metrics, recap benchmarks, and hook the reader by alluding to cool experiments. Make sure to reference all the key plots, ideally in a list, so the reader knows what to anticipate.

Subsections: Each subsection should correspond to a new experimental domain. In the first paragraph, describe the setup and why the experiment is interesting and is a good setup to rigorously test your algorithm. Use the same notation as in the problem statement.

Next, reference each Figure in order and show why they support your claim that your algorithm is better. Start with **aggregate metrics over several trials** such as boxplots/barplots of accuracy, control cost etc. Then, start with **qualitative, illustrative analyses**, such as visualizing timeseries and images. A good paper always gives the key take-aways and insights rather than simply stating your method is $X\times$ better than benchmarks. Follow the same structure and show the exact same metrics across diverse experimental setups.

Limitations: In 1-2 sentences, describe key assumptions and cases where you expect your algorithm will not do well. This should be a note to the practitioner and say you will handle them in future work. Reviewers often ask for this.

B. Guidelines

1) Figure and Caption Guidelines: All figures should be described and referenced in order in the text. Each caption should be descriptive and have a short title in **bold** describing the key take-away. The caption should describe why your method is better than benchmarks etc. Never have short, obvious captions that describe the plot title, such as "Plot of Accuracy vs. Latency".

2) Plots: The plots should be done using a combination of Matplotlib and the Seaborn Packages in python. All raw data to create the plot should be stored and backed up in GIT as a csv. Axes tick marks, legends, and axes labels should be *at least* **20 point font**.

If your axes titles or legends describe a mathematical quantity, first state the English name and the variable in LaTeX. For example, **Acceleration** a.

The following guidelines are for lines/curves:

1) Line width/thickness **3** or larger.
2) Different color **and** line style (dashed, dots, etc.) for people who print out the paper in black and white.
3) Use a descriptive legend.

VI. Discussion and Conclusions

A. Section Objective

Paragraph 1: Re-cap of novelty and key results

Highlight the key technical insight, contributions, and key numbers that describe why your method works so well compared to benchmarks. This should be similar to the contributions, but stress intuition and key novelty/insights.

Paragraph 2: Future Work

Describe next steps, open research questions, and why this line of work is important for the research community. Remind readers if the code or dataset is open-source and link to the URL. If you have read this far and are Sandeep's student, send him an email describing something you have learned from this document and only then start outlining your paper.

VII. Acknowledgements

Copy the NSF funding acknowledgement (or for your funding agency) here.

References

[1] S. Chinchali, A. Sharma, J. Harrison, A. Elhafsi, D. Kang, E. Pergament, E. Cidon, S. Katti, and M. Pavone. Network offloading policies for cloud robotics: a learning-based approach. In *Robotics: Science and Systems*, Freiburg im Breisgau, Germany, June 2019. *Finalist for Best Student Paper and Best Systems Paper.*

[2] S. Chinchali, A. Sharma, J. Harrison, A. Elhafsi, D. Kang, E. Pergament, E. Cidon, S. Katti, and M. Pavone. Network offloading policies for cloud robotics: a learning-based approach. *Autonomous Robots*, pages 1–16, 2021.

[3] M. Nakanoya, S. Chinchali, A. Anemogiannis, A. Datta, S. Katti, and M. Pavone. Co-design of communication and machine inference for cloud robotics. In *Robotics: Science and Systems*, Online, June 2021.

Appendix

The appendix should have detailed experimental settings, model architectures and hyper-parameters for deep learning experiments, and links to all URLs for datasets.

The appendix should also have the **full proof** for any theorems you cannot fit into the main document.