Advisor Guide: Answers to Common Questions

ThanhVu Nguyen
Roars Lab · George Mason University

*This document, adapted from these questions, aims to help current and future students in my lab understand my advising style, expectations, and practices. If you're interested in working with me, you can read this guide to see if my lab would be a **good fit**, which is crucial for a successful PhD advising. More general PhD admission advice can be found in the PhD Admission Demystify book.*

Contents

1. Advisor Style & Expectations 4
   1.1. Does the professor have tenure yet? [engagement level, PhD might get interrupted] 4
   1.2. What is the professor's formal training / background / PhD? [helps contextualize problems/approaches] 4
   1.3. What have previous lab members done after getting their PhD? [Gone to industry?/Post-doc?/Professor?] 4
2. Group Composition & Structure 4
   2.1. Do you work with undergraduate students? 4
   2.2. How many students are in the group? 5
   2.3. What is the lab structure? 5
   2.4. Do students mostly work with senior students or directly with professor? 5
3. Advising Style & Meetings 5
   3.1. Does the advisor consider themselves a "hands-on" or "hands-off" advisor? 5
   3.2. How does the advisor give feedback on papers? 6
   3.3. Are there lab meetings? What are other meetings you will see your advisor in a group with other people? 6
   3.4. What does a group/lab meeting look like? [Or other relevant meetings] 6
   3.5. How often does the advisor meet with their students? [1:1 or all together? Daily guidance by PI or post-doc?] 7
   3.6. How often are students expected to be contactable by their advisor? 7
4. Research Fit & Projects 7
   4.1. How directly applicable will your future technical skills be to the roles you want after graduating? [If set on industry] 7
   4.2. What research methods does the lab use? (What types of papers / contributions / conferences targeted) 7
   4.3. What are some of the projects that you and your students are currently working on? 8
   4.4. In general, do you tend to give your students projects or have them select their own? 8
   4.5. Do you have particular projects that you see me working on? 8
   4.6. How much freedom do you think I'd have in selecting my own projects? 8
   4.7. Are there other students you are interested in working with? If so, what would they be working on project-wise? 9
   4.8. Would they have their own line of work or contribute to a bigger project/someone else's project? 9
5. Expectations & Progress 9

5.1. What progress does the advisor generally expect from a student in the course of a semester? [Submission/Publication pace] ... 9
5.2. What other expectations does the advisor have for their students: time, vacation, paper, project, experiment wise etc.? ... 9
5.3. When have you given a letter of concern? Why? ... 10
6. Funding & Obligations ... 10
6.1. Where does their funding primarily come from? [If military/industry-focused funding bothers you, figure this out] ... 10
6.2. What are their constraints from their funding source? [Some restrict research topic and change final deliverables. Some add work — writing progress reports, traveling, preparing presentations for the funding source or engineering overhead for integration] ... 10
6.3. If your advisor made you work on a project in their area that you are least interested in (e.g., for a grant) would you still be excited doing work? [Useful for choosing between advisors] ... 10
6.4. If you run out of your primary funding for a student how do you expect the student to handle that? [advisor's responsibility / you'll have to write a grant with me / dept will cover the student / you have to find their own funding] ... 10
7. Program Requirements ... 11
7.1. What does the quals process look like? ... 11
7.2. Is there a TA requirement? / How often would I be expected to TA? ... 11
8. Recruitment & Fit / Placement ... 11
8.1. Are you taking a student? Do you have funding to take students in this year (or, for which projects)? ... 11
8.2. What factors will affect whether or not you take a student? ... 11
8.3. Do you think our research interests are a good match? ... 12
8.4. I'm interested in working with you. Do you think I'd have a good chance of working with you if I come to your university? ... 12
9. Co-Advising & External Collaborations ... 12
9.1. Would the advisor be interested in co-advising? ... 12
9.2. How do you handle it when a student outgrows your expertise — they know more than you? ... 12
10. Lab Resources & Working Environment ... 13
10.1. What equipment are provided to students? ... 13
10.2. Do students work together in common space? Do students often get meals together? ... 13
10.3. Do students often work late? [Often / only before conference deadlines.] ... 13
10.4. Are there snacks in the lab? ... 13
11. Conferences, Travel & Internships ... 13
11.1. How often do grad students get to attend conferences? [Pace + What constraints] ... 13
11.2. How many conferences are students expected to target a year? [remember pubs != submissions] ... 14
11.3. Is the professor in an 80/20 with a company? Are students' research projects tied to industry funding? ... 14
11.4. Will it be acceptable/encouraged to intern at a company during the summer? [Does this change with seniority?] ... 14
12. Culture, Time Off & Social Life ... 14
12.1. How often do students take time off? Are there lab/department outings/events? ... 14
13. Support, Conflicts & Student Well-Being ... 15
13.1. If one of your students is being treated unfairly — by a collaborator, committee member, or even within the department — how do you handle it? ... 15

13.2. How do you deal with students who are struggling? Do you "let go" of students who are not making progress? 15
14. Additional Questions to Understand the Advisor 15
14.1. What do you think about course work and grades when evaluating prospective PhD students? If these are not important then what do you look for instead? 15
14.2. What do you think about safeguarding ideas and results so that they don't get scooped? .16
14.3. If someone contributes little to a paper, do they deserve co-authorship? 16
14.4. Do we submit papers to see what the community thinks and get feedback? 17
14.5. How do you approach rebuttals? Strategy? [Do you respond to every reviewer comment, even unreasonable ones? How do you handle clearly incorrect reviews?] 17
14.6. How should a student get sufficient background for their project? Do you expect them to take courses, read papers, or something else? 18
14.7. What is your stand on AI usage in research? [e.g., ChatGPT, Claude, etc.] 19
15. Related Resources 19

1. Advisor Style & Expectations

1.1. Does the professor have tenure yet? [engagement level, PhD might get interrupted]

Yes, I am tenured. If you are wondering if I will be around for a while, the answer is yes. Northern Virginia is a great place to live and work, and CS@GMU is growing fast and getting better every year.

More importantly, having tenure allows me to take risks, pick problems I care about, and ignore the *publish-or-perish* pressure that many non-tenured faculty face. That also means my students have to be independent and take initiative in their work because I won't be pushing them to do things (actually, regardless of tenure, I have always been a hands-off advisor and expect my students to be independent).

1.2. What is the professor's formal training / background / PhD? [helps contextualize problems/approaches]

I did my undergraduate and Master's in CS at Penn State (in fact, all of my cousins went to Penn State and I have 15+ Nittany Lions in my family). I did my PhD in CS at the University of New Mexico-Albuquerque (with Stephanie Forrest and the late Deepak Kapur as my advisors) and then a postdoc at the University of Maryland (PLUM lab with Jeff Foster). I started wanting to work on evolutionary computing, and then gradually changed to software engineering and formal methods (automated reasoning). Outside of academia I have worked at the Naval Research Lab and Lockheed Martin.

Outside of work, I have a family with three young kids. My wife and I run the Chess club at my kids' elementary school (and the club consistently ranks as the top team in the county!).

You can see my CV and other details from Roars Lab page and my homepage.

1.3. What have previous lab members done after getting their PhD? [Gone to industry?/Post-doc?/Professor?]

So far I have graduated two PhDs: Guolong (2022, now a faculty at Minjiang University in China) and Didier (2026). Hai is also on track to graduate in 2027. I also graduated an MS student, Alex, who now works at Oracle, and an undergrad, KimHao, who works at Jump Trading.

2. Group Composition & Structure

2.1. Do you work with undergraduate students?

Yes, I have worked with many undergraduates over the years. Typically at any time I have 2–3 undergraduates working in the lab.

My undergrads are often supported either through research grants (e.g., NSF REU and university funding) or hourly pay. I **treat my undergrads as PhD students** by giving them flexibility to find solutions but also pushing them (*way*) beyond their comfort zone to realize their potential.

As examples, Linhan started as an undergrad at UNL and continued as a PhD student at GMU, KimHao published *9 papers at top conferences and journals* as an undergrad at UNL, Stefania won the Outstanding Undergraduate Research Award at GMU, and Azan---a first year freshman when joining our lab---built CSPicks within just a week of joining.

See the Roars People section for current undergrad members.

2.2. How many students are in the group?

Check out the Roars People section. Typically we have around 4–5 PhD students and a couple of undergrads. We have yet to have a postdoc, but my students collaborate with postdocs at other institutions, e.g., Hai works with Thanh Le at NICT (Japan) and Guolong with Chanh Le. We call our collaborators *"honorary"* lab members and they are very much part of our lab family.

2.3. What is the lab structure?

Each student—regardless of undergrad or PhD—has projects that they lead. However, everyone is encouraged to collaborate and help each other. In weekly lab meetings, you will hear about the progress of other projects and can contribute ideas. We also have a lab server on Discord where we chat about research and other random things.

As examples, Didier worked on Complexity analysis. KimHao worked analyzing build systems, and also complexity analysis and invariant generation. Hai works on neural network verification. Linhan works on neural network testing. Hai, Linhan, and Nguyen often talk to each other as their projects involve neural networks. Long does not work with verification but contributes ideas and expertise as he knows ML systems well. And everyone contributes to fun web apps (e.g., CSPicks).

2.4. Do students mostly work with senior students or directly with professor?

They work directly with me, but I encourage new lab members to work with senior students with similar interests. Our lab members are very close and collaborative, and they challenge each other's ideas and help each other with problems.

3. Advising Style & Meetings

3.1. Does the advisor consider themselves a "hands-on" or "hands-off" advisor?

I am mostly a ***hands-off*** advisor, especially when it comes to technical research. Students who need a lot of guidance and hand-holding *will not* enjoy working with me. I will suggest directions and ideas, but I expect students to work independently and take ownership of their projects. My students —including undergrads—need to be self-motivated and take initiative.

For new students, I provide help with project directions and writing. As students become more senior and mature, I would gradually transition to a hands-off advisor. However, if I see new members capable of working independently from day one, I will let them do so from day one---I don't get in the way of capable students.

Here's a real example from Discord. This was my response to a technical question from a new *undergrad* who impressed me by building a solid prototype in their first few weeks:

> *"I am not sure what would be best. Do what you think is best and that would be best."*

However, I do step in when it's really needed. If a student is truly stuck, facing an urgent problem, or approaching a deadline, I instantly switch into a very hands-on (or protector) mode. This only happens a small fraction of the time, but in those moments I will work closely with the student to get through the challenge.

Also, as mentioned in Section 2, I *highly* encourage my students to talk to other lab members for help and guidance. More experienced students can help new ones with ideas and research guidance.

3.2. How does the advisor give feedback on papers?

For the students' first papers, I ask for drafts and revise them iteratively with the students (e.g., through Overleaf). In some cases, I rewrite large portions of early drafts, especially the Intro, Motivating Example, and Evaluation sections. This helps the students see how papers are written and also learn how I expect papers to be written.

Same thing with paper rebuttals (Section 14.5), I will work with new students and help them revise the writing directly.

As the students have more experience, I will let go more and more. By the time the students can write and successfully publish the paper by themselves without much revision and editing from me, then I know they are ready to graduate. This is typically how I determine **when a student is ready to graduate**: not by specific time or number of papers, but when they can write and publish top papers independently without much help from me.

3.3. Are there lab meetings? What are other meetings you will see your advisor in a group with other people?

Yes, we meet weekly in the conference room on Thursday afternoon in the CS conference room. Everyone speaks, everyone contributes, no one is silent. Even the introverts in our lab are very active and engaged in our lab meetings. This is because they are confident about their research and their ideas are respected by others. Typically, the students talk about what they have been doing in the past week and what they will do next week. They also talk about issues they are facing and others contribute ideas to help solve the problems.

Sometimes, after status updates, a student will present their work in depth. The meeting usually lasts about 1–2 hours depending on how much we have to discuss. I am usually available after the meeting to chat with students individually if they have more questions.

Occasionally we all come and work together in a large conference room (e.g., when we are close to a paper deadline), and then have lunch together. This Google Photos album shows some of our lab gatherings.

3.4. What does a group/lab meeting look like? [Or other relevant meetings]

We start talking about our status, e.g., each person talks about their work for about 5 mins. This is inspired by the SCRUM method used in the PLUM Lab at UMD.

Then we go in depth in some topic, e.g., a student might talk about some problem they are working on, show their computation on the board, or present results. Sometimes we read papers or look at some existing tools/techniques. We plan to use about an hour for this but usually it goes beyond that. I usually have to go pick up my kids at that time, but the students want to continue and they keep going.

Our lab is highly collaborative and social among lab members. They often help each other with problems, which I highly encourage. There are several self-identified introverts in the lab, and while they do not speak much outside the lab, they are very active and engaged in our lab meetings. This is in part because they are confident about their research and their ideas are respected by others.

Sometimes we just relax and watch a movie (e.g., the PhD Movie).

Also, most of our communications happen on Discord server outside the meetings where we chat about research and other random things (e.g., plan for Thanksgiving party).

3.5. How often does the advisor meet with their students? [1:1 or all together? Daily guidance by PI or post-doc?]

In addition to weekly lab meetings described above, students can talk to me individually about any thing they want (usually we would talk right after regular lab meetings and quite a bit on Discord). And during paper deadlines, we sometimes meet several times a day on Zoom.

Many of my students work late at night and so do I (after kids have gone to bed!). So I've been making myself available to work with my students at night whenever they need, often 10PM--1 AM (much later when I was younger).

That said, I highly value independence, and so as long as it works for you and you're productive, it *doesn't matter* to me how/when you work.

3.6. How often are students expected to be contactable by their advisor?

Most of our communications happen over Discord. I sometimes contact students through email (e.g., forwarding messages from the university). For such occasions, I expect you to respond to my emails quickly. However, in general, communications through Discord are more casual and you can respond when you see them — though I find that all my students are quite responsive on Discord.

4. Research Fit & Projects

4.1. How directly applicable will your future technical skills be to the roles you want after graduating? [If set on industry]

During PhD we train you to be researchers, and so if your post-graduate career is not related to research, then you are not directly using the skills you learn in the PhD.

Still, many other skills that you learn—being independent, problem solving, read/write technical papers, endure my advising style, etc—are applicable to many careers. More specifically, skills you learn in my group (formal reasoning, verification, program analysis, being independent, writing well, etc) are real skills that are applicable to both academia, industry, and everywhere else. We go for hard problems and they often coincide with real problems in the industry. We didn't set out to solve "industry problems" per se, but we often end up solving them anyway. For example, KimHao's work on build systems is directly applicable to large-scale software development at Facebook. Hai's work on DNN verification is applicable to many systems that use neural networks, which we believe will soon be everywhere (e.g., self-driving cars).

4.2. What research methods does the lab use? (What types of papers / contributions / conferences targeted)

Our work is often on developing new techniques/algorithms and building tools. We target top-tier conferences in the field (e.g., ICSE, FSE, ASE, ISSTA, OOPSLA, PLDI, CAV) and more recently also in AI/ML conferences (e.g., NeurIPS, ICML, ICLR). For work we wish to extend, we also publish in journals such as IEEE TSE.

Ok, that was a boring neutral "academic" answer. The fact is that we don't just publish papers or produce boring incremental results, we focus on developing tools that actually work, win competitions, and receive awards. Our competitors are the best in the world and we enjoy **beating them**. For

example, NeuralSAT is often considered one of the best at neural network verification competition and outperforms many other state-of-the-art tools.

We also explore new areas and venues if they are relevant to our research. For example, since we work on neural networks, we also publish in AI and ML conferences (e.g., NeurIPS, ICML, ICLR). We have been quite successful in those venues as well (e.g., Highlight and Spotlight papers at NeurIPS'25, CVPR'26, ICML'26).

4.3. What are some of the projects that you and your students are currently working on?

We are working on various software analysis projects, including formalizing and proving mathematical theorems, neural network verification and inference, analysis on highly-configurable software, and program testing, analysis, and repair.

Ok, cross out ~~program repair~~ — I moved on from that. Nowadays I am more interested in applying formal methods and program analysis techniques to verifying and reasoning about neural networks and also formalizing and verifying software and maths using theorem provers (e.g., Lean 4).

4.4. In general, do you tend to give your students projects or have them select their own?

I give them research topics or directions to explore. For new students, usually, I give them more specific projects to start with so they can get familiar with the research area and techniques. As they are more familiar, they will be able to come up with their own projects. For example, Linhan bounced around several ideas before settling on his work on DNN benchmarking, and Hai and Long work with several students (some not even in our lab) on various projects.

I also aim to determine what my students are good at, and help them find projects that fit them. This is not just for research, but also for other things. E.g., in our Thanksgiving party, which I host at my house every year for my lab and their families, I delegate tasks to students based on their strengths and interests: I found out that Linhan is very good at cooking, so I appointed him as our turkey "czar" responsible for a 20+ lb turkey roasting for 30 people!

4.5. Do you have particular projects that you see me working on?

It depends on your background and interests, but I do have many ideas to try. In recent years, I have been focusing more on DNN reasoning and verification, so something related to that.

4.6. How much freedom do you think I'd have in selecting my own projects?

We need to find projects or research directions that we are both interested in. Once we have agreed on a research direction you will have a lot of freedom to explore and develop your own projects.

However, if you want to work on some areas that I am not interested in, then I would rather you find another faculty because I can't guide a topic I don't know or care about.

That said, I am open to new ideas and directions *if* you can convince me you can make them work. The best way to do that is as follows: if you come up with an idea that is quite different than what you're working on, **do not** tell me about it, especially if you don't have good things to back it up, because I will likely shoot it down. Instead, just pursue it (while still working on your main project), and build *a prototype* and achieve some *good results*. If you can show me your ideas work with concrete results, I will much more open to them. This happens multiple times in my lab, and probably some of my students are currently working on secret ideas that I don't know about yet!

4.7. Are there other students you are interested in working with? If so, what would they be working on project-wise?

Not sure what "other students" means. But if the question is whether you have to compete with other students in the lab, then the answer is NO. As mentioned in Section 4.4, in the beginning you will likely work on projects with other students, but eventually you will have your own projects. Moreover, ROARS members are very collaborative and they help each other with ideas and problems.

Now if the question is if I'd spend time working equally with all my students, then the answer is also NO. I will spend more time with students that get me excited and involved in their work by showing me progress and results. I made this explicitly to my students: get my attention by showing me results and progress! Don't just talk about ideas (or change them frequently), but show me results!

4.8. Would they have their own line of work or contribute to a bigger project/ someone else's project?

All of my students have their own line of work. They also collaborate with other lab members and in the beginning, they likely work on projects with other students. But eventually, they will **own** their projects.

5. Expectations & Progress

5.1. What progress does the advisor generally expect from a student in the course of a semester? [Submission/Publication pace]

I expect the student to have enough results that they can **submit** a paper to one of the top conferences each year. It does not have to be accepted because the reviewing could be noisy, but it is crucial to complete the work and submit it.

You can see how ROARs' students publish in the ROARs' Publications section.

5.2. What other expectations does the advisor have for their students: time, vacation, paper, project, experiment wise etc.?

- I want you to *be independent* and *take initiative* in your work. Don't just wait for me to tell you what to do. If you have an idea, try it out yourself and show me the results.
- You will have *freedom to come up with solutions and explore ideas in your own way*. Your projects need to be related to my interest, but beyond that, you can do things your way.
- I want you to *ask questions* and *be open to learning new things from others* (e.g., your lab mates, who are probably better resources to learn from than me).
- I want you to listen to me, but also *be independent* and *challenge me*. I am not always right but for you to challenge me, you must be able to convince me, through concrete evidence and results, that your idea is better than mine.

As an example, if I tell you to try X, but you think Y is better, then you should do both X and Y and show me both results. It doesn't matter if X or Y is better, but that you believe in your idea and can get results to support it.

5.3. When have you given a letter of concern? Why?

I haven't given a letter of concern yet, and I hope I never have to. But I would remind the student and give plenty of warnings if they need to get on track or are not meeting expectations. There is no surprise in my lab and you will know if you're not doing well.

Here is the expected progress for my PhD student. By the end of the second year, you should publish at least a second-author paper. You are doing well if you publish a first-author paper at that time. By the end of the third year, you are considered behind if you do not publish a first-author paper and are seriously behind if no good submissions were made. A typical student in SE/PL will have 4–5 publications, students in ROARS lab often have a lot more than that.

6. Funding & Obligations

6.1. Where does their funding primarily come from? [If military/industry-focused funding bothers you, figure this out]

Most of my funding sources are from NSF and some are from the industry (e.g., gifts from Amazon and Facebook). All of my funded projects are on basic/fundamental scientific research.

If the question is whether I would get funding from DoD, the answer is yes, I do not have any issue with DoD or military funding. However, while I also apply to DoD funding, I try to avoid putting my students—especially new ones—on "contract" projects as these often have less flexibility for students (e.g., require deliverables, reports, and publications are not the focus).

6.2. What are their constraints from their funding source? [Some restrict research topic and change final deliverables. Some add work — writing progress reports, traveling, preparing presentations for the funding source or engineering overhead for integration]

Since my funding sources are mainly from NSF or unrestricted from industry, our work has few constraints. Occasionally, I may need your assistance on progress reports and presentations if you are funded through specific grants.

6.3. If your advisor made you work on a project in their area that you are least interested in (e.g., for a grant) would you still be excited doing work? [Useful for choosing between advisors]

It is unlikely that you have to work on something you're not interested in.

6.4. If you run out of your primary funding for a student how do you expect the student to handle that? [advisor's responsibility / you'll have to write a grant with me / dept will cover the student / you have to find their own funding]

One of my responsibilities is finding funding to support my students, and I have been quite lucky to have sufficient funding to support my students (including summer). However, as a fallback, the department can also cover the student with TA-ship. CS@GMU is *very good* at providing PhD students TA-ships.

I also encourage students to apply for fellowships. I will help them with the applications (e.g., give feedback on your statements and provide LORs). These fellowships are not necessary for funding, but

they are prestigious and can help your career. I take pride in my students' achievements—sometimes more than in my own. When a student wins a fellowship or an award, it's also a win for me and the entire lab!

7. Program Requirements

7.1. What does the quals process look like?

The department has a very specific guideline and timeline that students need to follow for the comprehensive exam. It's quite straightforward and the student needs to write a paper describing the research area and problem they want to work on and present that paper.

I typically won't let my students go for the quals without having a paper submitted. While the exam is mainly a formality, it is also a checkpoint to make sure the student is on track and has a clear research direction (in which a paper submission is a good indicator of that).

7.2. Is there a TA requirement? / How often would I be expected to TA?

There is no TA requirement from the university or the department. Nevertheless, I strongly recommend doing a TA at least once or twice during your PhD. TA is a great way to get introduced to teaching, and can really help if you're interested in an academic career. More practically, in some cases not having to support you as a GRA during the normal Spring/Fall semesters would allow me to support you during the Summer.

8. Recruitment & Fit / Placement

8.1. Are you taking a student? Do you have funding to take students in this year (or, for which projects)?

I am **not actively recruiting**. My lab already has quite a few students. In fact, this advising guide is actually not meant for recruiting, but to help my current and future students understand my advising style and expectations.

That said, I am always looking out for **standout** students. Standing out is difficult to define and subjective—I can only know it when I see it (e.g., see the Reddit example in Section 8.2). So read this guide and if you think you would do well in my lab, then reach out to me.

8.2. What factors will affect whether or not you take a student?

Here's a concrete example: One day the CS faculty were talking about a GMU Reddit post from someone who built an opensource web app to identify **"easy A"** classes at GMU (because they find `RateMyProfessors` useless). Some profs. found it problematic (and also complained that it doesn't represent their own classes fairly). But I was *intrigued*.

I didn't care much about the app itself, but the initiative and technical skill caught my attention: the student finds a problem, comes up with a solution using an unconventional way (from publicly available FOIA data that no one pays attention to), and implements an app that works (and has a very good UI).

That same evening, I talked about the app with my lab and tracked down the developer. I reached out to him that evening, we talked the next day, and I hired him on the spot.

8.3. Do you think our research interests are a good match?

The question should be: are **my** research interests a good match for **you**? As mentioned, I will unlikely work on something that I am not interested in. So if you want to work with me, you need to be interested in my research areas.

8.4. I'm interested in working with you. Do you think I'd have a good chance of working with you if I come to your university?

The common way for me to know you (and vice versa) is taking my classes, standing out, and impressing me. I recruited several students (e.g., Linhan, Guolong, and Didier) from my classes that way.

9. Co-Advising & External Collaborations

9.1. Would the advisor be interested in co-advising?

I was co-advised during my PhD and it worked out, but I **prefer to be your main or sole advisor**, and other faculty members can be collaborators (but not co-advisor). This way, you can get more attention from me while still being mentored by other experts. For example, Hai is advised by me but also collaborates (and therefore mentored) by Matt Dwyer at UVA through our NeuralSAT project. It's also easier for me because I do not want to deal with conflicts between co-advisors, e.g., if they want you to do something that I don't want you to do (and vice versa).

In addition, I am **not** into broad **interdisciplinary research**, which would be beneficial for co-advising. I prefer to work on problems in my fields of SE/PL/FM and related ones (e.g., logics, mathematical proofs). So if you want to do interdisciplinary research in areas outside my core expertise (e.g., CS + Bio), then I may not be the best fit for you. While interdisciplinary work has many values (and I admire people who do it well), I have determined that it is not my strength and I prefer to focus on my core areas.

If you'd like to do interdisciplinary work that I am not too interested in, then I would not want to be your main advisor. I would rather you find another faculty member who is more interested in your work and can help you with it (e.g., the CSPicks tool can help explore faculty interests). I think this is a better fit for both of us.

9.2. How do you handle it when a student outgrows your expertise — they know more than you?

I **expect** this to happen. At some point, you will know more than me in their topic of interest. Depending on the projects, I may introduce you to people who are experts in that area and we can collaborate with them. For example, KimHao works with Quoc-Sang at Facebook on build systems, and Hai works with Matt Dwyer at UVA on DNN verification. In short, if you become better than me in your area, then I've done my job and you're ready to graduate.

Note that if you want to change or try another research direction that I don't have much expertise or interest in, then I can't help you much there, so we may need to discuss switching advisors. This is common in academia (almost happened to me during my PhD) and I have no issue with you doing that (in fact, some of my students came to me after switching from other advisors).

10. Lab Resources & Working Environment

10.1. What equipment are provided to students?

I purchase computers and electronics for my students (e.g., computers, laptops, monitors, hard drives, memory, headsets, keyboards, etc). Our lab has several powerful servers such as Nvidia Spark DGX (a gift from NVidia), that are shared among the students. Let me put it this way: our lab machines are much better than most other labs and our experiments get done much faster than others.

10.2. Do students work together in common space? Do students often get meals together?

Sort of. Due to office space limitations, my students are sharing space with other SWE PhD students. There are advantages of that as you get to interact with more people outside our lab.

10.3. Do students often work late? [Often / only before conference deadlines.]

We work late mostly before the conference deadline. I also stay up quite late to chat with students if they need. Though it really does not matter to me when you work as long as you are productive.

Work-life balance is a strange thing in CS research. During paper deadlines, we work very hard and long hours—and I do expect my students who are involved in the paper to do so as well. But outside of deadlines, we have a lot of flexibility, e.g., feel lazy and want to take a break for several days? No problem. I don't care and you don't even have to tell me (in fact, you should not tell me!). In short, creativity comes from freedom and flexibility, and in this regard, I give my students a lot of freedom.

10.4. Are there snacks in the lab?

Not sure, sometimes I see some snacks. When I was a grad student, I often stole pizza and food from other labs (I have plenty of stories to tell). I often take my groups out to eat (e.g., when working on papers, or end-of-semester gatherings; in fact, I am writing this while preparing for traditional lab Thanksgiving dinner).

More generally, we work hard but also have lots of fun, many of which involve eating and gathering.

11. Conferences, Travel & Internships

11.1. How often do grad students get to attend conferences? [Pace + What constraints]

If you have a paper at a conference, even not as first author, I will support your travel to the conference to present it. I also encourage you and other lab members to go to local conferences or workshops (e.g., PL meetings at schools in the Northeast region such as Stevens, Princeton, UPenn). I often send my students to the annual, week-long Formal Techniques Summer School in San Francisco (CA).

For full disclosure, I am not a big fan of attending conferences. I think they are useful but I am just not into traveling. I would rather stay at home and play with my kids and work on my research. But this is just me and I wholeheartedly support my students to attend conferences.

11.2. How many conferences are students expected to target a year? [remember pubs != submissions]

On the second year, I expect my students to *submit one* first-author paper to a *major* conference each year (e.g., ICSE, FSE, ASE, ISSTA, PLDI, OOPSLA, CAV). On the third year onwards, I expect you to *submit* at least *two* papers each year.

Essentially, I expect you to get some results to present after a full year of research, and continue and improve that pace throughout your PhD.

My students often publish more than what was mentioned above, e.g., KimHao published 9 papers as an undergrad, Guolong published 8 papers during his PhD, and Hai already has 7 papers by his 4th year, etc.

I strongly believe that students do not know what they are capable of until they push themselves, and they should not remain in their comfort zone (of course, not the level of craziness of David Goggins).

11.3. Is the professor in an 80/20 with a company? Are students' research projects tied to industry funding?

No, but I collaborate with researchers from the industry (e.g., Facebook and Amazon) and have industrial grants/awards (e.g., Amazon). Some students' projects are directly applicable to industrial problems, and they get opportunities to do internships at cool companies. However, we do not have any contractual relationship with any company, and so there are no obligations or constraints from the industry on our research.

11.4. Will it be acceptable/encouraged to intern at a company during the summer? [Does this change with seniority?]

Definitely, and in fact I encourage you to do that. E.g., Hai interns at Amazon AWS and KimHao at Facebook (twice) during the summer. Not only this save me money, it also provides you practical experiences that are quite important to find jobs when you graduate.

Note that, as PhD students, you should do research internships (e.g., at research labs in the industry) rather than engineering internships. Best if you can do something that is related to your research and can lead to publications and things you can use to support your dissertation. But if you want to do an engineering internship, I will support you as well.

12. Culture, Time Off & Social Life

12.1. How often do students take time off? Are there lab/department outings/events?

Time off: Academia is flexible and I am flexible with my students (as long as they get their work done). However I expect my students to be in the lab at least 3 days a week (part of school requirement otherwise they will lose their space, which I think is fair). But if they need to take time off (e.g., health appointments, personal/family events), they can always tell me and do so.

Lab dinner/gathering: Usually twice a semester (e.g., group gathering at the beginning and the end of a semester and during Thanksgiving). Once a year we also hang out with other labs at Round One (bowling and arcade).

Department outings/events: There are various events throughout the semester in both the SWE group and the department. I encourage students to attend these to have fun and know other people.

13. Support, Conflicts & Student Well-Being

13.1. If one of your students is being treated unfairly — by a collaborator, committee member, or even within the department — how do you handle it?

I am protective of my students. I can be hard on them when they do research with me, but I will stand up for them. Not only obvious cases like unfair treatment but also when I feel they need support, e.g., I would shield them from being overworked or being taken advantage of by others. My priority is the well-being of my students.

13.2. How do you deal with students who are struggling? Do you "let go" of students who are not making progress?

I have never let go of a PhD student yet, and I hope I never have to. Once I take in a PhD student, I see it as my responsibility to get them through the program and into a good career (job that they like).

Every PhD student will struggle at some point, but I believe my students are capable of pushing through challenges, and I will work with them to make sure that happens. So no, I don't give up on students easily. It's usually easier for them to give up than it is for me to give up on them.

14. Additional Questions to Understand the Advisor

Questions that are not categorized into the above sections but can help you understand my advising style and philosophy.

14.1. What do you think about course work and grades when evaluating prospective PhD students? If these are not important then what do you look for instead?

In my PhD Admission Demystifying book, I emphasize that for CS PhD admission, a good GPA rarely matters (you can read more about that in the guideline). There are much more reliable indicators of a student's potential for research; LoRs and research experience and publications of course, but even small things that are seemingly unrelated to research can be good indicators, e.g., the easy-A scraper example or someone who climbed the Himalayas! It doesn't have to be crazy thing (though it can be), but should be something that shows the student is a self-starter, has initiative, and can get things done.

The quote below is a snippet from a LoR I wrote for my undergrad student to nominate her for an Outstanding Undergrad Researcher award. At that moment, she had not done much research, but her volunteer activities showed self-motivation and initiative, which impressed me so much that I hired her to be my undergrad researcher and then later nominated her for the award (which she did get---Outstanding Undergraduate Researcher Award in 2025).

*"At NOVA, Stefania initiated the Girls Who Code club and volunteered in various activities during COVID-19. These include **tutoring elderly individuals** on using Zoom, ordering groceries online, and collecting old computers to refurbish and **donate to underprivileged families**.*

These efforts were not connected to any specific class or organization; she just did them with her friends out of a desire to help. It is worth noting that such activities are often mentioned in NSF CAREER proposals; the main difference is that most only talk about doing them while she actually did them."

Figure 1: Stefania's LoR snippet.

I would go further and say if even at the PhD admission level GPA is already not that important then during the PhD program it is *even less important*. I rarely ask about course work and grades from my PhD students. In fact, if they are not doing well in research but doing very well in course work, then I would question that as that can be a sign of them struggling in research and using course work as a way of coping or just trying to get by. In short, *if you're my PhD student and you get 4.0 GPA in your classes, then I might question why you spend so much time on coursework instead of research.* Just do enough to get by course work and focus on research.

14.2. What do you think about safeguarding ideas and results so that they don't get scooped?

We have many ideas, so if someone wants to steal some of them, it's not a big deal, and in fact, we should be **amused, flattered, and consider it a compliment!**

Ideas are meant to be executed. If we fail to execute the ideas, then we are the one to be blamed, not the ones who take the ideas and execute them. If the ideas are ours then we want to quickly realize them, publish them, and get the credit for them. If we don't/can't get something to work, we might as well give them to others. And they can acknowledge us if they want – or not (doesn't matter to me).

If people—even our competitors in the same research area—are interested or curious about our work, then give it to them; no need to hide the "secrets" sauce or anything important. Give people details, point them to our code base, e.g., project repos, which are *always* open-source and publicly available, to see all the cool things we're doing. And if they want some code or benchmarks etc, help them out and encourage them to use our work and even build on top of it.

Of course, this might clash with collaborators, funding sources, or projects that have restrictions on sharing. In those cases, we will re-examine, but I will not be happy about it. So far our projects, even those with industry collaborators, are all open-source and publicly available.

14.3. If someone contributes little to a paper, do they deserve co-authorship?

"Little" is subjective! We don't seek co-authors to help with our work, but if it happens that someone can help make the work better, we will share the credits. Even if we did 90% of the work, but can't do the other 10%, and someone else can help with that 10%, then we should offer them co-authorship (the person who did most of the work is still the first author). The point is that if we can't ship the work without that 10%, then that 10% is highly difficult and important and therefore deserves co-authorship. Thus, it's not about giving co-authorship to someone who contributed a "little", but rather about giving co-authorship to someone who contributed something that we can't do ourselves that is necessary to ship the work.

In addition, as co-author, the person can claim the work is theirs and do what they want with it subsequently. They don't need to include or involve us. We should not pay attention to that. We

just want to do good work and get it out. In fact, if they use it and talk a lot about it, that's great—free advertisement for us. And if they completely "steal" it, we should be flattered and consider it a compliment (Section 14.2)!

14.4. Do we submit papers to see what the community thinks and get feedback?

No, we don't send out "feelers" to see if the community likes our work. We should not use the volunteering reviewers as guinea pigs to test our work (besides, the comments from a couple of reviewers do not represent the community). Moreover, we don't need to send out papers just to get feedback, if we do enough homework we should have a good idea of how the community would react to our work. In fact, if our work is solid, we might be able to change the community's mind and get them to like it. So just do the work, get good results, then write and submit a good paper about it.

Note that this *does not* mean we should always aim for full papers to top conferences—which are the main venue for computer science. Our lab submits to a variety of venues, e.g., journals as extensions of our conference papers, short papers to specialized workshops or to introduce early ideas (e.g., New Ideas and Emerging Results track at ICSE), invited papers, etc. We also submit to places that are not our usual venues if we think they are a better fit for the work, e.g., for our work on verifying computer vision models, we submitted to CVPR instead of our usual SE/PL/FM venues. So we are open to exploring different venues and formats; we just don't submit papers just to get feedback or to see if the community likes it.

14.5. How do you approach rebuttals? Strategy? [Do you respond to every reviewer comment, even unreasonable ones? How do you handle clearly incorrect reviews?]

If the reviews are very negative (e.g., not even a borderline), we should just **withdraw the paper**. This saves both ours and the reviewers' time. However, even if we withdraw, we should still **thank the reviewers for their time and feedback** and say that we will withdraw the paper and try to address the concerns in the future. I am adamant about this---whether we find the feedback useful or completely garbage, we will acknowledge and thank the reviewers for their time and feedback.

In most cases, however, our papers are not so bad that everyone hates them (i.e., we don't submit papers when they are not ready as mentioned in Section 14.4), so we often get to do a rebuttal. In those cases, the students, typically the main author(s), will draft the rebuttal (usually on shared Google Docs), and I will give feedback and revise the writing. There are many common and good practices (e.g., don't over-promise, don't be defensive, etc). But the most important thing to me is to treat rebuttal as an *opportunity to improve the paper*, and not to worry about the outcome or trying to please the reviewers.

- **Always acknowledge the reviewers' time and feedback**, even if you think they are unreasonable or won't care (most likely they don't), but we need to do it because it's the right thing to do---even if we really hate their questions and believe nothing will change their mind.
- Don't do the rebuttal when we are emotionally upset (e.g., after reading the reviews). Take a break, calm down, and then come back to it. **The comments---no matter how bad they initially seem---typically have some truths in them** (e.g., if they ask a really "dumb" basic question, this could be because we assume too much and don't explain things well enough; if they don't like the problem, then may be because we didn't motivate it well enough or we submitted to the wrong venue, etc).

- **Don't try to find the champion and the defectors**. This might be counterintuitive as it is often suggested to identify the reviewers who clearly like the paper and provide them with the ammunition to fight against the defectors. However, I don't follow this strategy because, as mentioned, even the most negative comments have some truths in them and therefore can help improve the paper.

  Moreover, while a reviewer might appear to be strongly for or against the paper, they might actually be the opposite (e.g., a champion might not actually do anything to help the paper while a defector can their mind and advocate for acceptance if we can address their concerns). So just address the comments with the purpose to improve the paper, and don't worry about who is a champion or a defector.

In most cases, especially at top confs., rebuttals do not help and papers *will* get rejected; and in such cases we improve and resubmit and somtimes get best paper award, e.g., our compositional verification work was rejected at CAV'25 and got a Research Spotlight Award at NeurIPS'25. In some rare cases, however, rebuttal can actually change reviewers' mind as shown in Figure 2. So embrace the uncertainties of the review process; *treat rebuttal as an opportunity to improve our writing and work (think of them as "bug reports" that make your code better)*, and don't worry about the rest.

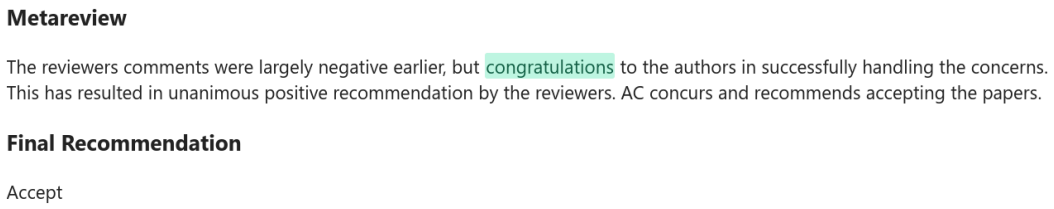


Figure 2: Even with largely negative reviews, we had a very positive rebuttal that changed the reviewers' mind and got the paper accepted at CVPR'26 (updated: and was given a **Highlight** Paper Award).

14.6. How should a student get sufficient background for their project? Do you expect them to take courses, read papers, or something else?

I actually don't encourage my students to take courses or read many papers before they start working on their project. I strongly believe that the student should dig right into (existing) code, make changes, and break something. If it helps, reimplement it from scratch. I'd like you to get frustrated with existing work and think of ways to improve it. Reinventing the wheel is a great way to learn and understand something deeply—and understanding something deeply is the key to doing research on it.

Reading papers won't reveal issues or limitations with the work, and might even demotivate the students (*"too many good work already, how do I come up with something better?"*). Every work has limitations, but if you understand the code very well (e.g., by reimplement it), you will find all the issues and think of ways to improve the work—that's how we come up with ideas and advance the state of the art.

Not only does this apply to engineering or tool building projects, but also to theoretical work—find the flaws in the proofs and theorems (like Paul Wolfskehl), and come up with ways to fix them and develop more elgance ones (and while we are at it, formalize the proof and check it with a proof assistant like Lean).

Once you understand something well, then you will be motivated to read papers in the area to see what others have done and how you can improve on them. You also will read papers in a different way—not with the "wow" or "impostor syndrome" feeling, but with the ability to understand the work

deeply, appreciate the good stuff, and identify and improve the bad stuff—usually hidden in the polished writing.

It's worth noting that I apply the same thing for the kids in my Chess club: they don't learn openings and strategies *when they start* — just play lots of games, go to tournaments (then cry a lot and then win a lot), and those who are really interested in sticking with it and want to improve themselves can study openings and endgames later.

14.7. What is your stand on AI usage in research? [e.g., ChatGPT, Claude, etc.]

I leverage AI *extensively* in my research, and I encourage my students to do so as well! It is a great tool that allows us to focus on the more important and creative parts of research, rather than spending time on tedious and repetitive tasks (e.g., writing code, debugging). I have "vibe-coded" multiple projects, including CSPicks and VietProfs, which would not have been possible without AI tools. Thus, not only do I encourage my students, I also expect them to know and use AI tools to help with their research and productivity.

However, I expect my students to understand the work and be responsible for it (e.g., so don't just use a couple of prompts and done). I *do not* expect them to read AI generated code, I expect them to iteratively interact with AI to understand how things work (the algorithm or technique behind the code). As example, for **CSPicks** or **VietProfs**, while I don't read the AI-generated code, it took me hundreds of iterations with AI to get things designed and built to my liking, and even then I still actively work on improving them and adding new features.

Furthermore, as AI often makes mistakes and we cannot manually check all the code or proofs, we should aim to formalize our implementation and specifications and use a proof assistant like Lean to check the correctness of the code and proofs (nothing is guarantee as Lean implementation itself can be buggy, but at least it is a lot more reliable than humans).

Similarly, while I encourage students to use AI to help with writing, I expect them to actually read and understand the content and *rewrite* to make it "theirs" (and also to avoid the AI generic flowering writing style). For my own writing, after each paragraph drafted with AI, I put a **`sorry`** (a Lean thing indicating something is not yet fully done) to remind me that I still need to work through that paragraph: reread it carefully, edit it, and usually rewrite pretty much the whole thing in my own words. Only after doing that do I remove the `sorry` for that paragraph.

Interesting, I use AI extensively these days, but at the same time try to find time to "retreat" from AI and do some "old-school" research, e.g., reading on printed papers, writing in pure Emacs without any AI assistance, and actually learning new programming languages like Lean by manually writing code and compile them.

15. Related Resources

- Jia-Bin Huang's answers
- IDEAL Lab's answers
- Qian Lou's answers
- Andrew Kuznetsov