Université de Montréal

Mila

Institut Courtois

Advanced Machine Learning Projects

Projets avancés en apprentissage automatique

IFT6759, Winter 2026

Lecture 1: Introduction to the course

Noam Aigerman (he/him/il)

Lecture outline

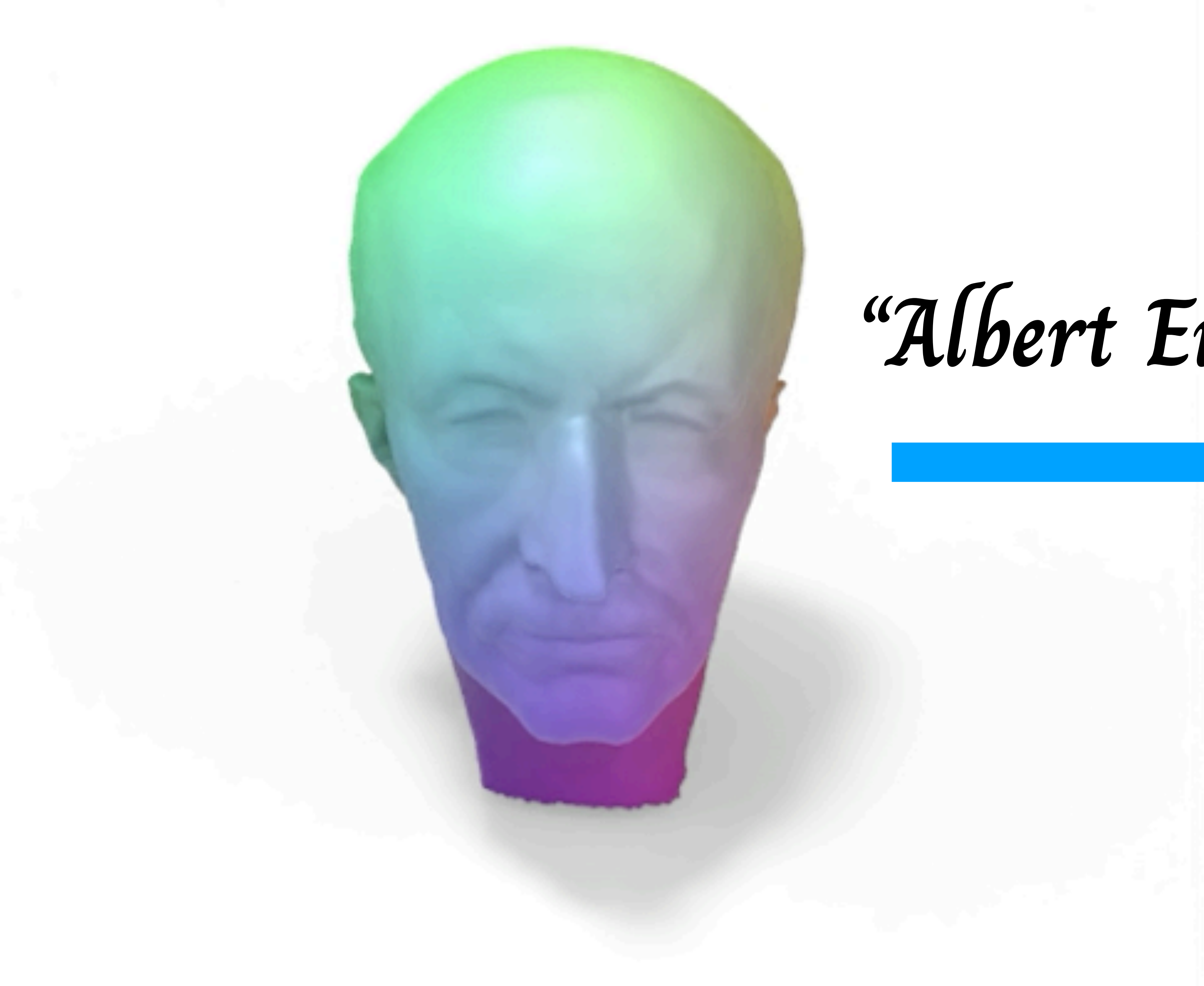
1. Small preface
2. Course logistics
3. Project guidelines
4. Resources

3 About me

- Area of research: ML, focused on 3D vision/geometry/graphics

Adobe

Université de Montréal

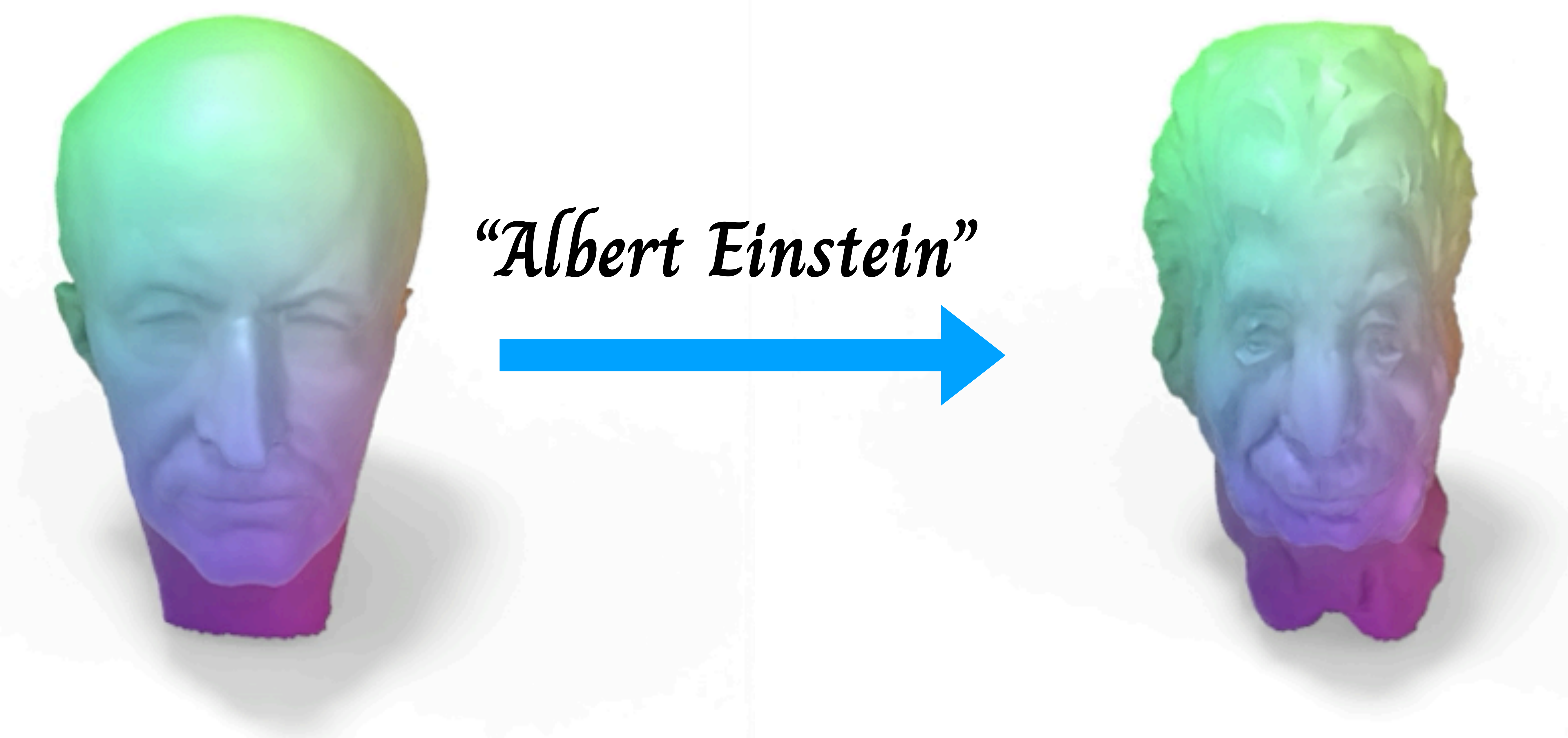


"Albert Einstein"

About me

- Area of research: ML, focused on 3D vision/geometry/graphics

Adobe → Université de Montréal



About me

- Passionate about teaching
  - Want to make this course the best it can be
  - Always in dialogue with you - let’s make this into a discussion!
  - Try to engage and don’t be shy

About me

- I Don't believe in "Professor is omniscient" approach
  - Will admit my own shortcomings
  - (Almost) no one knows everything, for instance:
    - I have zero experience in ML for audio
    - I wrote papers using GANs, but never trained one myself
  - But:

    Doesn't mean I don't know what I am talking about :P

  - In each of your projects, I (hope) to learn something new
    - Maybe new programming trick
    - Maybe new theory

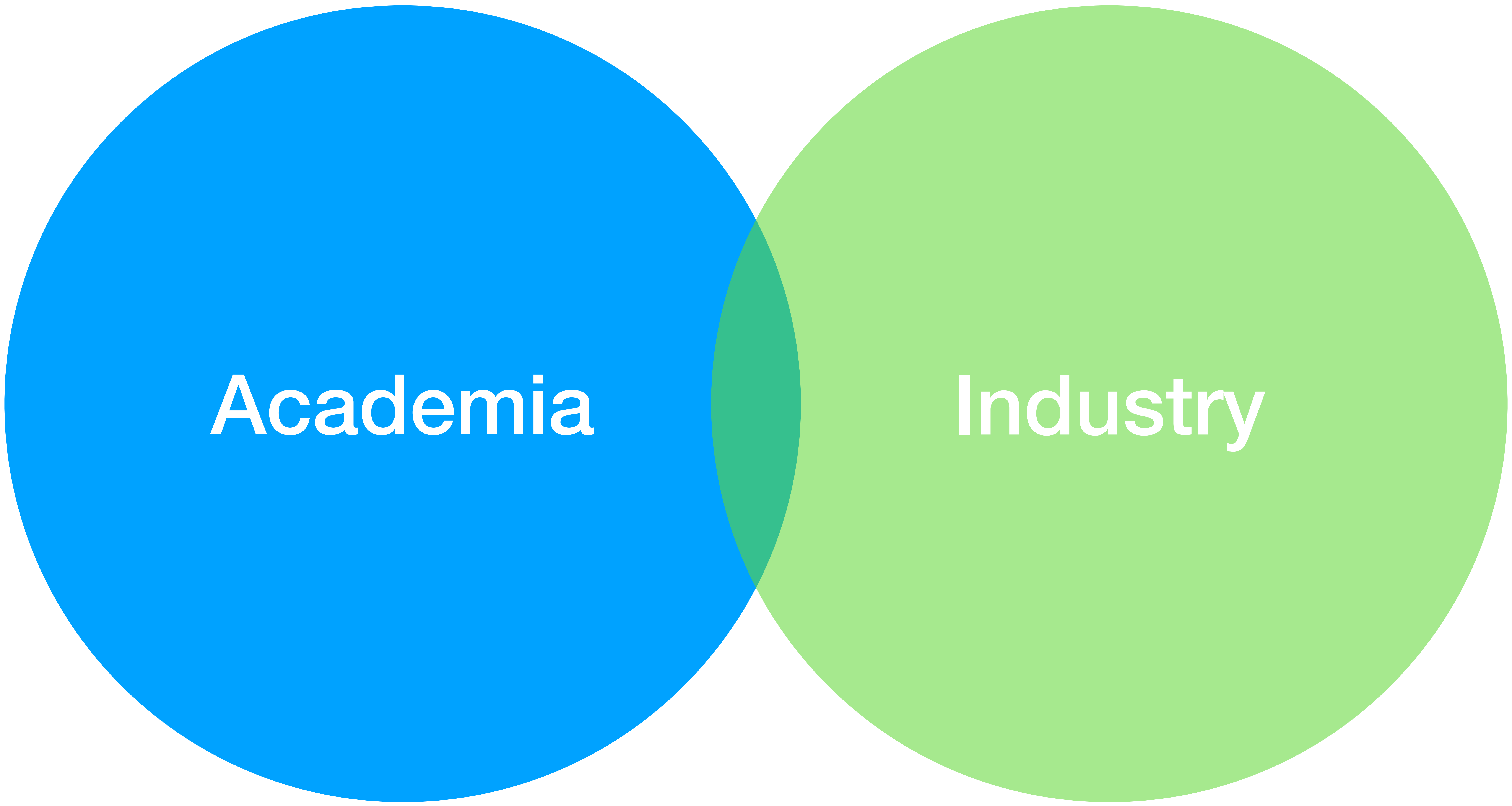
Regarding the course

- A bit of a weird course
  - Mainly: students working on projects
  - Not built around frontal lectures

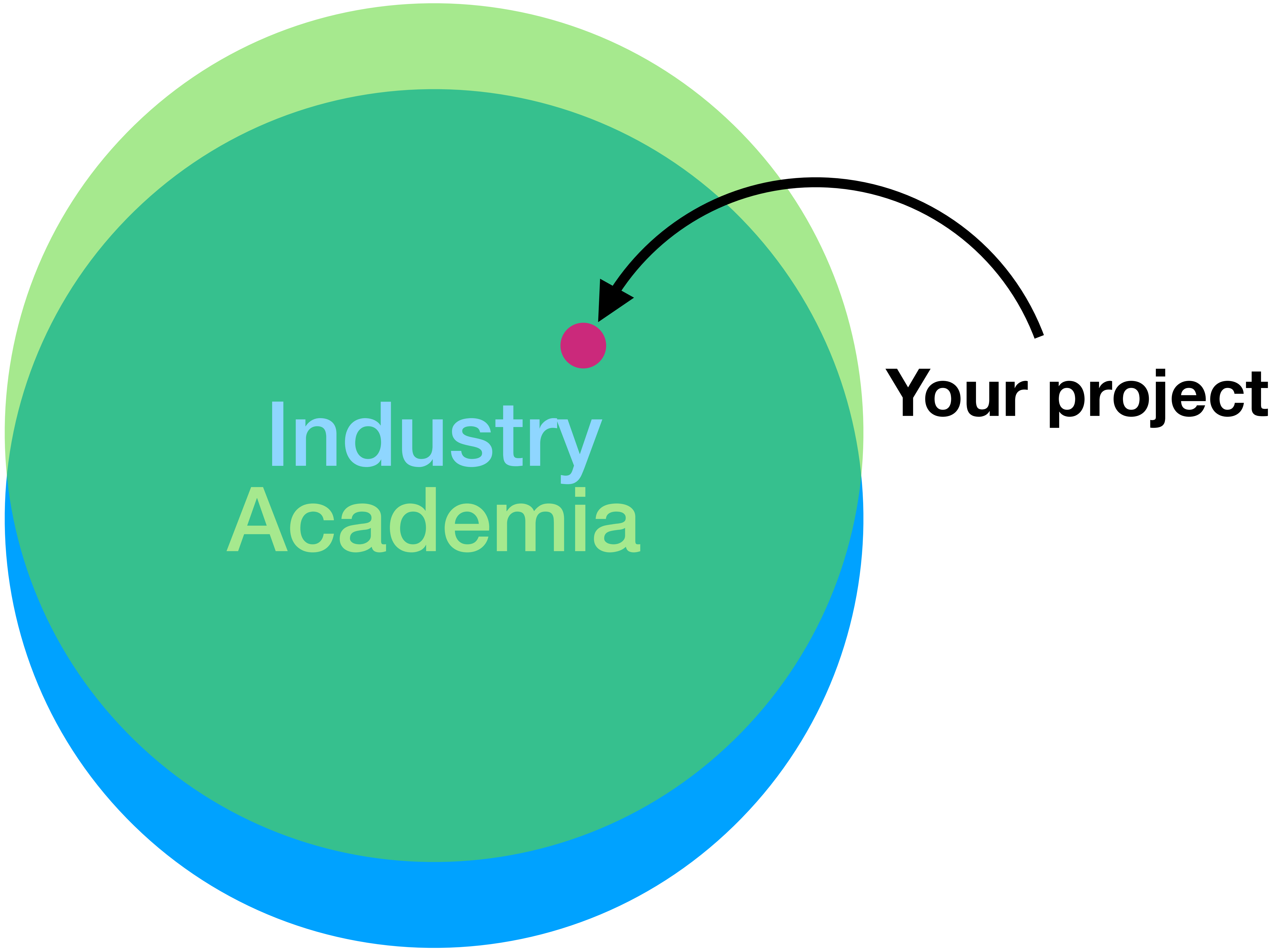
- Will try to add some useful bits
  - How to find relevant works
  - Paper reading
  - Presenting

8

Research in general



9 AI research



What is this course about?

- Experience complete lifecycle of an AI project

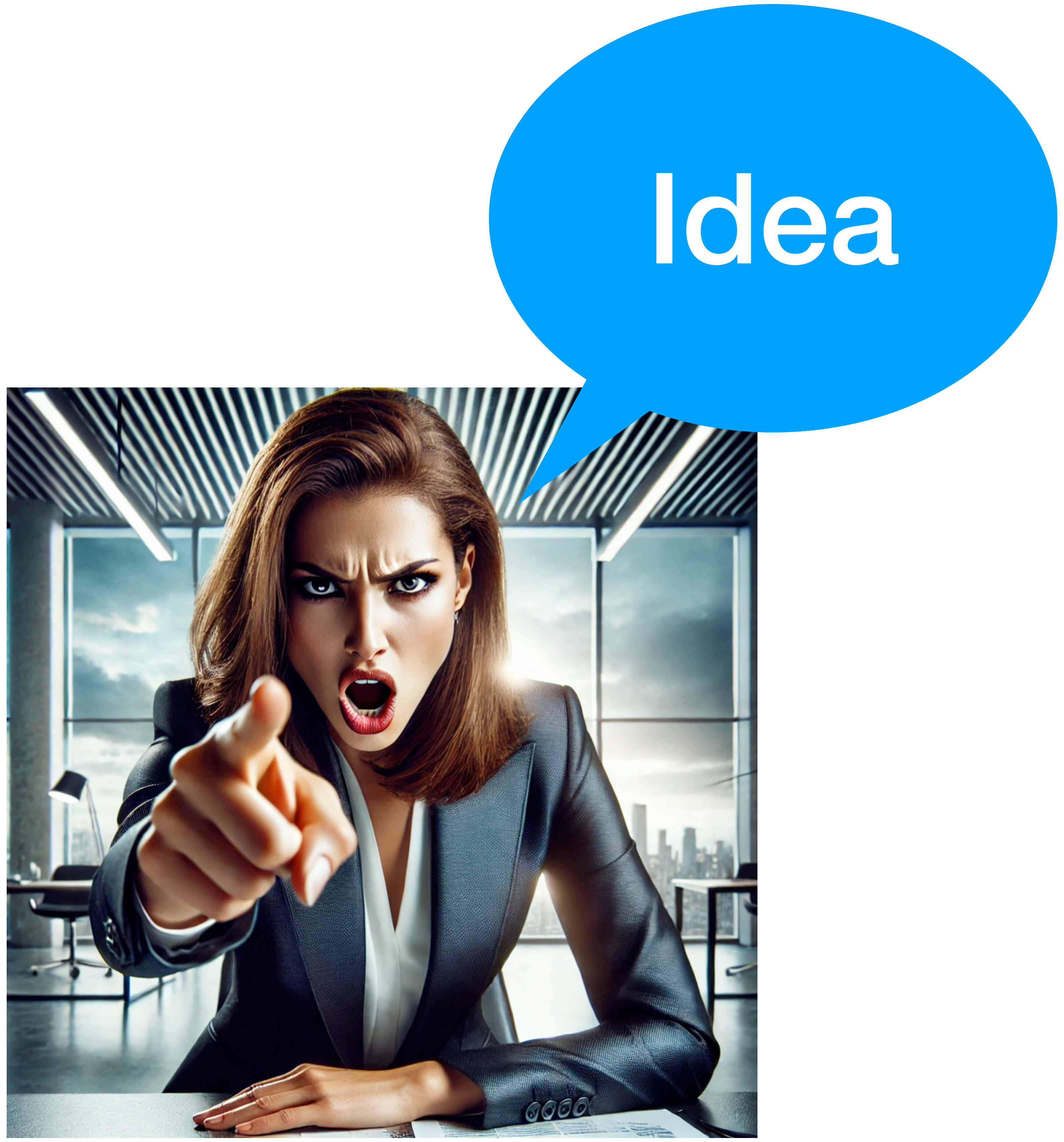
Your project

11 How I see this course

- Deep learning is a lot of black magic
  - Hard to teach in courses
  - Not discussed in papers
  - Scattered in blog posts/tweets
  - Mostly passed on by word of mouth

- Skills for deep learning are “soft”, learned through experience
  - Tricks of the trade
  - Technical skills (debugging)
  - Research skills (how to google/chatGPT, read papers)
  - Also: communication

⑫ Experience complete “lifecycle” of a non-trivial project



A project lifecycle

- What’s in a project?

A project lifecycle: preparation

- Ideation

- Finding relevant material
  - Papers
  - Tutorials
  - Blog posts
  - Github repos

- Reading the material
  - Understanding the method
  - Understanding pros/cons
  - Feasibility (compute?)

- Conveying information to other researchers and to "customers"
  - Summarize papers
  - explain important key points

A project lifecycle: execution

- Coding…
- Running experiments
  - Is it working, and why?
- Debugging
  - Is it a bug, or just bad results?
  - How to pinpoint the bug
- Managing compute resources
- Communicating in a group
- Show results
  - Iterate with client
  - Show final product/paper

Take this as an opportunity

- Learn the above skills
  - By practicing
  - By discussing with others
  - By looking for solutions online

- Choose a project that:
  - Interests you
  - Challenges you
  - Is still feasible (so you don’t get stuck)

Course logistics

Course logistics in brief

- **Structure:**
  - 4 weeks of introductory and review **lectures**: machine learning, deep learning, git and GitHub, Linux, Python and PyTorch, …
  - 10 weeks of **project work in teams** and **student presentations**.
- **At the end of the course, we expect you to be able to:**
  - Code and train machine learning and deep learning models.
  - Analyze the experimental results and draw conclusions.
  - Communicate written and orally the main aspects of the project.

Teacher and teaching assistants

- **Teacher**
  - Noam Aigerman (noam.aigerman@umontreal.ca)

- **Teaching assistants**
  - TBD

Class schedule

- **When and where:**
  - Mondays: 10:30 am - 12:29 pm (Y-115 Pav. Roger-Gaudry)
  - Thursdays: 10:30 am - 12:29 pm (1177 Pav. Andre-Aisenstadt)

- **Tentative outline:**
  - First 4 weeks:
    - Introduction (today)
    - Machine learning review
    - Deep learning review
    - Git and GitHub tutorial
    - Linux and Shell
    - Python and PyTorch
    - DL tricks
    - HPC clusters

Class schedule

- **Tentative outline:**
  - Weeks 5~14:
    - Work on **projects** in teams
    - **Office hours** during class time, or on request
    - **Student presentations** (progress, classical papers in machine learning and deep learning, cutting-edge research works, …)
    - Additional **tutorials** may be scheduled for support

Practical information

- **Important announcements:** **StudiUM**

- **Updated course materials:** **StudiUM**, and will also be upload to **my course webpage**.

- **Daily / informal communication:** slack
  - (https://join.slack.com/t/ift6759advanc-9fa7327/shared_invite/zt-2x76h2ogn-Y_N2aYpHPmb5LmiltU2_GA)
  - E-mail me if you cannot access

- **Also Piazza**: https://piazza.com/umontreal.ca/winter2026/ift6759b

- **Feel free to contact the teacher and TAs by email**

Course work and grading policy

- **Course project:**
  - Each team consists of **3~4 students**.
  - You can select one of the projects we provided, or define your own project (discuss with us before finalizing your project).
  - **3 milestones** to track the progress.
  - One **final presentation** (in teams).
  - One **final report** (in teams).
  - **Paper presentation** in class (individual).

Evaluation criteria

- Students will be evaluated **entirely according to their work on projects**. The final evaluation will take the following into account:
  - Difficulty of the project
  - Performance of the developed algorithms
  - Oral presentation
  - Written report
  - Quality of the code

- The grade will be **binary (pass or fail)**, not in a letter scale.

Computing resources

- **Computing Resources**
  - Google Colab, Google Cloud, Calc Québec, etc.
  - Experiments may take up to hours
  - Recommendation: start early (team up, define your projects, …)

Project Guidelines

What is a project?

- For 10 weeks (week 5 to week 14), you will work on a project in teams. The projects of the course are aimed to resemble as much as possible what real-world machine learning projects look like in either industry or research.

- Projects comprise the following stages:
  1. Literature review
  2. Planning
  3. Development
  4. Analysis of results
  5. Written report
  6. Oral presentation

Working in teams

- **Important:** The projects will be developed in teams of 3-4 students. Working in teams of at least 3 students is a requirement of the course. Larger teams may be accepted too.

- Working in teams does not necessarily mean that every teammate contributes equally to every part of the project, but we do expect every student to engage in all stages of the project (literature review, planning, coding, analysis, writing, presentation, etc.).

- Mechanisms to facilitate teamwork:
  - Students are free to organize themselves and propose teams to work on specific projects.
  - Otherwise, teams will be formed according to the preferences provided by the students.
  - Team-wise meetings with the teacher and TAs, to assess both the progress and the functioning of the team.
  - Questionnaire about the functioning of the team and the contributions at the end of the course.

Teamwork is a HUGE skill

- From experience, one of the first thing asked about a job candidate

- Learn to:
  - Be chill and friendly
  - Cooperative
  - Resolve disagreements
  - Work with people who are more/less experienced

- Don’t try to optimize your team to be the most advanced
  - Very senior folks are expected to do more advanced work
  - Undergrads get a break

Some general ideas

- Applications:
  - NLP
  - Signal Processing:
  - Non-trivial work with audio
  - Non-trivial work with vision

- Architectures:
  - Explainability
  - Fine-tuning
  - Latent-space design
  - Mutlimodality

Example projects:

- From last year:
  - Measuring the Opioid Crisis Using the NAMCS Dataset
  - World Cup Prediction
  - Sports Analysis for Soccer Games
  - Diagnose Medical Images for COVID-19
  - Cryptocurrency Movement Forecasting
  - Event Causality Identification
  - Pre-trained RNNs vs Pre-trained Transformers
  - Intended Sarcasm Detection
  - Procedural Knowledge Extraction with Language Models

From previous years:

- Predicting Stock Market Trends
- Automated Music Generation
- Smart Healthcare – ECG Anomaly Detection
- Autonomous Driving Simulation
- Fake News Detection
- Smart Home Energy Management
- Sign Language Recognition
- Climate Change Impact Prediction

Proposing your own project

In order to foster creativity and allow you to work on projects of your interest, **you are welcome to propose your own projects**. Nonetheless, **the project proposal must be accepted by the instructors** and the decision will be based on **the following criteria**:

- The project must involve the **use of advanced machine learning methods**.
- Works developed prior to this class will **not** be accepted.
- It must be **feasible** in terms of computational resources and time constraints.
- The data must be **publicly** available.
- The project should not raise serious **ethical** concerns.

If you have a project in mind, speak with the instructors as soon as possible!

How to plan a project

- What general area do I want to work on?
- What datasets are available?
- What are works everyone is excited about?
- Is it feasible with the compute I have?
- Is it notoriously difficult?

How to divide into teams

- Propose a direction you want to build a team around
- Find other people and figure out a direction together
- You can organically group up to 3 people
    - Waiting list for the 4th person, to ensure everyone gets a slot

Evaluation criteria

- The final grade will be based on the continuous work on the project, and the composition of the team, as discussed before), following these criteria, all with equivalent weight:
  - **Difficulty** of the project
  - **Quality and performance** of the developed algorithms: suitability of methods, technical rigour, results, etc.
  - **Written report** completeness, clarity, technical soundness, analysis, etc.
  - **Oral presentation** effectiveness, clarity of the presentation, etc.
  - **Code** clarity, documentation, modularity, extendability, etc.

- The baseline evaluation will be the same for all team members. However, the individual grades may be adjusted if necessary in the case of participation imbalance.

Pre-requisites

- **Why pre-requisites?**
  - This is **not** an Introduction to machine learning course, but **(Advanced)** machine learning projects.
  - All students must be able to contribute to the team.

- **What are the pre-requisites?**
  - Core machine learning concepts: there will be a review session.
  - Basic deep learning concepts: there will be a review session.
  - Familiarity with Python (And you will need PyTorch/Tensorflow/Jax, Hugging Face Transformers)
  - Basic Linux commands: there will be a tutorial session.
  - Familiarity with `git` and GitHub: there will be a tutorial session.

- **Important:** if you believe you do not have enough experience with any of these skills, we suggest you to make an effort to catch up during the first two weeks. Contact the instructor for supporting material. We are here to help!

Resources

Resources

- No textbook is required. But the following resources that can be read free online are helpful.
  - Ian Goodfellow, Yoshua Bengio, and Aaron Courville (2016). Deep Learning. MIT press.
  - Abu-Mostafa, Y. S., Magdon-Ismail, M., & Lin, H. T. (2012). Learning from Data. AMLBook.
  - Anish, Jose, Jon (last seen on Dec. 2021). The Missing Semester of Your CS Education. CSAIL MIT.
  - Aston Zhang, Zack C. Lipton, Mu Li, Alex J. Smola. Dive into Deep Learning.

- The following book is helpful to give you more background about neural networks and natural language processing.
  - Michael A. Nielsen. Neural Networks and Deep Learning

- The following book is helpful to give you more background about natural language processing if your project is related to this domain.
  - Li Deng, Yang Liu (2018). Deep Learning in Natural Language Processing. Springer.

Resources

- The following software or libraries can be helpful.
  - PyTorch an open-source deep learning library.
  - DGL an open-source library for deep learning on graphs.
  - HuggingFace Transformers an open-source library containing PyTorch and Tensorflow implementations, pre-trained model weights, usage scripts and conversion utilities for a variety of pre-trained language models.

Todo

- Check the StudiUM and **course webpage**.
- Fill in the survey
- **Start preparing your term project early**: build your team (up to 4 people), read the project proposal instructions

Next lecture: Useful skills for conducting ML research + Machine Learning Basics

Thanks! Q&A

Noam Aigerman
Email: noam.aigerman@umontreal.ca
Homepage: https://noamaig.github.io/

Université de Montréal

Mila

Something unorthodox

- Because this is a class about team building

- Tiny bit of introduction:
  - Something you like outside of work (hobby, animal, TikTok influencer, whatever)
  - Your dream ML project