Teaching Statement

Nghia Nguyen-Trung
Ph.D. Candidate, Statistics and Operations Research, University of North Carolina at Chapel Hill
nghiant@unc.edu · nghiant01.github.io · *Last updated: November 2026*

Why I teach

The student I remember best was never in one of my classes. In 2023, a first-year student in STOR 113 asked me to tutor him after a first midterm that went badly. He could do the homework, but the applied problems on the exam were another matter, and early on I saw that he still made mistakes in basic arithmetic with fractions. So we started there. Twice a week until the final exam, I gave him problems he could already solve, then problems only one step harder. When we reached derivatives, I did not start with the rules; I asked a smaller question first: is the variable in the base or in the exponent? He finished the course with an A−, and wrote that it "wouldn't have been possible without your help."

I have been doing this since I was fifteen, when I was on my school's physics competition team and taught physics to friends on the chemistry team. I then tutored mathematics, physics, and chemistry for eight years, from 2014 until I came to UNC in 2022. Those years left me with two convictions. Students learn mathematics by getting their hands dirty: working problems themselves, making mistakes, and learning from them, so that they do not repeat those mistakes later. And the next problem should be only one step harder than the last; my job, in a course and in research mentoring alike, is to find where that step is.

In the classroom

I have taught four summer courses as instructor of record (STOR 155 in 2024, STOR 113 in 2025 and 2026) and assisted in seven courses, from STOR 113 to the Ph.D. course STOR 612, including three Python lab sections for STOR 120. In STOR 113, a calculus-based course for business and economics students, I build each class from worked examples that grow one step at a time, then let students try the next one, sometimes at the board. One student wrote that working "practice problems on the chalk board… helped identify common mistakes." I take examples from the students' world: profit, pricing, and the cost of a coffee cup in STOR 113, and medical studies in STOR 155, which a pre-med student singled out in an evaluation.

I also draw a lot, because a picture is easier to rebuild than a rule is to remember. When I introduce the derivative, I open a graphing tool, draw a parabola, fix a point on it, and draw the tangent line; then I add a second point that slides along the curve, with the secant line through both points. As the second point slides toward the first, the secant line swings closer and closer to the tangent, and students watch the limit happen before they see it written in symbols. Similarly, I do not ask students to memorize the first-derivative test: going up and then down means you have just passed a hilltop. Later, I heard students reason exactly this way out loud while working a problem with me.

The steps get bigger as students advance. In STOR 415H, an honors optimization course, a student came to my office hours regularly with all kinds of questions; for one about the simplex method, we drew the feasible region of a two-variable problem and followed each pivot from one corner to the next. In STOR 612, the Ph.D. course I assist in now, we usually discuss the idea behind an argument rather than each line.

Listening and adjusting

After the first midterm in every course I teach, I run a short survey, with a small bonus for completing it, about the lectures, the materials, the exam, and whether office hours fit students' schedules; end-of-term evaluations come too late for the students who needed a change. Some students said that exam questions look different from the homework. Part of that difference is by design: homework practices one topic at a time, while an exam question may combine topics, because at university I want students to reason, not only to match a formula to a section of the book. What I can do is make sure students do not meet that jump for the first time on the exam. Before every exam, they get a sample exam and practice problems in its style; at the end of summer 2025, one student wrote that the practice problems "need to be kept" because they showed "the types of questions that would be featured on the

exam." In that session, the class mean rose from 69 on the first midterm to 76 on the second, and the lowest score rose from 30 to 64.

Teaching STOR 113 twice in one summer also let me change the course itself: for the second session, I cut some derivations, such as proving the logarithm rules, and used the time for applied examples. Before an exam goes out, my TA works through it to check that it is correct and can be finished in the time given, and I ask fellow instructors for comments; when grading, I give credit for correct reasoning and do not penalize the same mistake twice.

My surveys also showed me who my students are: in summer 2025, one worked full time, another worked from 6:30 to 11 a.m., and others had practices during my office hours. So I meet students at other times when they need it; one wrote that "he took his time and explained the concepts to me until i understood even if it took multiple different methods of explanation." I post annotated lecture notes and drop the lowest homework scores. In introductory courses, I do not ban AI as a learning tool, but graded work stays AI-free, and I ask students to use AI to learn how to solve a problem, not to solve it for them. In STOR 612, where the department has started using oral assessment, I ask a randomly chosen student each week to walk me through a homework problem. The department gave me its Excellence in Teaching Assistance and Instruction Award in 2024 and again in 2025; the faculty choose recipients based on student and faculty evaluations. Unless noted otherwise, the student quotations in this statement come from end-of-course evaluations.

Advising and mentoring

Since September 2026, I have been mentoring two undergraduate students as they begin a research project on robust bond portfolio optimization, where the data behind the model are uncertain. They are reviewing the literature, deciding what the objective should be, and collecting data. This is different from teaching a course. In a course, I have an answer key for every question; here, neither of us does, and our meetings are about which questions are worth asking: which uncertainties to model, and how to estimate them from data.

My advising will follow the path my own Ph.D. took. I began on projects that my advisor initiated, where I developed and refined the convergence proofs and carried out all numerical experiments, and I moved to first-author projects in which I led the problem formulation, algorithm design, analysis, and implementation. I would give each Ph.D. student a similar path: a first, well-defined problem close to tools we already have, such as extending my decentralized methods to directed networks, and then growing ownership until the student proposes and leads the questions. Undergraduate and master's students can start with implementation and simulation, where progress is visible early, and move toward formulation as their confidence grows. I plan to meet each student regularly, to have students write up arguments and present their work early, including at meetings such as INFORMS and SIAM conferences, and to connect them with collaborators outside the group, as my work with Ion Necoara has done for me. Mentoring also means being honest about what I do not know. When the mother of the student I tutored later asked whether I could tutor another course, I declined because it was outside what I know; with research students, I would say the same and point them to someone who knows more.

Teaching interests, and what I am still learning

I am ready to teach introductory statistics, calculus and linear algebra, business calculus and decision models, data science with Python, and undergraduate optimization (linear, nonlinear, and convex, and optimization for machine learning). At the graduate level, I can teach courses in linear, nonlinear, and convex optimization, and I would like to build a course on first-order methods for large-scale optimization, the area of my dissertation, running from gradient and proximal methods to monotone operators and splitting, acceleration, variance reduction, coordinate methods, and distributed algorithms.

The part of teaching I am still learning is the unexpected: a question from a direction I did not prepare for, or a class that does not go as planned. Here experience matters more than preparation, so I watch how senior colleagues handle such moments and ask them about it. I also want to work with colleagues on how assessment should change now that every student has AI; the oral check-ins in STOR 612 are my first step. What has not changed since I was fifteen is the part I enjoy most: helping someone do something they could not do the day before.