

# Research Practice Guidelines

Computation and Cognition Research how-to for new students

Last updated: April 29, 2026

## Contents

---

<b>1</b>	<b>Overview</b>	<b>3</b>
<b>2</b>	<b>Logs, Data Storage and Reproducibility</b>	<b>3</b>
2.1	Save everything . . . . .	3
2.2	File naming . . . . .	4
2.3	Directory structure . . . . .	4
2.4	Scripts: post-processing and visualization . . . . .	5
<b>3</b>	<b>GitHub practices</b>	<b>5</b>
3.1	Maintaining a code repository for paper submissions . . . . .	5
<b>4</b>	<b>Human Participant Data</b>	<b>6</b>
4.1	What gets stored and what gets shared . . . . .	6
4.2	Ethical treatment of participants . . . . .	6
4.3	Checklist: human experiments . . . . .	7
<b>5</b>	<b>Modelling Practices</b>	<b>7</b>
5.1	Model design documentation . . . . .	7
<b>6</b>	<b>Managing Compute Costs</b>	<b>8</b>
<b>7</b>	<b>Presenting Your Work</b>	<b>8</b>
7.1	Presenting at weekly meetings . . . . .	9
7.2	Presenting to collaborators . . . . .	9
7.3	Presenting at conferences . . . . .	9
7.4	Conference travel and reimbursement . . . . .	10
<b>8</b>	<b>How to read papers</b>	<b>11</b>
8.1	Paper reading strategy tips . . . . .	11
<b>9</b>	<b>Figure design</b>	<b>11</b>
9.1	Figure semantics is highly stereotyped . . . . .	12
9.2	Each figure makes exactly one claim . . . . .	12
9.3	Attributes encode semantics . . . . .	12
9.4	Remove redundancy . . . . .	12

9.5	Control the gaze . . . . .	13
9.6	Iterate with feedback . . . . .	13
<b>10</b>	<b>Getting Published</b>	<b>13</b>
10.1	How projects move through conference submission cycles . . . . .	13
10.2	Conference submission and post-submission revision . . . . .	14
10.3	Conference Rebuttal vs Journal Revision . . . . .	14
10.4	Project ownership, authorship and responsibilities . . . . .	15
10.5	Reference letters . . . . .	15
10.6	Contributing to and leading publications . . . . .	15
10.7	How the authorship is determined . . . . .	17
<b>11</b>	<b>The art of writing a paper</b>	<b>18</b>
11.1	Writing tips . . . . .	19
11.2	Writing the Introduction . . . . .	19
11.3	The title and abstract . . . . .	19
<b>12</b>	<b>Learning from others in your field</b>	<b>19</b>

## 1. Overview

---

The Computation and Cognition lab studies how intelligent systems — humans and machines — represent knowledge, learn from limited data, and reason about the world. We build computational models, run human and simulation experiments to test them, and use the results to advance fundamental research. The lab is highly interdisciplinary and collaborative, with projects involving people from multiple institutions and disciplines. The students who do best here are the ones who are comfortable with theoretical, empirical and computational approaches, self-motivated, and genuinely interested in how intelligence works.

The lab offers excellent professional development opportunities: mentorship from the PI and our collaborators, authorship credit commensurate with contribution, reference letters, networking opportunities, conference travel support, and hands-on experience with state-of-the-art research. To make your time here as productive and rewarding as possible, it is important to understand conventions and expectations by which research labs work.

This document is an attempt to compress the implicit knowledge that researchers absorb over years of experience, so that new students can get there faster with least trial and error.

*How to read this document* The section order approximately follows the order in which new students encounter each situation, but is not intended to be read in a liner order – feel free to treat it as a FAQ. The initial sections describe technical skills most effective in getting started, and the last sections describe social skills.

## 2. Logs, Data Storage and Reproducibility

---

Every experiment generates data that may be needed again — for follow-up analyses, reviewer requests, or replication. If data is lost, poorly named, or only partially saved, the experiment must be re-run. Workarounds, like reconstructing values from figures (e.g., reading off a histogram) should never be necessary. This practice applies to both data from human experiments, and data from any model simulations that cost money, time, and compute to generate the data.

**Assume that** any analysis in a paper must be reproducible by someone with access to the repository and no other information.

### 2.1 Save everything

For any experiments that cost time and money to run, we need to have a detailed logging procedure in place. For any experiment, where compute, API tokens, or human subject labor costs money, assume that the analysis you plan now is not the only analysis this data will ever need to support. Reviewers ask follow-up questions; new hypotheses emerge; others may extend the work. Data collection procedure should be rich enough to cover all such possibilities.

Examples of things we may want to reasonably record:

- **Computational experiments:** timestamps, model name, parameters, LLM prompts, diagnostic information, etc.
- **Human experiments:** Response values, times and accuracy, per trial; Full trial sequence,

including any randomization and counterbalancing conditions; Timestamps at key events (experiment start, trial onset, response, experiment end; Web-browser metadata, cursor movement, clicks, etc. Cursor movement data in particular helps to detect AI bots.

- **Both:** a processed CSV derived from the raw data, used for analysis and plotting should be stored in addition to the full log.

If you are unsure whether to save something, save it.

## 2.2 File naming

File names must be self-describing. Anyone reading your filenames — including you 6 months from now — should be able to identify the contents without opening the file.

For example, your file format can include fields like model name and version, key parameter values, date (YYYY-MM-DD), and a brief content descriptor:

```
{model}_{param1}-{val1}_{param2}-{val2}_{YYYY-MM-DD}_{descriptor}.{ext}
```

### Examples:

```
gpt5.1_theta_0.7_n100_2025-03-12_LLM_PS_S.json  
llama3_70b_temp_1.0_n50_2025-04-01_baseline.csv
```

Avoid names like `data.csv` or `output.json`, unless the model and parameter information is already conveyed by the directory structure.

## 2.3 Directory structure

Organize files hierarchically, reflecting the experimental space, for instance:

```
data/  
  {model_name}/  
    {parameter_set}/  
      raw/  
        {filename}.json  
      processed/  
        {filename}.csv
```

### Example:

```
data/  
  gpt4o/  
    temp-0.7_n-100/  
      raw/  
        gpt4o_theta_0.7_n100_2025-03-12_LLM_PS_S.json  
      processed/  
        gpt4o_theta_0.7_n100_2025-03-12_LLM_PS_S.csv
```

Each category (or parameter setting) gets its own subdirectory. Keep track of such logs in a well organized local directory, unless disk space becomes a problem.

## 2.4 Scripts: post-processing and visualization

During a project you will accumulate many scripts — data formatters, agent-generated figure scripts, exploratory notebooks with simulated model predictions. Store them in a well-organized local directory structure.

Two categories of scripts need to be tracked on github:

- **Post-processing scripts** that transforms raw logs into the CSVs used for analysis – i.e. the code an AI agent gave you to process the logs, or the prompts. Name such scripts clearly and keep them with the data they process.
- **Visualization scripts** that read data and produce figures appearing in the paper.

Everything that we tried but did not use in the paper – exploratory analysis, failed models, meeting notes – can be kept in a local scratch folder – just be mindful that you might still need it for a resubmission, or a reviewer request, and keep them well organized.

## 3. GitHub practices

---

During a project you will accumulate many logs, scripts, human pilot data, and exploratory models. Any models, logs and scripts that may be used by a collaborator should be committed to the lab repository continuously.

Make directories for `models/`, `scripts/` and `logs/` in the github repository and place pipeline-critical scripts there as you produce them. The repository structure should be designed so as to require minimal work when we are cleaning up the code-base for paper submission.

### 3.1 Maintaining a code repository for paper submissions

When submitting a paper to a conference or a journal, the logs produced by the final model version, the ones you’ve actually used to produce the results, need to be provided as part of the code repository. The goal of this repository is to help readers (or students who come to the project later) understand the results without re-running any expensive experiments.

When deciding what to leave out and what to include with paper submission, think what someone may need to quickly understand the paper. Specifically, it is helpful for the shared repository to include complete, traceable path from raw logs to every figure and table in the paper. The shared repository should not include: meeting notes, random logs, or exploratory models that did not make it into the final analysis.

A clean and organized repository signals intentional and high-quality work to the reviewers, and optimizes likelihood of acceptance. A minimal submission-ready repository looks like this:

```
data/  
  {model}/{params}/raw/*.json  
  {model}/{params}/processed/*.csv  
scripts/  
  process_logs.py # raw JSON -> processed CSV  
  figures.py # processed CSV -> paper figures  
figures/  
  fig1.pdf  
  fig2.pdf  
models/  
  {model}/model_fitting.ipynb
```

```
{model}/model.py  
README.txt
```

The README should document which scripts to run, in what order, to reproduce all figures from raw logs.

**Project management is a learned skill.** AI projects often iterate fast and with tight deadlines, meaning that a perfect project cleanup is not always feasible before submission. During each paper submission, we aim to approach the best practice the best we can. If a submitted paper is not perfectly organized, this should be treated as an opportunity for learning what we can do better next time.

## 4. Human Participant Data

---

Human data introduces ethical obligations: **IRB ethics compliance** (data storage, sharing, and participant treatment) and **scientific ethics** (sufficiency for replication and the value of the research).

### 4.1 What gets stored and what gets shared

We collect way more data than we share.

**Metadata (IP address, browser fingerprint, device info)** collected by web applications can be used for debug purpose only and accessible to the person who deployed the web application.

**Other potentially identifying data** include name, email, Prolific ID, can be collected during recruitment is likewise only accessible only to the person collecting data. These types of data are never committed to GitHub, never included in a shared dataset, and not passed to others.

**What gets analyzed and shared** is the anonymized behavioral record: responses, reaction times, trial sequences, and related measures, keyed only to a randomly generated session ID.

### 4.2 Ethical treatment of participants

**Instructions and interface design.** Instructions must be clear, respectful, and concise. Every design decision should be evaluated against: *is this serving the experiment?* If a task is confusing in piloting, that is a design problem to fix.

**Deception.** If the experiment withholds its true purpose or uses a cover story, participants must receive a debriefing at the end of the session explaining what was actually studied and why the deception was necessary. This should be written clearly.

**Compensation.** Online participants recruited through Prolific must be paid at least minimum wage, which we estimate from tracking completion time. If a study runs longer than expected Discuss with PI, as we may need to adjust this payment.

**What we do not collect by default:** webcam and eye-tracking data. Eye-tracking would be scientifically useful in many experiments, but introduces significant privacy concerns and would be difficult to give an IRB justification.

**The shared dataset for a paper** should contain every behavioral variable collected, stripped of identifying information, with a data dictionary defining each column. A reader should be able to reproduce the analysis from that file without contacting us.

### 4.3 Checklist: human experiments

- Metadata (IP, browser) not included in analysis files or repository
- Non-anonymized raw data accessible only to the person who collected it
- Participant identifier is session ID only — no names, emails, or platform IDs
- Anonymized behavioral data committed to repository at submission
- Data dictionary provided describing every column in the shared dataset
- Instructions reviewed for clarity; interface reviewed for unnecessary burden
- Debriefing included if any deception was used
- Prolific compensation verified against actual median completion time
- Reaction times recorded per trial
- Cursor movement collected

## 5. Modelling Practices

---

### 5.1 Model design documentation

Every new modeling analysis used in a project must be accompanied by documentation. This can take three forms, depending on what you did, and what conveys your ideas most effectively:

- **presentation slides** listing modeling steps and illustrating results with figures (minimal)
- a **modeling note** is a standalone  $\text{\LaTeX}$  PDF document listing any assumptions and model design decisions, describing how to run any accompanying code, and plots. This can include math or detailed verbal instructions you'd give to a coding agent to replicate your results.
- An `.ipynb` file with code cells providing implementation, and markdown cells documenting modeling assumptions. Do this when implementation-level understanding is essential to your claims.

**Best practice:** Write the modeling note in  $\text{\LaTeX}$  from the start to document your decisions, and share it with the project group as part of your project updates.

A convenient consequence is that any equations, notation, or framing precise enough to go directly into a paper or supplementary information (SI) can be copy-pasted from such notes with minor editing. In practice, well-maintained modeling notes are frequently copied into SI or methods sections with minimal change. Informal notes (Slack messages, google doc) are helpful, but are not a substitute.

**Do not assume** that you can turn up to group meetings with your code on your laptop screen, and that people will follow it and provide constructive feedback. Implementation-level code review is a last-resort intervention that we **sometimes** do to help new students, but if this does happen, modeling note, detailed logs and slides must accompany any such process.

**A modeling note should cover:**

- Any assumptions and formal principles used

- Free parameters stated formally, their priors (or fitting procedure), and the range of values considered.
- Fixed parameters and their values, with justification for each choice.
- Any approximations made and why they are acceptable.
- Any known failure modes or boundary conditions where the model breaks down.
- Version history and what changed, if this is a revision.
- Date

These notes should be either kept in the (private) project repository under `notes/` compiled to PDF, or shared with the group on a common channel in slack.

## 6. Managing Compute Costs

---

Running experiments against live APIs or cloud incurs costs.

If live API (or GPU hours) were used as part of dev workflow, expenses would add up quickly and consume a project’s budget before we even get to see exciting results.

Being mindful of compute costs is part of being a responsible lab member.

### **Rule 1: do not debug with live compute**

Anything that can be diagnosed using existing data — making a new plot, adding a data processing step, debugging pipeline — must be done using **saved logs**, and never by re-running against a live API or re-launching a compute job. Live API calls and GPU runs are for generating new data only.

In practice this means:

- Everything that could be debugged against saved logs should be debugged against saved logs.
- If something looks wrong in your results, check the logs first. Re-running to see if the problem disappears is not debugging.

This rule is followable because we save everything (see Sections 2 and 4) at a level where existing data can be used to ask a new question without spending new money. If you find yourself saying “I need to re-run this to check something,” that is a signal that logs were not correctly designed. To avoid disk space issues, consider structuring logs more effectively at the time of design.

### **Rule 2: API keys are only for the project they were issued for**

API keys may be provided to you for project use. These keys are linked to project budget, and should be used only for that project only. If you are unsure whether a planned API key use is appropriate, ask before running it.

### **Coding tools**

Are you using a personal paid coding agent subscription (i.e. Claude Code) on the lab’s projects? Please let us know as this may be reimbursable.

## 7. Presenting Your Work

---

Resources:

Patrick Winston, *How to Speak* <https://www.youtube.com/watch?v=Unzc731iCUY>

Presenting your work is as important as the work itself, as other people *will only know as much as has been presented to them*. This extends across all professional settings: in industry, your ability to communicate what you have done determines promotion; in academia, this determines career outcomes, collaboration opportunities and citations.

Common misconceptions about presenting research work:

- *Good work will speak for itself*. This does not generalize to research because mental work is invisible (unlike a physical skill that take minutes to show). To appreciate your work, people need to be shown what you did, explained how you did it, what were the issues you resolved, and why this is important.
- *High effort translates to high reward*. High effort can be made visible, but does not reliably predict good results on its own (unlike factory or implementation work). What is valued the most is research work that is helpful and effective – does the right thing at the right time, and moves the project along.
- *Only my PI needs to know about my work*. That may be the case for small things, but for example, if you are sharing slides with new results, then making them available in the general channel (1) allows others to learn from you (2) makes others aware of your work, and (3) opens discussion that often leads to productive feedback.
- *Smart people will follow a correct argument regardless of how it is presented*. Think about how many times you've spent an hour trying to understand someone's equations, only to get it in seconds once the intent was explained. Because of this, argument framing and presentation design are a major part of every research project, often developed via many iterations of discussion.

Because presentation is learned with practice, the lab aims to provide many such opportunities for students — from weekly updates to international conferences. The conventions below apply across all these formats.

### **7.1 Presenting at weekly meetings**

The slides you present at weekly meetings should start with what you tried, show the result, any conclusions this implies for the project, and what you'd plan to do next. This is a good time to consider any questions you'd like to discuss with the group (i.e. would model A or model B be the best course of action; which claims should we structure around).

### **7.2 Presenting to collaborators**

Presenting to collaborators from other labs, disciplines, and institutions, who are not part of your regular project meetings is a great way to collect feedback. Presenting to this audience requires more context; make sure to include a few slides explaining the problem area before technical details, and lead with explaining the research question, before presenting the method.

### **7.3 Presenting at conferences**

#### **Poster presentation**

A poster is a conversation starter. Its job is to attract someone walking past, give them the core claim in under a minute, and give you a "static whiteboard" that you talk about as you explain the

project.

A poster presentation is a networking and feedback opportunity. It is a success if person(s) coming to your poster express interest in reading the paper (a poster should have a QR link), express interest in working together, or ask interesting questions that we did not think about.

As a new student, you may find yourself presenting a paper where the research question, the framing, and possibly much of the modeling was done by others – a situation that becomes uncommon as your career progresses and you increasingly lead projects on your own. In such situations your job is to aim to understand the paper the best you can, including related work and theory that motivates the research question. Ask yourself: what someone would have needed to know to have come up with this project, and which of this is a critical background that should I know about? When interacting with the audience, notice which questions do people ask. What is most interesting to them? What is confusing? After discussing your project ask the person about their own work, to understand which types of perspectives motivate people’s interest in the project.

### Poster design guidelines

Designing a conference poster is the first step to prepare for your conference presentation.

**Method:** Use a presentation software, such as KEYNOTE. Drop figures and add text on a single slide –this will become your poster. Poster visual design principles are similar to figure design, and additionally, we make sure that the poster is readable from arms length. If you have never done this before – ask your PI for poster examples.

Prepare a draft, gather feedback, iterate and refine. Once done, export your final slide as a .pdf. Make sure you know the correct poster dimensions, which will be announced on the conference website. Print your poster in advance (!) – preferably locally before you travel, or, if you are confident you can do this, at a FedEx location after you arrive. Keep your receipts to reimburse.

### 7.4 Conference travel and reimbursement

The lab supports student travel to conferences where they are presenting work. Conference travel is a professional development opportunity for learning which questions are current research drivers in the field and building visibility as a researcher. Take it seriously.

Conference travel checklist:

- **Plan early.** Book flights and accommodation either as soon as registration opens or as soon as your presentation is confirmed. Last-minute travel is significantly more expensive.
- **Keep all receipts** as reimbursement requires documentation. This includes flights (economy only), accommodation (typically a hostel, an airbnb, or another economy option), registration fees, and public transport. While Uber is often a reasonable option (i.e. at night) it should not be the default when public transport is equally available. Meals are reimbursable up to a certain rate – ask PI before the trip.
- **Submit reimbursements after the trip** If you are unsure how to do this, ask the PI.
- **Example expenses that will not be covered** Flight upgrades, alcohol, candy, personal travel extending the trip beyond the conference dates, non-economy dining options.

## 8. How to read papers

---

Reading papers in your field is the best way to learn which research problems are important, which methods are typically used to solve them, and why do people care about solving these problems specifically. Importantly, the goal of reading research papers is to gather that information – it is not to simply *read* a paper end to end. Reading papers effectively is a learned skill.

To learn this skill, the first thing to notice is that **papers have a stereotyped structure**. Every publication venue (conference, journal) has its own conventions for how the sections are organized, but these conventions share a common structural logic. For instance NeurIPS (AI) and Cognitive Science Society (cogsci) follow similar logic, even if they differ superficially in length and style.

Typical paper sections include:

- **Abstract** — One paragraph. States the question, the approach, and the main finding. A reader should be able to decide whether to read the paper from the abstract alone.
- **Introduction** Motivates the research question, briefly situates it in the literature, and states the contribution. The goal of the introduction is to convince the reader that the question is important and the given methodological approach is logical and inevitable.
- **Methods** Describes what approach was used to solve the problem posed in the Introduction. Computational model and human experimental tasks are described here.
- **Results** Reports what was found, and shows figures illustrating how data supports these findings; typically boring, straightforward and optimized for clarity.
- **Related Work** Positions the paper relative to existing work to show how it is similar and different from all these works. In some venues this appears after the Introduction; in others, before the Discussion. The goal of related work is to convince the reader that (1) this research has not been done before, and (2) this research is topical: many others work in this area.
- **Discussion** Interprets the results, connects them to the introduction’s claims, discusses limitations and future work: what did we learn, why does it matter, and what should be done next?

### 8.1 Paper reading strategy tips

1. A highly recommended exercise is to make slides summarizing the papers you read and like. You can later use your slides as memory aides when using a paper for related work, or copy paste them into a presentation when presenting to others.
2. Use AI to give you an executive summary before reading a paper. The paper will be easier to follow once you know what to expect.

## 9. Figure design

---

*Recommended reading: Patrick Winston, "Make It Clear"*

Figure design is one of the highest-leverage skills in academic research. A well-designed paper is one where a reasonably informed reader can accurately reconstruct your main claims from the abstract and figures alone — without reading any body text. Reviewers frequently decide whether a paper is promising within minutes of reading; those minutes are spent on the abstract and Figure 1. Post-acceptance, a paper’s impact and citations are strongly determined by how clearly it communicates its contributions; figures constitute a large share of that clarity.

Therefore figure design is a core research communication skill. A figure with miniature fonts, visually cluttered, or ambiguous about which conclusions should be drawn from it will cost you reviewers, citations, and collaboration opportunities – regardless of the quality of code, models, or data that it is presenting. Therefore, figure design generally consumes a significant time and effort, often via multiple iterations of sketches and discussion with your project group. The following points will help you focus on what counts most in figure design.

### 9.1 Figure semantics is highly stereotyped

In AI/cogsci figures have distinct roles:

- Figure 1 (concept figure) explains what you did and why. This makes the main claim of the paper obvious.
- Figure 2 (experimental task or methods) explains how you did it. If the paper run a behavioral experiment: what did people do? If you designed a computational framework: what were the parts?
- Figure 3-4 (results) show that your claims worked; these figures provide evidence for your claims.

### 9.2 Each figure makes exactly one claim

Before designing a figure, write down in one sentence what it is supposed to show. If you cannot do this, the figure is not ready to be made.

**Fonts must be readable.** A figure that requires zooming is unacceptable. When sharing figures, even informally with collaborators, make it your default practice to ensure it has readable fonts; when making figure to be used in a paper, match fonts to the size text will have once your figure is embedded in a rendered PDF.

### 9.3 Attributes encode semantics

It often is a student’s natural instinct to make figures *pretty* and *colorful*. However, in a presentation slide or a figure, every visual attribute that can vary — color, stroke width, font weight, shape — carries semantic weight. For instance, if two things are a different color, they will be seen as a different type of thing. If they are not a different type of thing, they should not have different colors. Best practice is for every visual attribute that varies in a figure to correspond to exactly one semantic distinction.

**Color is powerful, so use it sparingly.** Assign color to at most two or three categories, and use it consistently throughout all figures in the paper. Do not use color to decorate.

**Simplicity is powerful** because it makes your reader feels smart. A simple, clear Figure 1 that can be read in a minute is a much more effective tool for conveying your claims than 3 elegant equations that the reader will appreciate after having spent an hour thinking about them.

### 9.4 Remove redundancy

For every element in the figure, ask: if I removed this, could the reader infer it from the rest? If yes, remove it. Labels that restate what an icon already shows, borders that whitespace already implies, and text that the caption covers are all candidates for removal.

## 9.5 Control the gaze

Readers scan left to right, top to bottom. Information should unfold in that order. Arrows that double back, columns that require reading right-to-left, or layouts that force the eye to jump around are all legibility failures.

Whitespace is the most versatile organizational tool available — use it to separate groups, imply hierarchy, and avoid clutter. Boxes and lines are often redundant once spacing is handled correctly.

## 9.6 Iterate with feedback

A figure typically goes through multiple critique rounds before submission. Expect to share early drafts with the group as you are working on them, and ask specifically: *what does this figure say to you?* If the answer differs from the one-sentence claim you wrote down, the figure needs revision. A good figure typically requires 4–6 rounds of critique and redesign – so if you are preparing a figure, plan accordingly, and allow yourself at least a week for reflection and redesign.

## 10. Getting Published

---

Our goal is to help lab members achieve the best possible outcomes — which, in practice, often means getting work into a competitive venue. In practice, this process involves following steps.

### 10.1 How projects move through conference submission cycles

Research projects are developed iteratively, where iterations are structured around conference deadlines. A typical trajectory:

- exploratory prototype (this is where you try something to see if it works, and may present it to PI and close collaborators)
- initial empirical results submitted as a non-proceedings conference abstract (i.e. 2-page abstract submitted to CCN)
- A non-proceedings conference or a workshop paper (i.e. 6-page paper for Cognitive Science Society)
- A conference paper at major AI proceedings conference (i.e. NeurIPS), which may be followed by a sequence of resubmissions (if the first is unsuccessful)
- a journal paper (once the project has matured)

Not every project follows all of these steps, and not all labs follow all these steps habitually, but the general progression of *prototype*, *abstract*, *workshop*, *conference proceedings*, *optionally journal* is a good general model to understand the stage your project is at.

**Conference abstract** Presenting research at conferences is the primary venue for feedback during active development. Submitting frequently, as results progress, keeps the work aligned with the field and topical. Abstracts can be 1-2 page long.

**Conference proceedings** A typical conference review cycle runs 3–6 months from submission to decision. During this window, work on the project continues toward the next phase. Do not assume that the project is done because it is submitted. Upon receiving an initial decision, we implement rebuttals and revisions, and upon receiving a final decision we may need to prepare a camera-ready version. Papers are 6-9 pages long.

**Journal papers** follow a similarly structured but a slower and less predictable cycle, sometimes

spanning years across submission, review, revision, and final publication – and it is partly for this reason that we target AI conferences before going for a journal (more prestige; could be way slower).

## 10.2 Conference submission and post-submission revision

Conference acceptance is a major checkpoint in a project cycle. Between initial submission and camera-ready, a project can be improved **as long as the improvement does not change material claims**.

Two channels support this.

**arXiv.** After submission, we may decide to post an arXiv preprint. The arXiv version gets indexed by Google Scholar and cited by people who find the work before it appears in proceedings. If you discover after submission that something should be corrected or clarified — a parameter value, a figure, a sentence in the methods — update the arXiv version. This is standard practice and the appropriate mechanism for fixing small things without waiting for camera-ready. It also keeps a record of what changed between versions (arXiv provides version history automatically).

**GitHub.** Post-submission is the appropriate time to ensure the repository is clean and complete (logs, scripts, data dictionary), and to incorporate any corrections identified during review. Do not treat the submission-time state of the repository as frozen.

**What this means in practice:** if after submitting you realize corrections are needed, update arXiv and the repository. This is best practices in the AI and cogsci ecosystems that ensure the record stays accurate.

However, if you'd like to make changes that would materially alter the scientific claims, this should be discussed with the PI. A material change is anything that affects what the paper asserts, such as a revised conclusion or a qualitatively different claim — as opposed to corrections that leave every claim intact. In practice, if this really is a material change, it will not be done to an existing paper, but it will affect how we write any new papers that builds on that work.

Camera-ready is an important deadline for the proceedings version. Use the time between submission and camera-ready to incorporate reviewer feedback, fix anything identified during that period, and ensure the repository and arXiv version are consistent with the final paper.

## 10.3 Conference Rebuttal vs Journal Revision

**Conferences often have a rebuttal phase**, where you respond to reviewers. This needs to be done in a short space of time, such as 2-3 weeks. In theory, rebuttal is an opportunity to increase the review score by responding to questions. In practice, the most likely outcomes of a rebuttal is that it either gains you 1-2 points which do not change the accept/reject outcome, or it gains nothing. The best practice is to still write one conscientiously: responding to a rebuttal provides a useful reflection toward preparing a resubmission; there's also a small chance that the reviewer is someone we know, and so engaging in good faith signals a constructive attitude.

How to do this: respond to every substantive point, be concise, do not be defensive, and if a reviewer misunderstood something the writing was probably unclear – this is an opportunity to think how we can fix it. Thank the reviewer. Do not argue. Regardless of what you believe of the situation, a rebuttal response has to be 100% dispassionate, professional, and in good faith.

Unlike a rebuttal, **a journal revision** for the most part provides a good chance to get the paper accepted. You will also have more time to complete it, but the reviewers will likely ask for more

extra work. Same engagement principles apply.

#### 10.4 Project ownership, authorship and responsibilities

There are several ways in which someone's contribution can be credited in a published work, each requiring different amount of commitment.

- **Acknowledgment section** Examples: productive discussion, contributing exploratory analysis which did not go into the paper, but was useful in some way, reviewing the draft and providing constructive feedback, sharing existing data, or a web-app used by the project;
- **Authorship** entails that the paper would be substantially different without that person's contribution. Examples: writing a section, building a new model, providing original data analysis, collecting data
- **Authorship order** means different things in different fields; In AI/cogsci first and last author positions carry the most weight. First author(s) is typically a student who has done the most implementation work. Last author(s) is the PI whose lab supervised the research.
- **Presenting research** as a talk or a poster. This can happen at different levels, and carries different weight. Presenting to collaborators  $\leq$  presenting to your department or campus  $\leq$  presenting at an international conference.

#### 10.5 Reference letters

Not all undergraduates contribute to published work, for instance, you may be working on a prototype project, or something that does not make it into a publication, etc. Even if you did not contribute to published work, **reference letters** sometimes can be a lot more valuable than authorship, and can happen with or without it.

The best way to get a good reference from our lab is to do your best at being helpful, responsible, motivated, and contributing materially to a project the best way you can. By doing so, you will be giving your PI(s) lots of material to work with and shape into a strong reference letter when you need it.

#### 10.6 Contributing to and leading publications

The level of responsibility (authorship credit) and project ownership (first authorship), that someone can expect, is directly determined by someone's *ability* and *motivation* to advance the project. In practice, this means being able to contribute materially within project timelines, where a **material contribution** implies that the final paper would have looked differently without you. Two points will help you understand how to do this most effectively to make your experience as productive and positive as possible.

#### Point 1: Successfully navigating deadlines

Nearly all active projects have deadlines and time constraints. If you are not sure what yours are - do ask.

**Exploratory prototype deadlines** New students are often given simple, exploratory tasks without an explicit deadline. This is done deliberately to avoid stress, and simply because we want to get to know how you work before entrusting a critical responsibility. The amount of responsibility and project ownership increases as your skills and experience grows. In practice "no deadline" does not mean that taking weeks or months is a reasonable option; such low priority tasks are your

opportunity to demonstrate your motivation, skills, and time-management. Students who do well will be included in impactful projects. When things do not go as expected, that provides useful information. For the PI, this helps to understand your motivation (did you try to do it?) and skill calibration (how did you approach the problem? which questions did you ask?). For the student, this helps to understand gaps in your skills or adjust your approach.

**Conference deadlines** are known in advance and are non-negotiable. This is a day by which the research paper needs to be complete and uploaded to the conference website.

**Journal deadlines** are flexible at submission time, but can be fixed at revision (i.e. provide analysis X by date Y).

Because research is unpredictable and deadlines are fixed to a non-negotiable date, the weeks immediately before a fixed deadline **may require** concentrated effort, weekend work, rapid iteration, etc. While working long hours before a deadline is not the default, it is not unusual. The cost of missing a conference deadline is  $\approx 6$  month delay before the next comparable submission opportunity. For AI-adjacent projects, a 6 month delay often means that most of the project has to be redone. Missing a deadline because of poor planning can cause significant loss of future opportunity, simply because the research ecosystem prefers reliable collaborators.

Therefore, at all times you should know (1) when the next deadline for your project is and (2) what is critical to do before then. For instance, if you know that *we are planning to submit to NeurIPS*, you are expected to look up when NeurIPS is, and treat that as a deadline signal. To make holidays and time off a positive experience, you need to take deadlines into account in your plans. It is important to avoid a situation where you have a deadline responsibility that you are unable to meet. If you are unsure if a certain project is compatible with your time constraints, discuss with PI. Emergencies should be communicated to the PI as soon as you can, so we can redirect effort (i.e. have someone else finish your work).

## Point 2: Project organization

**Structuring your work around weekly project meetings** is a central component of successful research trajectory. Weekly meetings provide structure for project members to present updates and discuss any outstanding action items for the project. Not everyone presents updates every time. If you have updates to present, please let the group know, and do not wait to be asked.

Lab members are expected to make **slides** or a modeling note when presenting updates. It should take you less than an hour to make slides. Slides are a highly effective way to help others understand you, and to help you keep track of your own work. Always share your slides (or modeling note) on slack before the meeting, and save the slides you've presented for your own records, clearly labeled with project name and date. A common misunderstanding for new students is coming to a group meeting with code on their screen, and expect others to read it and fix things online.

Outside of these regular project meetings, every lab-member can request to meet with the PI to discuss your plans and how the lab can best support your work. At the beginning of your time in the lab, we may setup a regular time for such meetings, to discuss papers that you read and find interesting and find a research direction. If you do not have a regular meeting time, it is ok to ask to meet anytime!

### Point 3: Time management and project organization

**An adequate time commitment** is a pre-condition to success.

**Undergraduates are expected to commit on average 10 hours per week** under normal conditions. In the 6–8 weeks before a deadline, that time commitment may be higher, depending on project need.

**Graduate school is a full-time job.** This is worth stating explicitly because it is frequently underestimated. The expectation is sustained engagement throughout the year, not during term time only. Time between formal academic semesters is not vacation — it is often the most productive period for research, free from coursework and TA obligations. Vacation should be planned deliberately, communicated in advance, and kept within reasonable bounds (this means - within 4 weeks a year) and concentrated around non-deadline periods. It is a misconception to treat semester breaks, or the summer term, as a holiday.

**Where to work.** The lab does not mandate physical presence or a fixed schedule. Lab members are expected to manage their own time: we trust that you are motivated, and you understand that the amount and quality of publications you produce determines your research career outcomes. Additionally, remote work is a normal part of how we operate given our international collaborations. However, working in proximity to others whether in shared space or via a regular online presence — accelerates learning and keeps you connected to the project. New students in particular benefit from being around more experienced researchers. If you are working fully / mostly remote, make an effort to stay visible on Slack and engaged in meetings. Disappearing between weekly project meetings is not the same as working independently.

#### 10.7 How the authorship is determined

**Authorship inclusion** should be easy for you to understand intuitively. Ask yourself: *how would this submission have looked without me?* The part that would have been missing is your contribution. If that part is substantial enough that the paper would be meaningfully different without it, you are an author. If your contribution was not at that threshold — attending meetings, giving feedback, piloting an idea that did not make it in — you will be acknowledged in the acknowledgements section. Both are valued.

**Authorship order** decisions are made by the PI(s) shortly before each submission, and reflect the degree of contribution and responsibility *toward that specific deadline* — not effort that someone spent, not intended contributions that did not materialize, and not seniority or time in the lab.

In AI and cognitive science:

- **First author** is usually the person who did the most implementation toward the submission – implementing the models, collecting the data, building any primary analysis, writing the paper, designing experiments, etc.
- **Middle authors** contribute meaningful components of the analysis, a proof, a dataset, a model, a figure, or a section of the paper.
- **Last author** is the supervising PI(s) whose lab led the project - designing the research question, conceptualizing design, writing and editing the paper. Funding acquisition can constitute a contribution to supervision; administrative supervision alone does not.
- **Shared author position** Any authorship position can be shared. When multiple persons

contribute equally, they share an *equal contribution* or similar footnote.

Authorship order **can change between submissions of the same project**. If a student led the first submission but steps back before the second, the order can shift to reflect who did the work for that deadline. This is a standard practice where submission windows are narrow and project contributions shift over time. When a student's planned contribution has to be delegated to someone else to meet a deadline, the authorship will reflect what was actually delivered, not what was intended.

**New students** usually begin as a middle authors on a project where the research question, framing, and much of the modeling were done by others. Your job in that role is to understand the paper deeply, including the related work and theoretical motivation, and to build an understanding what you'd need to do to lead your own project.

Sometimes, when all students in the group are new to research, PI(s) may contribute substantially to writing the paper, data analysis, experiment design, model implementation and so on. This happens not by design, but because PIs take over tasks that cannot be delegated. If this happens, a student may be still given first authorship; authorship decisions in such situations are made on case-by-case basis. Do not assume that there is a fixed playbook for this, because (1) the default mode of operation is that the first authorship entails intellectual ownership of a project, and (2) PIs often prefer last authorship position. The same person can not be both first and last.

**We use the CRediT taxonomy** (<https://credit.niso.org>) to decide who did what on each submission. Many journals require it; we often use it even if not formally required, because it makes contribution explicit and reduces ambiguity. Familiarize yourself with the taxonomy — it provides useful vocabulary for understanding your role. If you are unsure about your authorship status on a project, ask.

**Will I have a project of my own?** It's ok to discuss your *expected* contributions and credit at the beginning of your involvement in a project, however the final credit is determined by the actual outcome. We sometimes encounter a **misconception** that authorship position or project ownership can be **promised** to a student. Such beliefs arise from a PI saying something like "If you work on this you can be a first author.." – such statements express an intention, not a guarantee.

## 11. The art of writing a paper

---

*The best way to learn to write a good paper is to practice a lot of writing.*

I *almost never* have to say this, but – AI cannot generate original research contributions — novel claims, interpretations of results, theoretical framing, original writing. Attempting to use it for that wastes time and produces noise. AI-assisted writing works well for boiler-plate, generating latex tables, visualization, or answering simple forms.

When you first start as a research assistant or a new student, you will be contributing to papers written by someone else. Over time, you will be expected to build independence, and increasingly write papers of your own.

### 11.1 Writing tips

1. A common instinct is to write abstract first, this is okay to do to setup the context for further writing – however, abstract is nearly always rewritten once the paper is complete and it becomes clear what can be concluded from the results.
2. If you can anticipate who will be the reviewers on your work, make sure to cite them in the related Work and the Introduction. Additionally, cite key references in existing literature that much of the field will recognize.
3. If you have not yet contributed to paper reviews, ask the PI! As a reviewer, you will see a lot of good and bad papers. This will help you understand (1) what reviewers are looking for in your papers, and (2) how to know that you’ve done a good enough job (hint: upon submission, your paper should look like the good papers).
4. When reading a paper you like, notice the opening paragraph in the Introductions. Analyze the strategies they used to write it, and learn to simulate these strategies. Reflect on what made the paper convincing and how can you leverage the same approach.

### 11.2 Writing the Introduction

The job of the Introduction is to convince the reader that your research question is valuable, and it has a disproportionate weight on whether the paper is accepted. Introduction takes the most work to write, and should be designed to tell a story. Ask yourself, who are the audience you want to convince? Which kinds of openings will resonate with them? How to convince them that the work you’ve done is necessary?

### 11.3 The title and abstract

The title serves several purposes.

1. **Recruiting the right reviewer(s).** At venues with broad reviewer pools (e.g. NeurIPS), papers are routed by area, and reviewers get to “bid” on papers they are willing to review based on title and abstract. Being assigned a reviewer outside your expertise area, someone who will not understand your paper or does not care about your question, will certainly cost you points, and will likely cost you rejection. Therefore, the title and abstract should signal the subject area to the right reviewers who will be sympathetic to your work.
2. **Discoverability and citability** Search engines index papers primarily on title and keywords. A title optimized for the terms people search when looking for work to cite will be found more often and cited more. Your arXiv title may differ slightly from your submission title — and should, if the submission title optimized for the venue acceptance. Post-acceptance, you can again update the proceedings title to maximize discoverability.
3. **Notice how highly cited titles are structured.** Be strategic.
4. **GenAI training corpora.** Papers that are clearly written are more likely to be included in training corpora for language models. The AI systems trained on your paper will be more likely to recommend it to people using AI to scope related work.

## 12. Learning from others in your field

---

Many of our projects connect large groups of collaborators. Additionally, the lab aims to provide networking opportunities for students to attend high-profile conferences, and meet researchers who

may become collaborators and mentors. Learning research skills from others is an important and highly effective way to optimize your success. New students may be invited to join project meetings, even if they are not directly contributing to it, as a way of understanding the research ecosystem.

**The best practice**, which we aim to follow for all projects, is to structure a project around regular weekly meetings, with asynchronous discussion on SLACK. Following the discussion on slack, sharing your updates with the group, and engaging in that discussion to the extent you are able is a highly effective way to learn how projects are organized, and eventually own a project yourself. Examples of things you can learn by engaging with slack discussion:

- Deciding which claims should go into the paper
- Figure design – this is extremely important for Figure 1 (concept) and Result Figures (motivating your conclusion).
- Model and experiment design
- Paper framing
- Effective research communication and which ways to presenting updates work most effectively

If you see someone doing something exceptionally well:

- an intuitive figure design approach,
- a clear and convincing slide layout,
- a modelling note that can be dropped directly into SI,
- a conference poster that is particularly well organized and informative,
- a highly effective way to organize research notes,
- a smart way to use AI coding agents,
- a paper section that reads particularly well, –

Tell the person nicely what you liked about it, ask how they did it, and learn from that approach. The lab is a collaborative environment, and how much you get out of it scales with how much you engage.