Guide for PhD students (Pitt CS)

Hello! We are happy that you have chosen to pursue your PhD degree at Pitt CS!

Some of this content is also covered in CS 2001. This document was created to provide self-contained, reasonably concise advice. It also describes the faculty's expectations on some milestones which students often struggle with, such as the comprehensive exam.

Intro to research

The challenge in research is how to find problems that are (1) interesting, (2) important, and (3) feasible, i.e. no algorithms currently exist, or algorithms are not sufficient, but the task is not completely out of reach.

It is really important that you understand YOU lead your research projects, not your advisor. You have to take initiative. Of course, your advisor will be involved in the thought process, experiment design, and paper writing, but there are many mid-level tasks (specifying algorithm details) and low-level tasks (coding, registering paper abstracts by the conference deadline) that your advisor usually cannot be involved in. It is also important to clearly describe your research progress to your advisor, including what hypotheses you were testing in your work. More on this below.

It is also important to work hard, without becoming stressed. Computing research moves fast and you want to avoid someone else publishing a paper with your idea before you do. However, graduate school is also nice in that offers flexibility. Set ambitious goals for yourself for every day and every week and try to achieve them, but don't become overwhelmed if you cannot always achieve them.

It is also important to set clear expectations of your work with your advisor, as well as agree on protocols you are expected to follow.

Reading papers

A key aspect of being a good researcher is being well educated. This means knowing how to make a certain idea work, and also knowing what ideas other researchers in the field are currently exploring. You don't want to reinvent the wheel-- it is a waste of your time, it is not interesting to others, and you might be trying an idea that is known not to work well. So, you need to know the relevant literature well. This is just as important as knowing how to code.

Find out what are the good conferences in your area. Be proactive about finding papers to read; do not rely on your advisor to send you readings. However, you cannot read all papers in their

full detail. Thus, you have to choose a few topics that you're interested in, that you plan to follow. It is good to read for both "breadth" and "depth". Skim some papers so you have a high-level idea of the field. Then read some papers in great detail so you learn about the methods and evaluation (often just as important as the methods).

Many conferences have proceedings online. Also learn to use Google Scholar; for example, when you find a relevant paper, it is helpful to click on "Cited by" and look through the titles of papers that cite this relevant paper– some of those will be relevant too, and they're more recent. There are also arxiv, online platforms that show leaderboards for particular benchmarks (e.g. https://paperswithcode.com/), etc.

Code and experiments

When you write code and conduct experiments, make sure to regularly stop and (a) document this code, and (b) check this code. Mistakes get made but the time to discover these errors is before a paper gets submitted. Your evaluation pipeline especially should be set up early, a few weeks before paper submission. You should make sure this pipeline is fair and correct.

Record the settings used to conduct each experiment, for repeatability and paper writing. Do as much as possible with a script rather than manually, so that you have a record of the full process that was used to produce your results.

It is a good idea to conduct code reviews with your labmates. You should sit down with a labmate, and explain your code to them, step by step. If something doesn't make sense to them, try to explain it, but also look at the code carefully and check that you didn't make a mistake.

It is also a good idea to open-source your code, assuming this does not violate any policies. This helps your work be noticed and remembered. Please code with the intent to publish your code.

Computing resources

Your advisor might have dedicated machines for you to use. The CS department also has machines (https://tech.cs.pitt.edu/faqs) and the RIC cluster with 13 machines. Other resources include CRC (https://crc.pitt.edu/) which has training resources and workshops, and the Pittsburgh Supercomputing Center. Your advisor typically needs to be involved for you to get access to these. Cloud services are of course also available. Do not pay for any services out-of-pocket without discussing with your advisor, as there may be rules preventing these costs from being reimbursed.

Progress log and meetings with your advisor

It is a good idea to keep a log where you document your tasks and progress. You can share this log with your advisor, and use it to provide structure to the meetings.

When you meet with your advisor, summarize what tasks you tackled since the last meeting. Include your motivation– what did you do, what did you hope to learn, what approach did you take, what did you find out, etc.

If you run into problems between our meetings that prevent you from making progress, that you cannot resolve on your own, it is a good idea to reach out to your labmates or advisor, rather than stall and not make progress.

Writing papers

The importance of paper writing quality is greater than you might think. There have been plenty of papers whose idea, algorithm and results were good, but the papers got rejected because the authors didn't spend enough time on the paper text, or because they didn't know how to write a good paper.

A paper should tell a story. It should address the following questions: What is the current scientific "gap" that you want to address, i.e. how are existing methods for an important problem insufficient? Why is the problem important? In one sentence, what does your work contribute that previous works omitted? Why is your idea/method a good idea and why do you think it should work? What was your hypothesis, and how did you evaluate it?

The paper should be exciting to read, but the language you use should be neutral, i.e. avoid colloquialisms or exaggerations.

When writing a paper, start with an outline. What do you want to say or show in each section? In each paragraph? It should be clear from the start of each paragraph where that paragraph is "going".

When describing your approach, do not make the description too low-level. Talk about the ideas first, even when showing an equation-- before you show the equation, briefly explain what intuition/idea it captures.

Many conferences use LaTeX for typesetting, so it is good to learn it. Overleaf.com is a good resource and allows you to collaborate on writing and editing.

Make sure to agree with your advisor on when they expect to see a draft of a paper before conference submission (e.g. a detailed outline 3 weeks before, and a full paper draft 1 week before the conference deadline). It is good to also discuss expectations about the author list on papers whose development you are leading or involved with.

A good research paper:

1. Proposes a novel idea
2. Tells a coherent story
3. Backs up its claims (about novelty, method working well, etc) with clear evidence
4. Does a good job reviewing prior literature that is relevant
5. Describes the proposed method clearly, in reproducible fashion
6. Is consistent in terms of formatting

A good research paper does not have to:

1. Solve the problem completely
2. Describe related work in too extensive detail

Presenting work

Your presentation should show your excitement, and should excite others. In your oral or poster presentation, you don't need to give all details. You want to convey why what you worked on is exciting and important, how what you did advances the field, and briefly give the intuition about how the method works, and how you evaluated it. Good presentation skills take practice. Practice the talk in front of your colleagues and friends, or even if front of yourself, by recording a practice. It can be painful to watch but it helps. Try to prepare your presentation a week or two in advance, so you have time for multiple practices (with time away in between).

Attending conferences

Use conferences to network, i.e. meet new people. Knowing people is important for future opportunities for research collaborations, getting a job, getting invited to give talks, and even feeling like a part of the community.

Networking is hard if you're shy. It helps to do a partial version. Say on day 1 you met a new friend at the conference; on day 2, do hang out with this friend (i.e. have lunch with them), and meet people that *this friend* already knows. Thus you're expanding your network while being with people you know somewhat, and it feels less intimidating.

Students at large labs / large universities (e.g. Berkeley) have an advantage; you have a disadvantage because Pitt's grad cohort is relatively small. Thus, your "crowd" is naturally small, and you only have a few people that you know that you can hang out with. However, as you become more senior, you'll meet more and more people.

Think about how you present yourself at conferences (but don't *over*think it). Think about what kind of people you would like to meet-- probably ones who seem confident and smart, who are excited about the research they are doing, and who want to hear about what others are working on. So try to be that person-- kind, open, optimistic, and interested in others' work.

Should you introduce yourself to "big names" in the field? Yes, if there is a good opportunity, e.g. they're giving a talk at a workshop-- you can ask them a question about their work.

Being a good colleague

Getting a PhD and doing research is fun but can also be intimidating. You may feel great about it, or you may feel scared and uncertain at times. It helps to have a support group either way. Be kind to your labmates and everyone else you meet along the way. Be inquisitive and ask hard questions, but do not belittle or mock anyone, and do not ever try to make them feel like their work is not worthy, or like they are not smart or driven enough. Everyone is working on something they think is worth it-- have respect.

Do notice people around you. If one of your colleagues is feeling upset or down, ask them how they are doing and if they want to get lunch together, then try to see how they feel and whether you can help. If someone is happy, share their excitement. Always be supportive and encouraging.

Ask your labmates about their work and brainstorm with them if it feels appropriate.

Be smart but also generous with your time. Don't offer to help everyone all the time but do offer help sometimes. Spending a few hours helping someone will take away from your research but usually won't affect it in a significant way. And you can help someone else get their paper accepted because of your perspective and knowledge.

Comprehensive exam

In the Pitt CS comprehensive exam, you work with three faculty members (presumably your advisor and two more) to identify one topic area per faculty paper, on which you present essentially a survey of the literature. The topic area should not be too broad (e.g. machine learning) but something a bit more specific. Depending on the area you are working in, faculty may have different expectations.

In AI comprehensive exams, students typically choose 10-15 papers per topic, and share the list with the faculty for feedback (before the exam). The student's goal is to organize the papers in each area in some meaningful, interesting way. Show families of methods, discuss their similarities and differences, and their strengths and weaknesses. Do NOT show one slide per paper, and do not talk about any one paper in too much detail. Think of the comprehensive exam as writing a survey paper on the topic. You want to present the general directions researchers have explored, educate the audience about when they might choose one family of methods over another, and discuss limitations and open research challenges.

In some systems comps, each topic area has 3-4 papers, but the student should know these papers really well. Sometimes faculty choose seminal papers for the exam, and sometimes faculty request students to pick the important papers in the topic area. Students are responsible for knowing the ancillary/pre-requisite materials for these 3-4 papers, and are also supposed to know how to apply the ideas of the papers to solve other problems. Not too many papers, but a lot of thinking about the papers (about the techniques and how to apply them).

Thesis proposal and defense

Students typically propose when they have an established research direction, and likely some papers published on the topic. The proposal describes the completed work, and identifies one or two additional units of work to complete before the student can present their defense. Both milestones involve a significant writing effort, and you should give sufficient time to your advisor for feedback on your writing. It is a good idea to explicitly identify the hypotheses you investigate in your work.

Advice from the faculty (best part of this document!)

- "Get your hand dirty."
- "Anchor your sense of self in the flexibility rather than consistency. Let go of knowledge that no longer serving you well…" quote from the book *"Think Again" by Adam Grant (highly recommended)*
- "Research is not only about pushing the boundary of knowledge, it's also about pushing more people to the boundary of knowledge." Doing research to lower the barrier of entrance to your research community rather than set a high bar there.
- Your attitude determines how far you go. Ph.D. is not about finishing the jobs I give to you, it's about solving the problems you care most and make intellectual breakthroughs. Don't expect to get a todo list from me; Instead, I will guide you to define your own research agenda.
- Be skeptical when reading papers. Ask yourself: does this research problem make sense to me? Is there a simple way to solve this problem? Why conduct the experiment this way, not that way?
- Have long-term and short-term plans for your research.

- Writing Tips for System research: https://greatresearch.org/2013/10/11/storytelling-101-writing-tips-for-academics/ ii): https://www.usenix.org/legacy/events/samples/submit/advice.html iii):http://www.linzhong.org/opinions/writing.html

- Time management is an important skill to build:
    - When you talk to your advisor and other faculty, you will likely get about 1000 ideas to work on. You don't (and can't) do them all. Your job is to prioritize which is most important (timely, accomplish-able, practices a valuable skill) and work on that.
    - Try to "pop up a level" to describe what you're doing and why relatively frequently. That can help you avoid going down rabbit holes of debugging code forever or building small apps or endlessly tweaking datasets, etc, where there might be quicker and easier alternatives. Writing down weekly goals and accomplishments and discussing those with your advisor can help retain your focus on what matters.
    - Focus on progress, not perfection. Have something to show your advisor when you meet even if it's not perfect, or provide reasons for why you weren't able to make progress. For papers, try to write a rough draft early even if it's very poor. That's okay, writing can always be improved! The process of writing and presenting research often reveals areas where you might need to do some more coding or analysis. It's ideal to find those areas early.
    - There's a lot to balance in a PhD program: research, coursework, TAing, on top of moving to a new city for most students and navigating normal everyday life things. Look into time management strategies such as time blocking and backcasting. Find what works for you to gain some control over your time.
- Be open to all kinds of feedback about your research and try to avoid getting defensive or arguing when others offer advice. You don't necessarily have to take the advice, just consider it in light of what others have said and what you know about the work. Be honest and open about limitations of your work—all research has limitations and simplifying assumptions!
- Document code and record experimental settings and how datasets were filtered.

- Read lots of papers: keep up-to-date with developments in your field. Focus on understanding their contributions, problem statement (what problem are they addressing), novelty (what new perspective or new method are they introducing), etc. Reading paper guide: http://ccr.sigcomm.org/online/files/p83-keshavA.pdf
- Journal your ideas and discuss them with peers - discuss your ideas with PhD students to get different perspectives.
- Prioritize your research goals, including publishing papers.
- Time management - Manage your time well, and allocate time for your research and maintaining a healthy work-life balance.

- Keep asking the question: how can I solve this problem in a better, faster, easier, and/or more intuitive way?
- Be curious about other people's work, even if it is not "your topic/area" (you can learn from listening and even more from asking questions and giving feedback when appropriate)
- Don't forget that about half of the PhD is what you know (or develop, or prove, …) and the other half is what you show (presentation is key, making a good point, showing how different your stuff is from others, showing visually how your work is a notch above others, based on some important metrics)

- Always ask "why?" - don't accept the results from your own experiments or ones in papers you read at face value. Ask yourself what you would expect the results to look like, check whether they match your expectation, and make sure you understand *why* they look the way they do. You should expect to encounter results that don't match your expectations regularly. Sometimes this will help you find simple errors, and sometimes it will lead to interesting new research questions.
- Everything will take longer than you think. That's normal and ok! But you need to give yourself buffers to account for it. Plan to finish things well ahead of the actual deadlines.

- Read more papers
- Get hands dirty: test ideas quickly. Do not wait for a perfect solution before testing
- Talk with more people

- Skim lots of papers and pick the interesting/important ones to read in detail. When reading papers in detail, always act as a reviewer. What problem does the paper try to solve, what methods is the paper using, and what's the reason for accepting and rejecting the paper?
- In your research meeting with your advisor, think ahead and be prepared. Prepare to present your results, answer questions that your advisor might ask you, and propose the next steps.
- Keep experiment logs.
- Start writing papers early.
- Peer is important! Talk to your peers, and help each other out. Research can be a lonely process and it's the peers in your group who made the process much easier.
- Getting paper(s) rejected is normal; do not get discouraged, and your work does not define you.

- Prioritize research. Research is your key duty and also how the outside world will evaluate you when you leave here. Learning how to de-prioritize other things without annoying other people or hurting yourself is a skill.
- Quality over quantity. People will remember you for one great work, not a list of 20 papers. It is risky, but that's an inherent aspect of a PhD. Plus, we are all here to help you.

- Take big bets, but also take bets prudently. Make sure you have done your homework to secure yourself a safety net if you lose the bet. Always, always, have a plan B.
- Be a nice person. Academia is full of smart people, but being a nice person could help you go a long way.
- Help your peers, cheer for them, learn from them. They will be your greatest asset, for many years to come.

- Take a course at CMU as soon as possible.

- Stay connected to your (temporary) advisor, work closely with them to plan your coursework for the next two years, including summers and pay attention to the deadlines

- Verify the rule that your advisor told you with Keena

- Build a cohort of fellow doc students early. Lean on them for peer-mentorship, but also friendship. Otherwise, the PhD can be lonely journey.

- It is a marathon and not a sprint, ensure that you are pacing yourself accordingly.

- Focus on building foundational knowledge for your long-term career. Abraham Lincoln: "Give me six hours to chop down a tree and I will spend the first four sharpening the axe."

- Follow your heart and focus on things you feel most interesting and important

- Explore outside of your area. Your area now may not be your area eventually, and ideas from outside of your immediate area of research may influence your approach in unexpected ways. Go to talks, talk to other students, and keep an open mind.

- Importance of writing - the actual results are necessary but not sufficient

- Be prepared. Three examples:
    - Meetings: plan ahead (think of it as a presentation you need to make in a limited amount of time) AND make sure to have a way to take notes
    - Courses: follow along with readings, assignments, and projects – do not allow for a backlog
    - Deadlines: plan ahead and not react at the last minute

Possible discussion topics for student peer mentoring

1. How do you come up with research problems to tackle?
2. How do you stay on top of the relevant literature?

3. In how much detail should I read papers?
4. How should I find papers to read?
5. How should I set up my experiments?

When you're sick

Please stay home if you're coughing, have a fever, or just aren't feeling well. Viruses spread easily and it's greatly suboptimal to infect the whole lab because then no one can work. Rest so you can heal faster, and let your advisor (also your faculty supervisor and fellow TAs, if you're a TA) know!; a few days of not working won't kill your research, and working when you're sick is unpleasant, less productive, and can delay your return to full health. If you're feeling fine but still have some symptoms and want to meet with your advisor, ask for an online meeting.

Document authors

The document was initially drafted by Adriana Kovashka. It contains the contributions of Daniel Mosse, bill Garrison, Longfei Shangguan, Michael Miller Yoder, Stephen Lee, Amy Babay, Xiaowei Jia, Lorraine Li, Ryan Shi, Kirk Pruhs, Panos Chrysanthis, Jake Biehl, Aakash Gautam, Junyu Liu, Adam Lee, Diane Litman, and Alex Labrinidis.

July 2024