Amir Reza Peimani

Email: amir.peimani@utoronto.ca

Teaching Statement

My Teaching Approach: Creating Clarity, Confidence, and Capability

My teaching philosophy is rooted in a simple belief: students learn best when complex ideas are made clear, when they feel supported, and when they understand how concepts connect to real-world problems. Across more than a decade of teaching at the University of Toronto, Northeastern University, Seneca Polytechnic, the University of Waterloo, and York University, I have worked with students from a wide range of academic backgrounds, cultures, and levels of preparation. My goal is to meet every student, regardless of their starting point, with structured guidance, intuitive explanations, practical examples, and a learning environment that encourages questions, collaboration, and curiosity.

My approach is shaped by three foundational commitments:

- Make the complex understandable through structure, intuition, and examples.
- Create an inclusive environment where diverse students can succeed.
- Connect theory to practice through hands-on, relevant, real-world applications.

Whether I am teaching computer science (data structures, algorithms, theory of computation, Python programming), data science (foundations, statistics, large-scale data analysis), machine learning and deep learning, or emerging areas of AI (LLMs and transformer architectures, NLP, reinforcement learning, generative models), these principles guide how I design courses, teach concepts, and mentor students. My goal is for students not only to master the material, but to build the confidence and intuition needed to navigate rapidly evolving technologies and apply them thoughtfully in real-world contexts.

Teaching 70–700 Students: Structure, Transparency, and Active Engagement

I have taught classes ranging from small seminars of about 30–40 students to large-scale courses with 500–700+ students across multiple sections. In all contexts, organization and clarity are essential. Students consistently report appreciating my ability to break down complex topics into intuitive steps. In written feedback/surveys, they highlight that I am prepared, supportive, and approachable:

> "... he puts a lot of effort into preparing his own slides—which, honestly, can be more helpful than the official readings of the course. Great sense of humor, open to feedback, happy to answer questions." (Anonymous student)

> "Amir has been one of the best TAs I have worked with because it is very clear that he cares about giving students an excellent learning experience. He proactively seeks feedback from myself and his students ... this sets him apart from other TAs. All in all, he makes my job as an instructor easier." (Instructor, 2022)

> "He always took the time to explain concepts until I understood the material presented in class. His dedication to the Computer Science department at UofT is outstanding

and his amiable character makes him very approachable." (Anonymous student)

Informal end-of-term surveys show organization and knowledge ratings in the 4.6–5.0/5 range across multiple courses.

To keep large classes engaged and supported, I rely on several practices.

Active learning in every lecture. I incorporate live coding walkthroughs, think–pair–share problem solving, mini–quizzes and polls, on-the-fly debugging demonstrations (for programming, data science, and ML courses), step-by-step reasoning for proofs (as in theory of computation), and real-time visualizations for data science topics. In advanced AI courses, I extend this to interactive demonstrations of model behavior, such as attention maps in transformer models, gradient-flow visualizations in deep learning, and policy rollouts in reinforcement learning. Students remain active participants rather than passive listeners.

Scaffolding difficult concepts. I intentionally build lessons with conceptual overviews, simple motivating examples, progressively harder exercises, and a "big picture" synthesis at the end. This applies across levels, from recursive proofs and induction, to asymptotic analysis in algorithms, to eigenvalues and linear algebra in deep learning, to statistical modeling and evaluation in machine learning, to attention mechanisms and representation learning in LLMs and transformers, to policy optimization and value-based reasoning in reinforcement learning, to sequence modeling and linguistic structure in NLP systems, and to interpretability and alignment concepts across modern AI architectures.

Transparency and predictability. Large classes demand predictable structure. I clearly outline weekly expectations, assignment breakdowns, grading rubrics, learning objectives, and timelines for feedback. Students repeatedly comment that this structure reduces anxiety and helps them focus on learning rather than logistics.

Building Inclusive, Supportive, and Culturally-Responsive Classrooms

Teaching at large public universities means teaching students with very diverse backgrounds, including international learners, first-generation students, students with learning differences, and students with variable prior preparation in mathematics and programming. To meet these needs, I draw on several commitments.

Designing for multiple learning styles. I integrate written explanations, diagrams and visualizations, live coding, verbal reasoning, collaborative problem solving, and recorded walkthroughs when permitted. This multimodal approach ensures that students can choose the learning channels that work best for them, whether they are more comfortable with algebraic manipulation, visual intuition, or hands-on experimentation.

Normalizing questions and uncertainty. I adopt a tone that encourages questions, framing confusion as a normal part of learning: for example, "If this part seems confusing, you're in the right place, let's unpack it together." This has made my classes feel safe for students who may feel intimidated by technical subjects such as proofs, algorithms, or neural networks.

Accessibility and responsiveness. I offer consistent office hours, prompt responses on Ed Discussion or Piazza, additional example sheets, and optional weekly review sessions. Even in large classes, students often report feeling seen, supported, and comfortable approaching me for help.

Culturally aware and ethical teaching. In machine learning, deep learning, and AI courses, I intentionally integrate discussions of fairness, bias, responsible use of models, societal impacts of algorithms, data ethics and privacy, and reproducibility. These topics are central to modern computing and data science education and essential for preparing students to use technology responsibly.

Teaching Across Disciplines: CS, Data Science, AI, Engineering, and Health

My teaching spans multiple disciplines: computer science (CSCB63 data structures, CSC236 theory of computation, CSC108/CSCA08/CSC111 foundations), data science and analytics (GGR376 advanced data analysis, BAN210/230 predictive analytics and data mining, BME1510 data science for bioengineers), machine learning and deep learning (CSC311, CSC413/2516, APS360, CHL5230, CSC490), and engineering and computation (ECE1779 cloud computing, ECE448 biocomputation, MIE334 numerical methods, MIE1628 big data science).

This breadth allows me to teach:

- core undergraduate CS courses (intro programming, data structures, algorithms, theory of computation, discrete math for CS);
- data science and statistics courses (foundations of data science, probability and statistics for data science, applied regression, data visualization, SQL/ETL and databases);
- ML/AI courses (introductory ML, deep learning, AI, neural networks, ML for health and biomedical signals, LLMs and transformer models, NLP and sequence modeling, reinforcement learning, and modern generative AI);
- interdisciplinary and project-based courses that integrate data science and computing with neuroscience, geography, public health, and engineering.

Connecting Theory and Practice: Real-World Relevance

Students learn best when concepts feel useful. In my programming, data science, and ML courses, I integrate domain-specific datasets (health and physiology, neuroscience, environmental data, and business analytics), full end-to-end workflows (data cleaning → feature engineering → modeling → evaluation → interpretation), hands-on labs, open-source tools (Pandas, NumPy, scikit-learn, PyTorch, Spark), and real debugging workflows with authentic error messages.

For example, in applied ML courses I guide students through building and evaluating classifiers and regressors, comparing baselines and regularized models, exploring hyperparameters, and diagnosing overfitting and data leakage. In advanced AI courses, I extend this to modern topics such as fine-tuning and prompting LLMs, analyzing attention patterns in transformer models, training and evaluating RL agents, and implementing core NLP pipelines (tokenization, embeddings, sequence modeling). In data structures and theory courses, I emphasize how the mathematical abstractions students learn, such as asymptotic analysis, inductive proofs, and automata, directly influence the performance, correctness, and scalability of the systems they use and build.

Mentoring, Feedback, and Student Success

I see grading not just as evaluation, but as another opportunity for learning. I design rubrics that highlight conceptual steps, reward partial reasoning, and help students identify where their thinking broke down. I provide in-line comments and host "grading review" Q&A sessions. For students with anxiety around coding or proofs, I build one-on-one scaffolding sessions that gradually increase the level of challenge.

Faculty colleagues have noticed the impact of this approach. One instructor noted that, compared to previous offerings of the same course, they received far fewer complaints about difficulty and credited my guidance and support in tutorials. Students similarly report that tutorials and office hours were "essential" to their learning and that my explanations and patience helped them overcome challenges they had previously found overwhelming.

Courses I Am Prepared to Teach

Based on my background and teaching experience, I am prepared to teach a wide range of undergraduate and graduate courses, including:

- *Computer Science:* Introductory programming in Python; Data Structures and Algorithms; Theory of Computation and Automata; Discrete Mathematics for Computer Science; Foundations of Computer Science.
- *Data Science & Statistics:* Foundations of Data Science; Probability and Statistics for Data Science; Applied Regression; Data Visualization; Data Wrangling, SQL, and Databases; Capstone and project-based data science courses.
- *Machine Learning & AI:* Introduction to Machine Learning; Deep Learning; Artificial Intelligence; Neural Networks; ML for Health and Biomedical Data; Applied ML project and capstone courses; Large Language Models and Transformer Architectures; Natural Language Processing; Reinforcement Learning; Generative AI and modern representation learning.
- *Engineering & Interdisciplinary:* Cloud Computing; Numerical Methods; Biocomputation; Scientific Computing; interdisciplinary connector courses that link data science and computing with health, neuroscience, geography, or social sciences.

This flexibility allows me to contribute both to large foundational courses and to advanced or interdisciplinary offerings, depending on departmental needs.

My Commitment as an Educator

My goal as an instructor is to empower students with the tools, confidence, and curiosity needed to solve complex problems and think critically about technology. I strive to build equitable and inclusive classrooms, explain concepts with clarity and empathy, guide students through authentic hands-on workflows, and foster a community where questions and mistakes are welcome. I want students not only to know how to implement algorithms and models, but also to understand their assumptions, limitations, and broader societal impacts.

Teaching is one of the most meaningful parts of my academic career, and I approach every course, small or large, introductory or advanced, with enthusiasm, structure, and a deep commitment to student growth.