# Welcome!

AI for Algorithmic Reasoning and Optimization

# About me

Grew up in Lincoln, Vermont

BA: Columbia
*Math*

PhD: Carnegie Mellon
*Computer Science*

Faculty: Stanford
*Research:*

- *Machine learning*
- *Algorithm design and*
- *Interface between Econ and CS*

# Outline for today

1. **High-level overview**
2. Course outline
3. Course format
4. Policies
5. Project

# AI for Algorithmic Reasoning & Opt

## Core idea

- Models that **choose**, **design**, or **execute** algorithms
- Bridges LLMs, GNNs, optimization, and theory

## Core capabilities (examples)

- Translate natural language into mathematical formulations
- Learn heuristics, policies, and solver configurations
- Simulate algorithmic primitives (search, DP, …)
- Co-optimize models and solvers for performance gains

# AI for Algorithmic Reasoning & Opt

In this class, we'll cover…

## Evaluation principles

- Evaluate optimality, feasibility, generalization, efficiency
- Respect problem structure: graphs, constraints, objectives, …

## Applications and limits

- Power real-world systems: energy, logistics, markets, planning
- Recognize limits: NP-hardness, shifts, safety, reproducibility

# Why now?

**Advances in models**

- Transformers enable in-context algorithm selection
- GNNs approximate local algorithms with growing theory
- Diffusion models generate structured, combinatorial solutions

**Infrastructure**

- Benchmarks and datasets across routing, scheduling, …
- Hardware and libraries support large-scale experimentation

# Why now?

**Applications**

Power systems, logistics, etc. need learned decision systems

**Foundations**

Guarantees with predictions connect ML and classical theory

**Community momentum**

Seminars, workshops, tutorials, …

# Key archetypes to organize the field

**Select**

Choose algorithms, heuristics, or configurations per instance

**Simulate**

Learn algorithmic primitives (e.g., BFS, DP, local search)

**Co-optimize**

Couple models and solvers in hybrid methods

**Design**

Discover new algorithms and heuristics

# Outline for today

1. High-level overview
2. **Course outline**
3. Course format
4. Policies
5. Project

# Course map

**Foundations** (9/23 – 10/2)
Core ideas: GNNs, discrete optimization, approximation algorithms

**Transformers & LLMs** (10/7 – 10/16)

**Graph Neural Networks** (10/21 – 10/30)

**Mathematical Optimization** (11/6 – 11/13)

**Theoretical Guarantees** (11/18 – 11/20)

# Alg. reasoning with transformers & LLMs

**Goal:** uncover algorithmic behaviors inside large seq. models

**Focus:** regression ICL, algorithm selection, auto-design

**Core questions:**

- What algorithms emerge from in-context learning?
- How do transformers select among candidate algorithms?
- Can LLMs design new heuristics (beyond selection)?

# Algorithmic reasoning with GNNs

**Goal:** analyze how GNNs generate solutions to graph problems

**Focus:** classical algorithms to graph diffusion models

**Core questions:**

- How powerful can GNNs be as approximation algorithms?
- Do GNNs benefit from learning related problems together?
- Can diffusion models be used to solve hard graph problems?

# Mathematical optimization

**Goal:** connect ML/LLMs with optimization modeling & solving

**Focus:** NL→mathematical models, ML-guided search

**Core questions:**

- How can LLMs make optimization modeling more accessible?
- Can learning-guided search improve solver performance?
- What does it mean to make discrete opt. differentiable?

# Theoretical guarantees

**Goal:** study when ML-guided optimization is provably reliable

**Focus:** landscapes of solution samplers, apx. w/ predictions

**Core questions:**

- What can gradient methods guarantee for solution samplers?
- When can learned guidance be both useful and robust?

# Course map

**Foundations** (9/23 – 10/2)

Core ideas: GNNs, discrete optimization, approximation algorithms

**Transformers & LLMs** (10/7 – 10/16)

In-context learning, algorithm selection, auto-design

**Graph Neural Networks** (10/21 – 10/30)

Approximation algorithms, dual reasoning, graph diffusion solvers

**Mathematical Optimization** (11/6 – 11/13)

LLMs for modeling, ML-guided search, differentiable discrete opt.

**Theoretical Guarantees** (11/18 – 11/20)

Policy-gradient landscapes, approximation with predictions

# Big questions across the course

How do we **represent** algorithms in ML models?

How should we **evaluate** learned algorithms?

What are the **limits** – and **guarantees** – we can prove?

Where do these ideas matter in **practice**?

# Something for everyone!

If you like **theory**:

- Provable guarantees for ML-guided algorithms
- Explore limits of LLMs, GNNs, etc. under NP-hardness

If you like **machine learning**:

- Study how transformers, GNNs, etc. reason algorithmically
- See how ML extends beyond prediction into problem-solving

# Something for everyone!

If you like **optimization**:
- LLM-based modeling, ML-guided search, differentiable opt.
- Understand when ML improves—or fails to improve—solvers

If you like **applications**:
- Apply ideas to energy, logistics, finance, and markets
- Gain a toolkit for structuring real-world decision problems

# Outline for today

1. High-level overview
2. Course outline
3. **Course format**
4. Policies
5. Project

# Two types of class sessions

**Foundations lectures**

**Discussions**

# Foundations lectures

Cover the core background you'll need for the discussions

Topics include:

- Basics of graph neural networks
- Mathematical optimization
- Approximation algorithms
- Diffusion models

Format: Whiteboard

Research shows students retain concepts better this way

# Discussions: the heart of the course

Most meetings will be student-led discussions

Each discussion covers one well-received paper from ’23-’25

Some days you’ll be a presenter, some days a non-presenter

The goal: spark conversations; analyze and build on ideas

# Format of a discussion

**Presenters**:

- One of three roles (Archaeologist, Researcher, Reviewer)
- ~7 minutes presenting highlights + 3 minutes Q&A

**Non-presenters**:

- Post one discussion question on Ed by 1pm that day
- Join the class conversation

**Professor**:

- Closes each class with a 20-minute preview of the next paper
- Helps clarify what to focus on when reading

# Archeologist

Situate the paper in context

Find one **older paper** it cites and explain the connection

Find one **newer paper** that cites it and explain the extension

Show how both relate to the paper's main idea

# Researcher

Propose a follow-up project inspired by the paper

Ground it in the paper’s ideas, results, data, or code

# NeurIPS reviewer

Critique the paper as if you're writing a review

Provide ≥3 strengths and ≥3 weaknesses

Address: quality, clarity, significance, originality

# Non-presenter Ed discussion question

Characteristics of a good question:

- Specific: anchored in the paper’s setup
- Forward-looking: invites thinking beyond what’s tested
- Open-ended: there’s no “right” answer, good for conversation
- Accessible: everyone in class can weigh in

# Why this format?

**Build research skills**

- Learn to read papers quickly but deeply
- Practice asking sharp, thought-provoking questions
- From course evaluations: *"I'm a junior who had never done research before, but after this class, I feel equipped to do so."*

**Shape your own projects**

- See how to frame problems and methods
- Spot open directions for your final project

# Why this format?

## Join the research community

- Discussions mirror what happens at conferences

## Make the class fun

- Conversations are more memorable than lectures
- You'll learn from each other, not just from me

# Why this format?

- Format developed by Alec Jacobson & Colin Raffel (U of T)
- Great feedback here at Stanford!
  - *"Very helpful in developing my ability to read and digest research papers, in addition to engaging with them in an intellectually meaningful way."*
  - *"The course format [was] a great way to interact with the various papers we read."*
  - *"The seminar format led to some really cool conversations about potential extensions of the work, applications to other areas, etc. I wish more classes were structured like this."*

# Role logistics

Everyone will rotate through all three roles:

- Archaeologist
- Researcher
- Reviewer

Expect to be in each role once or twice during the quarter

- The exact count will depend on final enrollment

If the final enrollment is large, some roles may be done in teams

# Outline for today

1. High-level overview
2. Course outline
3. Course format
4. **Policies**
5. Project

# Attendance: Lectures

Attendance is optional but recommended

These are the “Foundations” sessions on the schedule

# Attendance: Discussions

Attendance is required

We know life happens (e.g., illness), so here's the policy:

- If scheduled as a Presenter:
  - You may miss one session
  - Must submit a video of your presentation within one week
- If scheduled as a Non-presenter:
  - You may miss up to two sessions
  - Still required to post your question on Ed

# Grading breakdown

**60 points: Discussion**

- E.g., if 6× Presenter and 6× Non-presenter:
  - Each presentation = 8 points → total 48 points
  - Each non-presenter assignment = 2 points → total 12 points
- Subject to change depending on final enrollment

**40 points: Project**

**Pass/fail option:** Only discussion, no project

# Outline for today

1. High-level overview
2. Course outline
3. Course format
4. Policies
5. **Project**

# Project overview

- Write a mini-paper (empirical, theoretical, or both)
- Option to work in groups (max 3 students) or solo
  - If in group: include specific contributions paragraph
- Groups expected to produce more work than solo projects
- Paper length: 3 + n pages (n = number of authors)
  - Excluding references & contributions
- Projects should show novelty
  - New application, method, or perspective

# Milestones/deadlines

- 10/10: Topic interests spreadsheet
  - Fill in topics you're excited about
  - If you want a project partner, reach out to like-minded students
- 10/31: Progress report
  - 1-2 pages describing project + partial results
- 12/4: Final presentation
  - Present your final project in class
- 12/12: Final writeup due

# Grading breakdown

Project matching spreadsheet – 3 points

Progress report – 7 points

Writing – 10 points (readability, completeness, context)

Novelty – 10 points (new idea, method, or perspective)

Final presentation – 10 points (clarity, insight into what you did)

# Course Assistant

- TBA!

- They will have 5+ hours a week of bookable office hours
  - Take advantage of this for the course project!
  - Every year, at least one project turns into a NeurIPS/ICML paper
  - Meet with them (and me!) often to push project

Looking forward to a great quarter!