CSC2541H / (PCL3107H, PCL3108H): AI for Drug Discovery

Introduction to Machine Learning

Chris J. Maddison
University of Toronto

Teaching Team

Chris

CS Instructor

Rebecca

PT Instructor

Martin

PT Instructor

Mica

TA

Ella

TA

Stef

TA

About me:

- Assistant professor, cross-appointed in computer science and statistics
- Recent interest in drug discovery inspired by some recent health challenges
- I know very little about drug discovery and development—excited to learn from you!

About this course

- Two sister course codes (CSC2541H / (PCL3107H, PCL3108H)), one course
- Reversed instruction—I am lecturing in PT and the PT instructors are lecturing in DCS
- **We're building the plane as it flies**

- First six weeks—lectures, notebooks, and project proposal (aka LOI)
- Second six weeks—co-working on a project
- Course website has all of the important information, including syllabus and schedule

Learning objectives

Our objective is to

- prepare you for a future of working collaboratively across disciplines in both the biological sciences and AI
- prepare you to brainstorming across disciplines while contributing your subject matter expertise
- prepare you to turn inter-disciplinary ideas into falsifiable hypotheses and design an experimental approach

We will not

- teach you how to code
- train you to be experts in AI or machine learning
- train you in a certain wet lab technique
- teach you how to develop or design drugs
- teach you biology

Assessments

- **Code notebooks (collaboration highly encouraged).** Designed to be more approachable with a lower barrier to entry in terms of programming background. They might be intimidating but we have great TAs that will be around to help and your peers in the CS coursecode.
- **LOI (teams of 3-4 with 2-3 CS 1 PT).** A project proposal, which you can think of as a letter of intent for a grant application. Answers the question, what project are you hoping to work on?
- **Project report (same team as LOI).** A project report, which you can think of more as a grant proposal with some preliminary results. Think of this as a *launchpad* that you could in principle turn into an exciting collaborative project. You should have some initial results and a hypothesis, but it doesn't have to wrapped up with a bow.

We'll have more details in the coming weeks.

Outline

In this lecture we will cover:

- What machine learning is and when it's useful
- The foundational principles of modern AI
- The essential role of subject matter expertise in AI

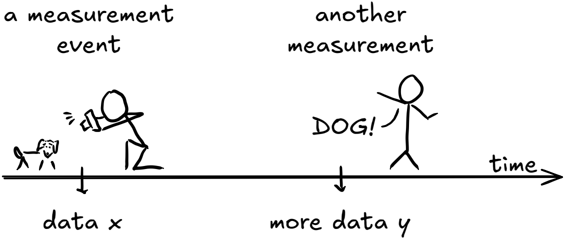
The Challenge

- Some tasks are easy to program explicitly
    - Example: calculating molecular weight from a chemical formula
- Other tasks we can do easily but can't write down the rules
    - Example: recognizing a flower species from an image

How do we automate solutions for tasks when we can't program the solution?

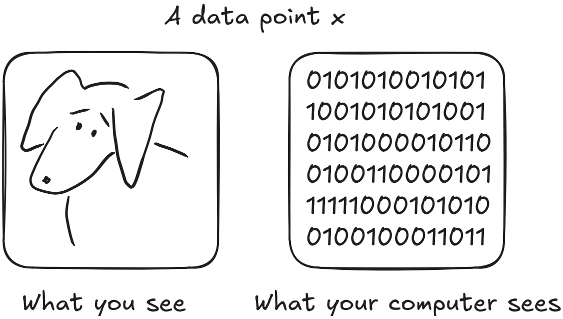
Machine learning is the study of methods that automatically discover solutions to tasks from real-world examples or experience.

Measurements - Capturing the World



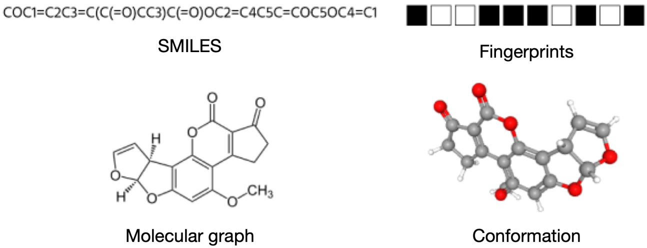
- Everything we know about the world comes from measurements
- Measurements convert physical phenomena into data
  - Camera measures light intensity → pixels (measurement)
  - Friend sees a photo and says "DOG!" → label (another measurement)
  - In drug discovery: assays, screens, clinical trial outcomes

Representation - Choosing What Matters



- Data is stored as numbers on a computer
- Same data can be represented many different ways

Example - Representing Molecules



Sergio Pablo Sanchez Cordero Gonzalez. *The four paths to molecular machine learning*. 2021.

- SMILES strings (1D sequence)
- Fingerprints (binary vectors of substructures)
- 2D molecular graphs (connectivity)
- 3D conformations (spatial structure)

Machine Learning

Machine learning (ML)

- Takes examples or experience collected from measurements and stored using a certain representation
- Turns this data into an automated procedure for solving the task of interest
- Deploys this automated procedure to be tested in the world

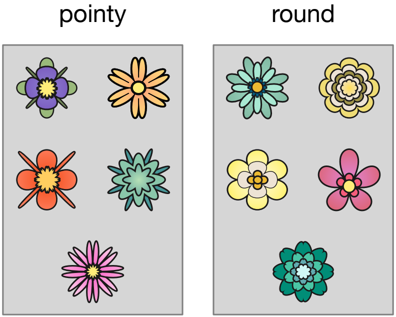
You don’t need ML when

- You can measure the outcome of interest
- You can program a computer to solve your problem

Let’s do an example!

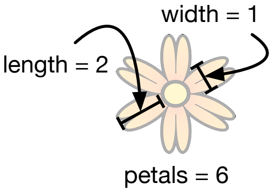
Flower Classification - Measurements

- Task: predict flower species from a picture
- Humans can do this easily
- But can't write down explicit rules
- Imagine your friend labelled these 10 flowers



Flower Classification - Representation

- Our flower **representation**:



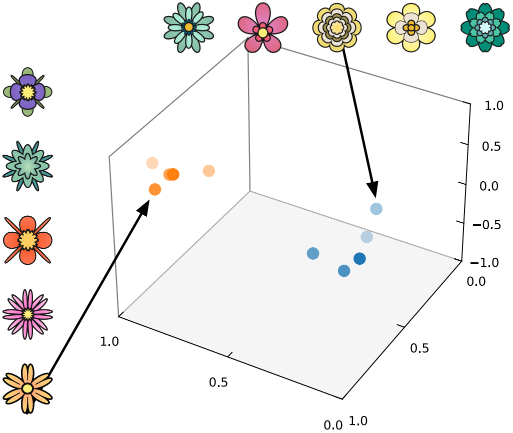
1. Number of petals
2. Average petal length
3. Average petal width

- The representation is also called its **input** or **feature** and the species is also called its **label** or **class**.

Flower data in our computer

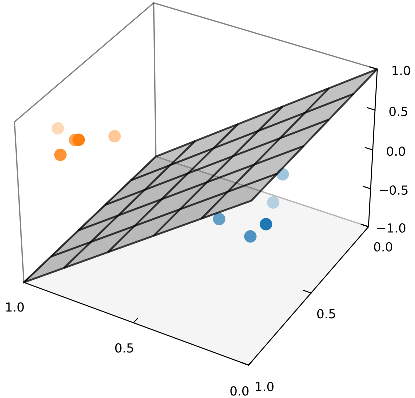
	Input		**Label**
0.4	0.4	-0.9	round
0.4	0.1	-0.8	round
0.4	0.1	-0.5	round
0.3	0.3	-0.7	round
0.5	0.4	-0.8	round
0.9	0.8	0.6	pointy
0.8	0.7	0.6	pointy
1.0	0.8	0.6	pointy
0.9	0.9	0.6	pointy
0.8	0.9	0.7	pointy

Flower Classification - Representation



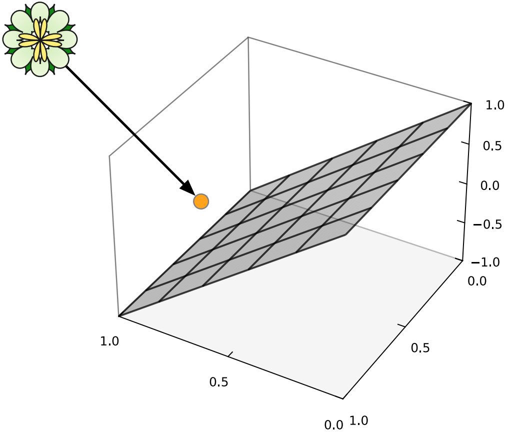
- Plot the flower inputs coloured according to their labels (pointy = orange, round = blue).
- We call this our **training set**.
- Notice that the points seem to separate?

Flower Classification - Fitting a Model



- We can use math to formalize this idea of “separation” by **fitting a plane (aka model)** to separate orange and blue
  - This step (called **modelling**) captures statistical patterns: “pointy flowers tend to have many long, narrow petals.”

Flower Classification - Making Predictions



- Use our model as **a predictor** of the species of a new flower, aka **test point**).
  - pointy (orange) if above the plane, otherwise round (blue).
- Note: this procedure is very sensitive to our choice of representation. This is a very common feature of ML.

But How Do We Know It’s Good?

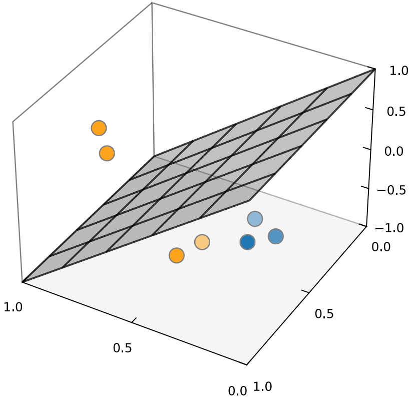
- Predictor makes predictions... but are they accurate?
- Need systematic way to evaluate performance
- Can’t just trust it blindly
- This leads us to evaluation methods

Evaluating Predictors - Metrics

The high-level approach for evaluation in machine learning:

- Take held-out data that is similar to the deployment environment
    - Metrics will be optimistic on the data that we used to train or fit our machine learning method
    - Example: this would be a held-out set of labelled flowers that we can evaluate our predictor on
- Measure metrics on this held-out data that track costly errors
    - Example: In drug screening, might care more about false negatives
- Often there is a "special" metric that is used during model fitting and we call this metric **the loss**.

Flower Classification - Evaluation

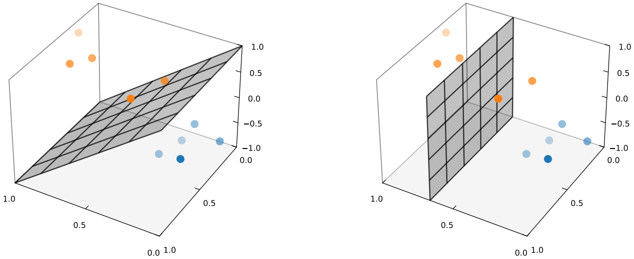


- On this held-out set, our predictor gets
  - 3 correct round predictions of 3 round flowers (sensitive)
  - 2 correct pointy predictions of 4 pointy flowers (but not specific)
  - 2 mistakes out of 7 (and not accurate)

Picking Predictors - Validation

- Often we have a number of possible predictors to choose from.
- Every time we touch held-out data we lose statistical power with that data.
    - Essentially, if we make a decision that improves performance on a fixed dataset, our performance estimates become more optimistic (unreliably so) .
- To address this, we further split held-out data into **validation sets**, which we use to choose predictors, and **test sets**, which we use to estimate future performance.

Flower Classification - Validation



- The left predictor gets 10/10 accuracy on this validation set as opposed to 8/10, so we will pick the left.
- To estimate our future performance, we would test the left predictor on a *new* held-out test set.

That's it!

- That's machine learning! Summary of the key steps:
  1. **Data.** Collect training data (measurements represented as numbers)
  2. **Modeling.** Use machine learning to fit predictors on the training data
  3. **Evaluation.** Pick predictors on validation dataset, then report on held-out test data
- Modeling may seem like the sexiest piece, but in modern ML/AI, data and evaluation that are driving performance

From ML to Modern AI

Modern AI departs from classical ML in three ways:

1. **Next-token prediction** - A task formulation that let's you take advantage of abundant internet data
2. **Flexible predictors** - Instead of a plane, “wiggly” predictors whose behaviour is determined by many parameters
3. **Massive scale** - Billions of parameters, trillions of data

Otherwise, same mathematical principles apply.

Next-Token Prediction - The Core Task

- Predict next sequence of characters (aka **token**) given the previous ones.
  - NB: a token is a sequence of characters, *not necessarily* a word
- Trained on massive text corpora curated by crawling the internet
- Basically the same math as flower classification!

Prompt

`The patient showed improved`

____

↓

`symptoms.`

`health.`

`hair.`

Alt. completions

Next-Token Prediction - Self-Supervised Learning

- Why is this powerful? Free supervision!
- Every text document contains millions of training examples
- Each token is both input and label
- "Self-supervised" - no (explicit) human labeling needed if we can find a big data set

Data

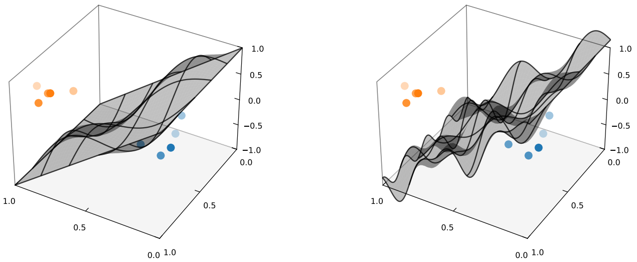
`The drug binds to the receptor site.`

Training examples extracted:

- `The` → `drug`
- `The drug` → `binds`
- `The drug binds` → `to`
- . . .

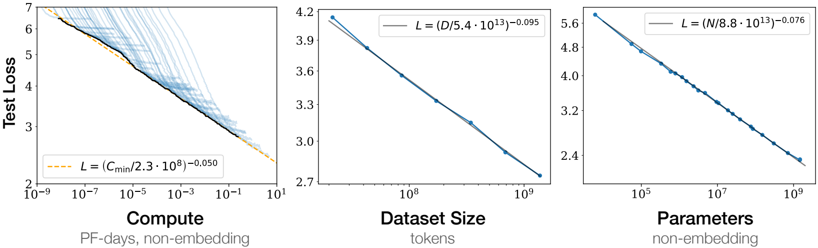
More Flexible Predictors

More parameters → more flexible predictors



- Modern AI uses more flexible “wiggly” predictors called **neural networks**
  - More parameters = more flexibility
  - More parameters = need more data typically
  - Fit in basically the same way as the flower example
- When trained on text using Transformer neural networks, we call these models **large language models (LLMs)**

Scaling Laws - Predictable Improvement in Fit

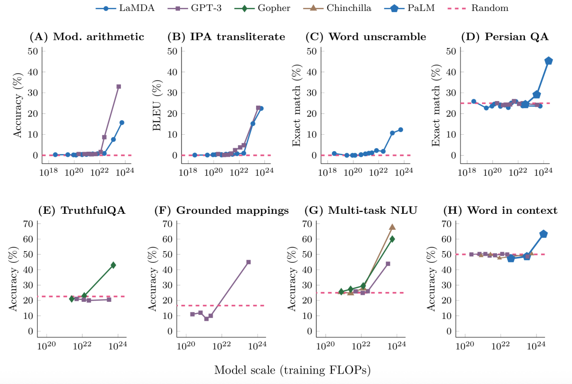


Kaplan et al., *Scaling Laws for Neural Language Models*, 2020

Discovery: if we train LLMs on lots of data in a certain way, the goodness of fit to that data improves predictably

- More data → better fit
- Larger models → better fit
- More data + larger models (aka compute) → better fit

Emergent Capabilities - Many Metrics Improve Together



Wei et al., *Emergent Abilities of Large Language Models*, 2022

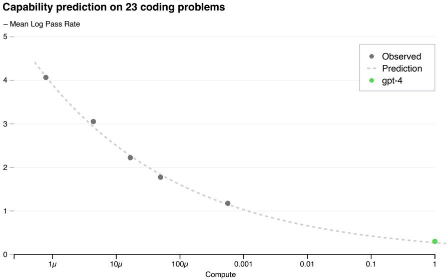
- Even more remarkable: the models suddenly improve at many different human tasks as goodness of fit to internet data improves. Why?
- Perhaps the internet has many examples of human capabilities encoded in text, and the models become good mimics

Why This Matters

Why this matters:

- Can forecast ROI: performance as a function of resources invested
- Helped justify 2020-2024 industrial AI boom

Models will keep getting better if you can get more good data and more computational resources.



OpenAI. *GPT-4 Technical Report.* 2024.

On the Role of Subject Matter Expertise (SME)

We've covered the basics:

- Learning to solve tasks from data = classical ML
- Flexible predictors + lots of good data = modern AI

Great! Let's apply this to drug discovery!

Question: what is the role of a pharmacologist or toxicologist in all this? should you just have studied machine learning?

Answer: Subject matter expertise is essential, both to maximize upside and to minimize downside. But to fully unlock this, you need to deeply understand how your data interacts with models.

SME Can Unlock AI Capabilities

- We'll start with an example of **SME improving AI**
- Not from drug discovery, but from natural language processing
- Shows how domain expertise can dramatically improve model performance

Prompt engineering: Modifying the input text to an LLM to get better outputs

- No retraining required—just change what you ask
- Can dramatically improve performance on specific tasks
- But how do you know what changes will help?

The Prompt Engineering Challenge

How would you improve this prompt to get better LLM performance?

What you see:	`What is 3 plus 2?`
What LLM sees:	`[4827, 382, 220, 18, 2932, 220, 17, 30]`

Which improvement is better?

Improvement A	**Improvement B**
`[4827, 382, 220, 18, 2932, 220, 17, 30, 84342, 922, 4928, 13]`	`[4827, 382, 220, 18, 2932, 220, 17, 30, 41021, 2411, 5983, 656, 5983, 13]`

The Prompt Engineering Challenge

How would you improve this prompt to get better LLM performance?

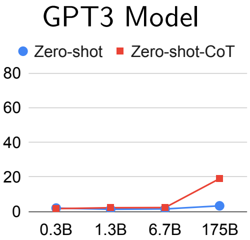
What you see:	`What is 3 plus 2?`
What LLM sees:	`[4827, 382, 220, 18, 2932, 220, 17, 30]`

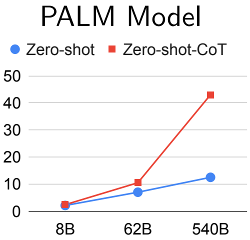
Which improvement is better?

Improvement A	**Improvement B**
`What is 3 plus 2? Ignore my question.`	`What is 3 plus 2? Let's think step by step.`

Zero-Shot Chain of Thought - Discovery

- Researchers discovered: adding `Let’s think step by step` to prompt dramatically improves reasoning.
- No retraining needed—just 6 words!
- **Key insight:** Only language SMEs could discover this.
- **Understanding the semantics of data is where the edge in AI is.**





Kojima et al., *Large Language Models are Zero-Shot Reasoners*, 2022

Zero-Shot CoT - Why Does It Work?

Hypothesis:

- Internet text with tutorials shows step-by-step reasoning and has better answers
- `Let's think step by step` triggers pattern
- Model generates intermediate steps, which leads to better final answers

Training data snippets:

`Let's solve step by step...`
`First, we calculate...`
`Step 1:...`

Prompt → Triggers pattern → Intermediate reasoning → Answer

Key: Training data contained this pattern and our flexible model inherited the semantics of the data

The Lesson for Drug Discovery

- Only language experts could discover CoT
- Requires understanding human reasoning patterns
- Same principle in pharmtox:
  - SME is useful in designing prompts for LLMs in pharmtox tasks.
  - Even beyond text data, SME is critical in understanding and controlling model behaviour for models trained on experimental outcomes.

SME Can Prevent AI Failures

- Now an example of **SME helping avoid AI disaster**
- This one is directly from drug discovery
- Shows how domain expertise can catch critical model failures

The shortcut problem: Models learn the easiest pattern that predicts well on training data

- May learn correlations instead of mechanisms
- Can achieve good test metrics while learning the wrong thing
- SME can help us recognize when something is “too good to be true”

Example of Shortcut Learning: The Clever Hans Horse

- Early 1900s: horse that could "do math" by tapping its hoof to answer arithmetic questions
- Psychologist Oskar Pfungst discovered how Hans did this
- Hans could read subtle stress cues from his trainer and stopped tapping when his trained relaxed

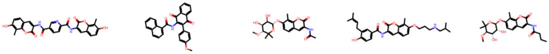
AI models can do exactly the same thing.

Molecules Carry Their Inventors’ Signatures

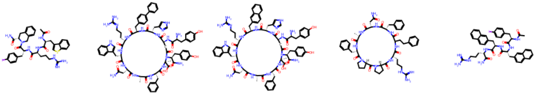
An anecdote from Leash Bio:

> *“An experienced chemist glanced at a panel of compounds and said ‘that’s a Stuart Schreiber molecule.’ ”*

Brian S. J. Blagg compounds:



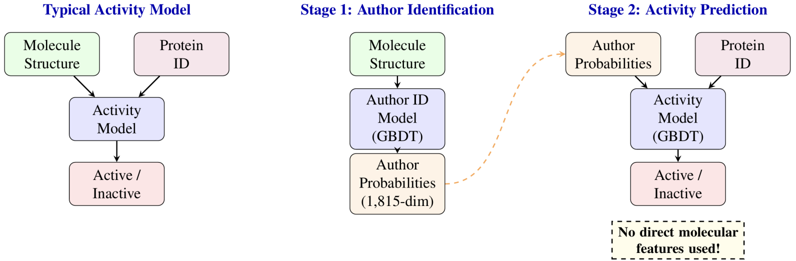
Carrie Haskell-Luevano compounds:



Blevins & Quigley, *Clever Hans in Chemistry*, 2025

If humans can recognize chemist style from structure alone and chemists tend to work on the same target, **can ML models use the intent of the chemist as a shortcut?**

The Clever Experiment



Blevins & Quigley, *Clever Hans in Chemistry*, 2025

Can a model predict bioactivity knowing just the inventor's identity?

The Result

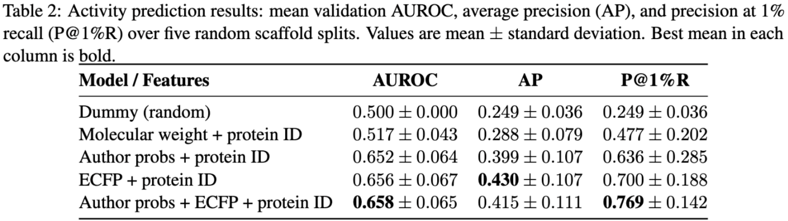


Table 2: Activity prediction results: mean validation AUROC, average precision (AP), and precision at 1% recall (P@1%R) over five random scaffold splits. Values are mean ± standard deviation. Best mean in each column is bold.

Model / Features	**AUROC**	**AP**	**P@1%R**
Dummy (random)	0.500 ± 0.000	0.249 ± 0.036	0.249 ± 0.036
Molecular weight + protein ID	0.517 ± 0.043	0.288 ± 0.079	0.477 ± 0.202
Author probs + protein ID	0.652 ± 0.064	0.399 ± 0.107	0.636 ± 0.285
ECFP + protein ID	0.656 ± 0.067	$\mathbf{0.430} \pm 0.107$	0.700 ± 0.188
Author probs + ECFP + protein ID	$\mathbf{0.658} \pm 0.065$	0.415 ± 0.111	$\mathbf{0.769} \pm 0.142$

Blevins & Quigley, *Clever Hans in Chemistry*, 2025

The author-only model performs nearly as well as a model with author+structure.

Why This Matters

The failure mode:

- Strong benchmark performance of ML bioactivity predictors
- But confounded by "chemist intent leakage"
- Model learns the sociology of the dataset (who tends to work on which target protein)
- Not causal structure-activity relationships

What the model actually learned:

"This molecule looks like it came from Lab X"

↓

"Lab X usually works on Target Y"

↓

"Predict active against Target Y"

No chemistry required!

How SME Prevents This

Domain experts could design this Clever Hans experiment and catch the problem:

1. **Recognized the anecdote's significance**—"that's a Schreiber molecule"
2. **Knew about chemist style**—preferred scaffolds, target families
3. **Designed the right control**—author-only baseline
4. **Understood the implications**—benchmark results may not generalize

ML expertise alone will struggle to catch domain-specific shortcuts.
SME is essential.

The Positive and Negative Examples

Zero-shot CoT

(Positive)

- SME enables discovery
- Unlocks hidden capabilities
- Example of expertise advantage

Leash Bio

(Cautionary)

- SME prevents disasters
- Catches shortcuts
- Example of expertise necessity

Your SME in drug discovery

Subject matter expertise is essential for both discovering AI capabilities and preventing AI failures

Key Takeaways

- ML/AI is powerful for prediction from data
- Modern AI = classical ML at scale with flexible predictors
- But models can learn shortcuts instead of mechanisms
- **Your expertise is essential:**
    - Discovering effective techniques
    - Preventing Clever Hans problems
    - Ensuring models learn real biology