Future of Probability

Chris Piech + Jerry Cain

2021 Abel Prize (~Nobel Prize Math)

Two Computer Scientists

One winner Avi Wigderson
Major theme: randomness in computer science

Eg Zero knowledge proof

https://www.nytimes.com/2021/03/17/science/abel-prize-mathematics.html

Abel Prize Chair: “This prize is on the applied side, toward computer science,” Dr. Munthe-Kaas said. “But it’s deep mathematics.”

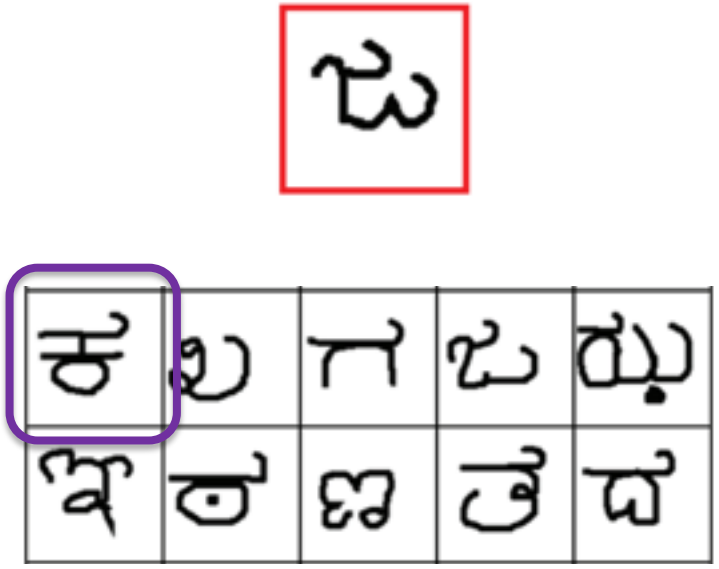
Open Problem: One Shot Learning

B Lake, R Salakhutdinov, J Tenenbaum. Science 2015.
Human-level concept learning through probabilistic program induction.

Current deep learning methods are not enough to move the needle as far as we want, **especially on socially relevant problems** that often do not have the benefit of massive public datasets. The best new ideas are coming from probability theory

Open Problem: One Shot Learning

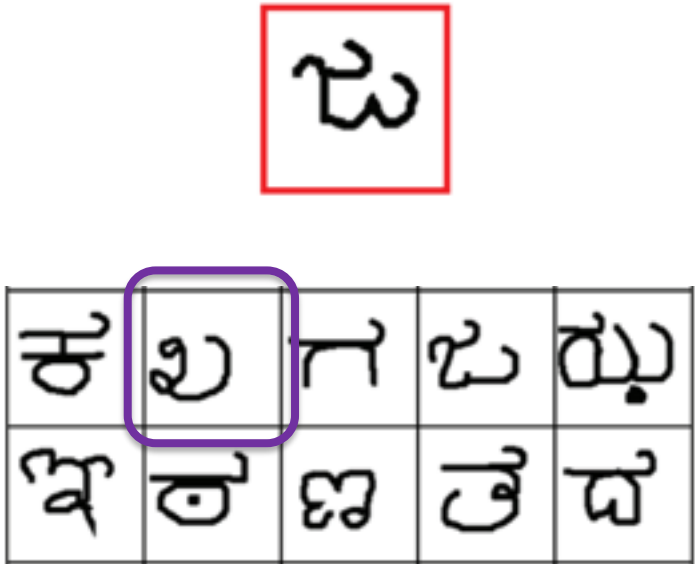
B Lake, R Salakhutdinov, J Tenenbaum. Science 2015.
Human-level concept learning through probabilistic program induction.



Current deep learning methods are not enough to move the needle as far as we want, **especially on socially relevant problems** that often do not have the benefit of massive public datasets. The best new ideas are coming from probability theory

Open Problem: One Shot Learning

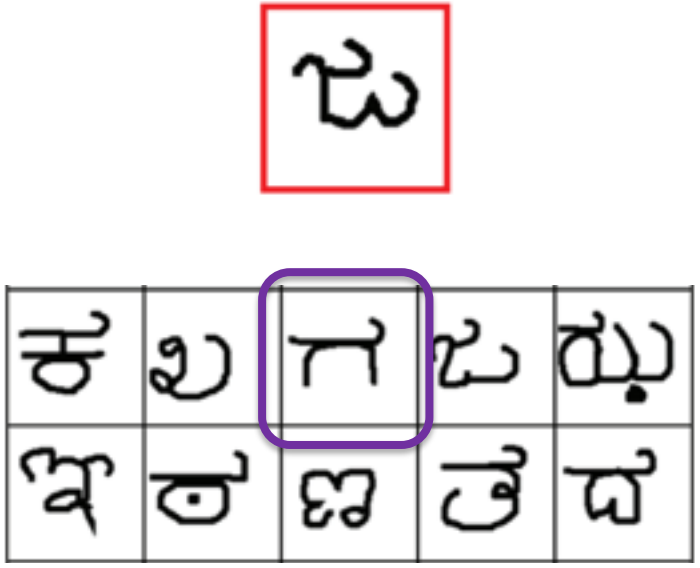
B Lake, R Salakhutdinov, J Tenenbaum. Science 2015.
Human-level concept learning through probabilistic program induction.



Current deep learning methods are not enough to move the needle as far as we want, **especially on socially relevant problems** that often do not have the benefit of massive public datasets. The best new ideas are coming from probability theory

Open Problem: One Shot Learning

B Lake, R Salakhutdinov, J Tenenbaum. Science 2015.
Human-level concept learning through probabilistic program induction.



Current deep learning methods are not enough to move the needle as far as we want, **especially on socially relevant problems** that often do not have the benefit of massive public datasets. The best new ideas are coming from probability theory

Open Problem: One Shot Learning

B Lake, R Salakhutdinov, J Tenenbaum. Science 2015.
Human-level concept learning through probabilistic program induction.

Current deep learning methods are not enough to move the needle as far as we want, **especially on socially relevant problems** that often do not have the benefit of massive public datasets. The best new ideas are coming from probability theory

Today

Probabilities digital future

Let me tell you a story about one particular problem.

It will give you a sense of what it takes to do research in **applied probability**, and our desire to solve the problems, will lead us to deep **theoretical** challenges in modern AI.

Application -> **Theory**

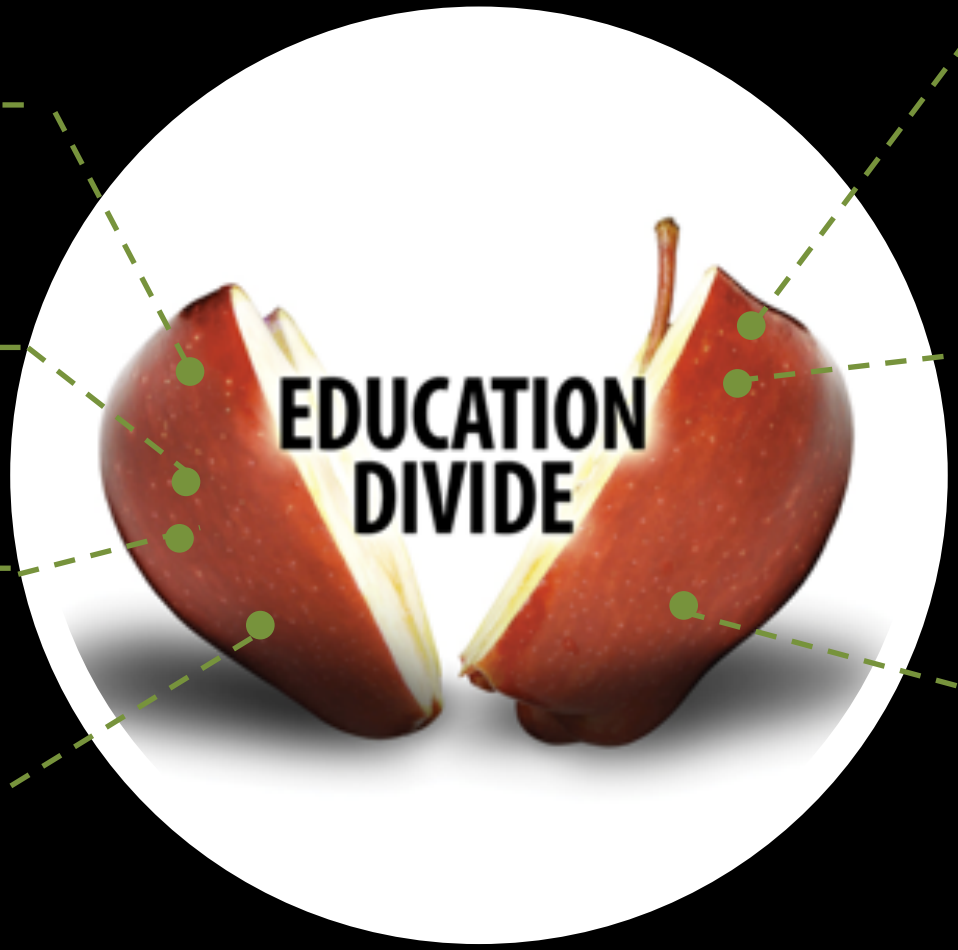
Quality Education Gap

90% of children enroll in primary education [1]

40% in secondary education [1]

20% in tertiary education [1]

Dramatic quality differences



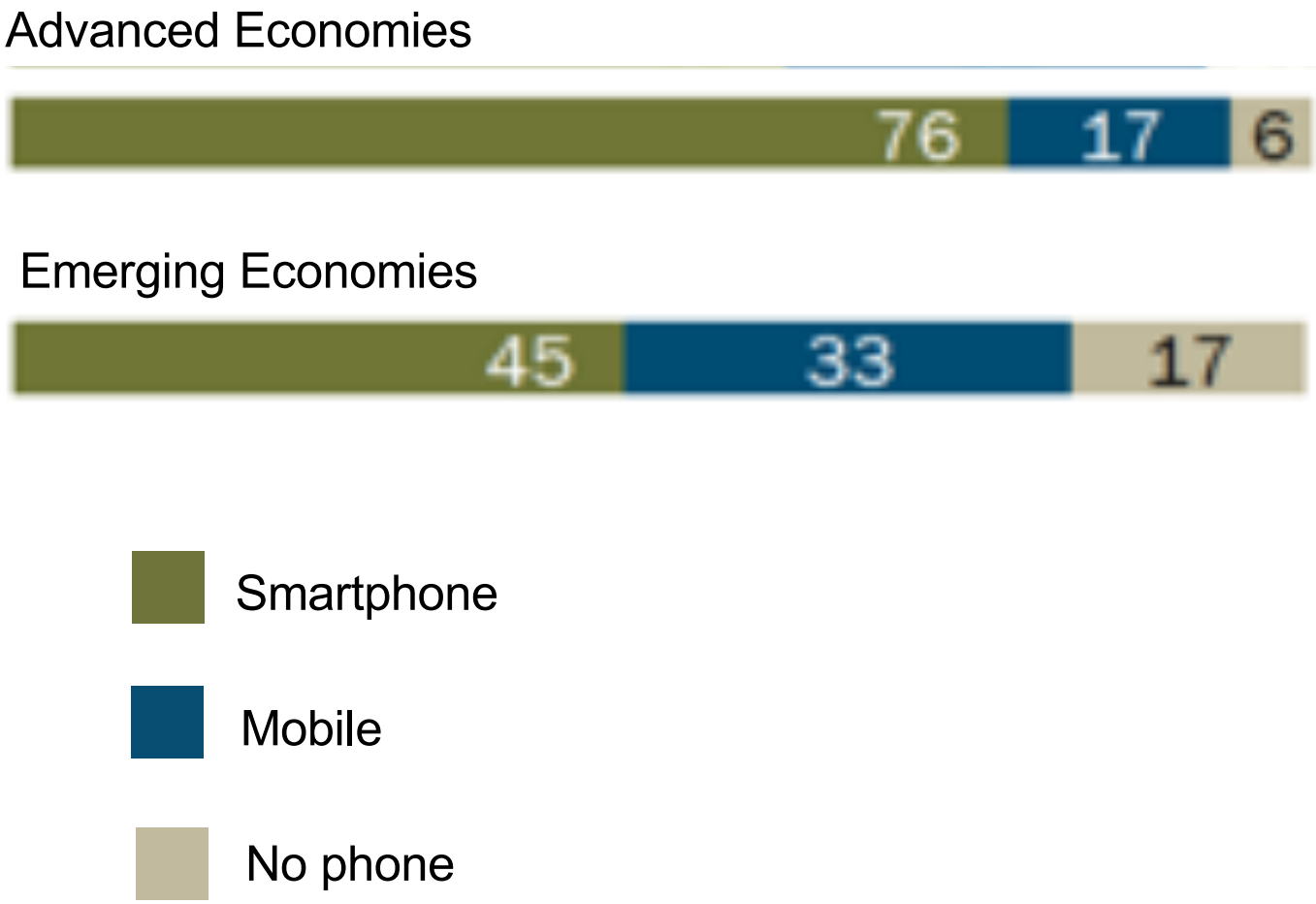
375 million workers need to be retrained by 2030 [3]

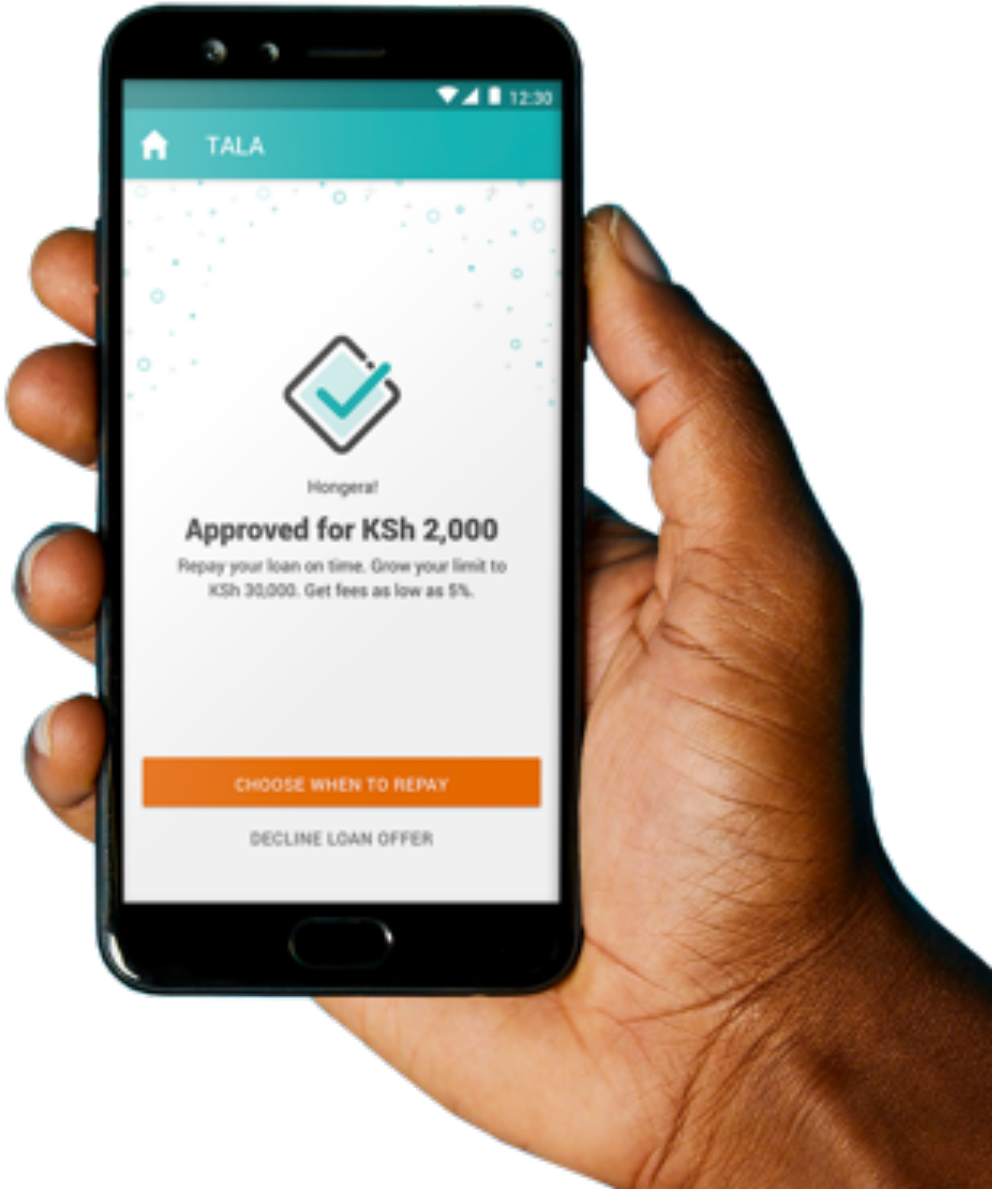
Half a million unfilled computer science jobs (60% of STEM jobs) [2]

For all learners we want *quality*.

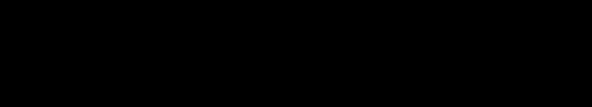
[1] *World development indicators 2015*. World Bank Publications, 2015.
[2] USA Bureau of Labor Statistics Employment Projections , 2016.
[3] Jobs lost, jobs gained. McKinsy Global Institute, 2017.

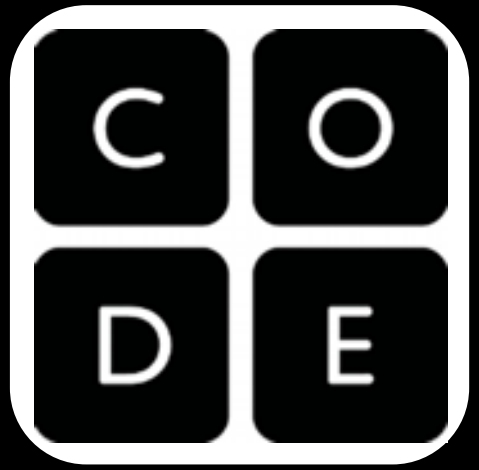
Smart Phone Access





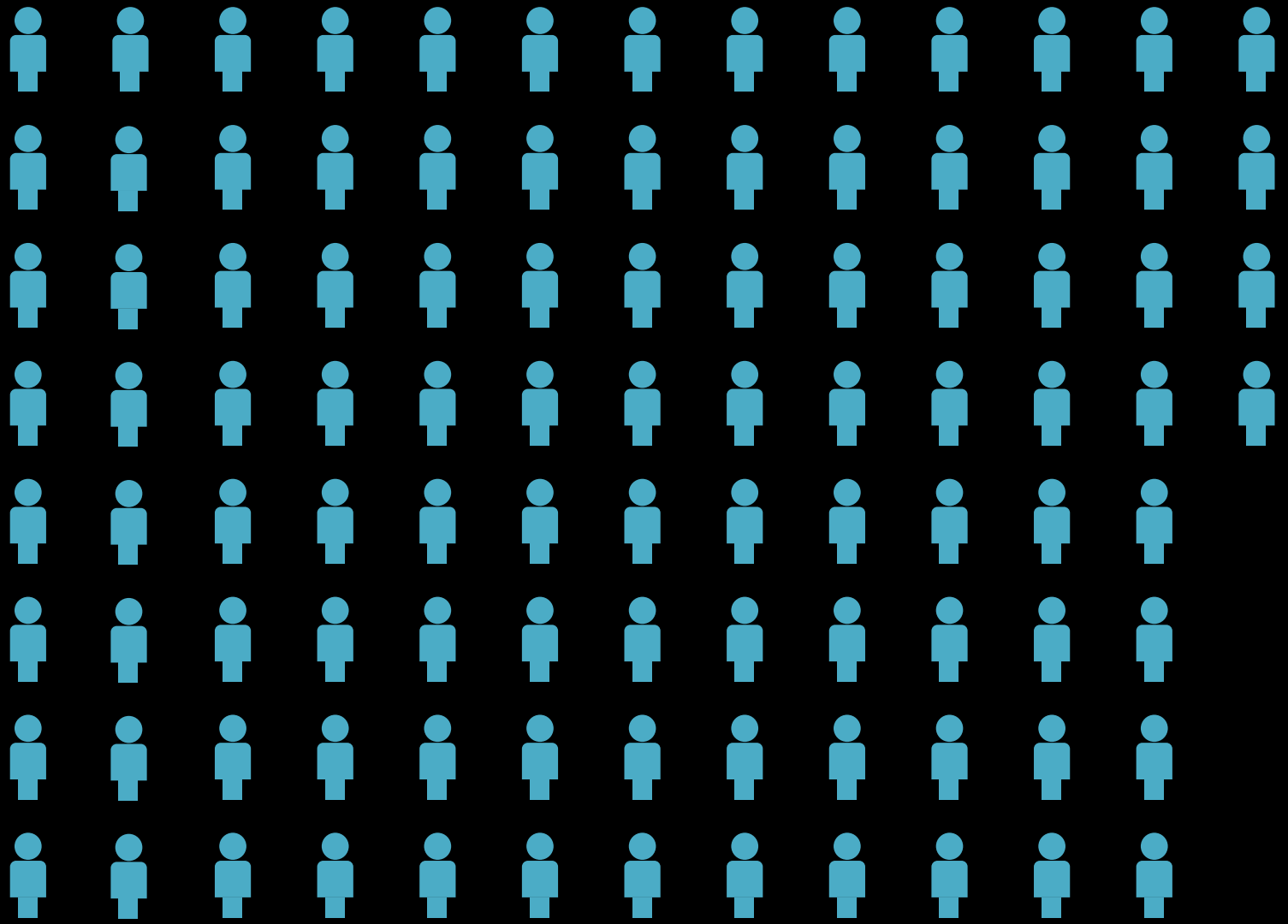
Unprecedented Data





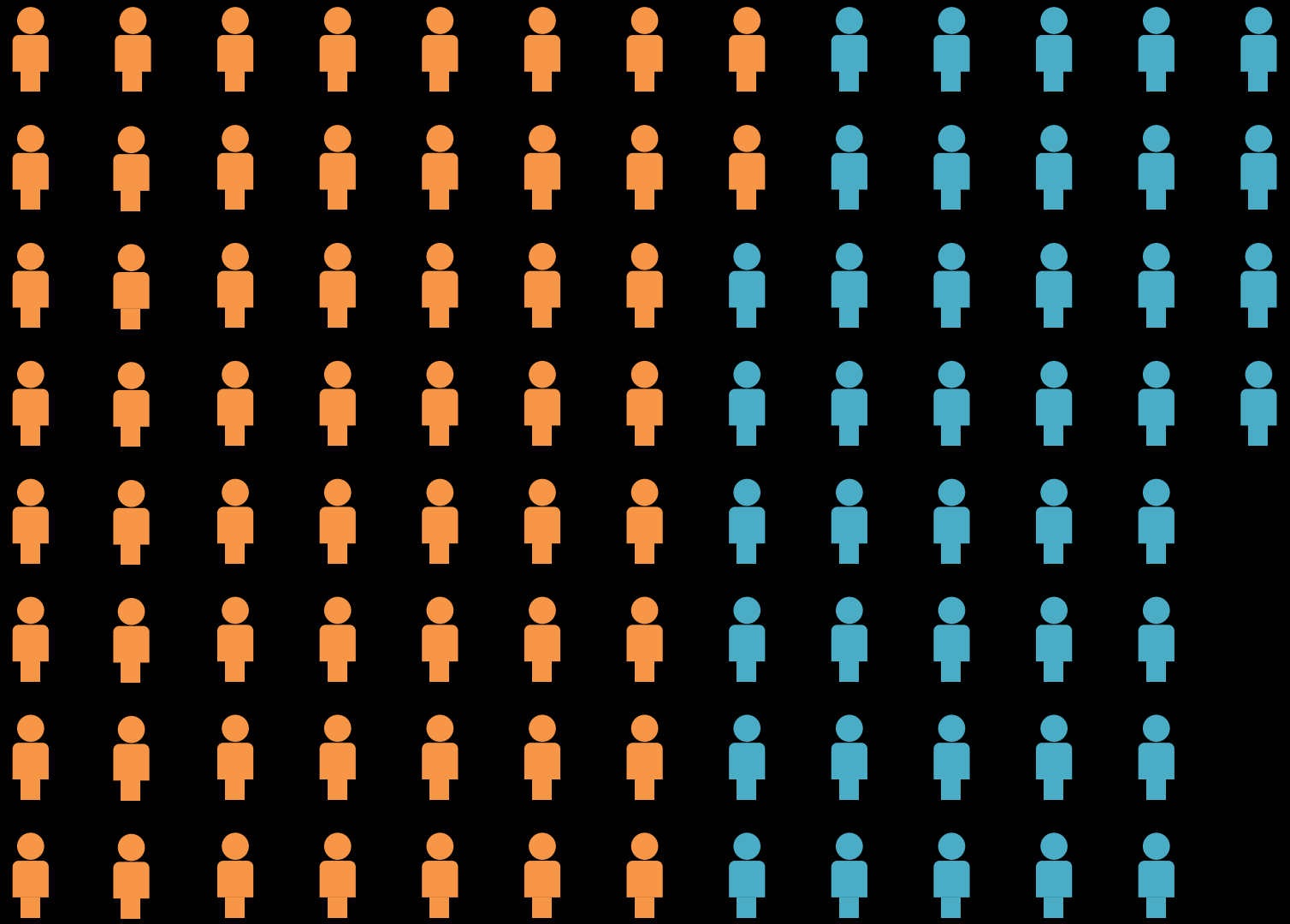
Over 50 million learners

US K-12 Students



= 500,000 learners

Code.org Students



= 500,000 learners

Code.org by the numbers

1,234,127 teachers

42M unique enrolled students

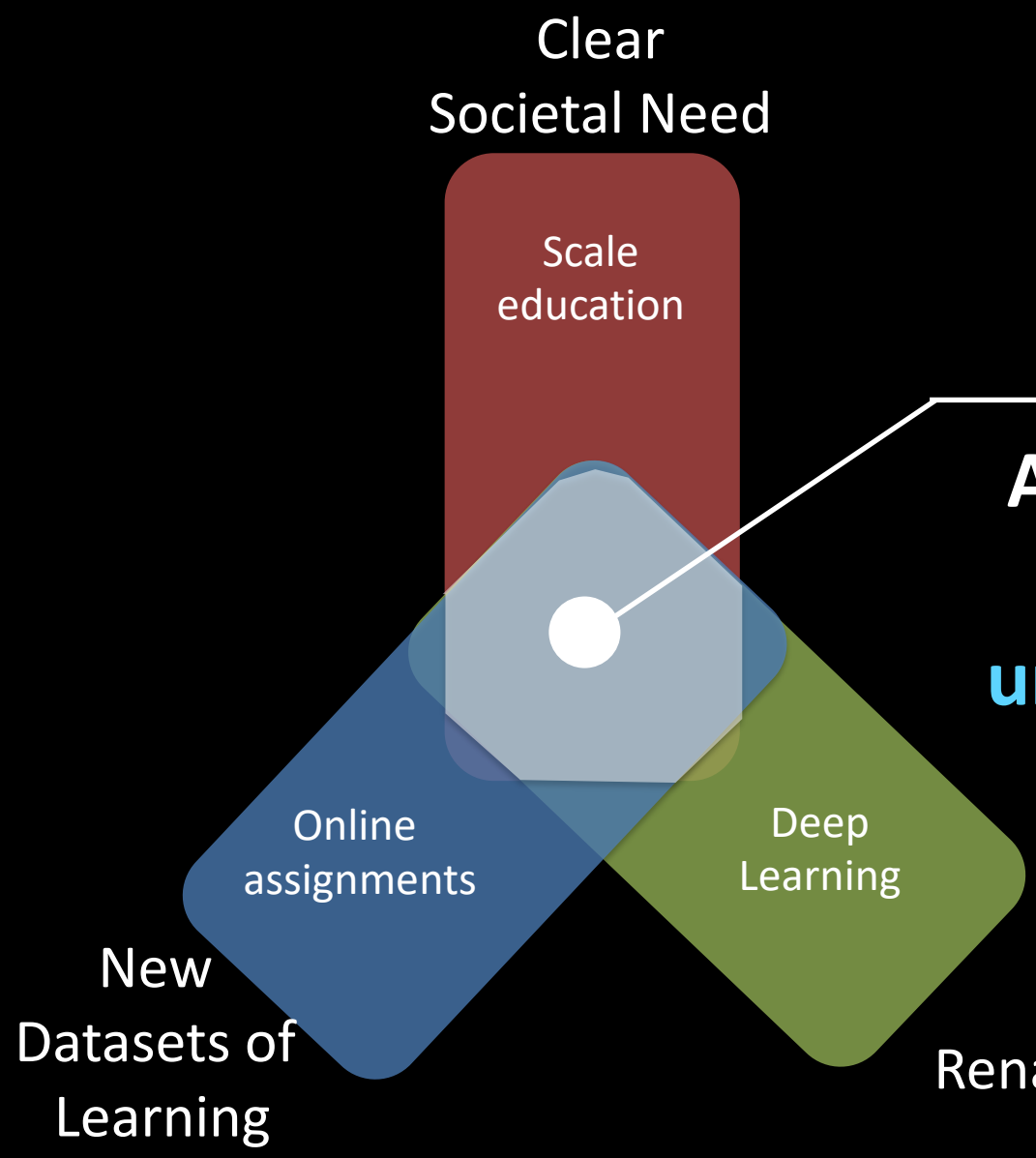
Used in **180+** countries

832M hour of code sessions

4 papers publish with our lab

50M K12 students in the US

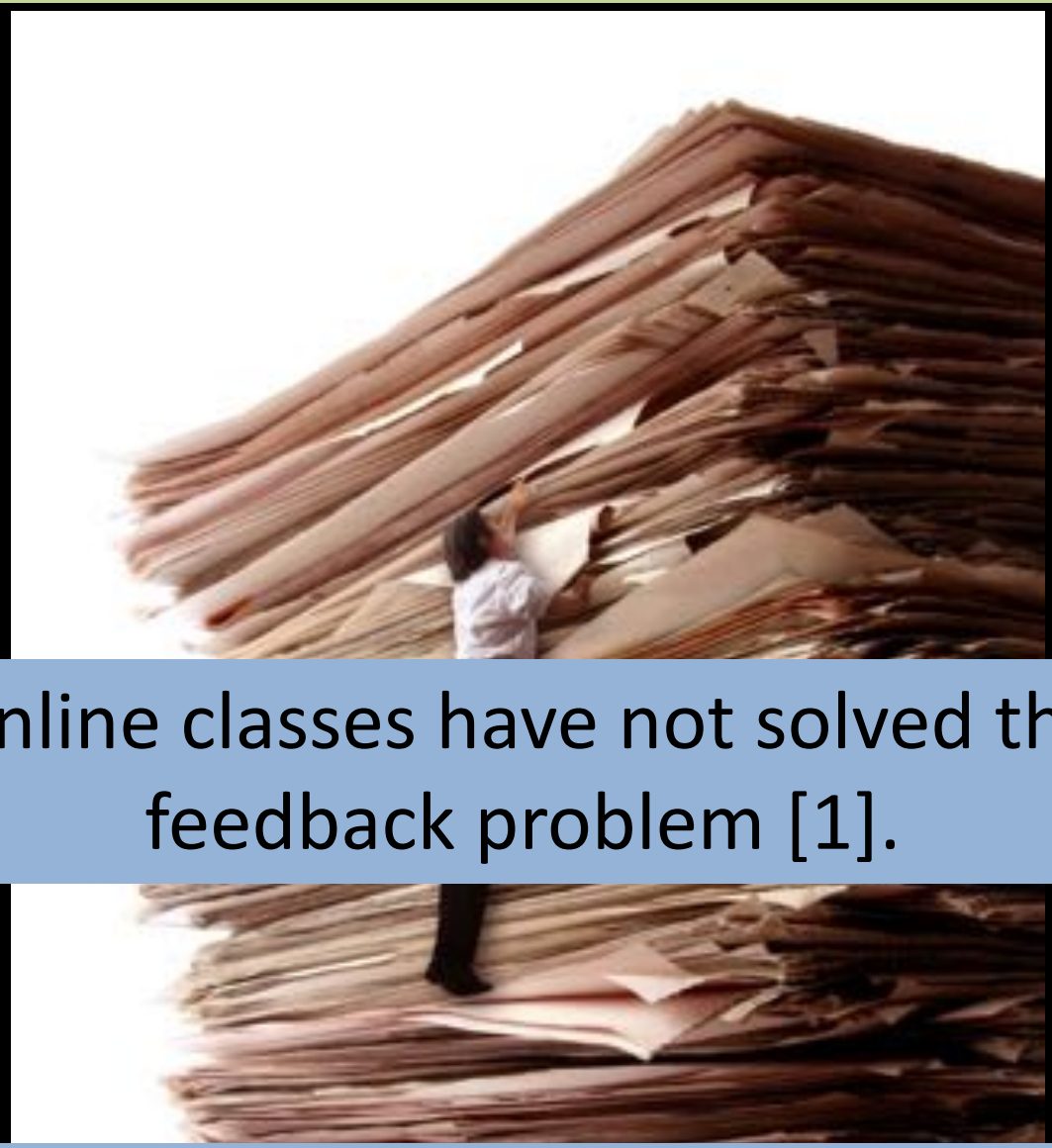
Speech Recognition



Grand Challenge
in Education

Autonomously support education by better understanding students.

Feedback is Labor Intensive

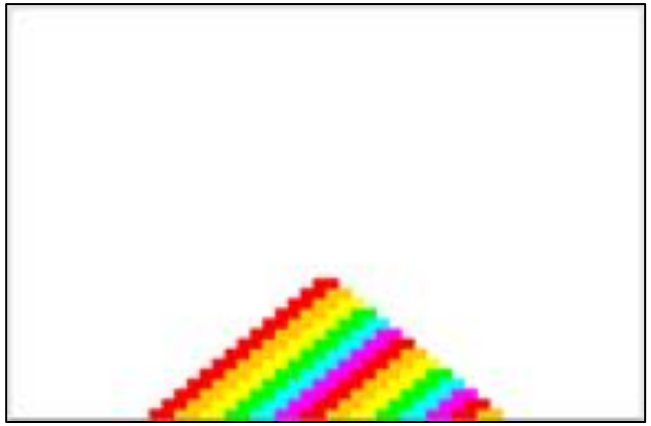


Online classes have not solved the feedback problem [1].

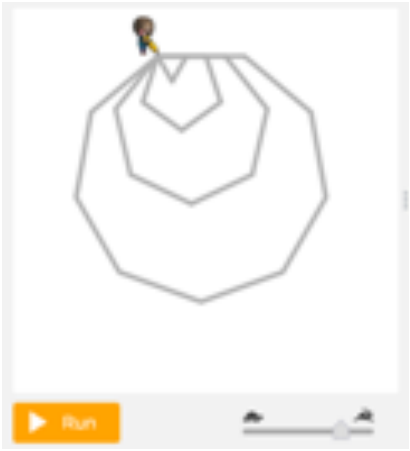
[1] Deconstructing Disengagement. Analysing learner subpopulations in MOOCs. Kizilcec, Piech, Schneider. Over 600 citations since 2013

Many domains of student work

$$
\begin{aligned}
&9 \times 6 \\
&= (10 - \square) \times 6 \\
&= 10 \times 6 - \square \times 6 \\
&= 60 - \square \\
&= \square
\end{aligned}
$$



Why did the original pilgrims come to America?



Chapter 0: Always start simple

First deep learning for education

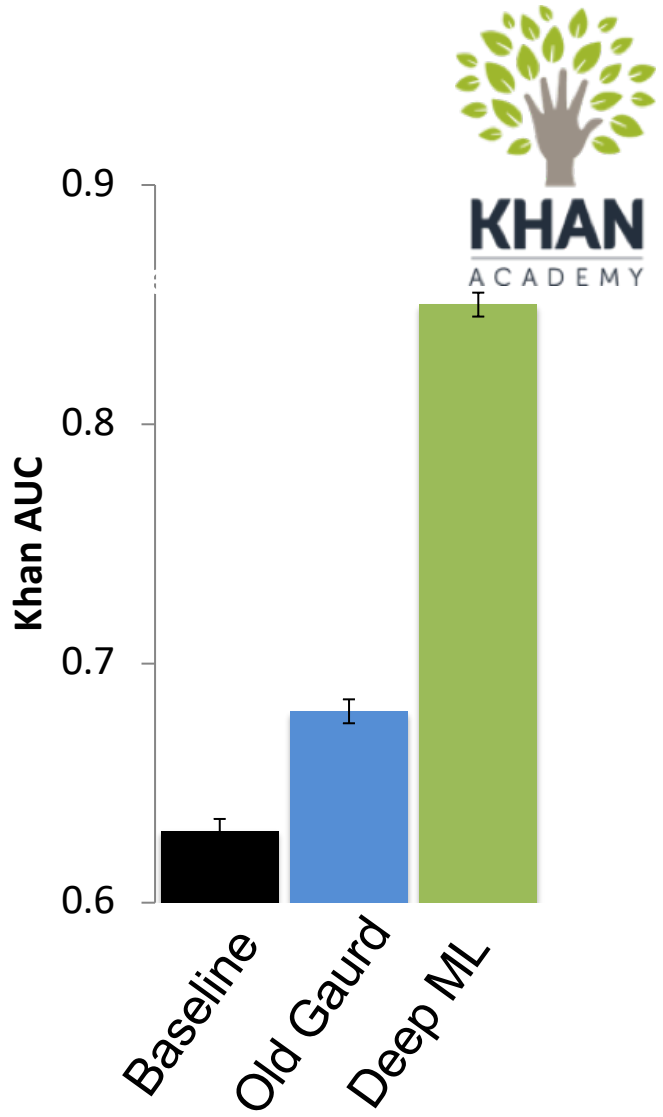
KHAN
Student

Exercise Type:

- Solving for x-intercept
- Solving for y-intercept
- Graphing linear equations
- Square roots
- Slope of a line

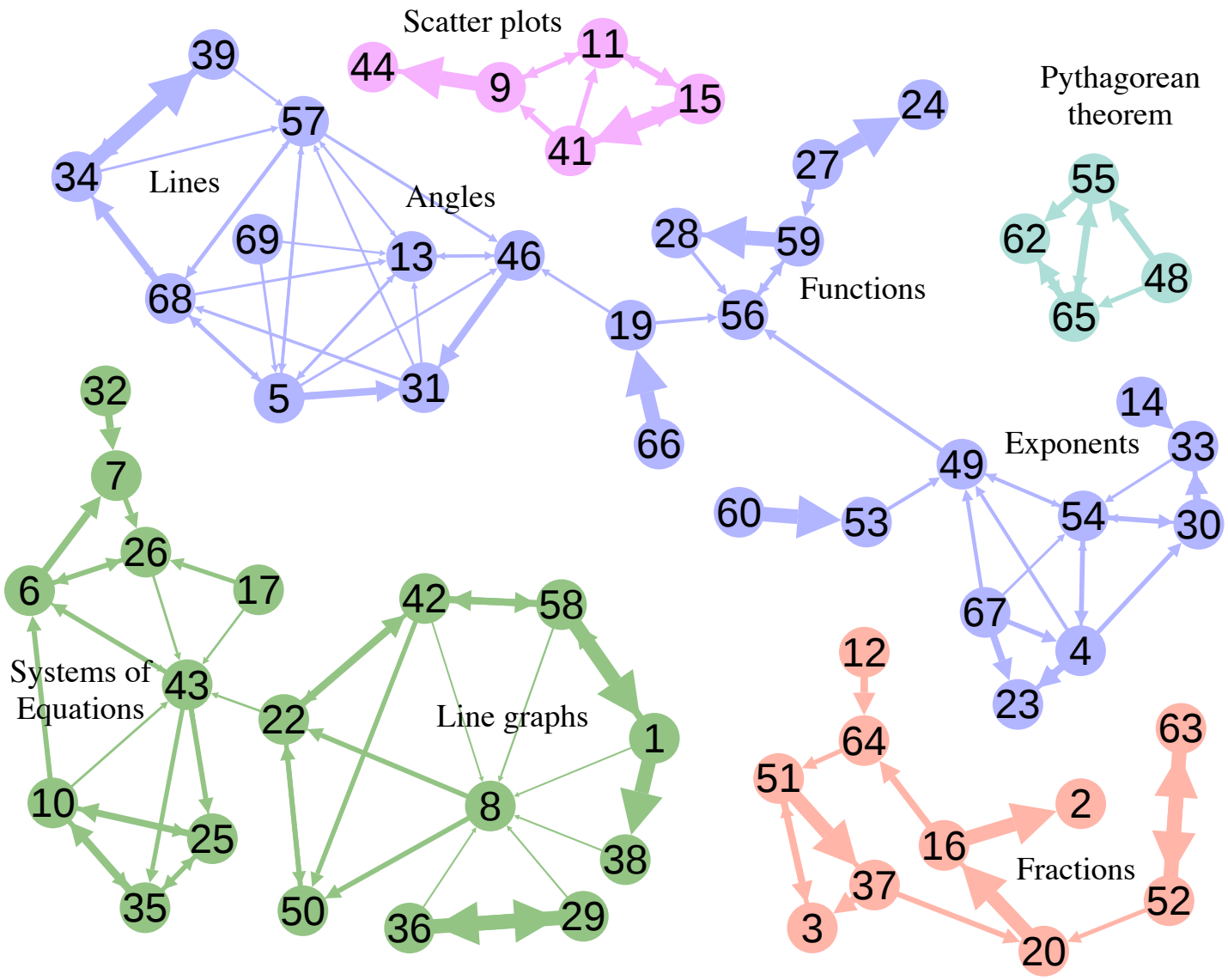
Answer:

- Correct
- Incorrect

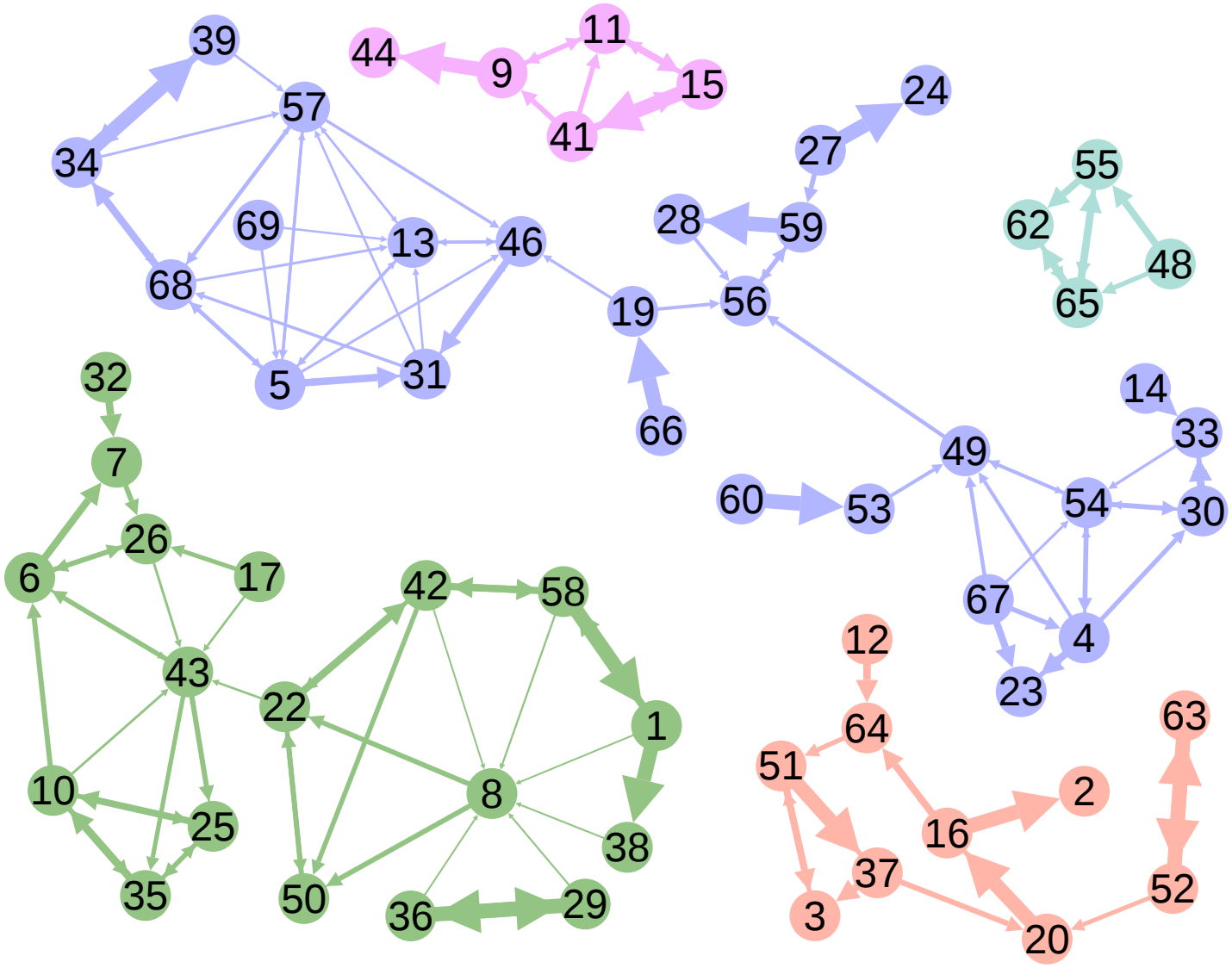


Old Problem

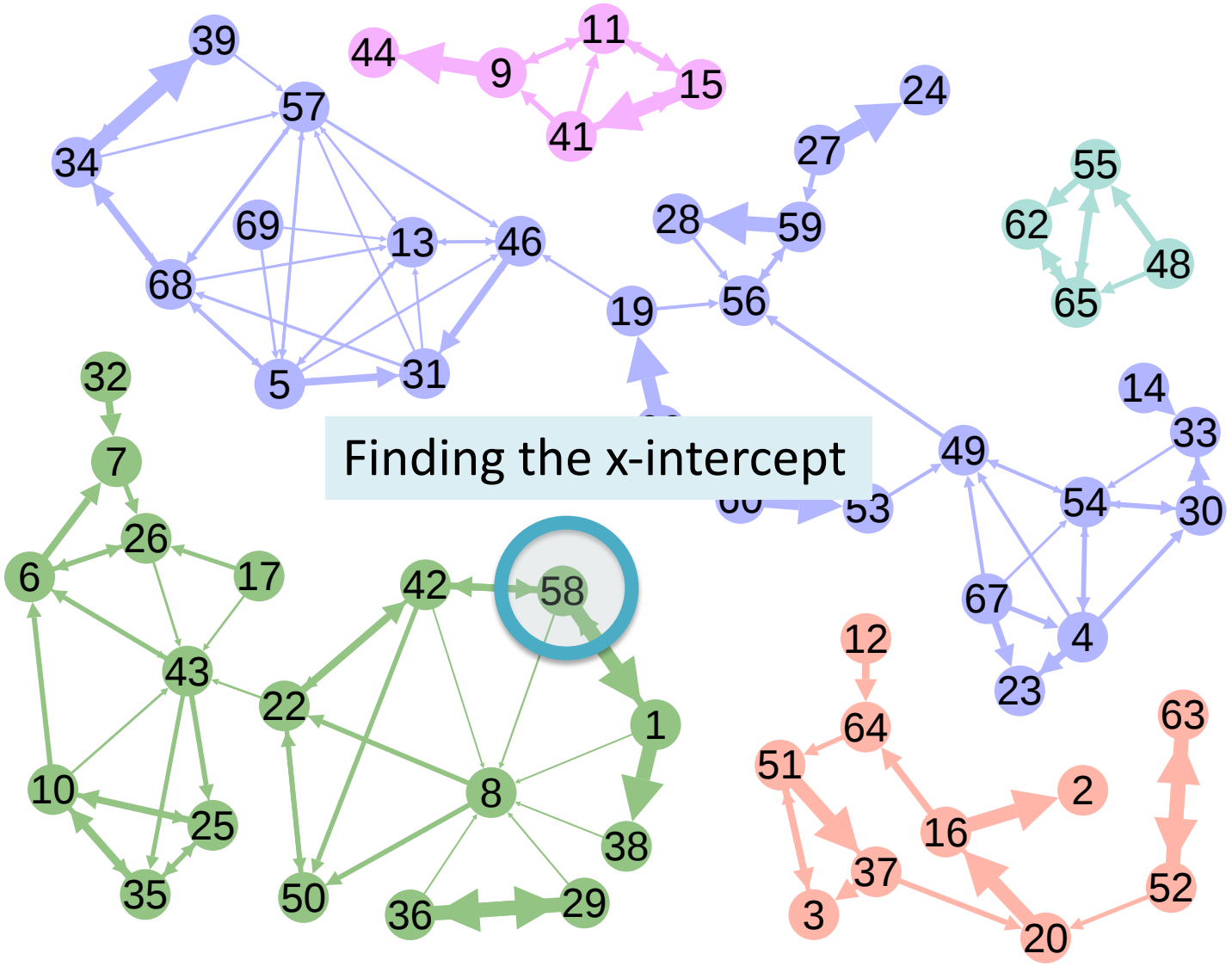
Learns Concept Relationships



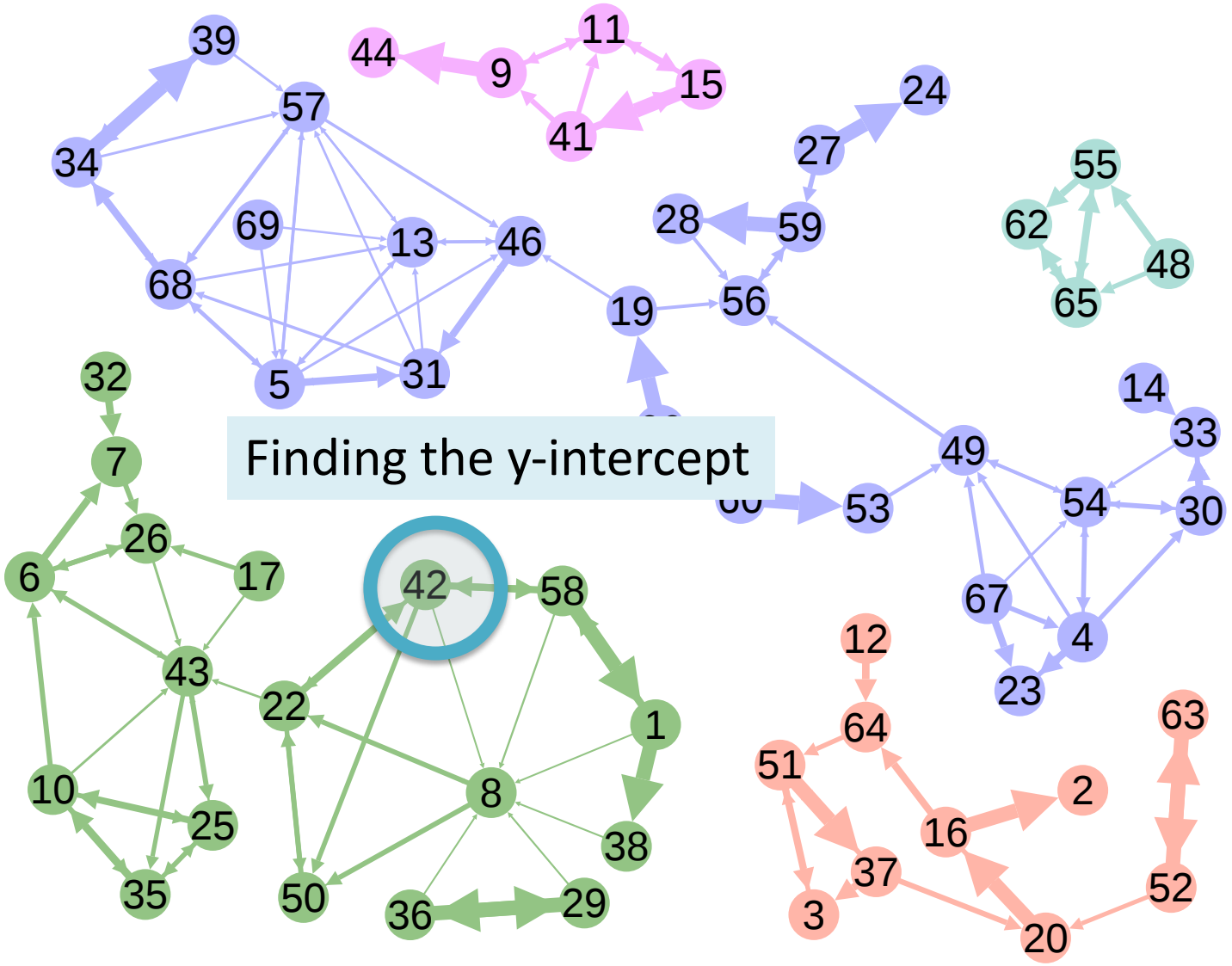
Learns Concept Relationships



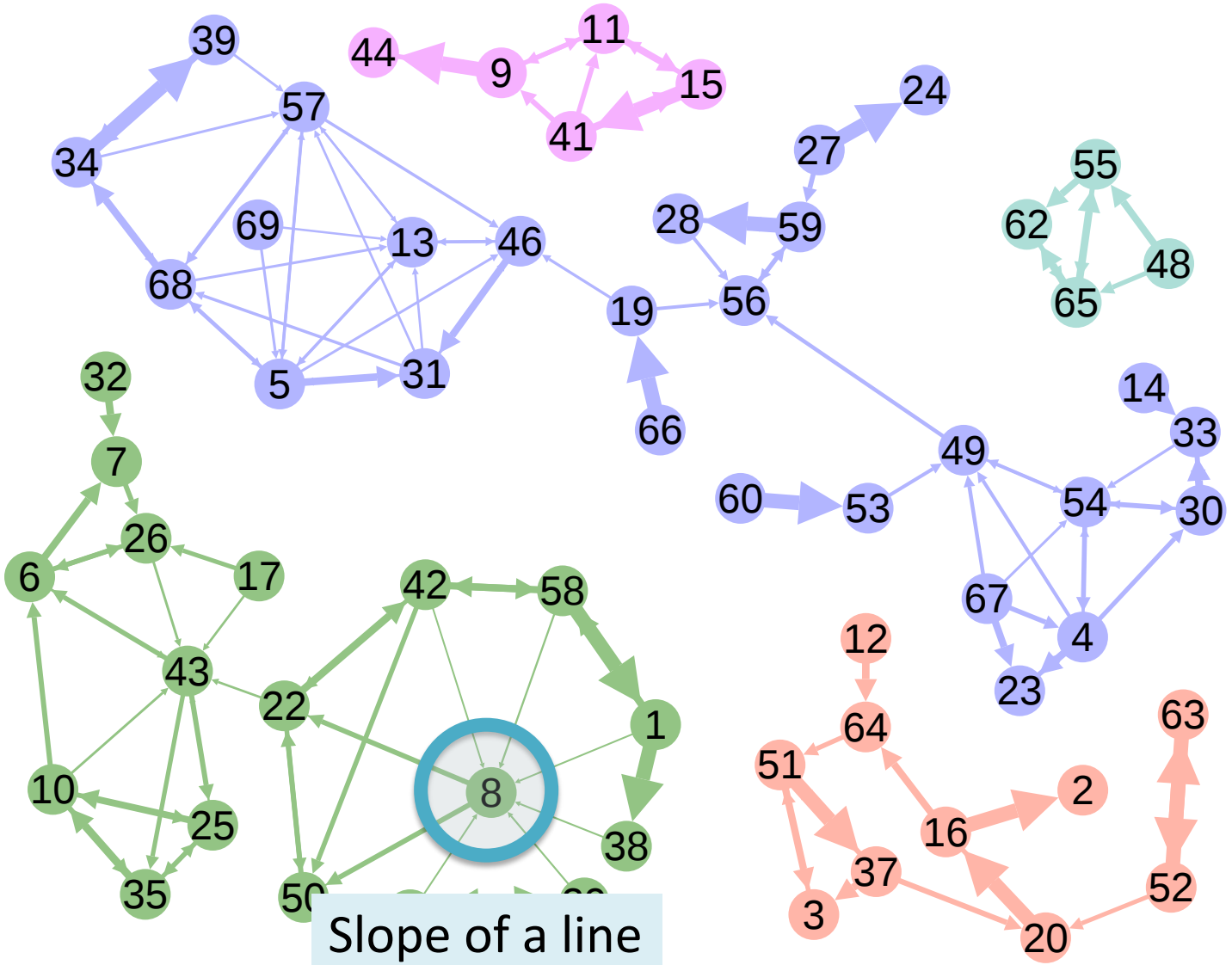
Learns Concept Relationships



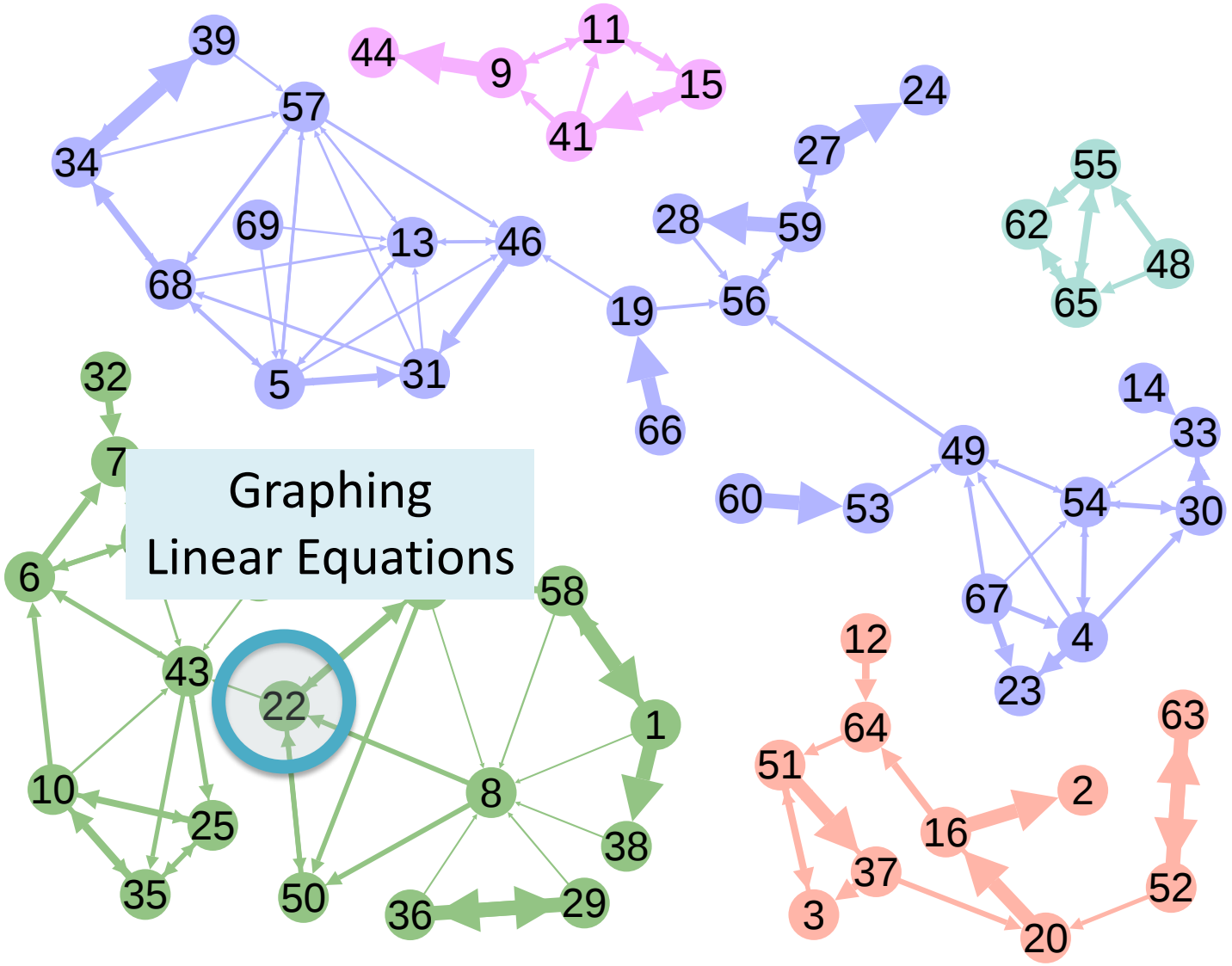
Learns Concept Relationships



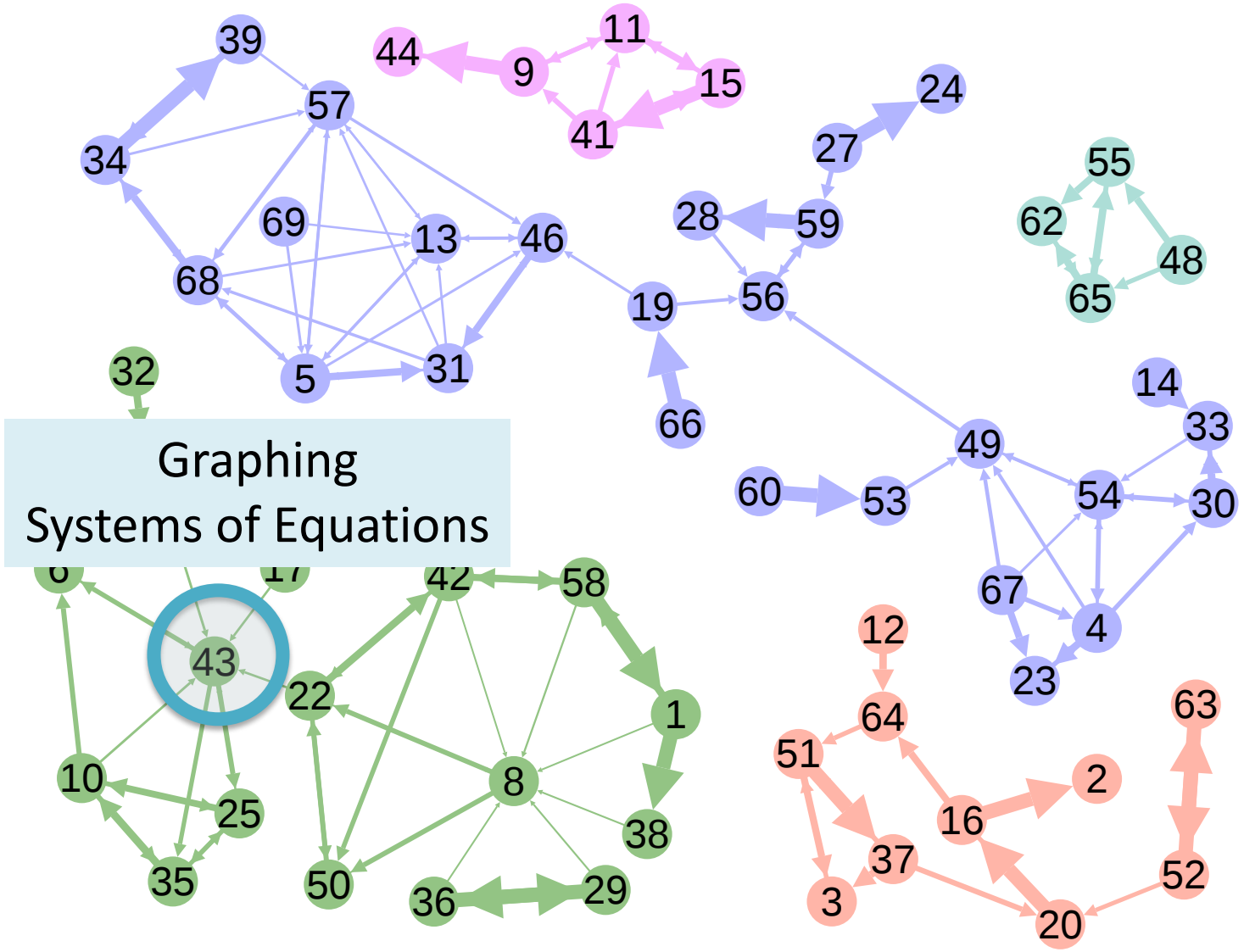
Learns Concept Relationships



Learns Concept Relationships

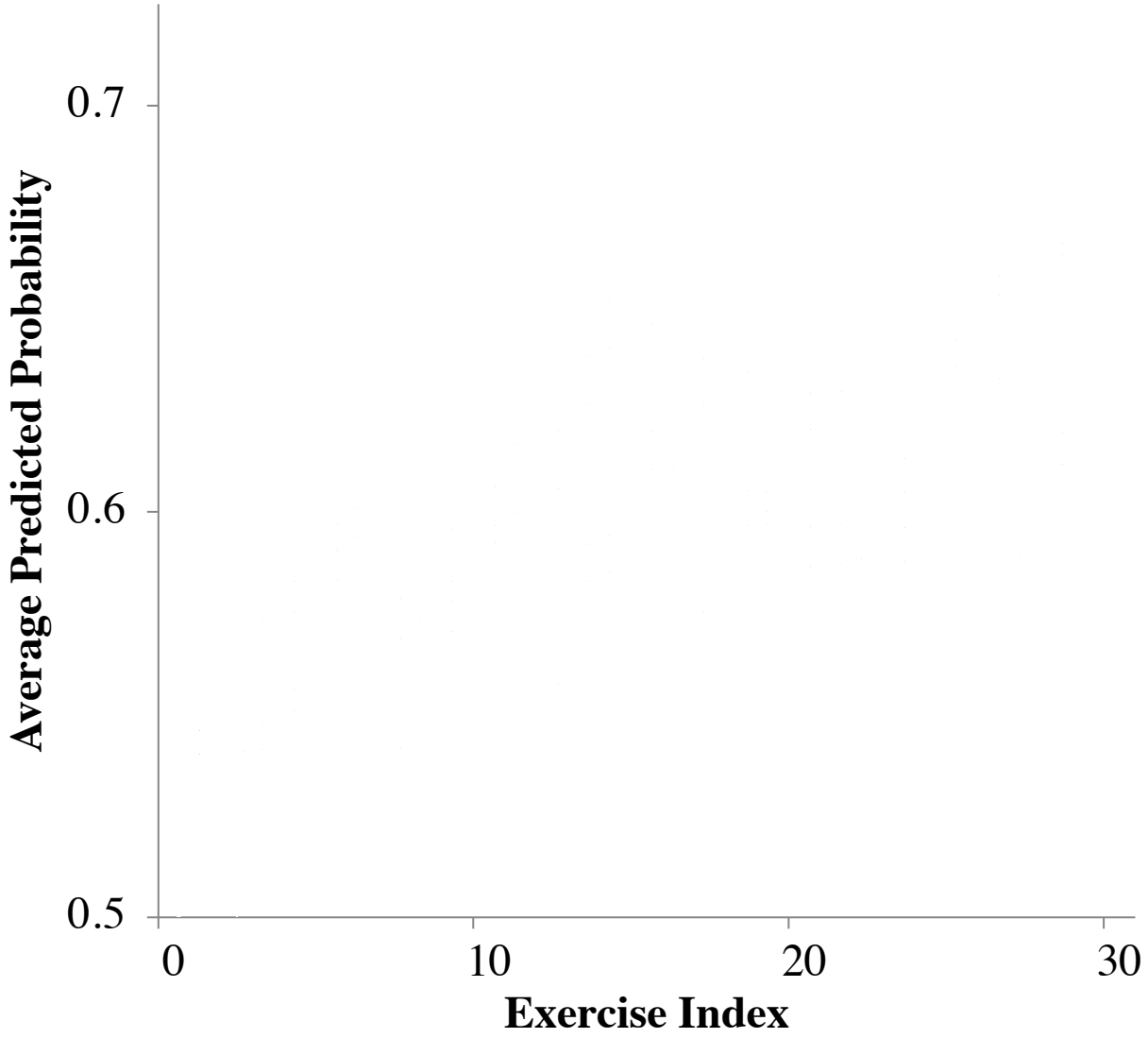


Learns Concept Relationships

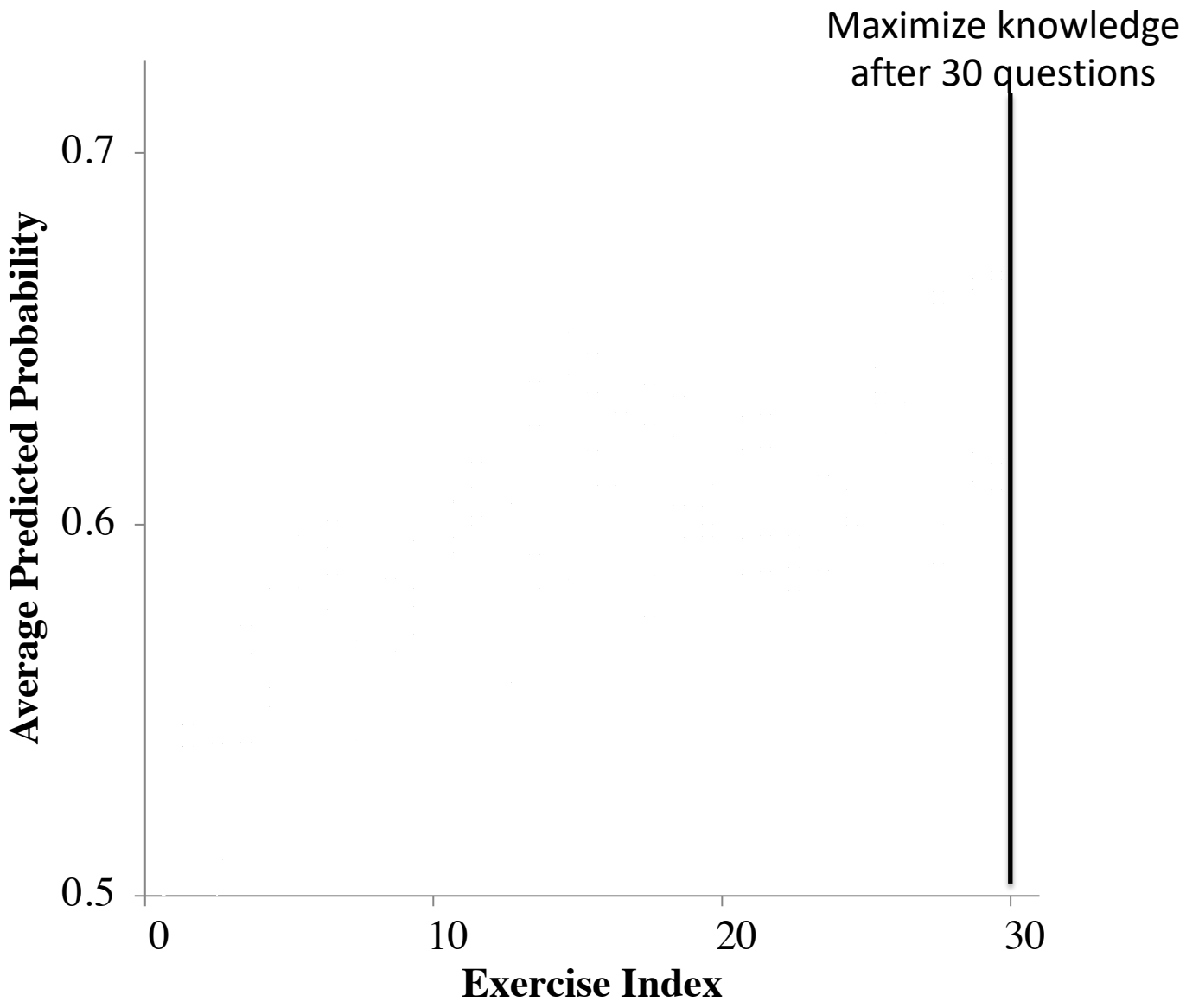


Graphing
Systems of Equations

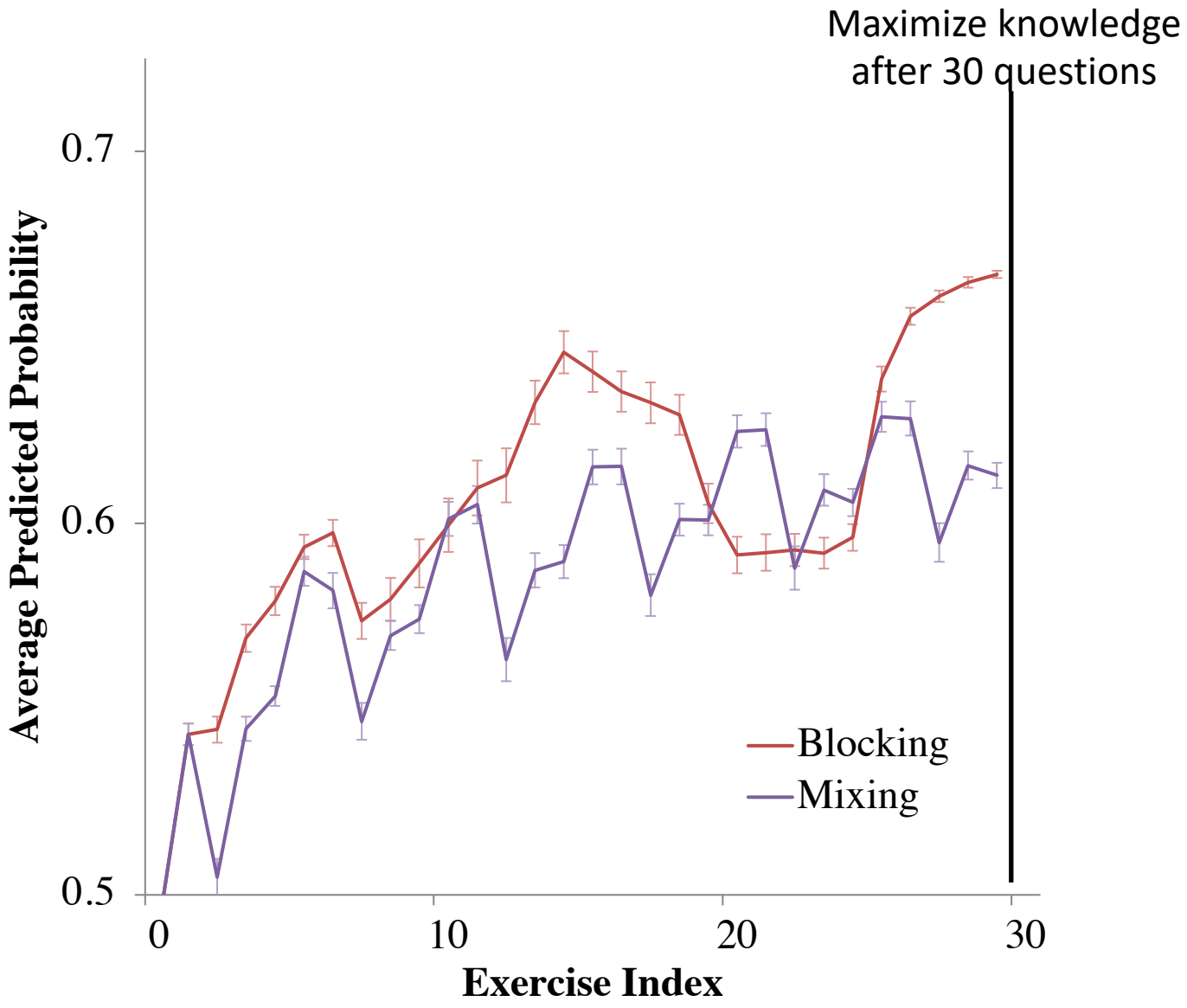
Optimal Teaching



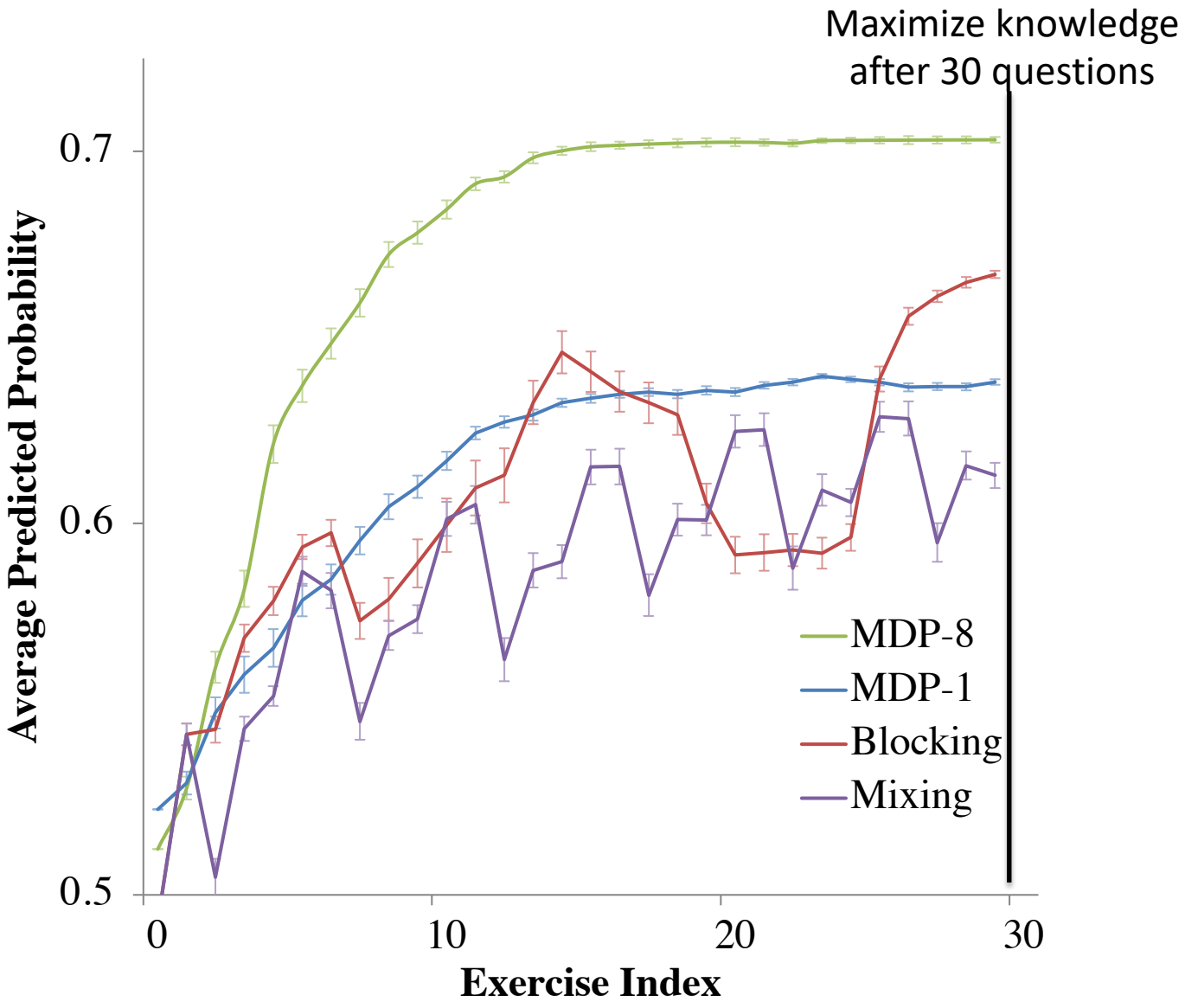
Optimal Teaching



Optimal Teaching

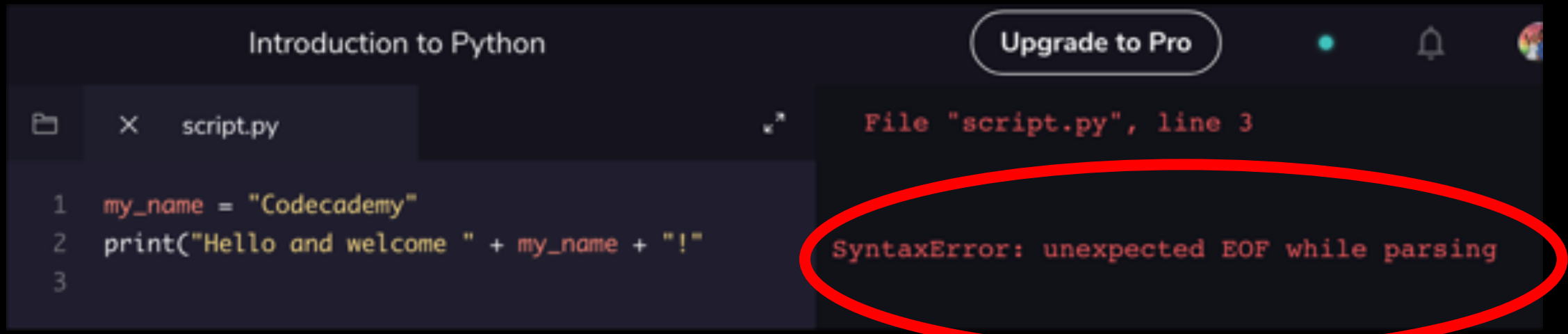


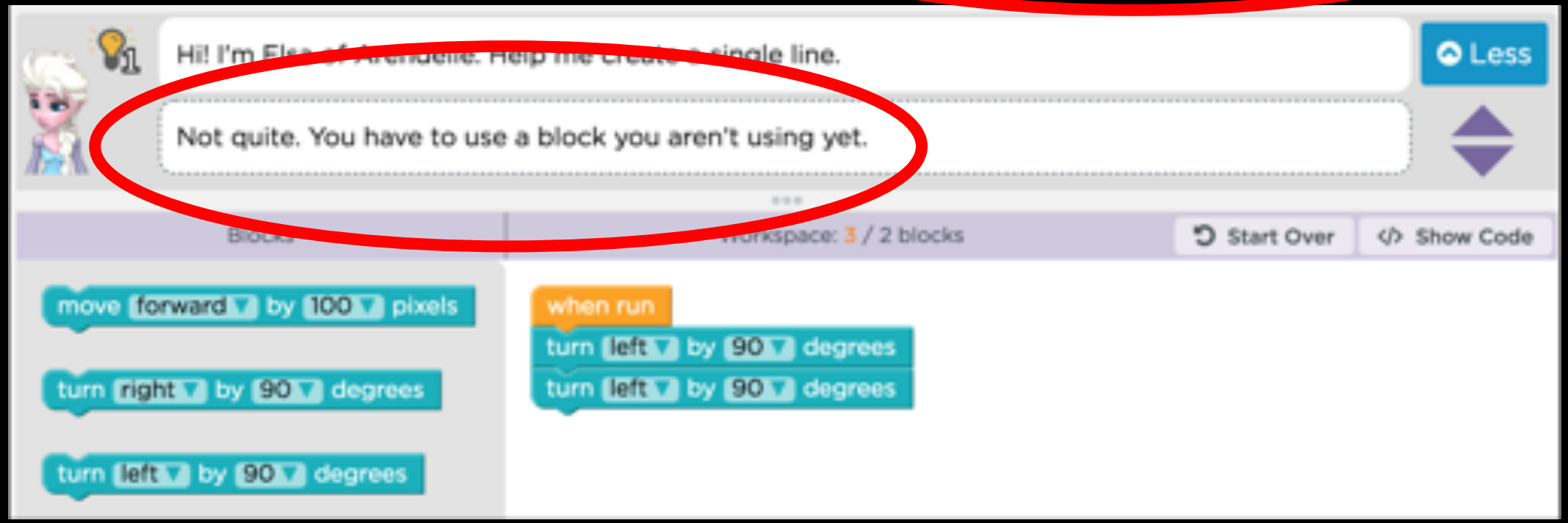
Optimal Teaching



We truly would rather move beyond correct / incorrect

Some domains are very hard





Can you understand this code?

Top Secret

Not Secure

Question | Solution

Instructions

- If there are many moves, focus on the first one
- Random code strategy is for when the student seems to be trying things randomly
- Lookout for students who don't get nesting or pre/post conditions. Often extra blocks in a body is an indication that they don't get that the post of the loop has to match the precondition

Question

```
import code.org.*;


public class MySoln {
  public void run() {
    move(50);
    for(int i=0; i<4; i++){
      if(frontIsClear()) {
        turnLeft(90);
      }
      for(int j=0; j<i; i++){
        move(i * 20);
        turnRight(120);
        move(10);
      }
    }
  }
}
```

Student 0

Label Console

✔ Num Done: 8273

Strategy

- Beeper Boundry (most people do this)
- Triangle Strategy
- Recursive Strategy

Looping

- Correct use of looping
- Doesn't use a while
- Doesn't have correct stop condition

Stanford TAs label 800 submissions

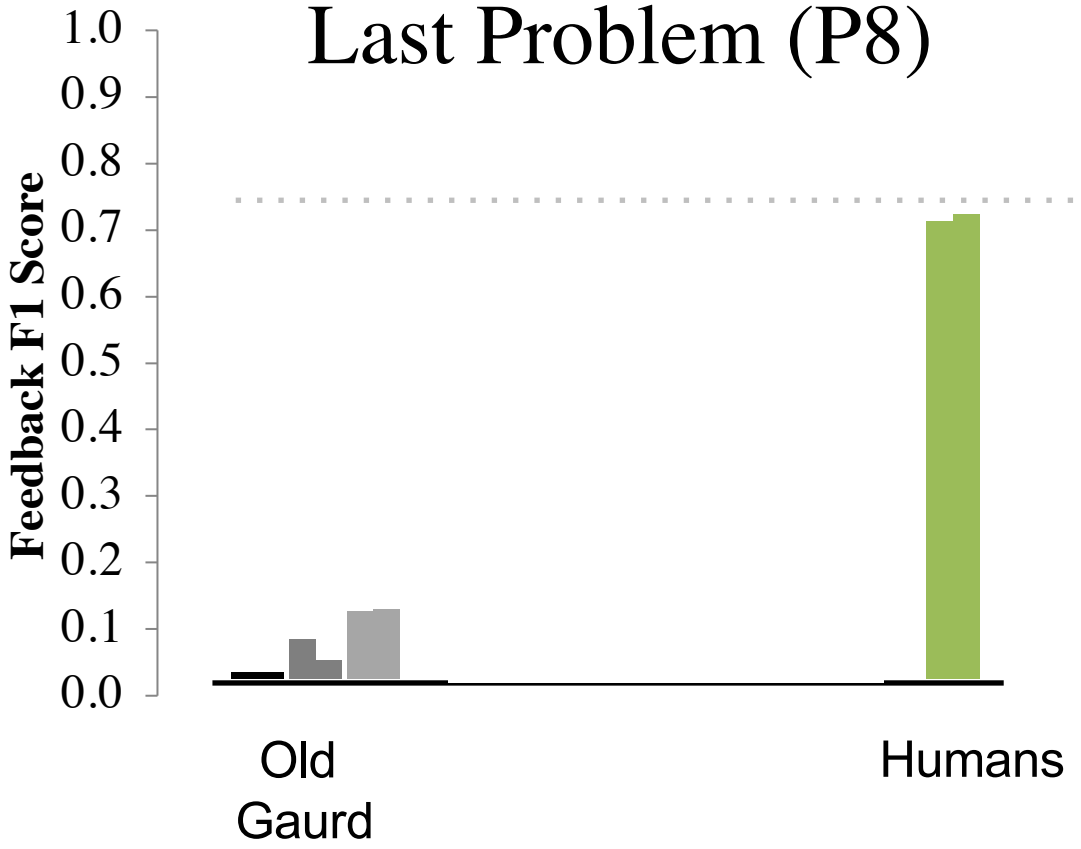
- Loop post condition doesn't match preconditi
- Repetition of bodies

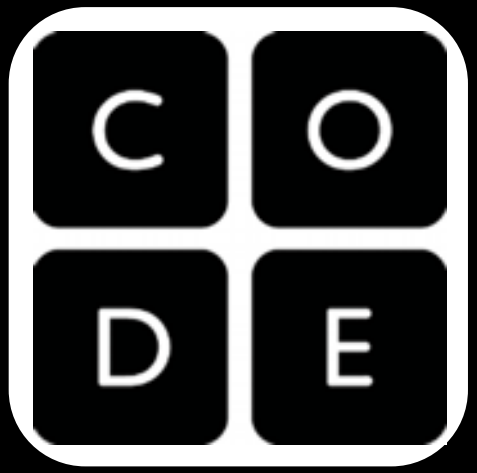
Cleanup

Record label

Traditional Deep Learning Doesn't Work

Label student code





Can we provide feedback by dynamic analysis?

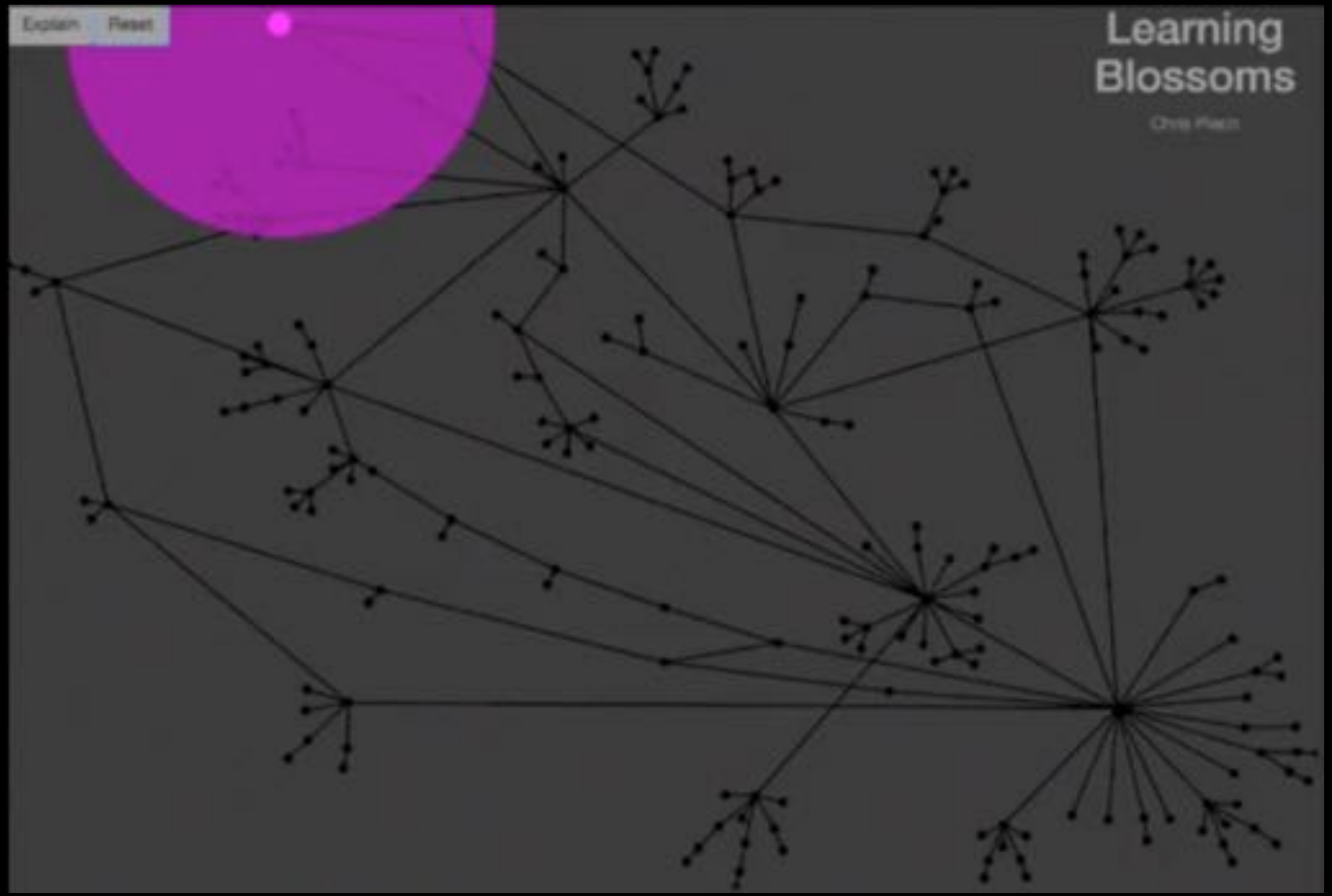
Starter code

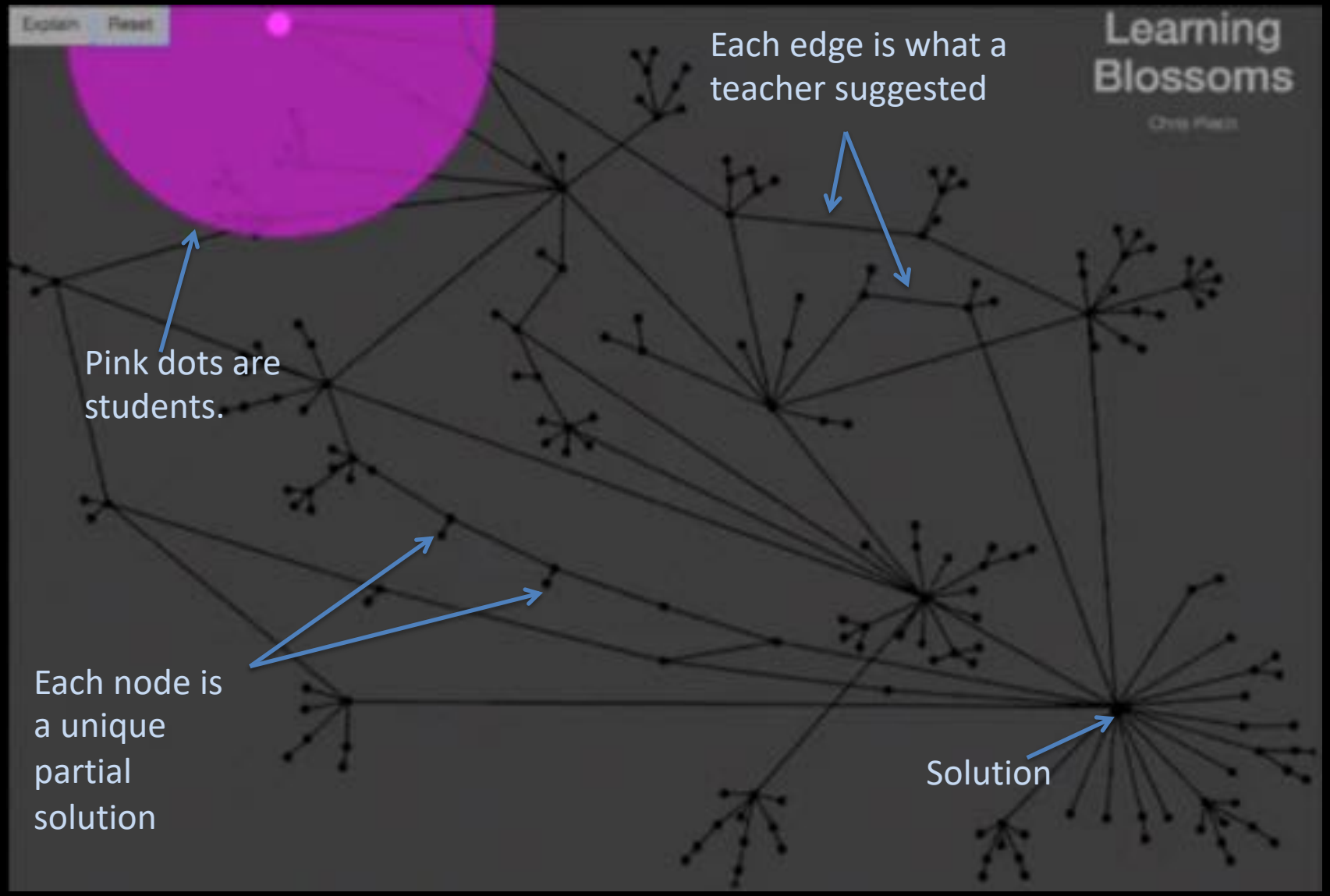
First attempt

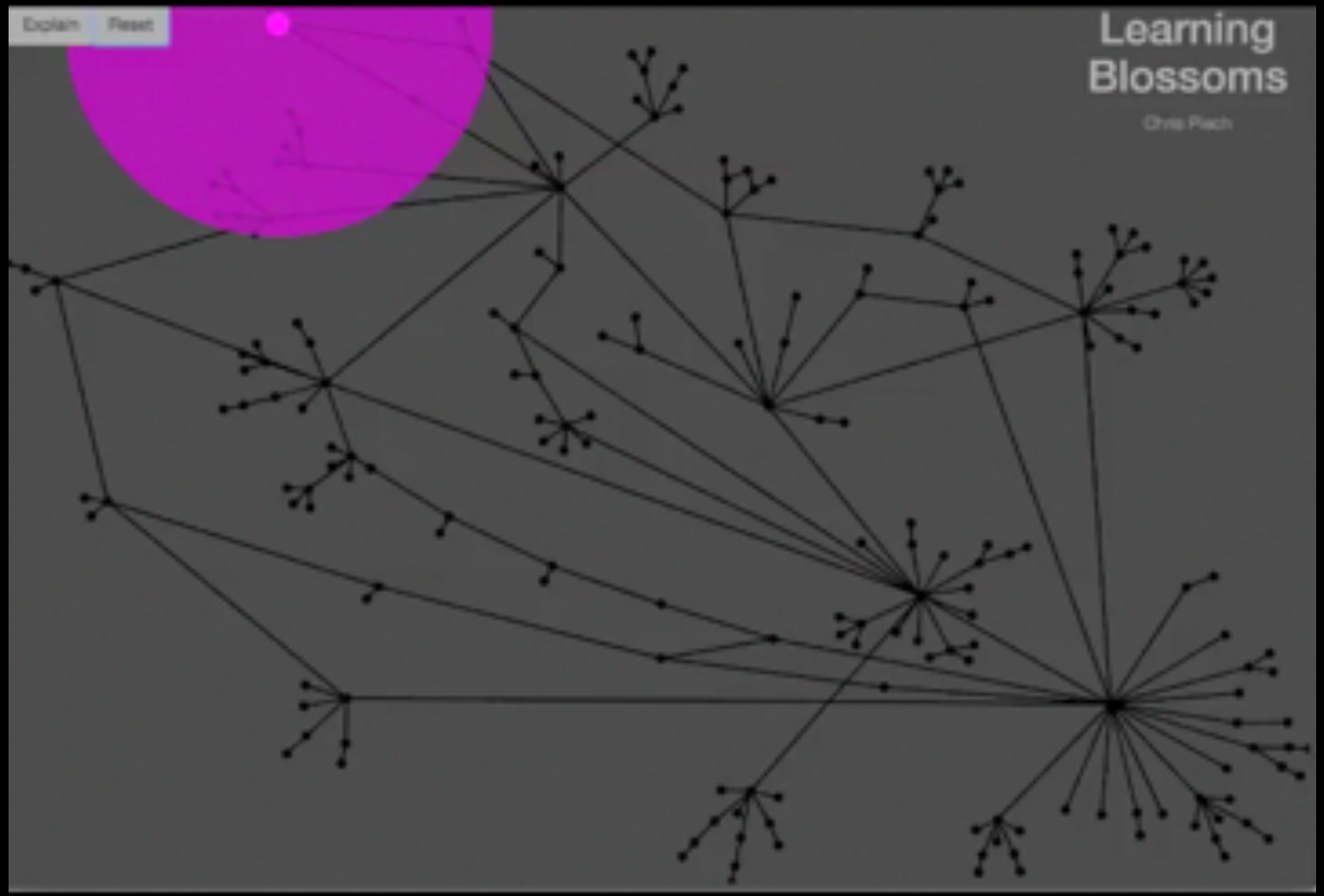
. . .

Final solution

Chapter 1: Better data source?


Explain
Reset
Learning
Blossoms
Chris Piech


Explain
Reset
Learning Blossoms
Each edge is what a teacher suggested
Pink dots are students.
Each node is a unique partial solution
Solution


Explain
Reset
Learning
Blossoms
Chris Piech

The Crowd is Un-wise

Temporal methods tried:
Shortest path
Min Time
Expected Success
Reinforcement learning
Most Common Next
Most Popular Path

18%

45%

12%

Desirable Path Algorithms

Poisson Common Path

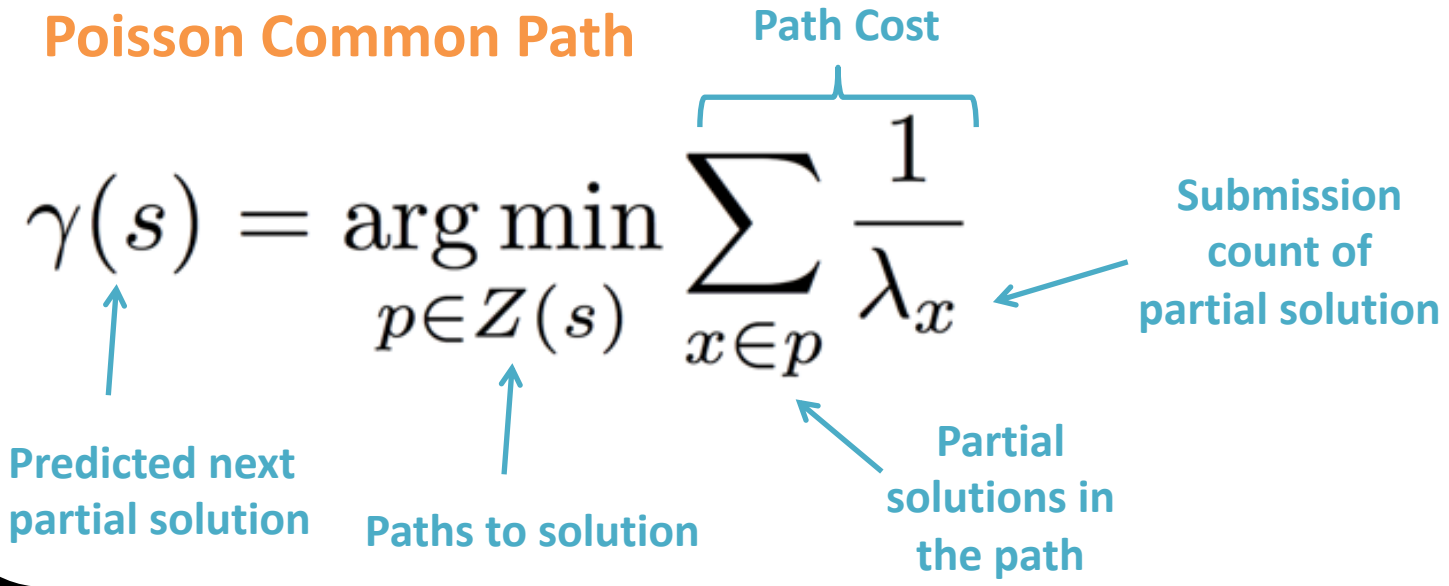
$$\gamma(s) =$$

Predicted next partial solution

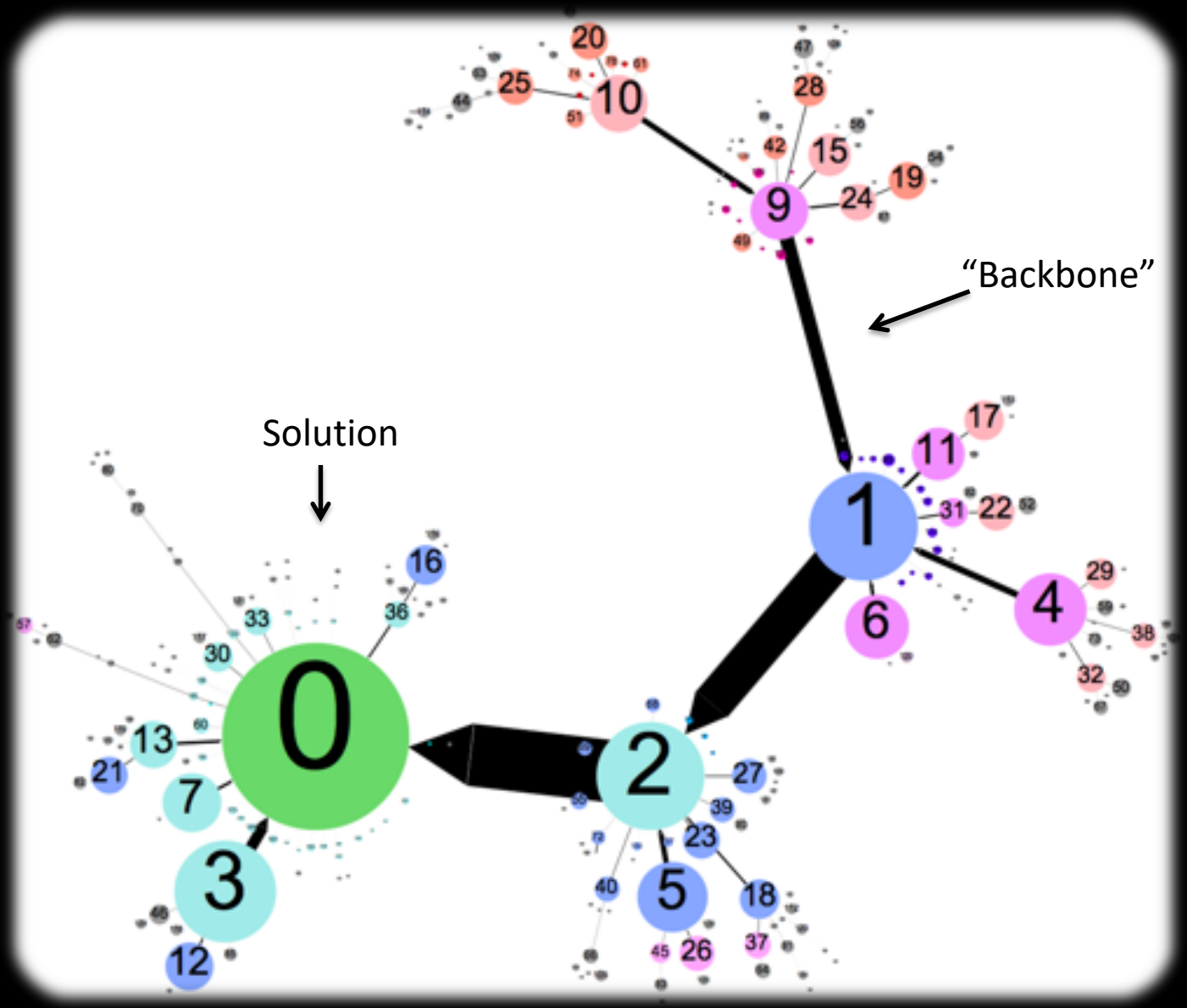
First step in the ***most frequent*** path to the solution from *s*, taken by ***average*** students. Assume poison process.

Desirable Path Algorithms

Poisson Common Path



Learned Problem Solving Policy



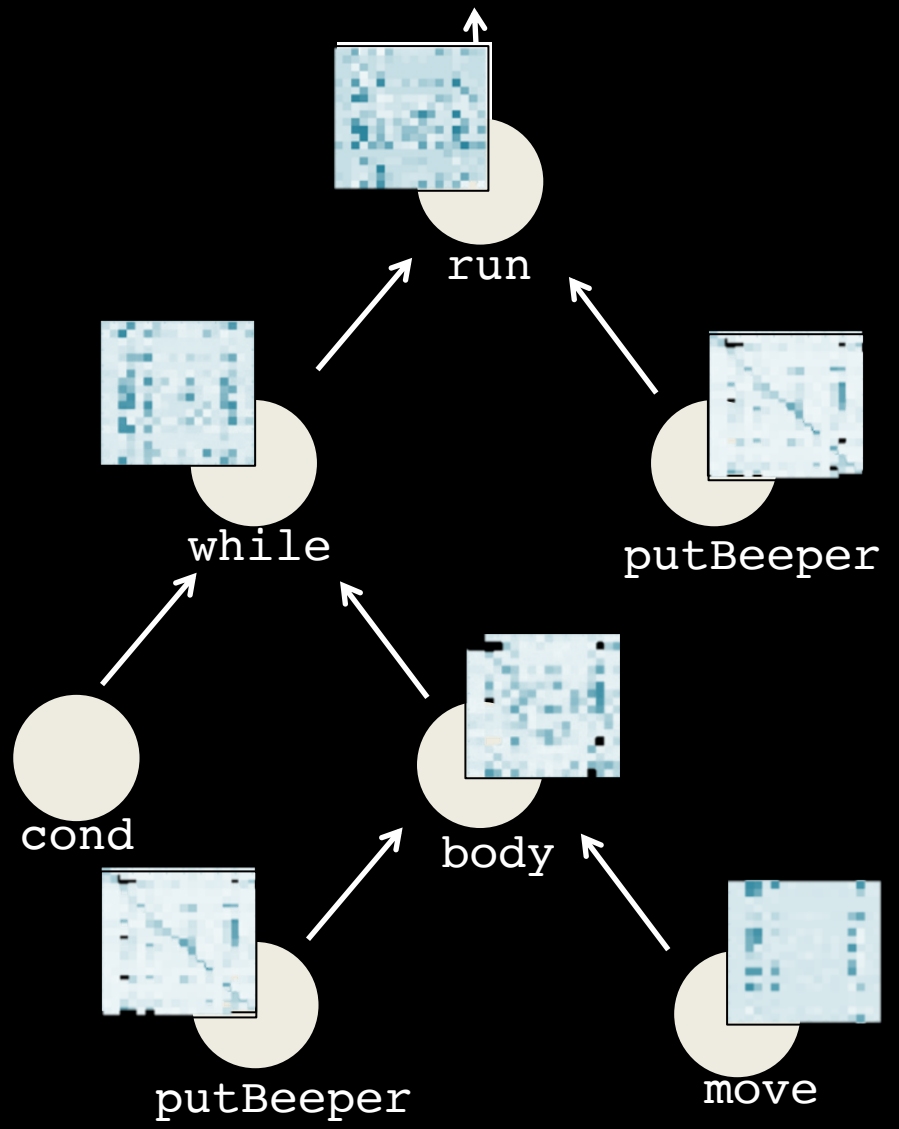
Only worked well for 6 line programs…

Chapter 2: Start to invent new algorithms…

Encode a Program

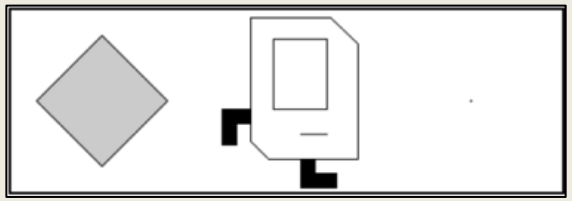
```
// User defined method
private void run() {
    while(isClear()){
        putBeeper();
        move();
    }
    putBeeper();
}
```

It looks like you have a fencepost error!



*Note: this was coded pre-tensor flow

Collect Triples



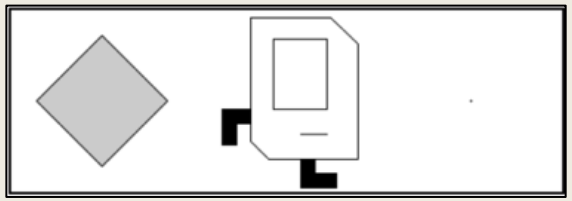
Precondition

```
putBeeper();
move();
```

Code

About 5 million triples per assignment

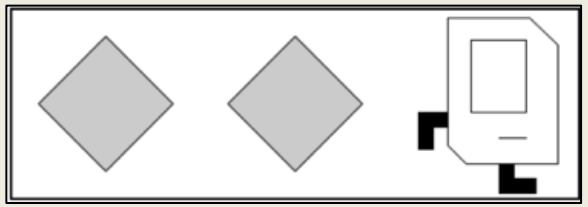
Collect Triples



Precondition

```
putBeeper();
move();
```

Code

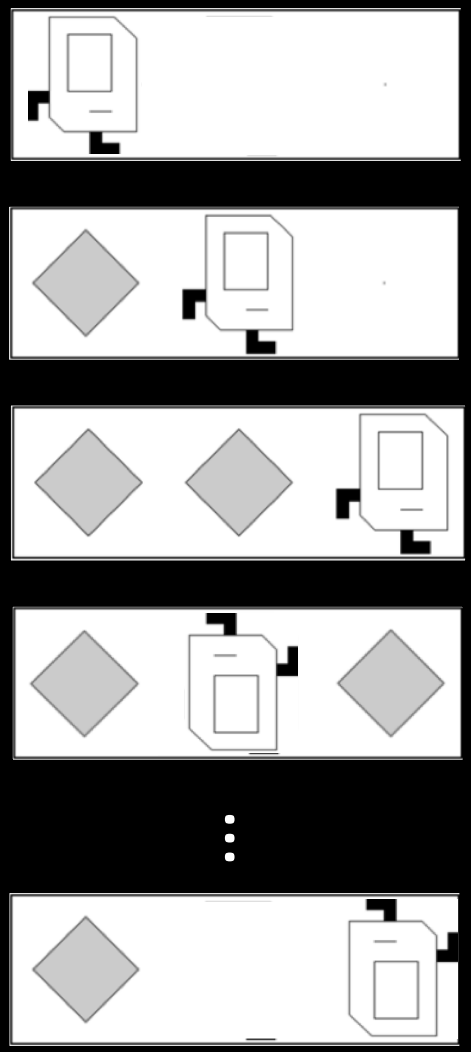


Postcondition

About 5 million triples per assignment

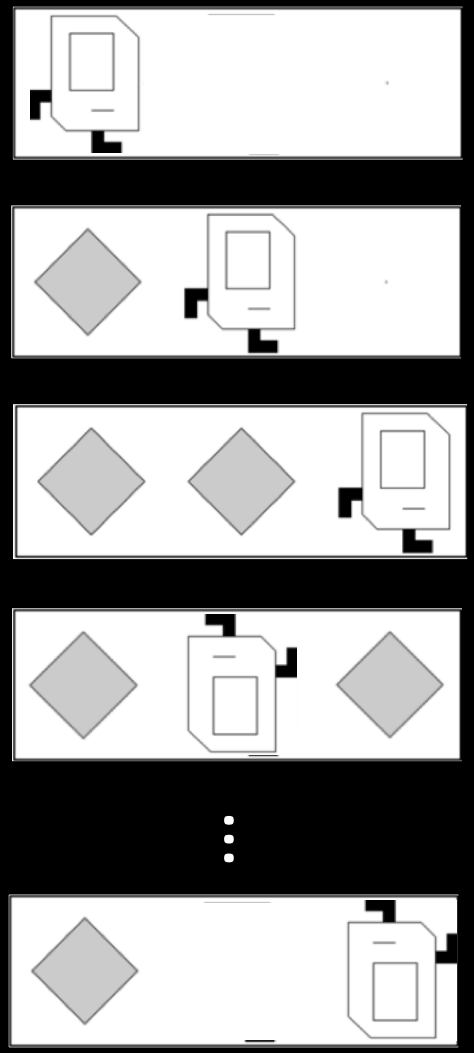
A Code Phrase is a Mapping

All possible preconditions

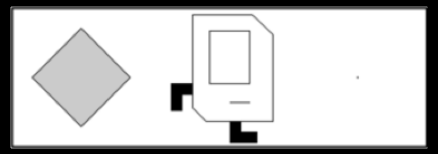


```
putBeeper();
move();
```

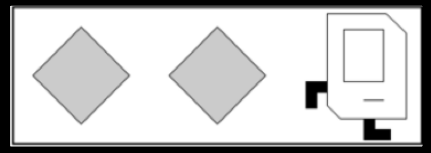
All possible postconditions



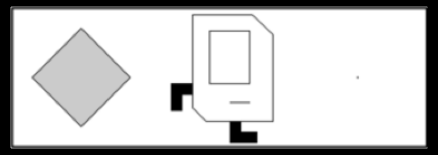
Neural Network for Programs



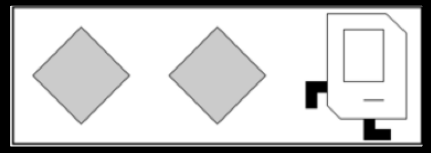
```
method step() {
    putBeeper();
    move();
}
```



Neural Network for Programs



```
method step() {
    putBeeper();
    move();
}
```

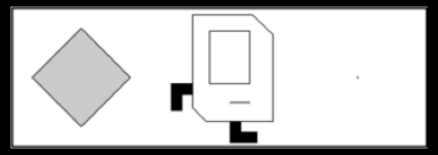


Raw Precondition

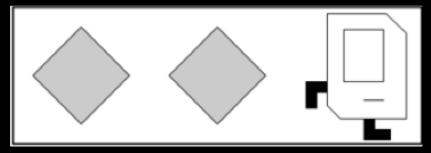
Program Matrix

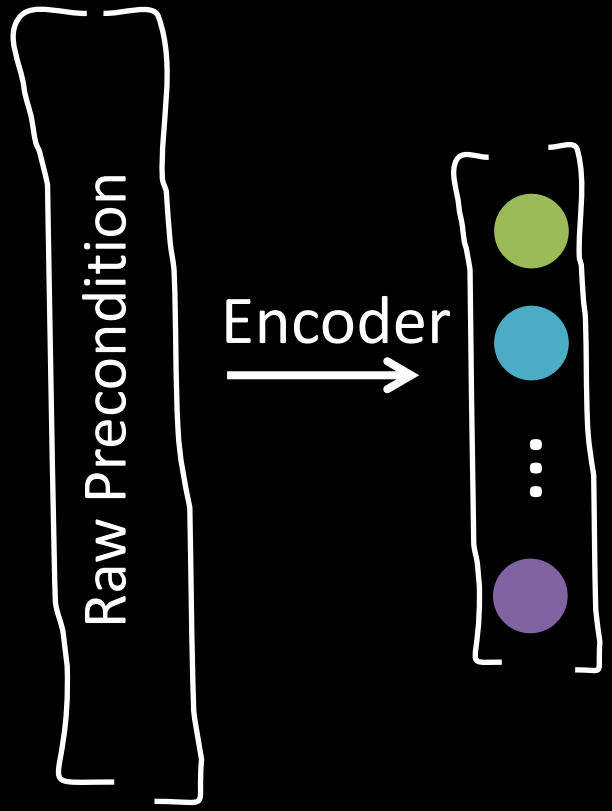
*coded pre-tensor flow

Neural Network for Programs



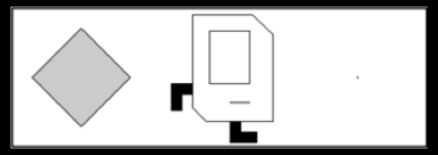
```
method step() {
    putBeeper();
    move();
}
```



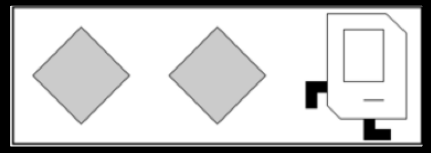


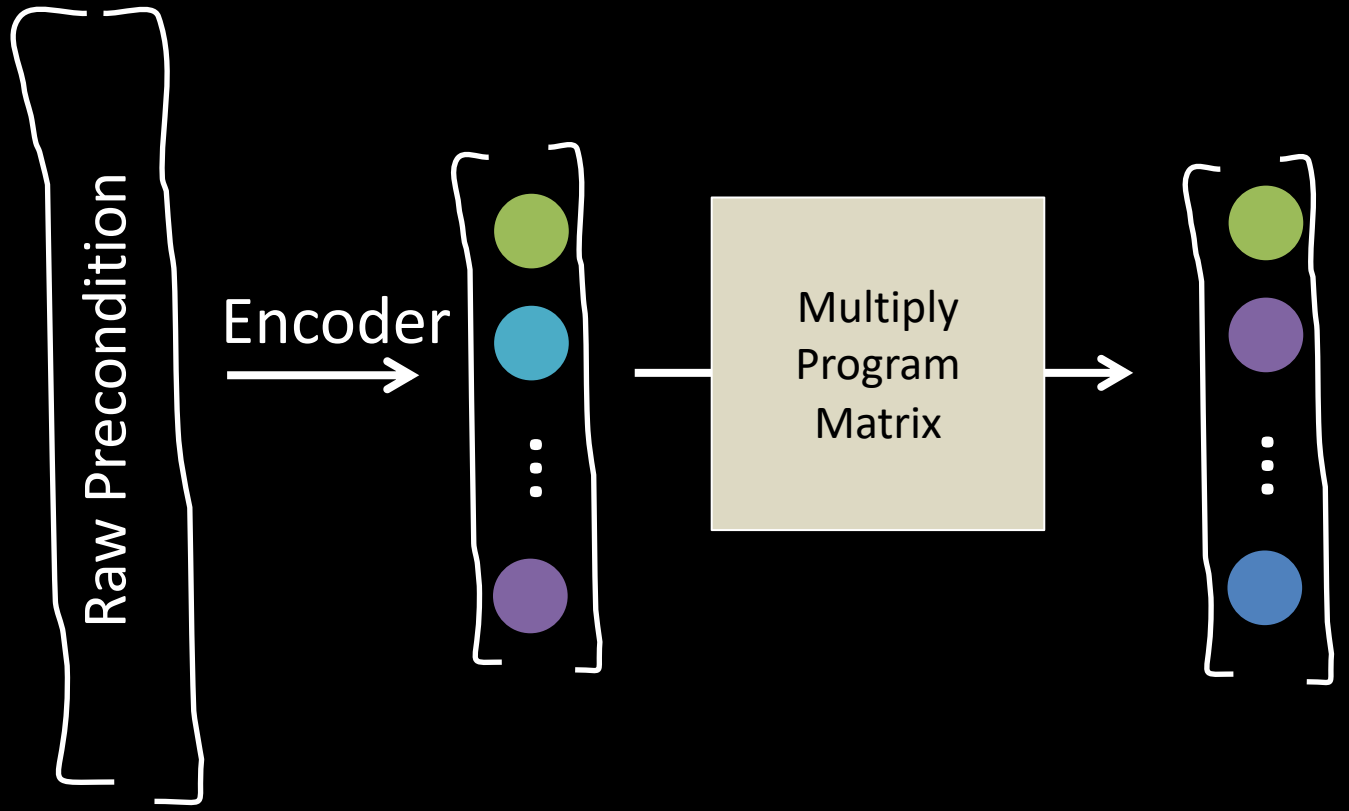
*coded pre-tensor flow

Neural Network for Programs



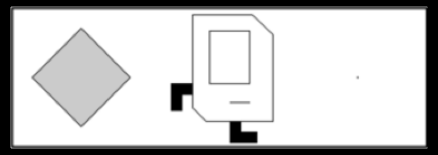
```
method step() {
    putBeeper();
    move();
}
```



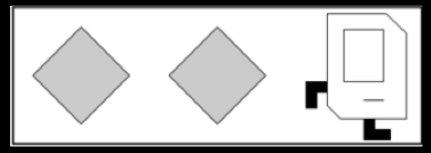


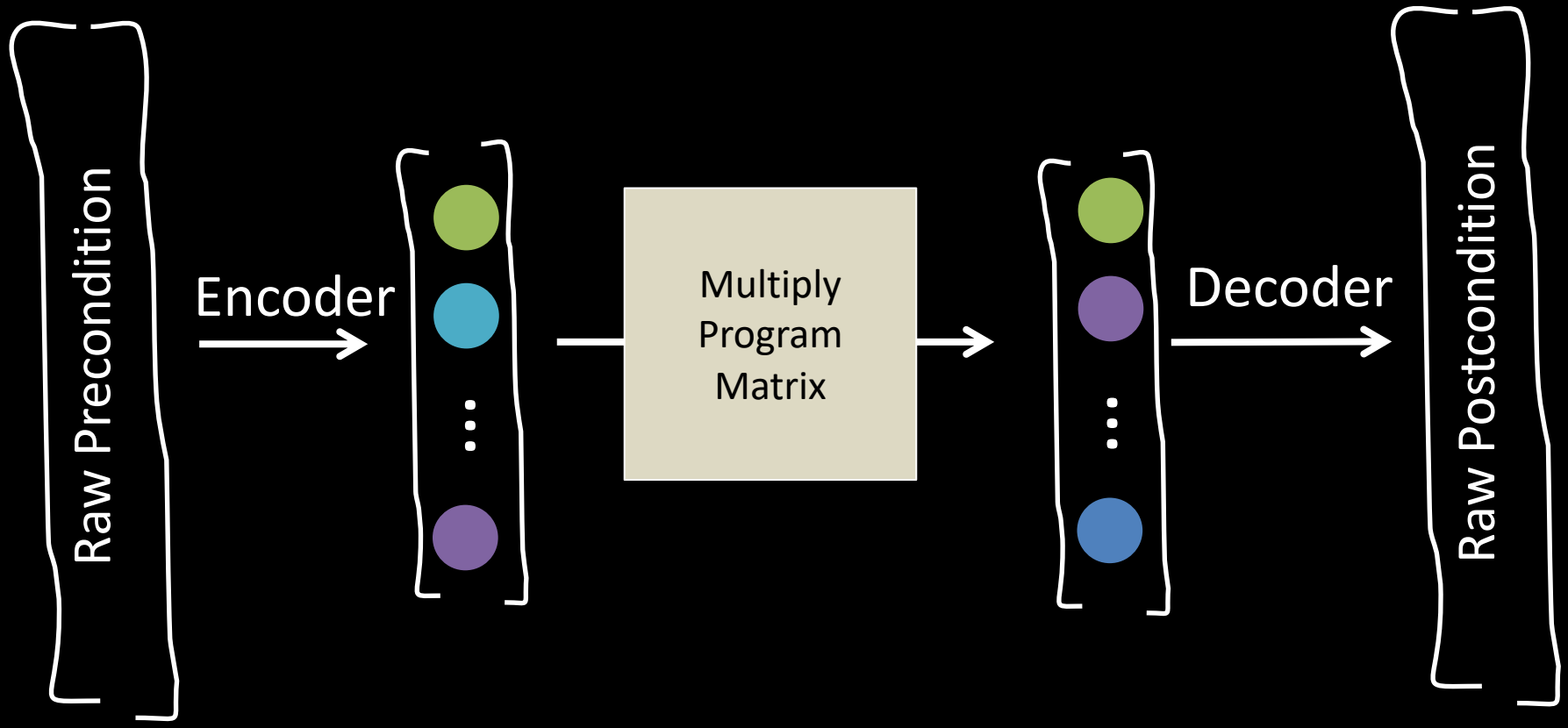
*coded pre-tensor flow

Neural Network for Programs



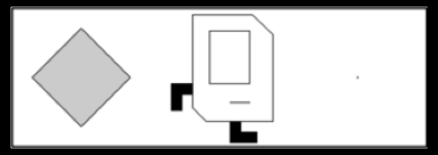
```
method step() {
    putBeeper();
    move();
}
```



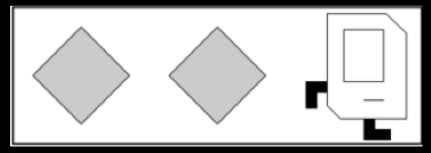


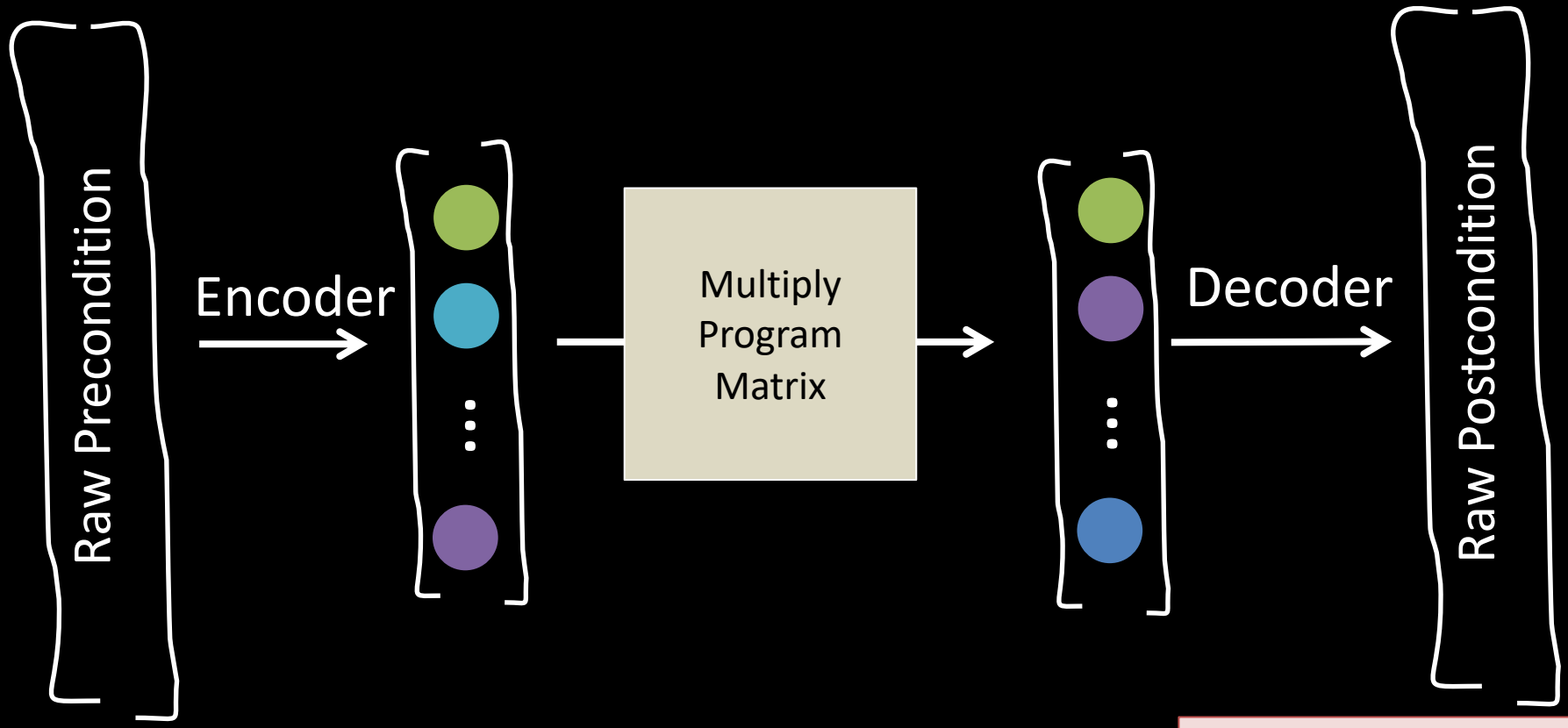
*coded pre-tensor flow

Neural Network for Programs



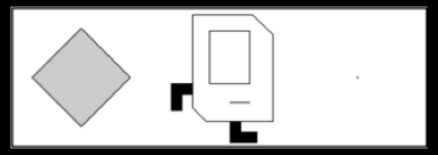
```
method step() {
    putBeeper();
    move();
}
```



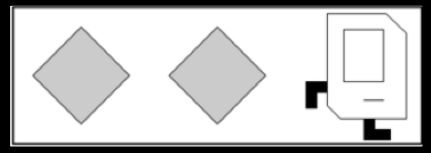


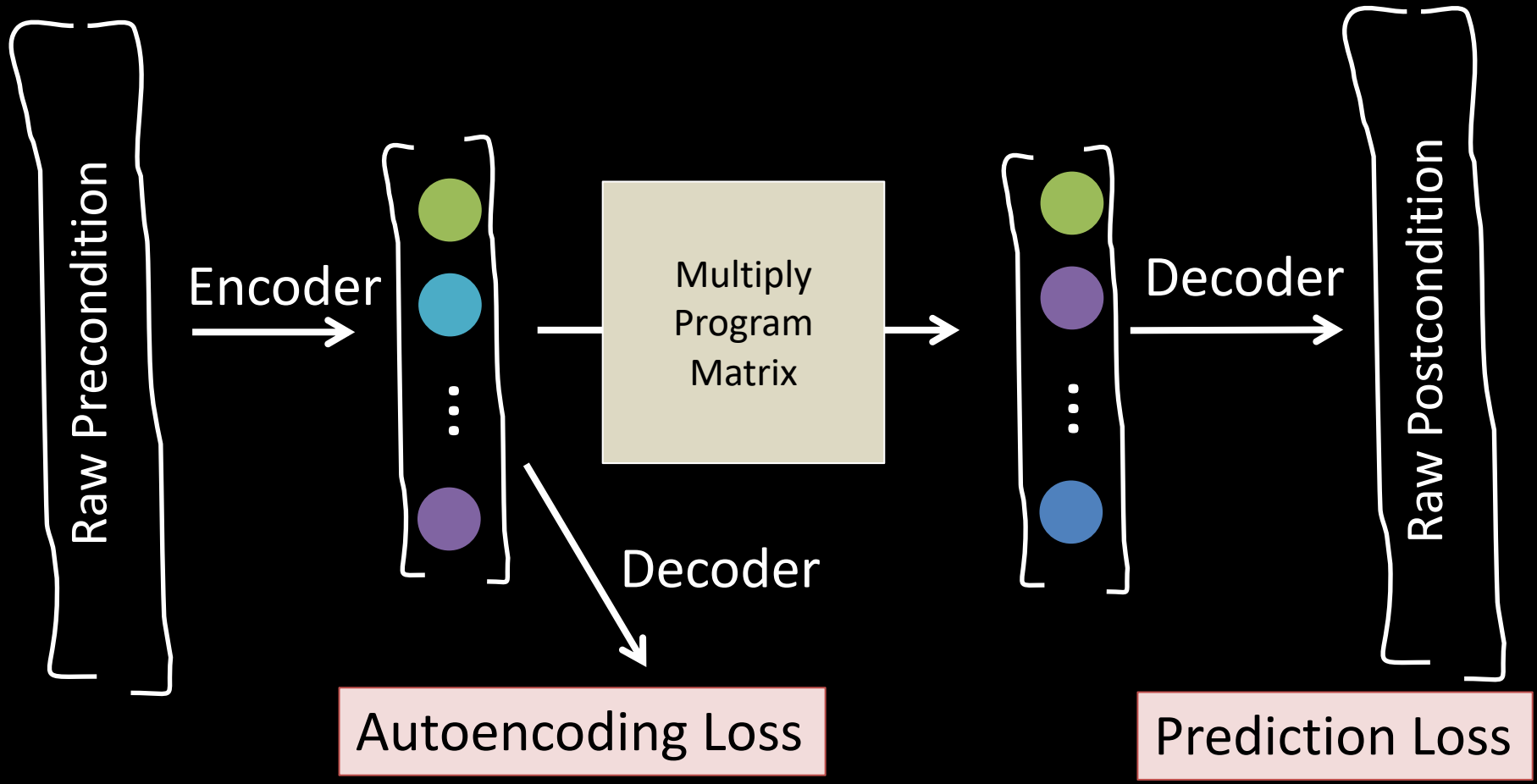
*coded pre-tensor flow

Neural Network for Programs



```
method step() {
    putBeeper();
    move();
}
```



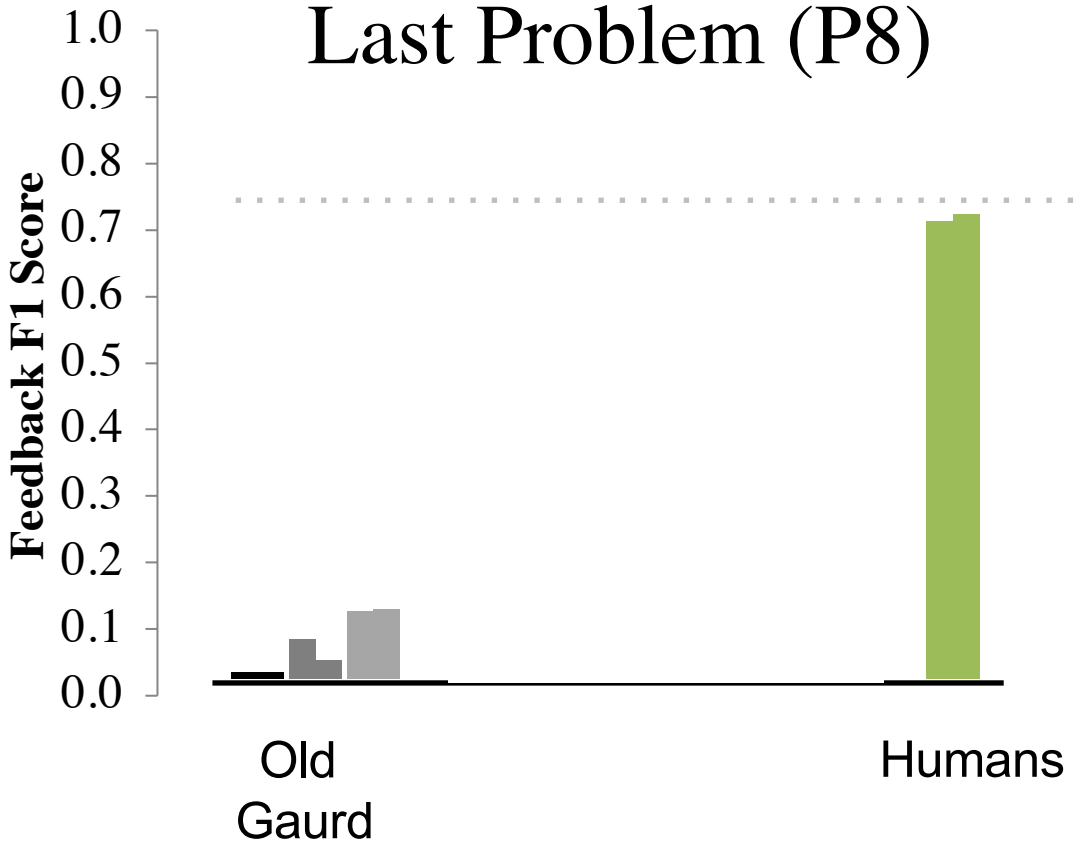


*coded pre-tensor flow

Does it work?

Traditional Deep Learning Doesn't Work

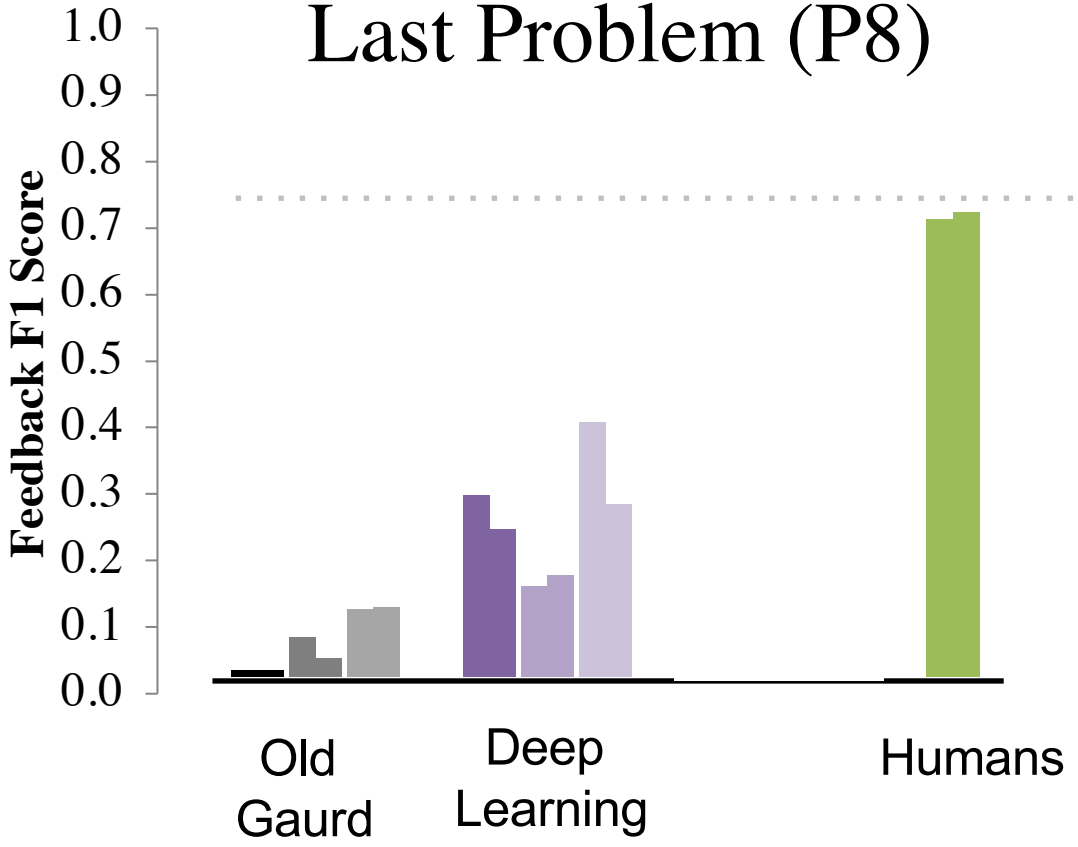
Label student code

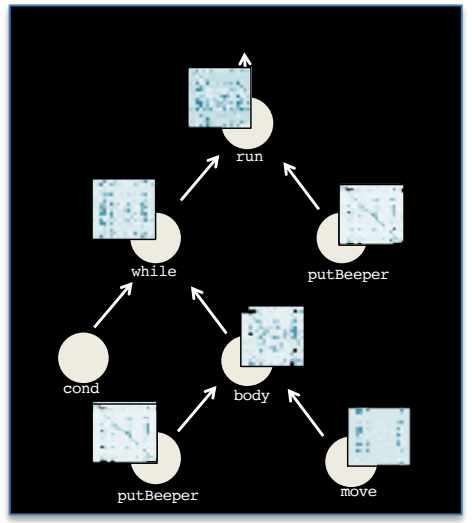


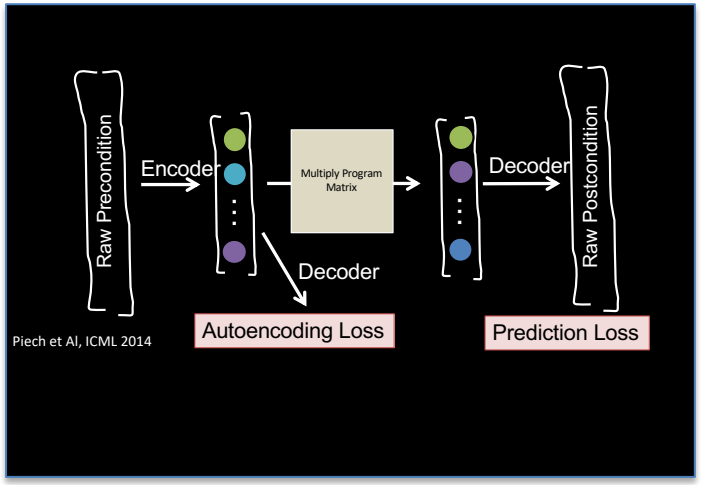
Inaccurate, Uninterpretable, and Data Hungry

Label student code

Last Problem (P8)

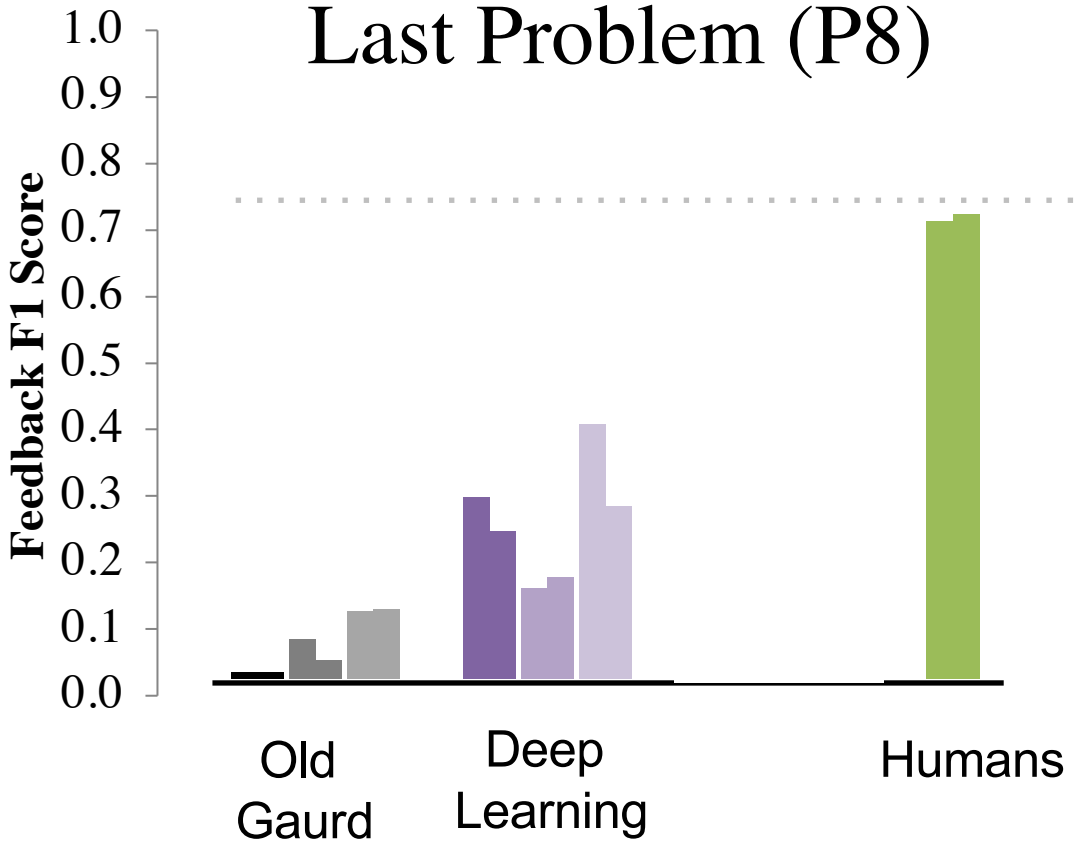






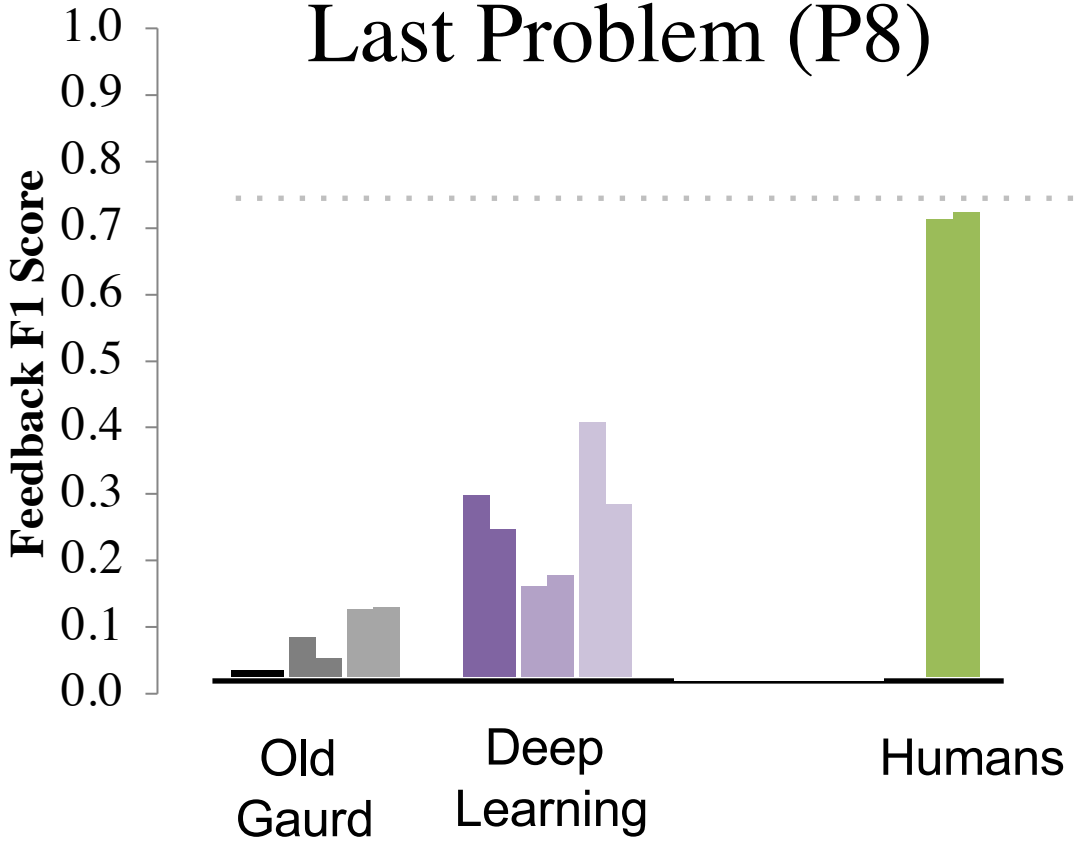
Inaccurate, Uninterpretable, and Data Hungry

Label student code



Data Hungry

Label student code



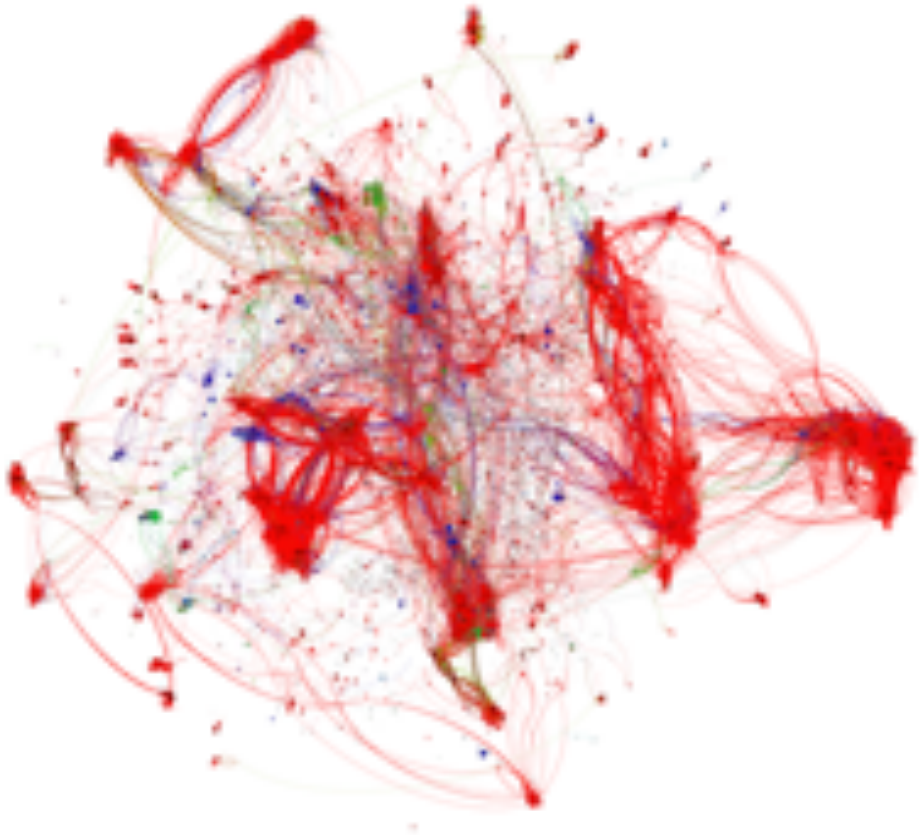
We need one shot learning

We need verifiability

Why is it so hard?

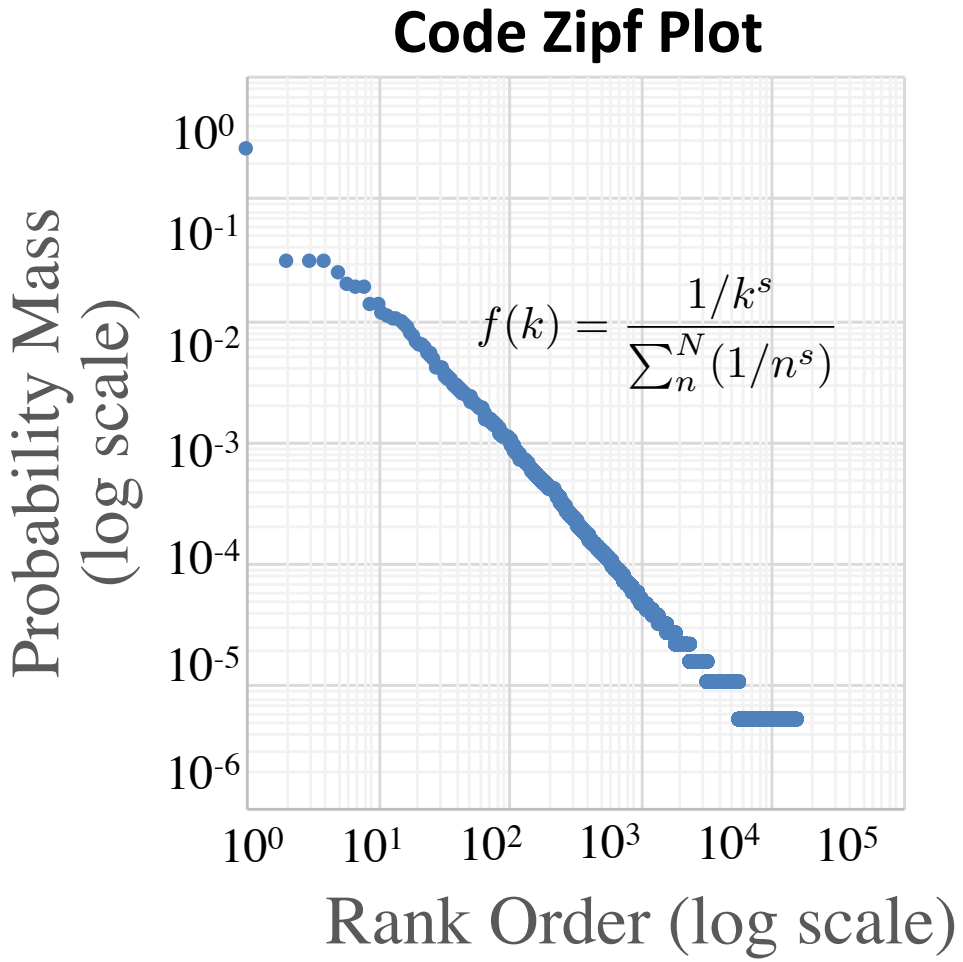
Hard Problem

Brute force solution?



1 million unique solutions to programming Linear Regression

WWW 2014



[Suspense]

Chapter 3: Back to the drawing board

Humans Don't Need Much Data

Single training example:

Test set:

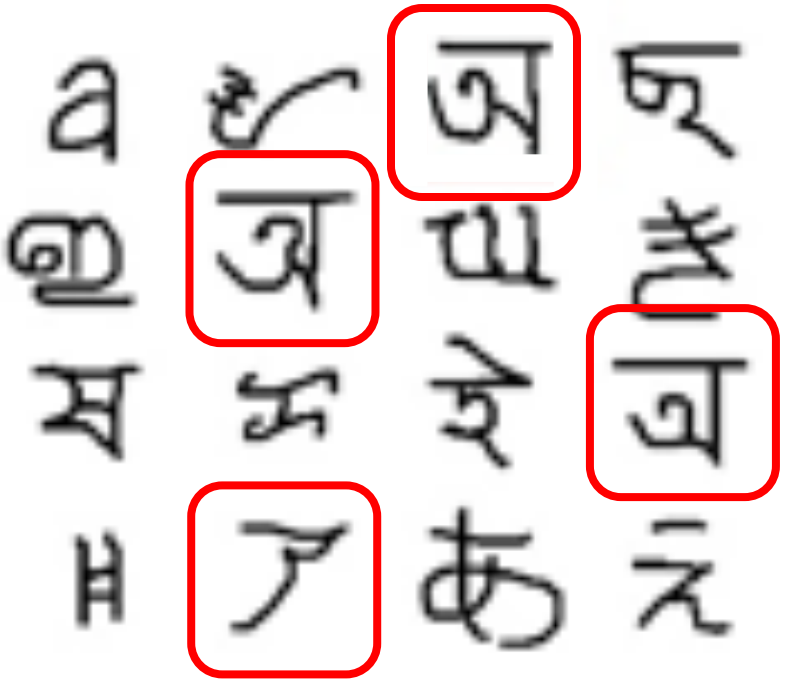
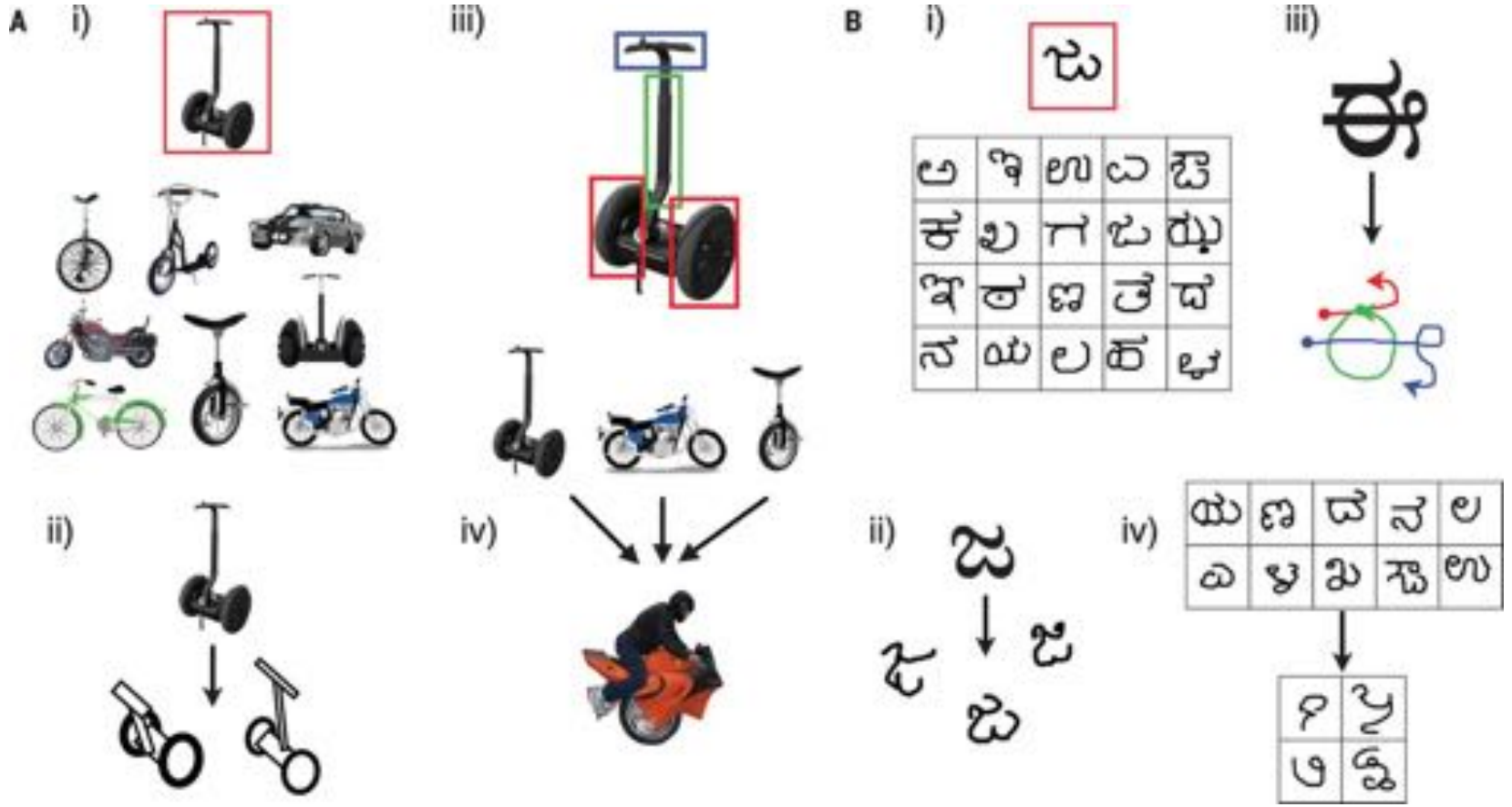


Fig. 1 People can learn rich concepts from limited data.



Brenden M. Lake et al. Science 2015;350:1332-1338

Fig. 2 Simple visual concepts for comparing human and machine learning.

Brenden M. Lake et al. Science 2015;350:1332-1338

Science
AAAS

Bayesian Program Learning

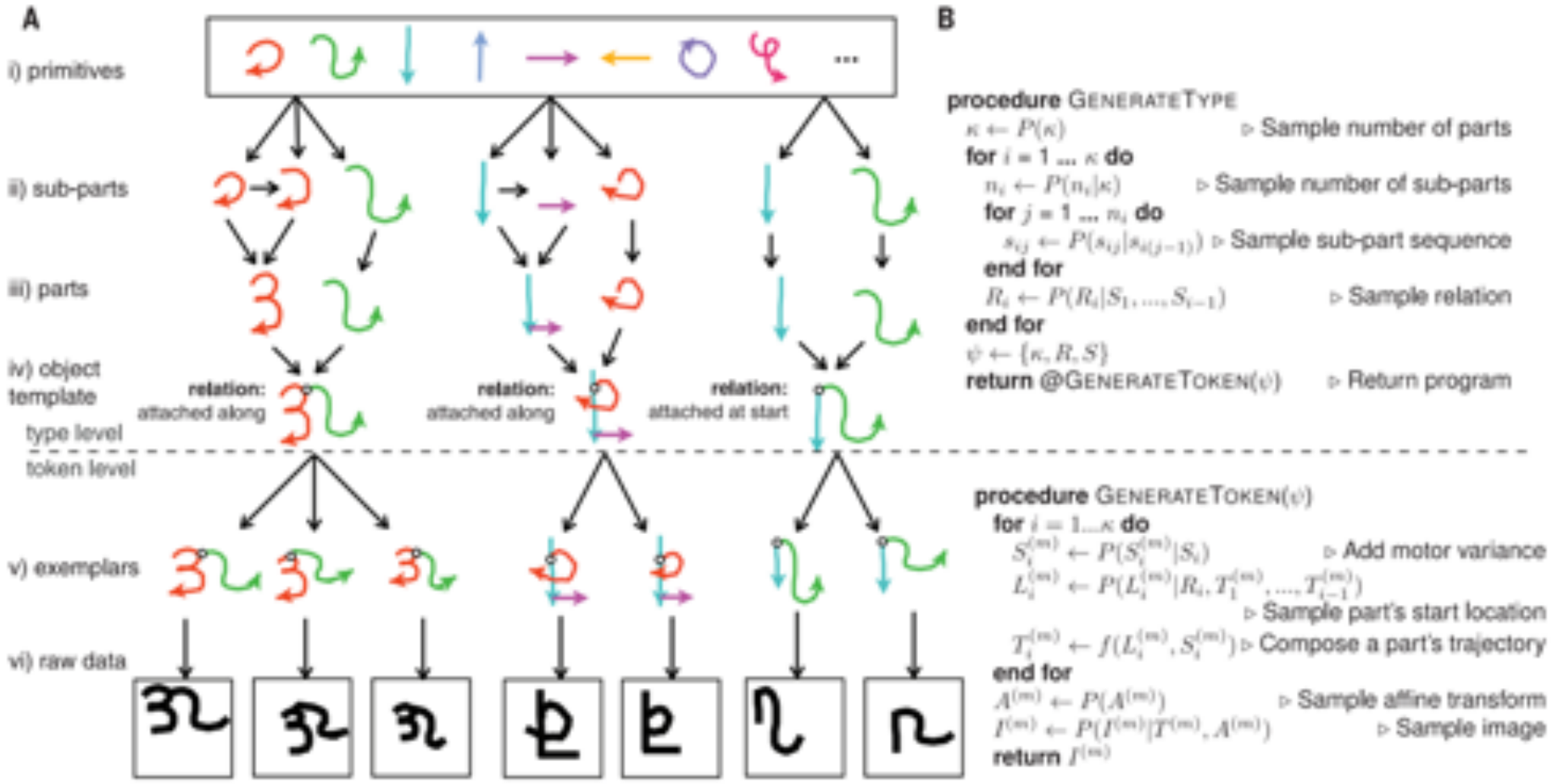
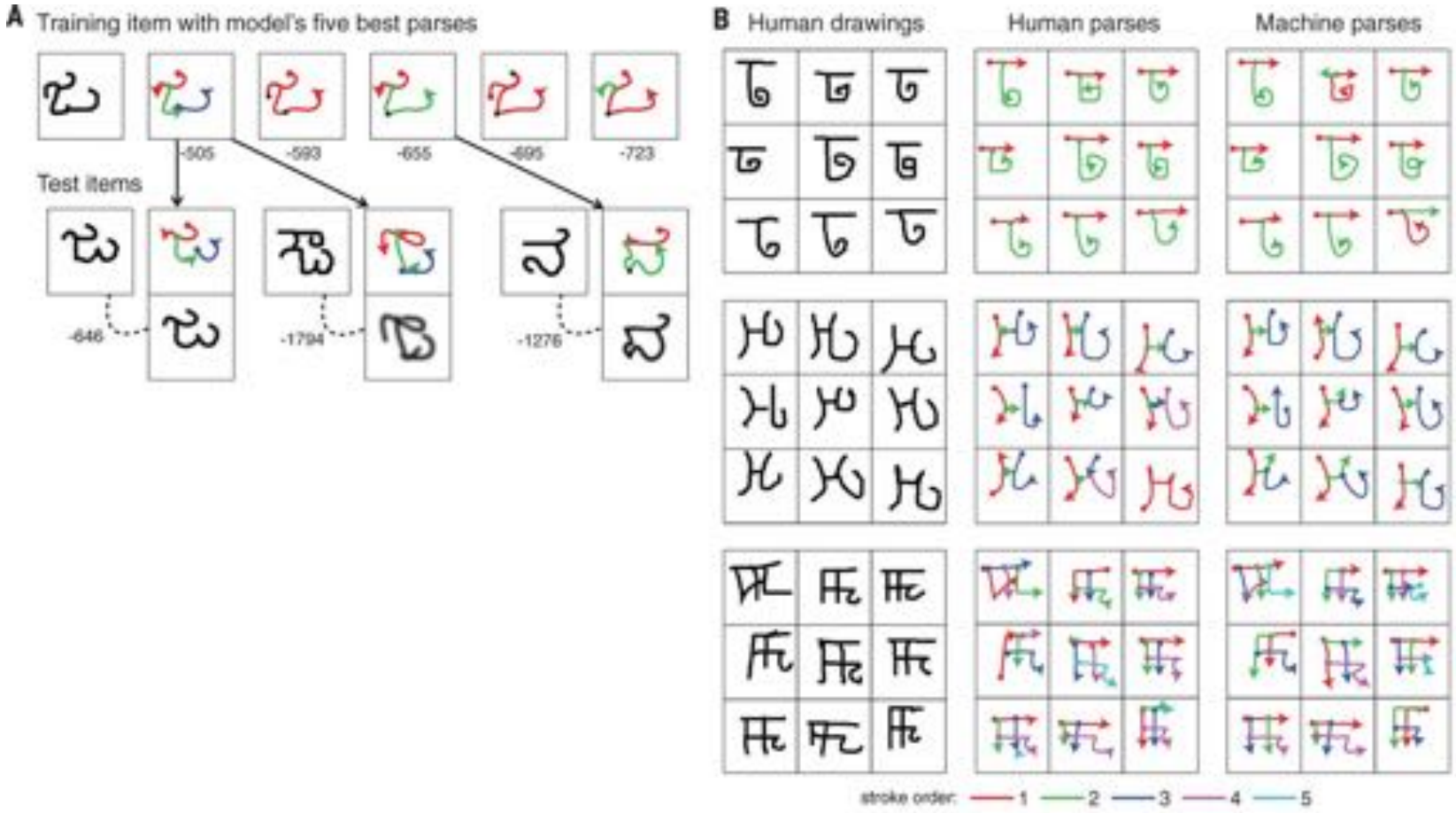
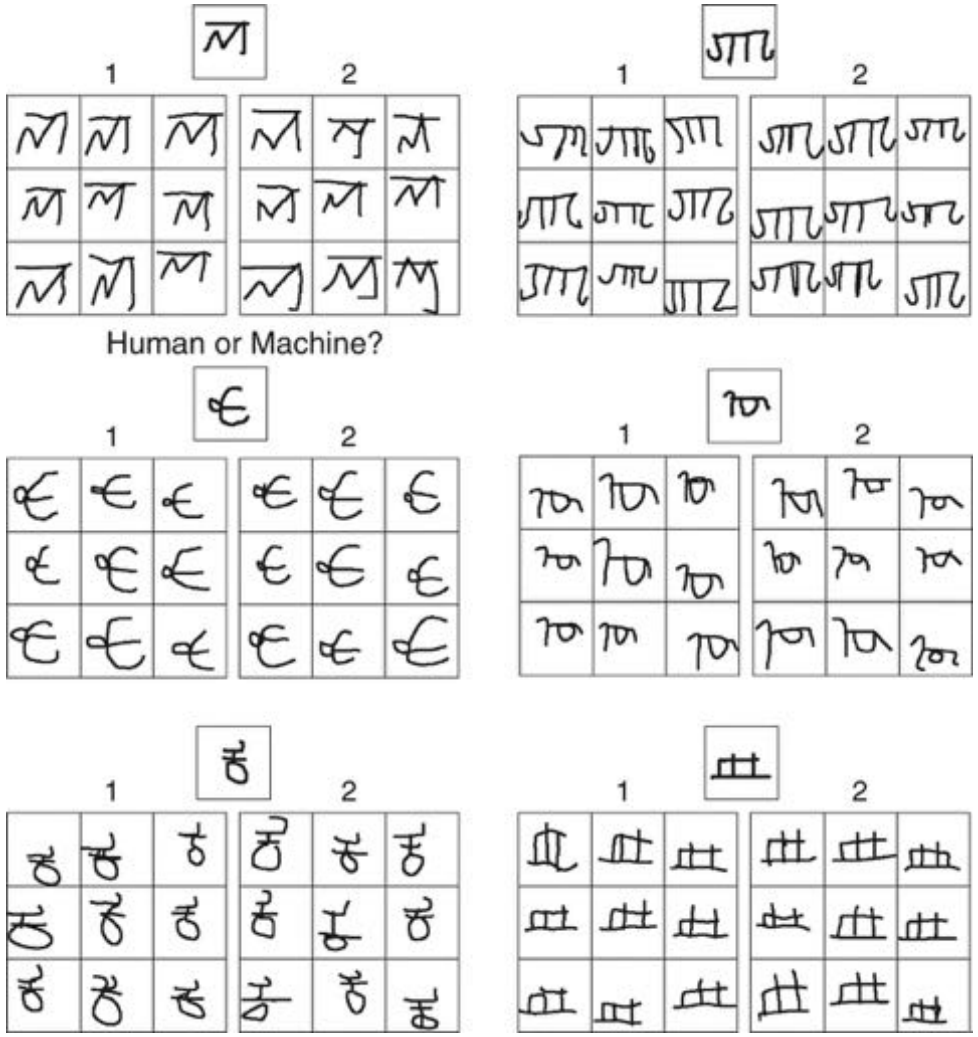


Fig. 4 Inferring motor programs from images.



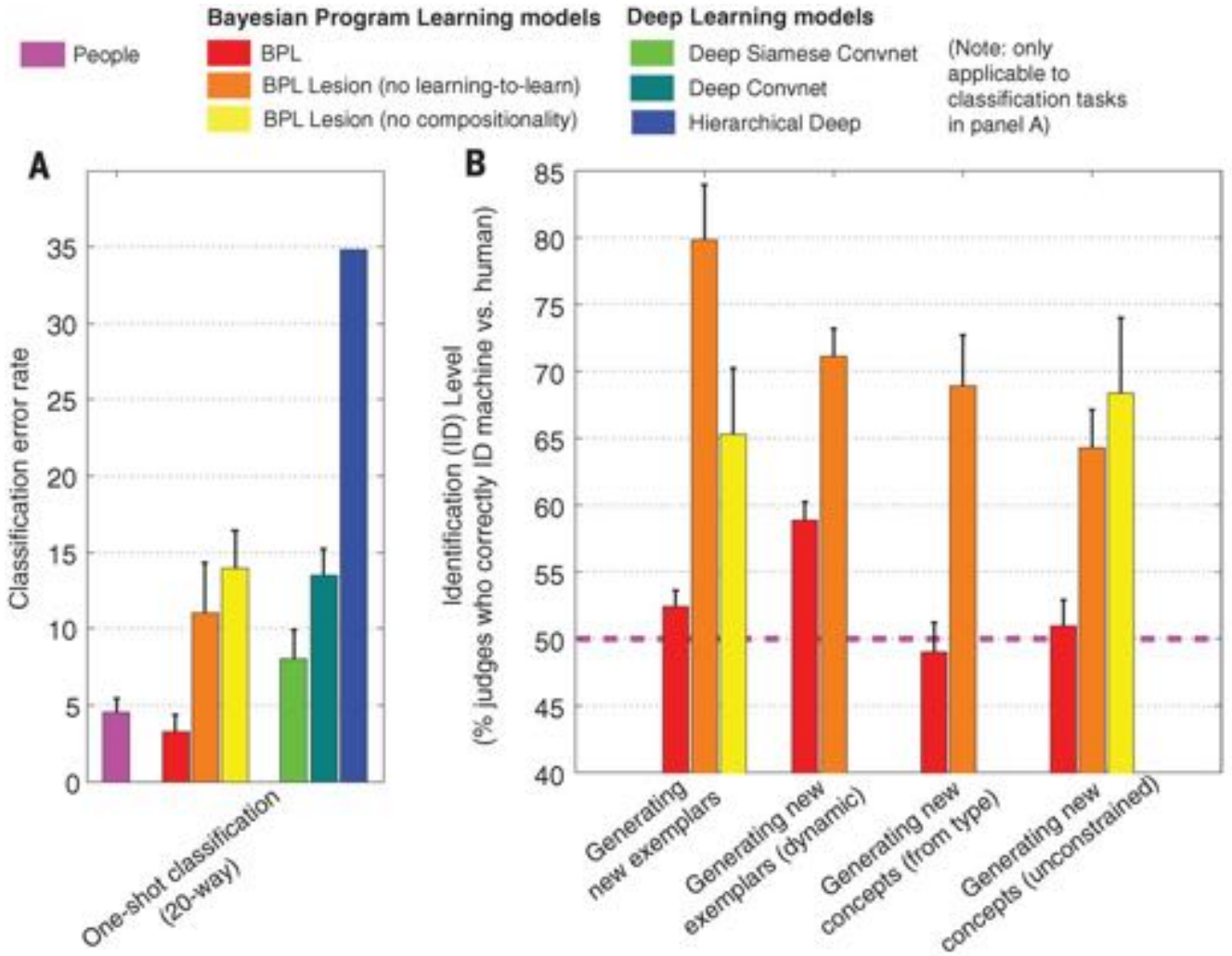
Brenden M. Lake et al. Science 2015;350:1332-1338

Fig. 5 Generating new exemplars.



Brenden M. Lake et al. Science 2015;350:1332-1338

Fig. 6 Human and machine performance was compared on (A) one-shot classification and (B) four generative tasks.

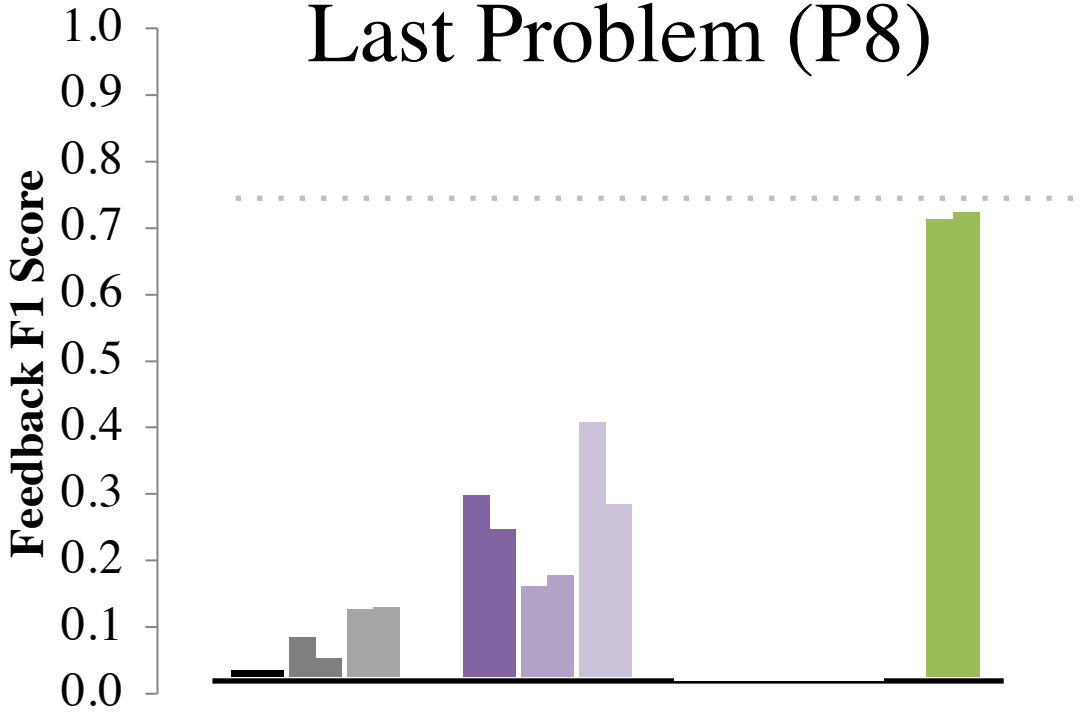


Brenden M. Lake et al. Science 2015;350:1332-1338

Generative Understanding

Label student code

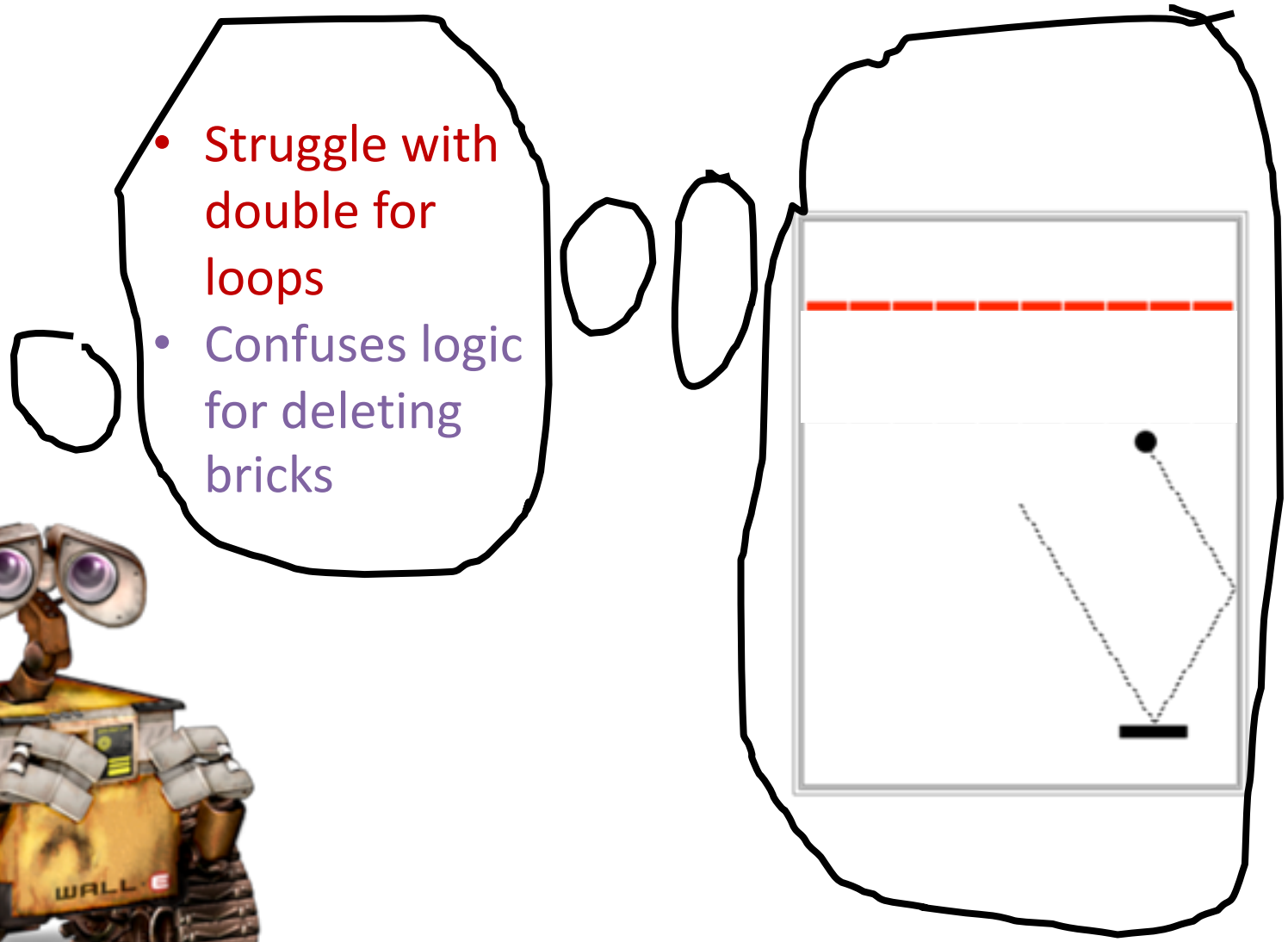
Last Problem (P8)



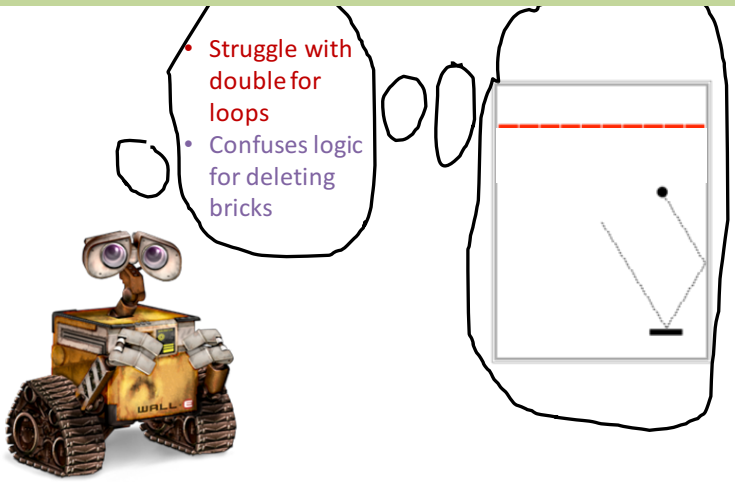
- Struggle with double for loops
- Confuses logic for deleting bricks

Imagine Students

- Struggle with double for loops
- Confuses logic for deleting bricks



Imagine Students



This is easy and exponential

This is hard and linear

A students "ability"

$$\Theta \sim \text{pythonSample}$$

A students "choices"

$$C \sim \text{pythonSample}|\Theta$$

The resulting code

$$\Pi \sim \text{pythonSample}|C$$

$$P(\Theta, C|\Pi)$$

Infer ability and choices from code

Bayesian Programming Language

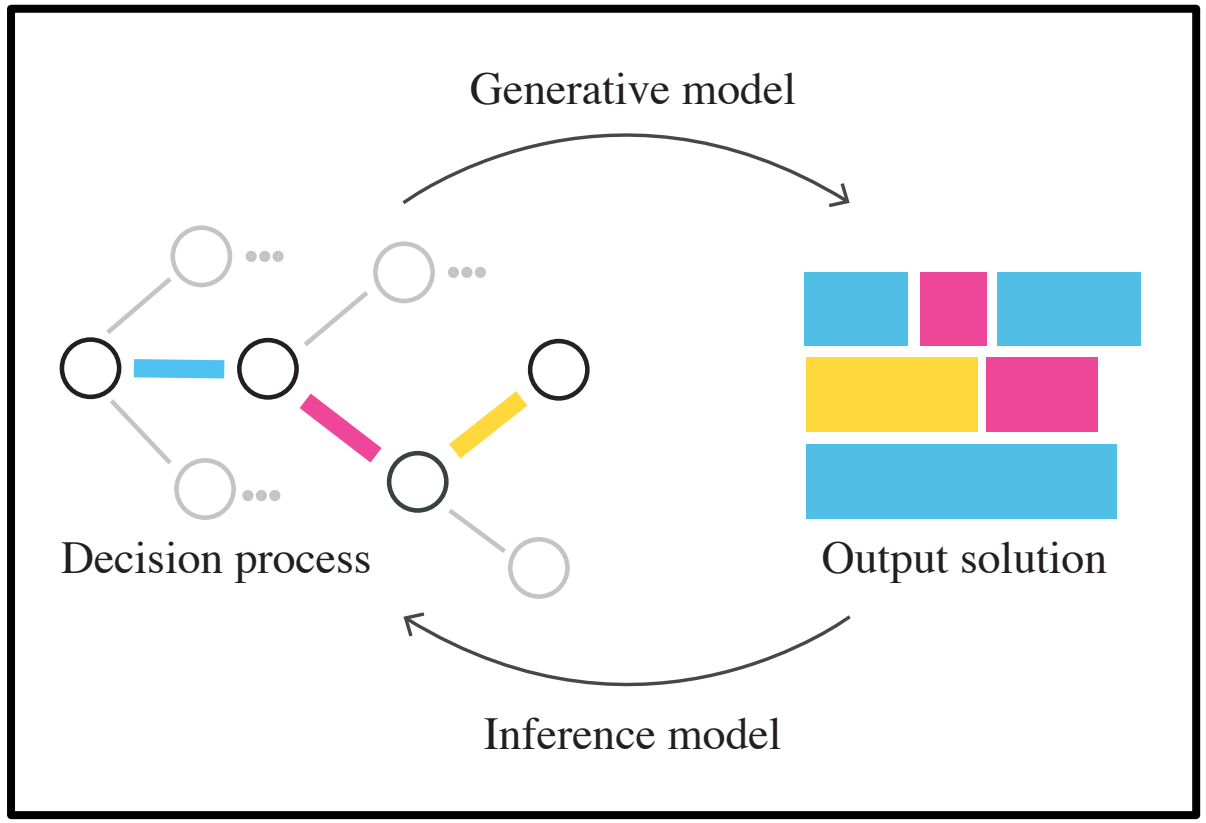
`ideaToText`

Teachers Articulate N misconceptions

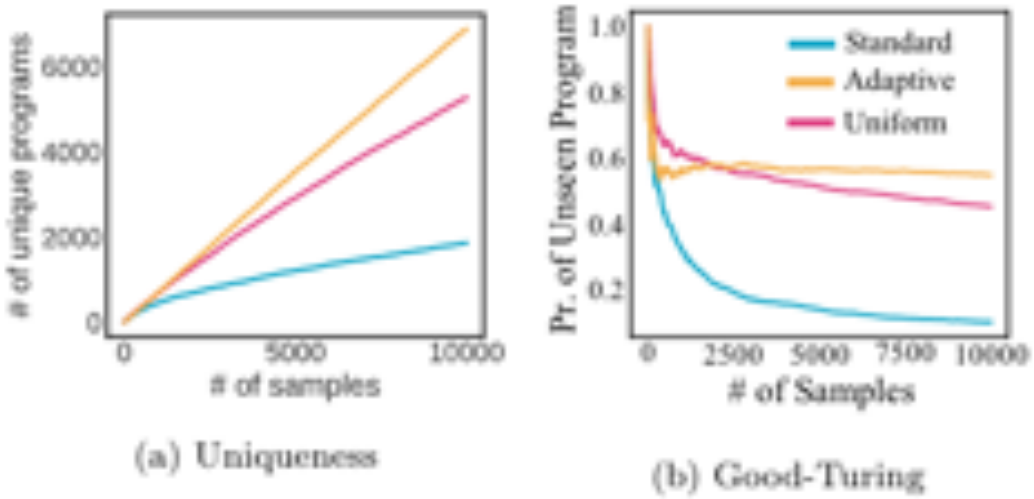
```
# This python class is a RubricSampling Decision
# it generates programs that print the numbers 10 -> 1
class Countdown(Decision):

  def registerChoices(self):
    # these are the main strategies for printing out a
    # countdown
    self.addRubricChoices('loop-style', {
      'for' : θ1 ,
      'while' : θ2,
      'none' : θ3
      'empty' : θ4
    })

  # we can make some grading choices based on which
  # strategy they chose (did they actually use a loop?)
  def processChoices(self):
    style = self.getChoice('loop-style')
    hasLoop = style != 'none' and style != 'empty'
    self.addLabel('rubric-hasLoop', hasLoop)

  # Based on their strategy render a different decision
  def renderCode(self):
    style = self.getChoice('loop-style')
    if style == 'for': return '{ForSoln}'
    if style == 'while': return '{WhileSoln}'
    if style == 'none': return '{NoLoopSoln}'
    if style == 'empty': return ''
```

1. This is code for a single decision point
2. Give a name to the choice that the student is making
3. How do those choices translate into feedback?
4. What does the code look like? Often evokes other decision points

Generative Understanding



Idea: (1) sample a ton, then(2) build a neural network to learn to predict decisions



Decision process

Next choice

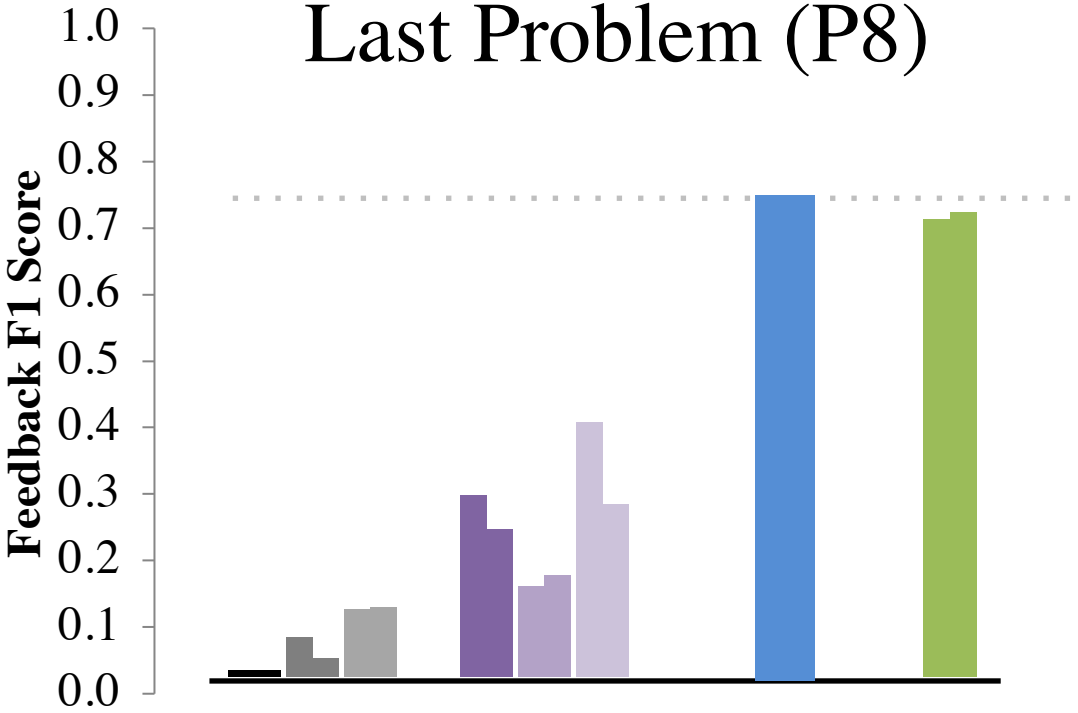
$$p_{\mathcal{G}}(x_{a_1}, \ldots x_{a_T}|y) = \prod_{t=1}^{T} p_{\mathcal{G}}(x_{a_t}|y, \mathbf{x}_{<a_t})$$

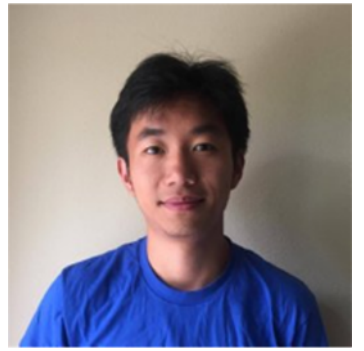
solution

Previous choices

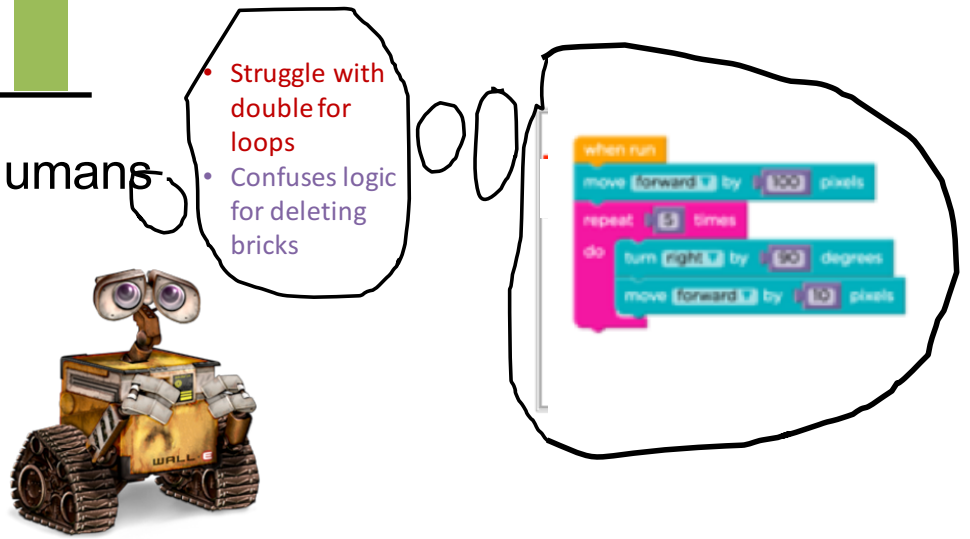
Generative Understanding

Label student code





Outstanding Student paper award, AAAI 2019



Not just for code

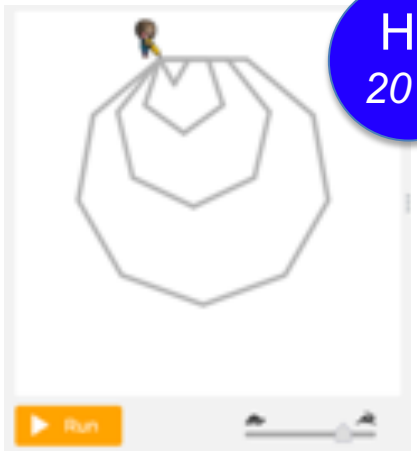
Results from early 2019

Many domains of student work

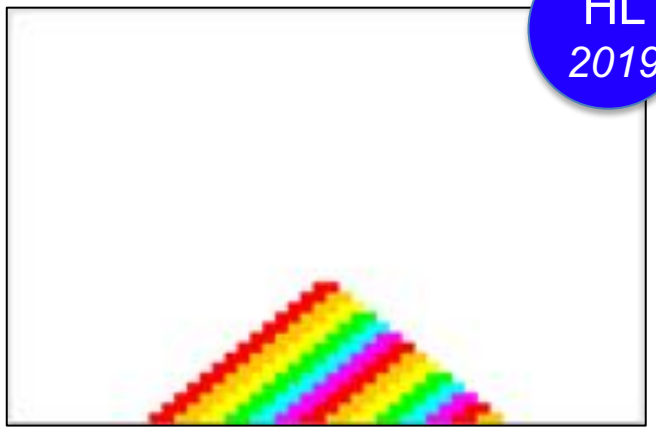
HL 2017

$$9 \times 6$$
$$= (10 - \square) \times 6$$
$$= 10 \times 6 - \square \times 6$$
$$= 60 - \square$$
$$= \square$$

HL 2019



HL 2019

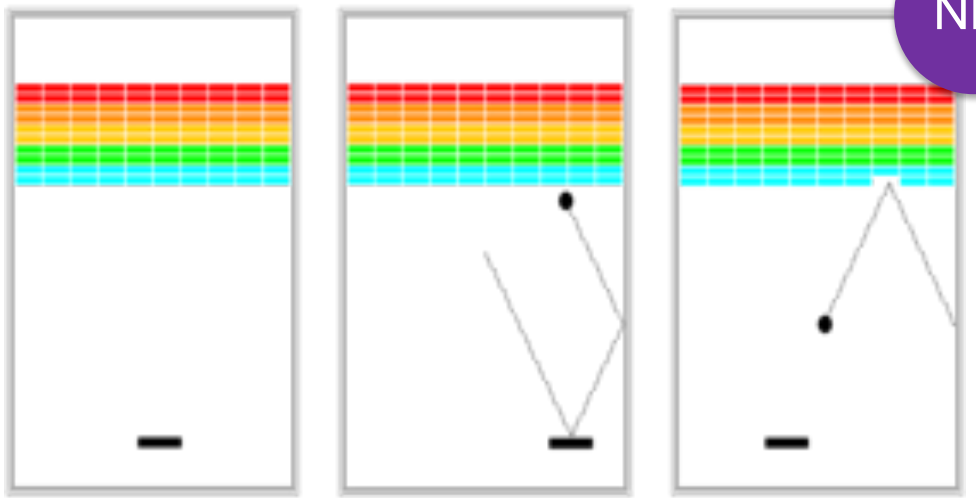


NH 2019

Why did the original colonists come to America?

SH

NH?



NH Near Human

HL Human Level

SH Super Human Level

So what?

What does this mean for me?!?

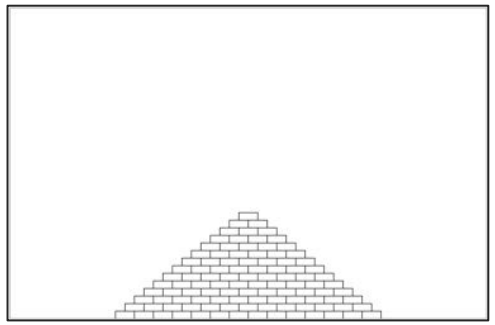
Cutting Edge @ Stanford

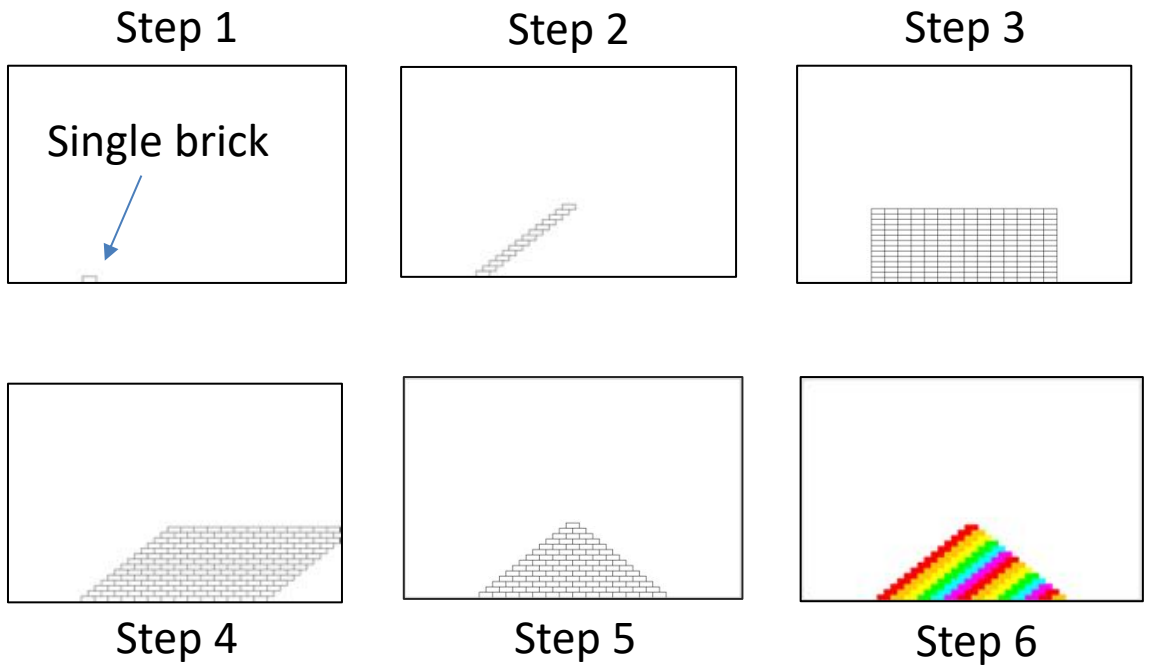
2,600 students

130,000 partial solutions

μ snapshots per student = 50

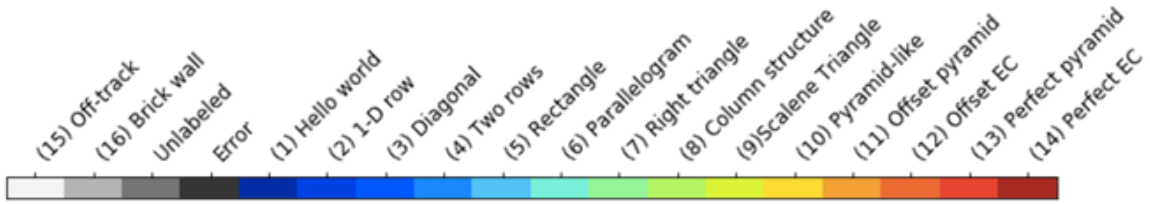
μ time per student = 2 hours



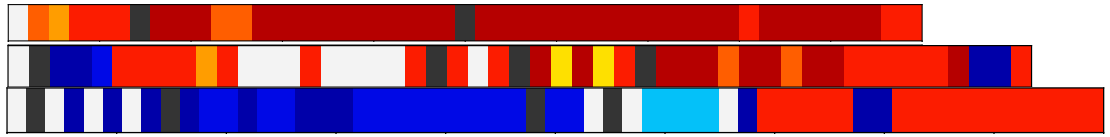


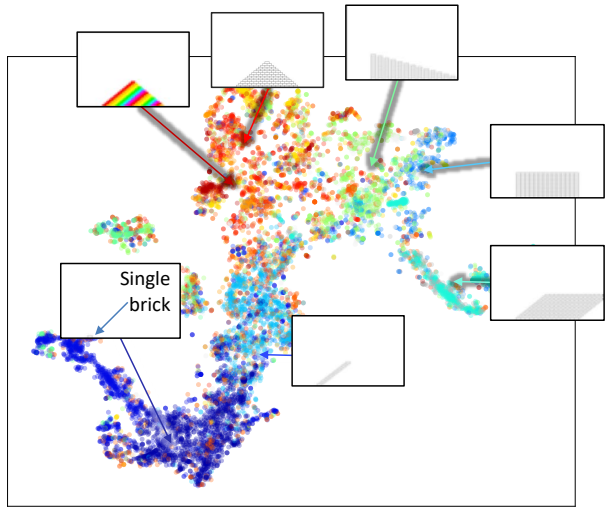
Lisa Yan, Nick McKeown, Chris Piech. Understanding students from visual output. SIGCSE 2019

Cutting Edge @ Stanford



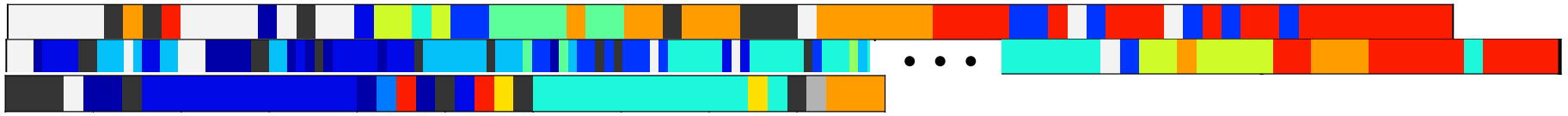
Students scoring in 99th percentile on midterm exam

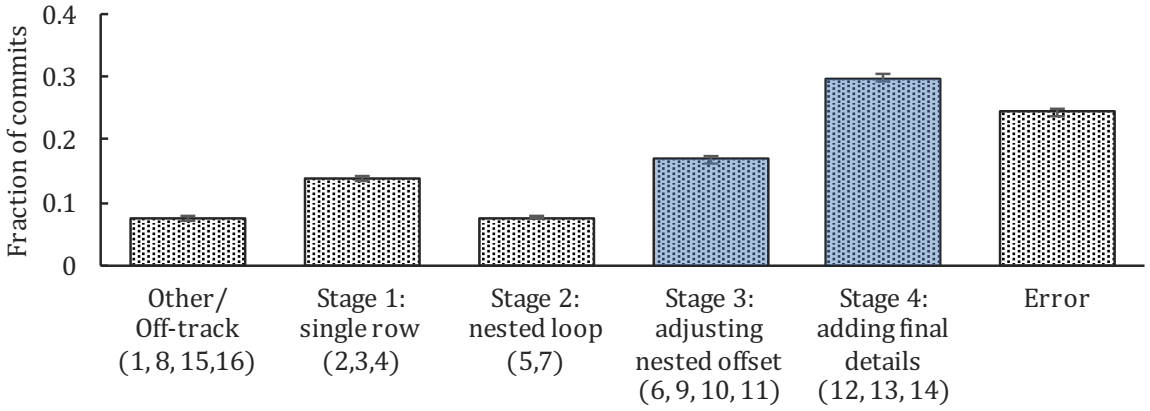




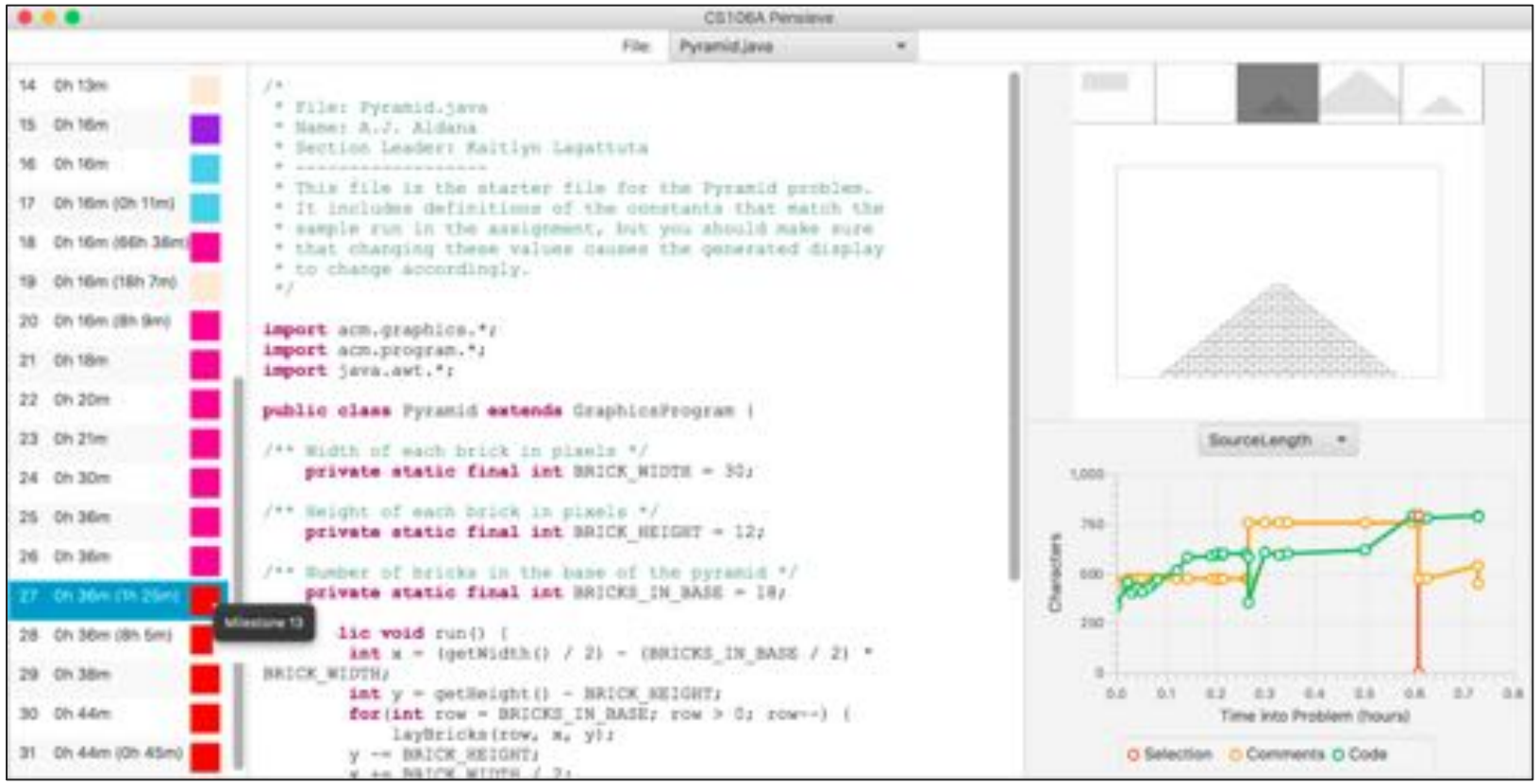
t-SNE embedding of 130,000 partial solutions

Students scoring in ≤3rd percentile on midterm exam





Cutting Edge @ Stanford

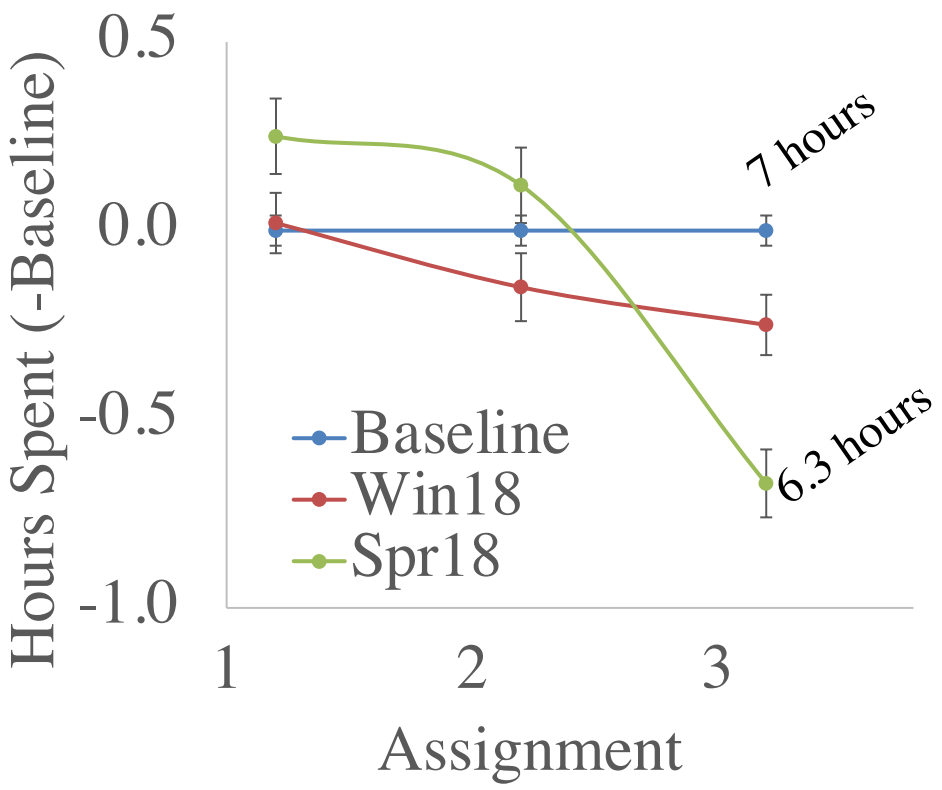


One quarter, all students were *shown* their progress:

- Early correction of bad habits.
- Chance to teach the art of programming.
- Academic dishonesty becomes much harder.

Cutting Edge @ Stanford

Using assignment *timing* as pre-post

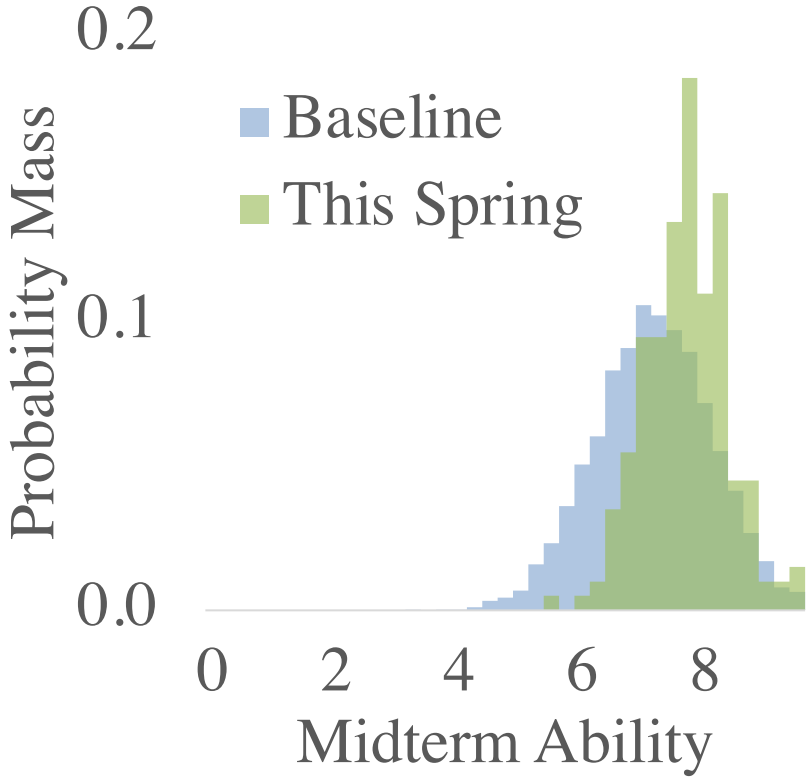


$$E[\hat{X}_3|X_1] - E[X_3] = 42 \text{ mins} \quad p < 0.00001$$

Predicted time — Actual time

Assignments are taking less time

Item Response Theory based ability assessment



$$S_{i,j} = n \cdot \sigma(a_i - d_j)$$

Score — points — ability — difficulty

Students perform better than expected

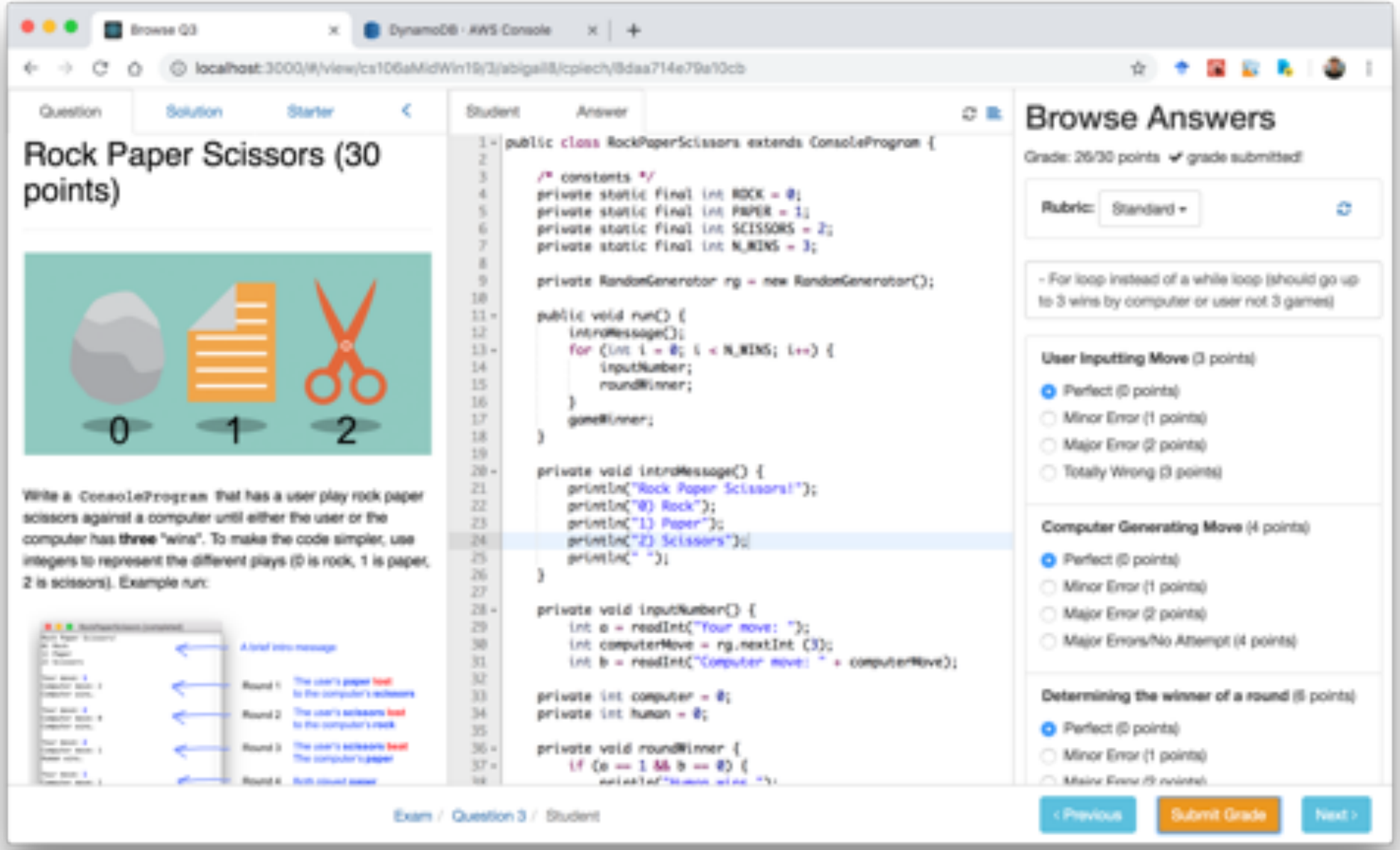
Next trial

Code in Place 2021

But its not perfect

Hit a ceiling with code > 20 lines

PurpleBook

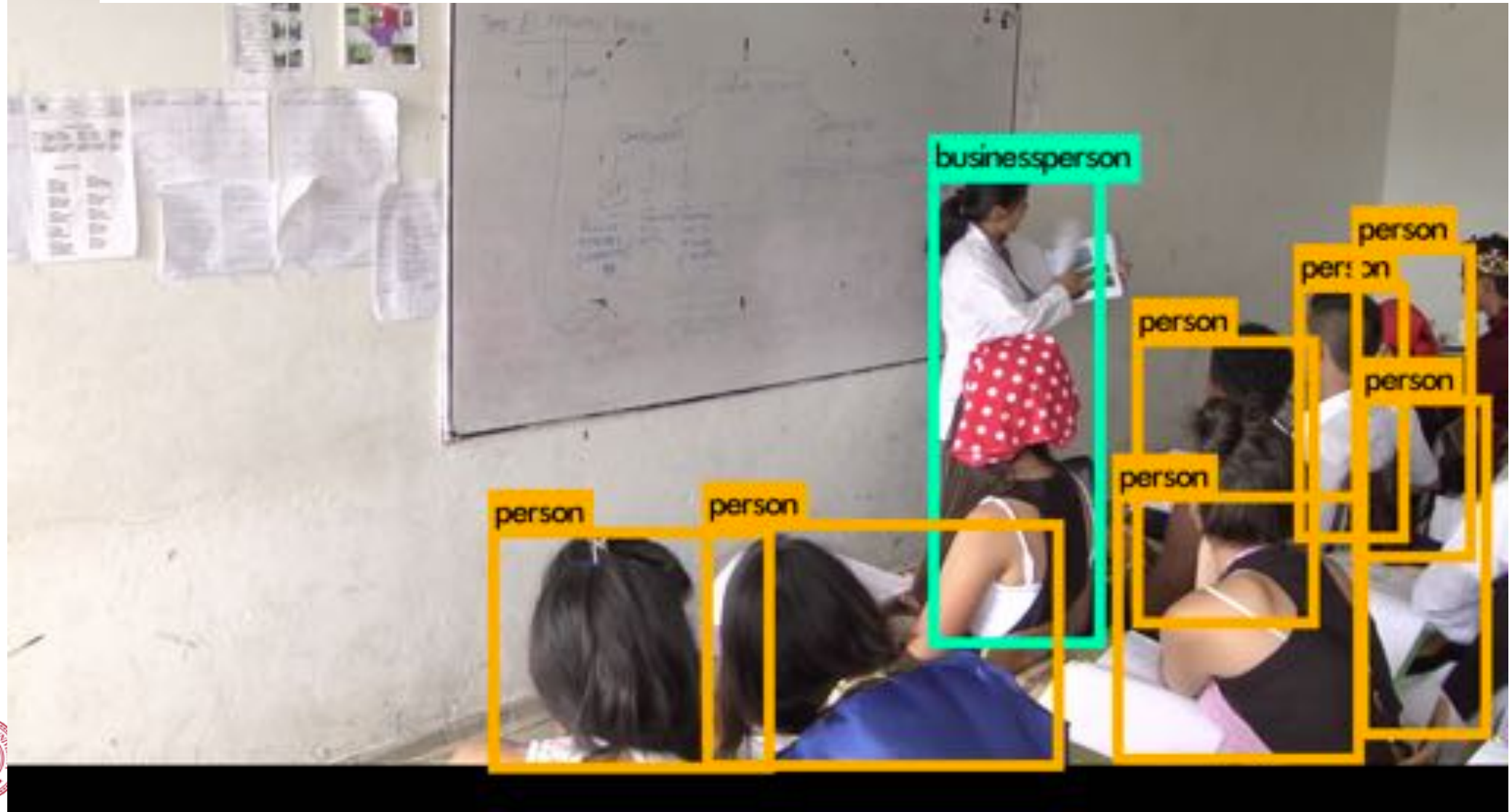

Browse Q3
DynamoDB - AWS Console
Question
Solution
Starter
Rock Paper Scissors (30 points)
0
1
2
Write a ConsolePrograms that has a user play rock paper scissors against a computer until either the user or the computer has three "wins". To make the code simpler, use integers to represent the different plays (0 is rock, 1 is paper, 2 is scissors). Example run:
Student
Answer
public class RockPaperScissors extends ConsoleProgram {
/* constants */
private static final int ROCK = 0;
private static final int PAPER = 1;
private static final int SCISSORS = 2;
private static final int N_WINS = 3;
private RandomGenerator rg = new RandomGenerator();
public void run() {
introMessage();
for (int i = 0; i < N_WINS; i++) {
inputNumber;
roundWinner;
}
gameWinner;
}
private void introMessage() {
println("Rock Paper Scissors!");
println("0) Rock");
println("1) Paper");
println("2) Scissors");
println(" ");
}
private void inputNumber() {
int a = readInt("Your move: ");
int computerMove = rg.nextInt (3);
int b = readInt("Computer move: " + computerMove);
private int computer = 0;
private int human = 0;
private void roundWinner {
if (a == 1 && b == 0) {
Browse Answers
Grade: 26/30 points ✔ grade submitted!
Rubric: Standard
- For loop instead of a while loop (should go up to 3 wins by computer or user not 3 games)
User Inputting Move (3 points)
Perfect (0 points)
Minor Error (1 points)
Major Error (2 points)
Totally Wrong (3 points)
Computer Generating Move (4 points)
Perfect (0 points)
Minor Error (1 points)
Major Error (2 points)
Major Errors/No Attempt (4 points)
Determining the winner of a round (6 points)
Perfect (0 points)
Minor Error (1 points)
Exam / Question 3 / Student
‹ Previous
Submit Grade
Next ›

Coming up

New problems

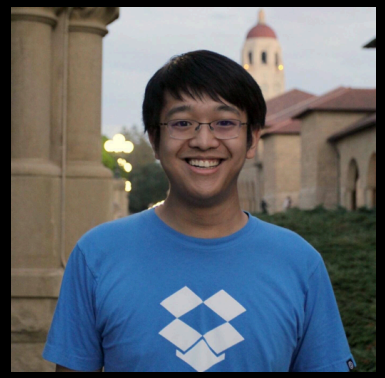
Feedback for Teachers

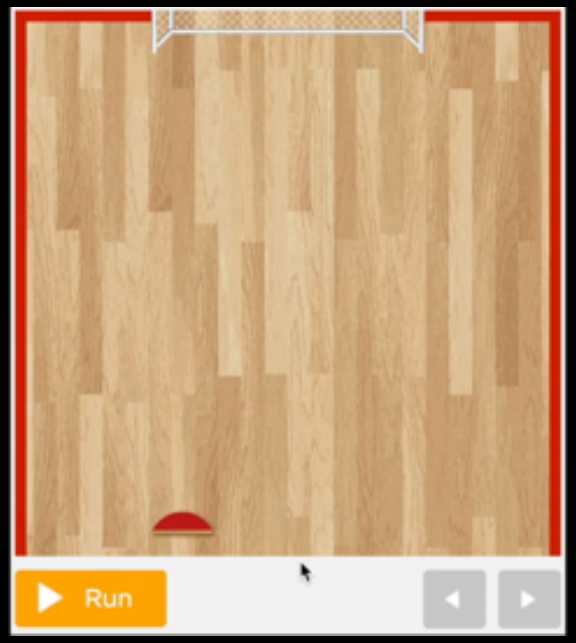
200,000 videos of teachers in Colombia, Chile and USA teaching

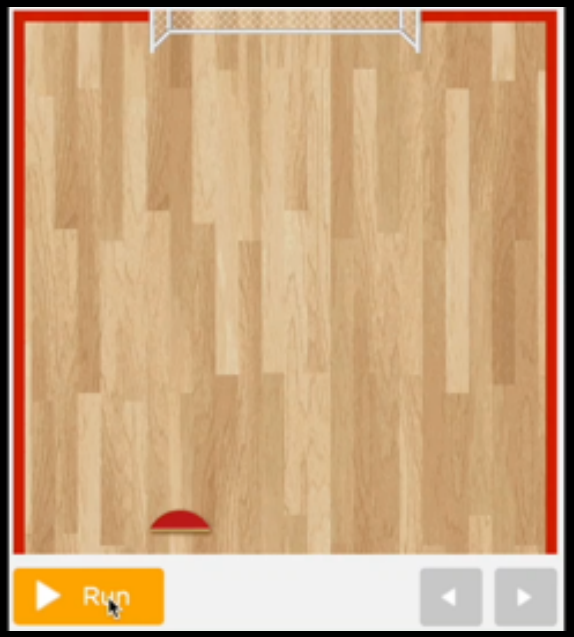


Rotation: play to grade

Allen Nie







Problem:
Grade the ~1M unique student implementations of this problem on code.org

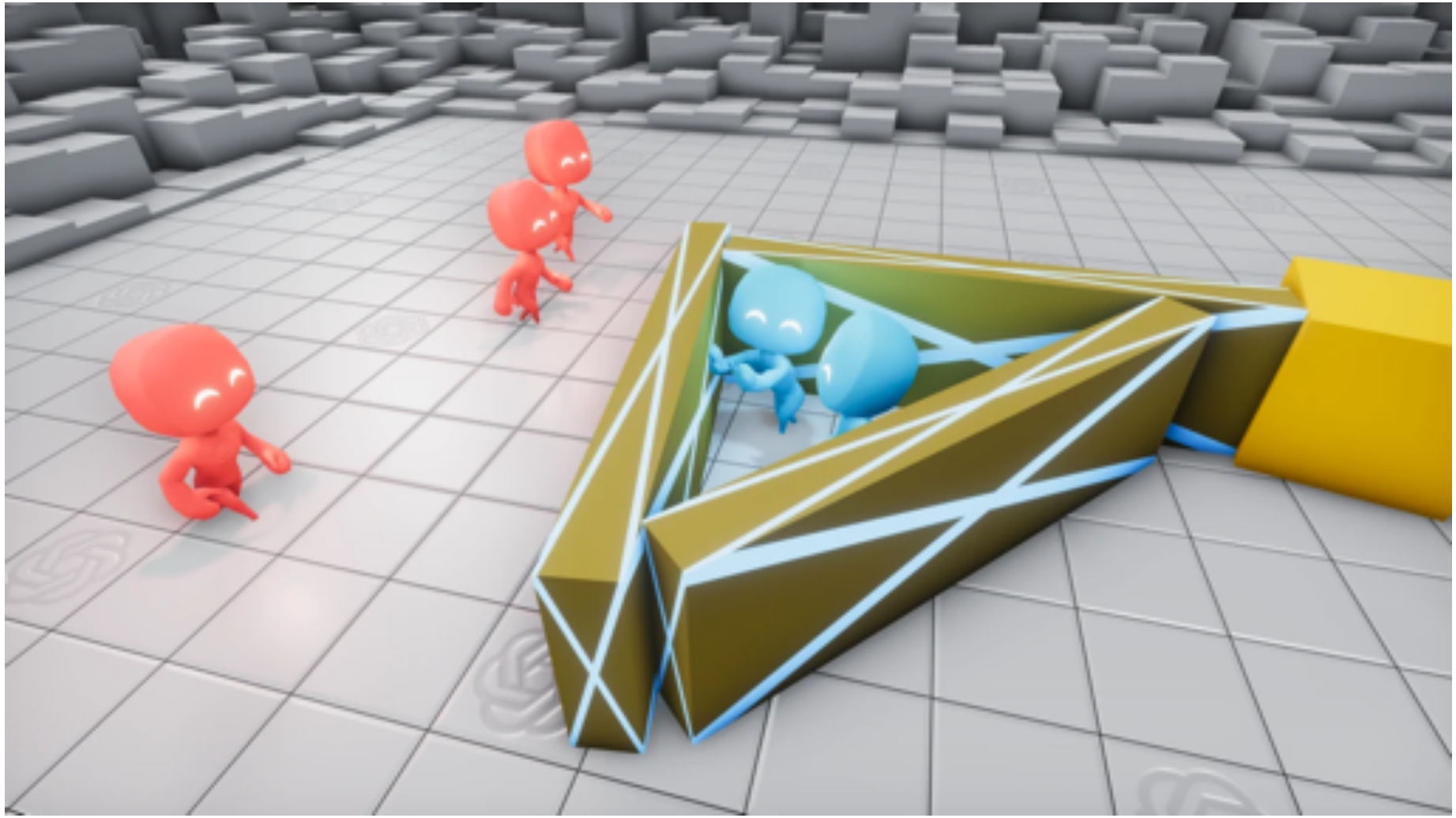
Input:
Teacher gives you one example of each mistake on their rubric and one example of invariances

Impact:
Immediately change what sort of assignments are auto-gradable

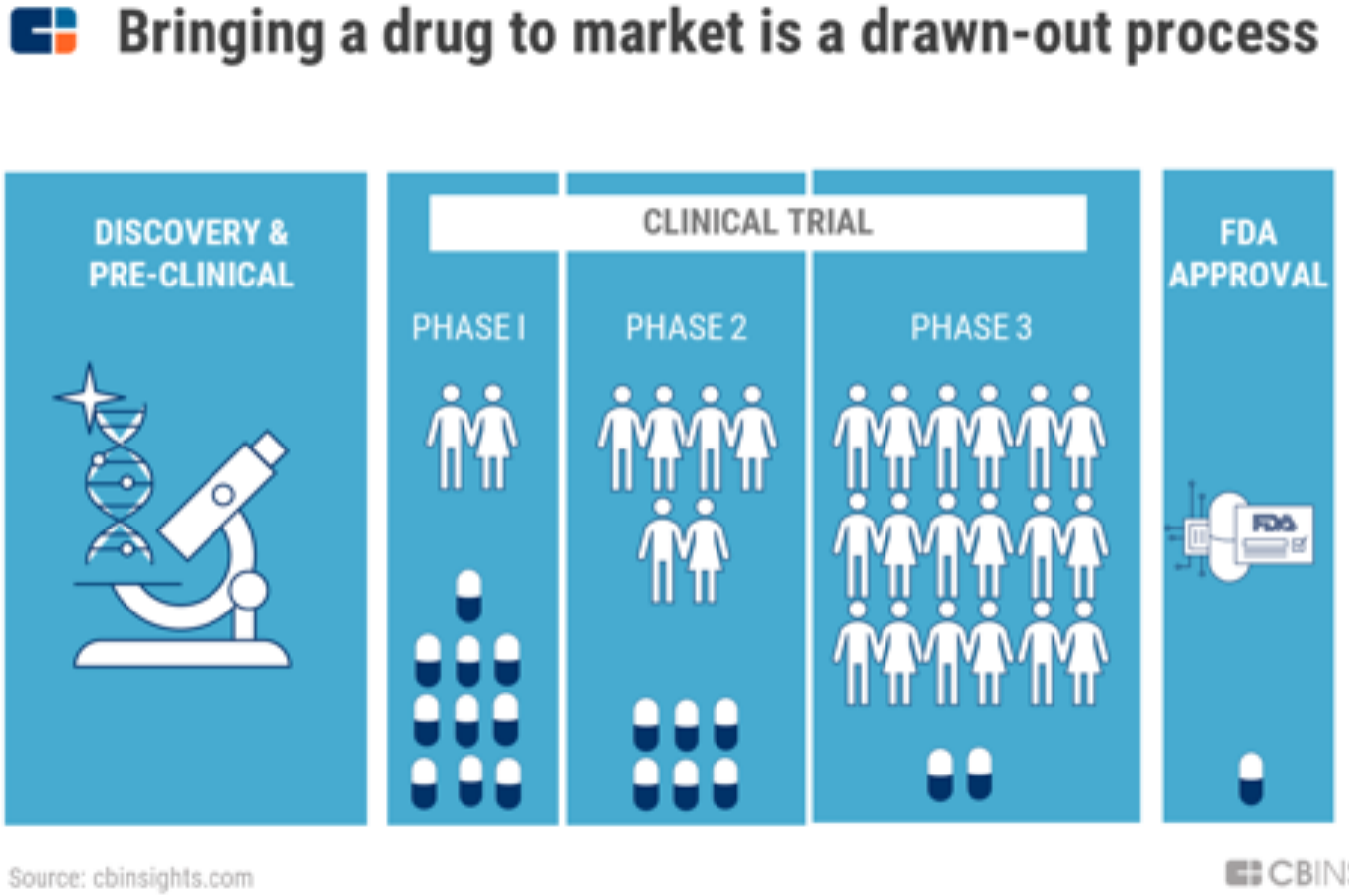
Theory contribution:
First model to build deep RL for a classification task . *Instead of learning an environment you are learning to test an environment*

Open Research Problems

Teacher Gym



More than education



Application -> Theory

Understand social science, especially with small data

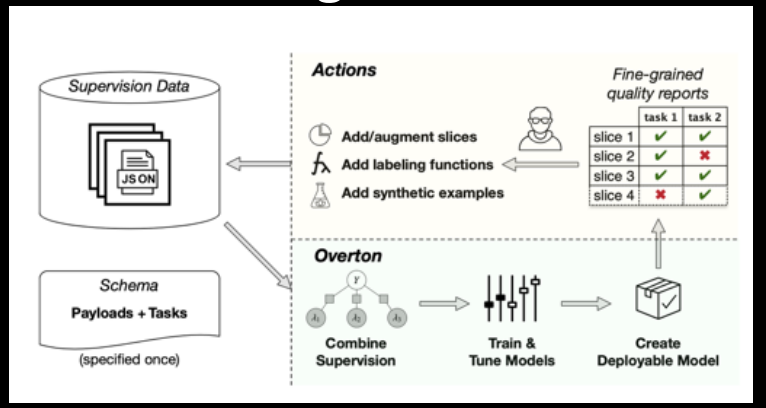
Explain why it made the choices it did

What are things that AI currently can't do?

What are things that AI currently can't do?

Teach humans based on what it has learned

Design itself



If time

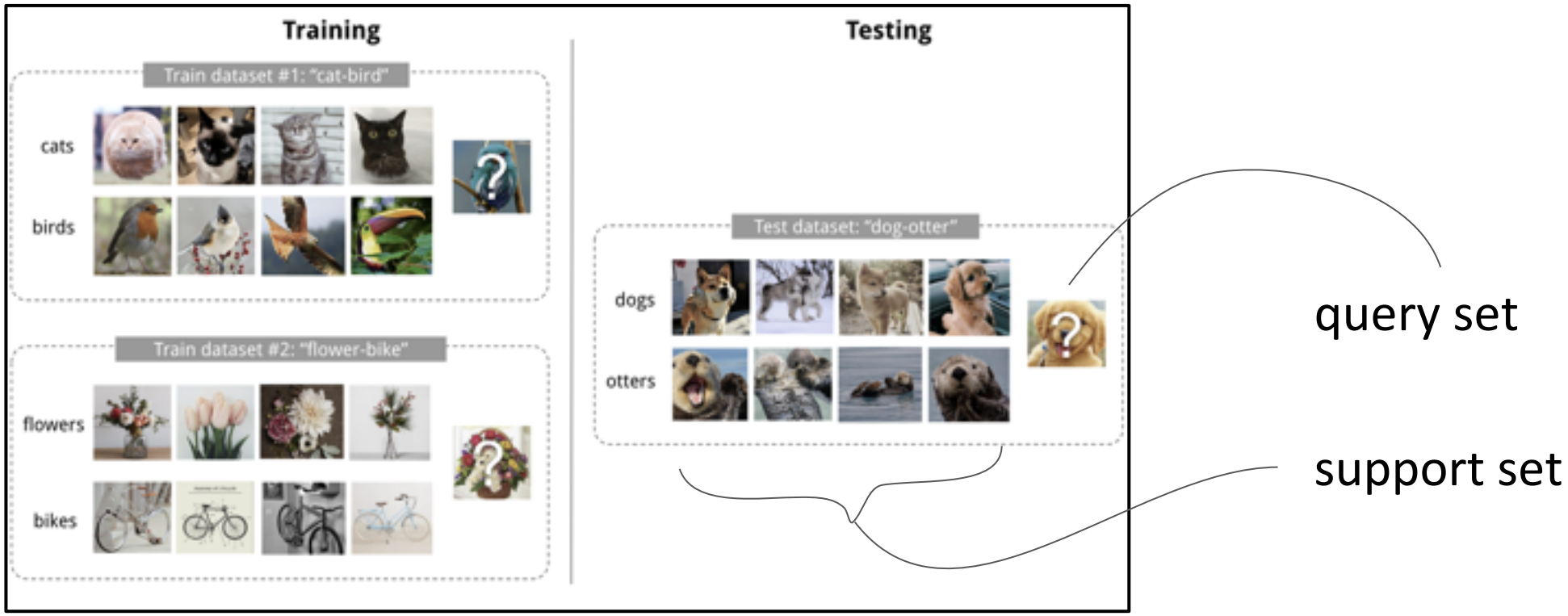
Meta Learning

https://lilianweng.github.io/lil-log/2018/11/30/meta-learning.html

Meta

Chelsea Finn

Goal: Divide **tasks** into meta-training and meta-test splits. Use the meta-training set to learn a model and evaluate performance on meta-test set.



Meta Grading AI

Two different splits (“easy” and “hard”).

Meta Grading AI

Two different splits (“easy” and “hard”).

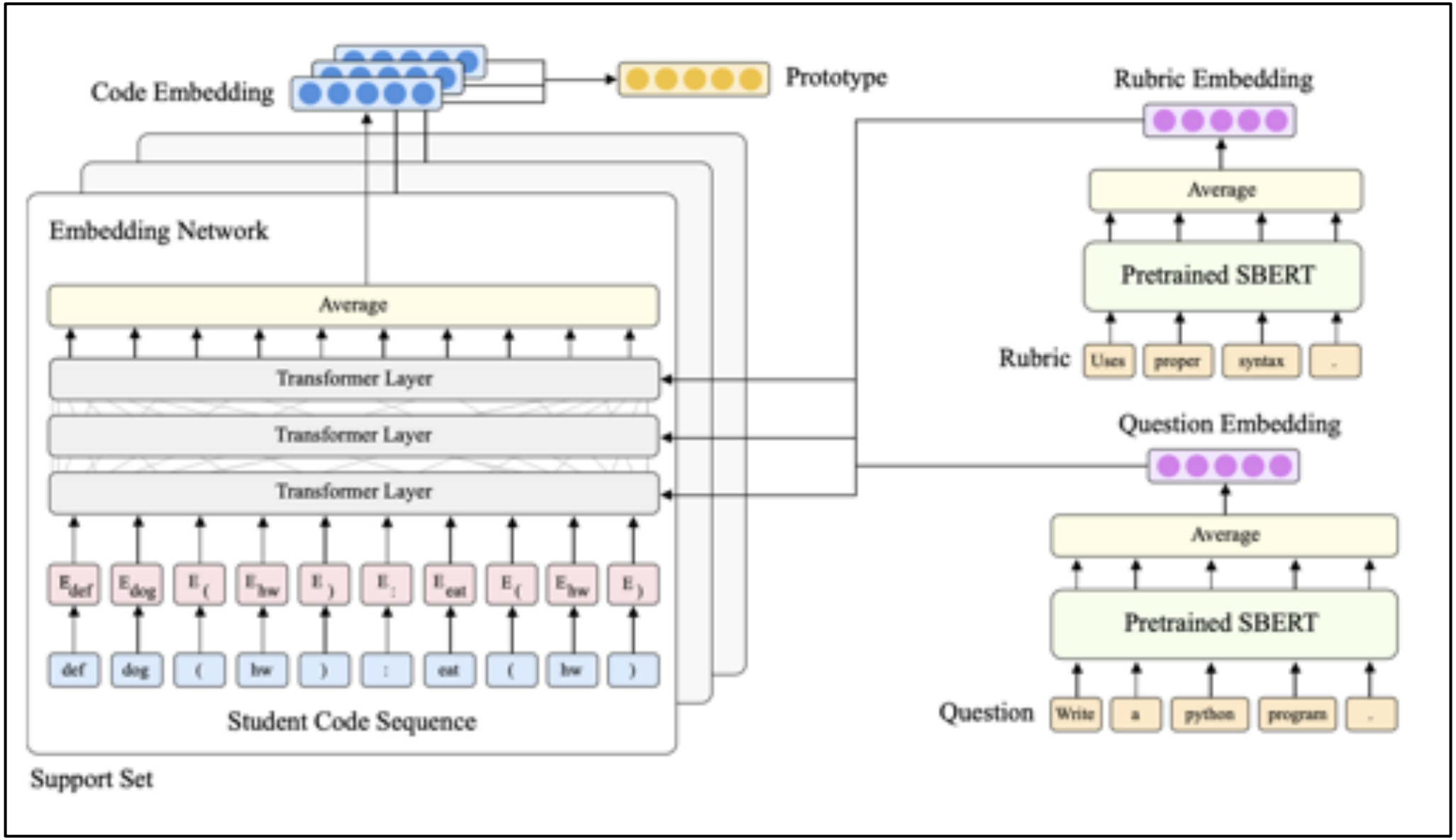
Held-out rubric items (“easy”): reserve 10% of rubric items for meta-test tasks. Same student programs can appear in both splits but for different items.

Meta Grading AI

Two different splits (“easy” and “hard”).

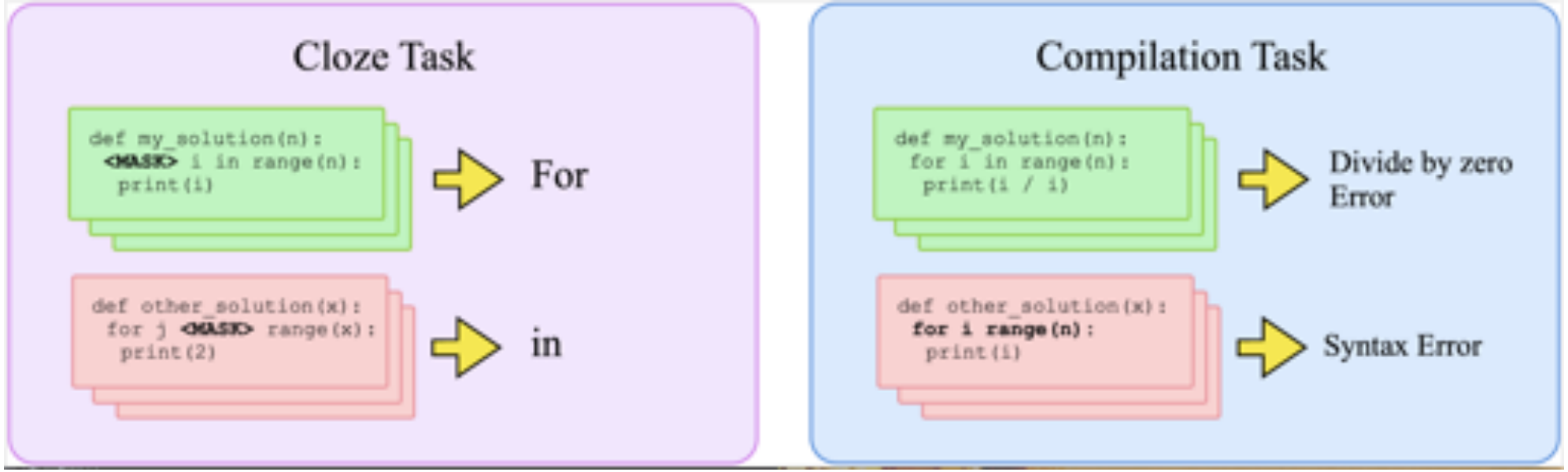
Held-out rubric items (“easy”): reserve 10% of rubric items for meta-test tasks. Same student programs can appear in both splits but for different items.

Held-out exams (“hard”): reserve 10% of exams for meta-test tasks. Truly unseen problems at test time.


Code Embedding
Prototype
Rubric Embedding
Average
Pretrained SBERT
Rubric
Uses
proper
syntax
.
Embedding Network
Average
Transformer Layer
Transformer Layer
Transformer Layer
Question Embedding
Average
Pretrained SBERT
Question
Write
a
python
program
.
def
dog
(
hw
)
:
eat
(
hw
)
Student Code Sequence
Support Set

Trick #1: Task Augmentation

259 is not a lot of tasks. Meta-learning usually operates on 1000s of tasks. We apply the “data augmentation” idea to coding tasks!



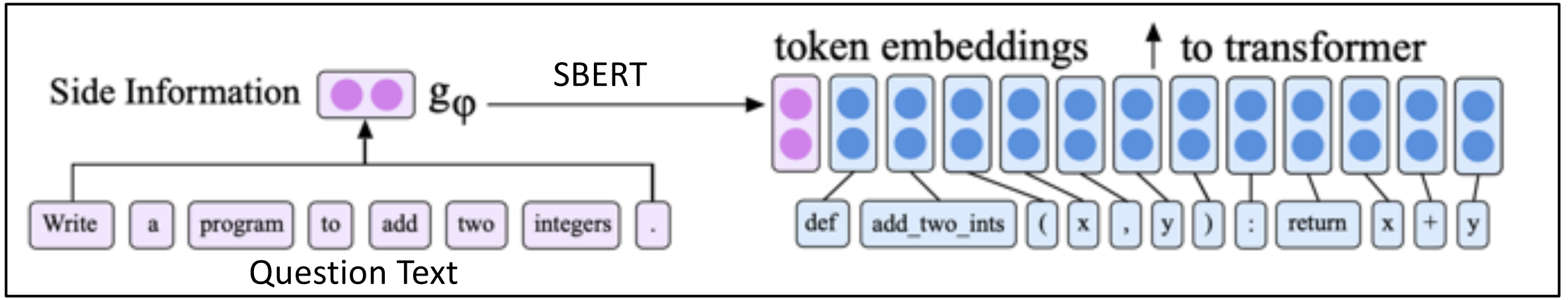
Trick #2: Side Information

A task is only composed of 10 or 20 examples, leaving a lot of ambiguity. Suppose we have "side information" $z = (z_1, z_2, ..., z_T)$ about each task: **rubric option name** and **question text**. How do we add this side information into our embedding function f_θ?

Few-shot Sequence Learning with Transformers (Logeswaran et. al. 2020)
Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks (Reimers and Gurevych 2019)

Trick #2: Side Information

A task is only composed of 10 or 20 examples, leaving a lot of ambiguity. Suppose we have "side information" $z = (z_1, z_2, ..., z_T)$ about each task: **rubric option name** and **question text**. How do we add this side information into our embedding function f_θ?

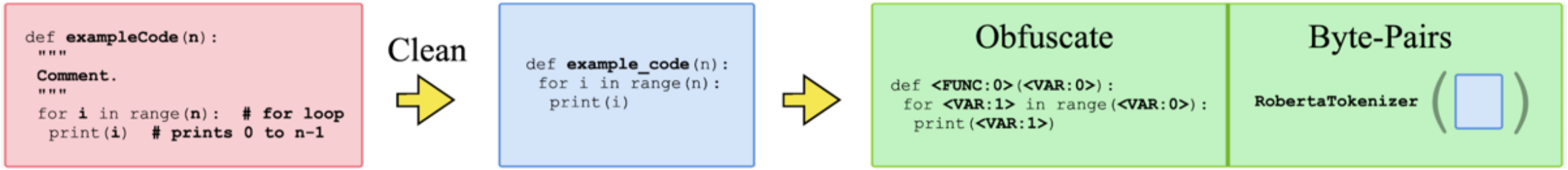


Prepend side information as a first token.

Trick #3: Variable and Function Names

99% of our vocabulary is for variables and functions. This makes learning hard so we try two approaches.

1. Convert names to snakecase + use bytepair encodings.
2. Allocate a fixed number of tokens that the model can use to represent variables and functions.



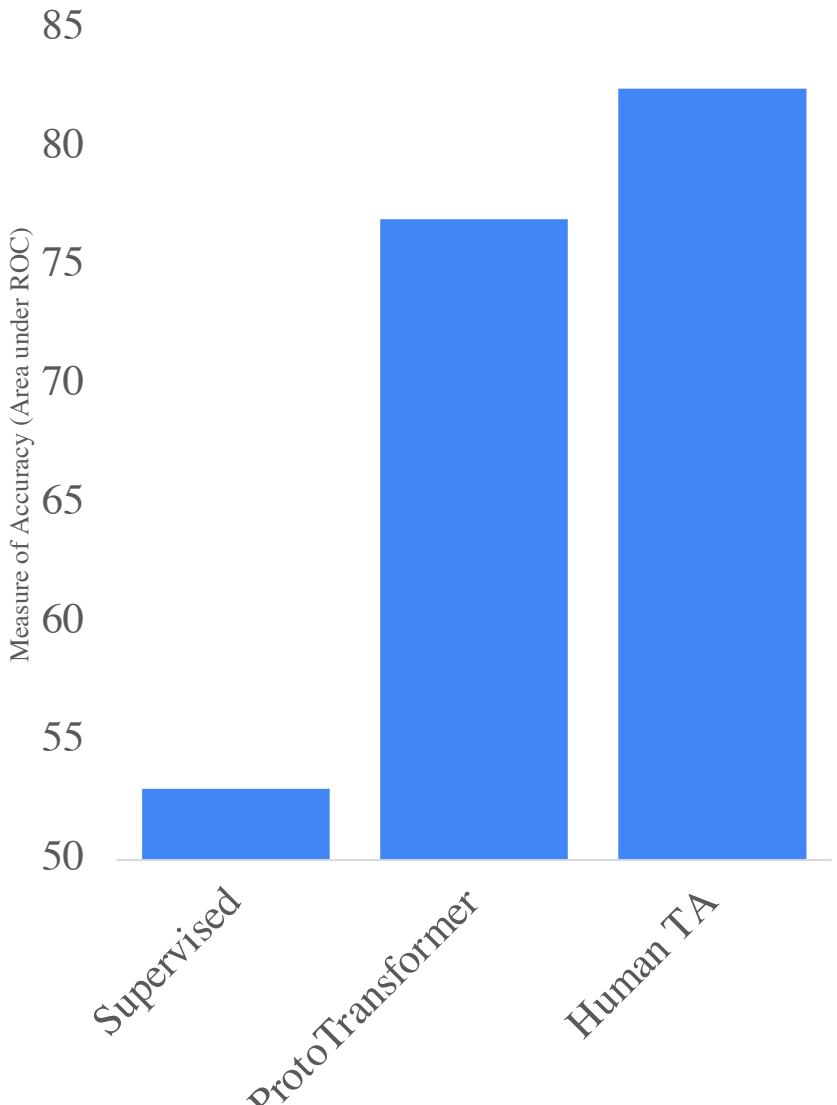
CodeBERT: A Pre-Trained Model for Programming and Natural Languages (Feng et. al. 2020
CodeSearchNet Challenge: Evaluating the State of Semantic Code Search (Husain et.al. 2020)

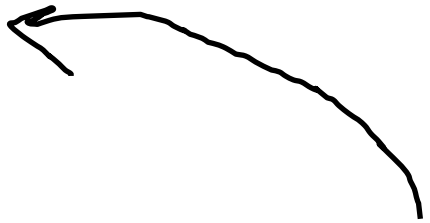
Trick #4: Code Pre-training

Can we utilize large unlabeled datasets of code to help the model learn a good prior for code?

In practice, we initialize the embedding network from pretrain weights and finetune top M layers.

Ability to Grade Unseen Exam (with 10 labels)



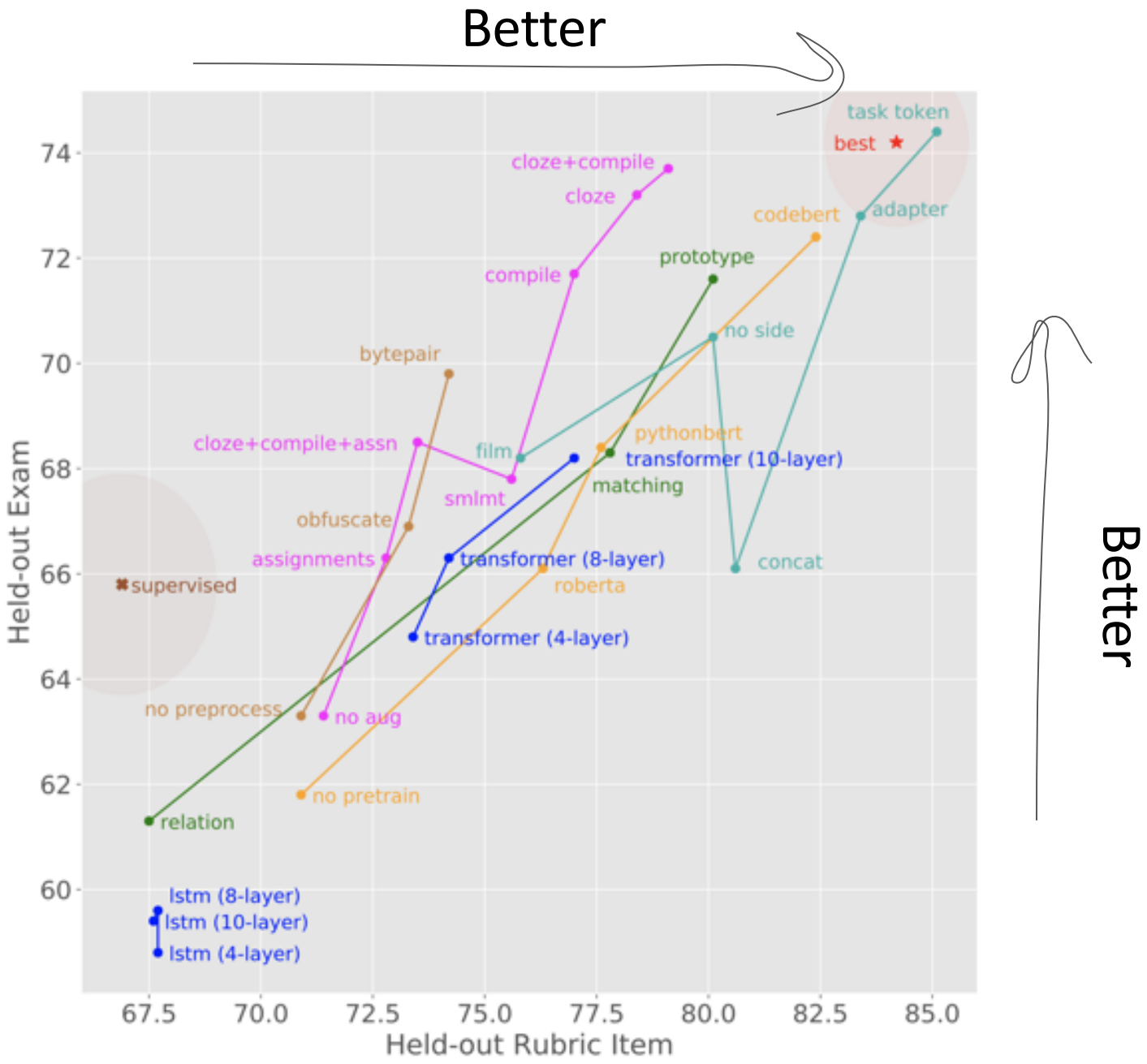


Room to grow!

Ablations

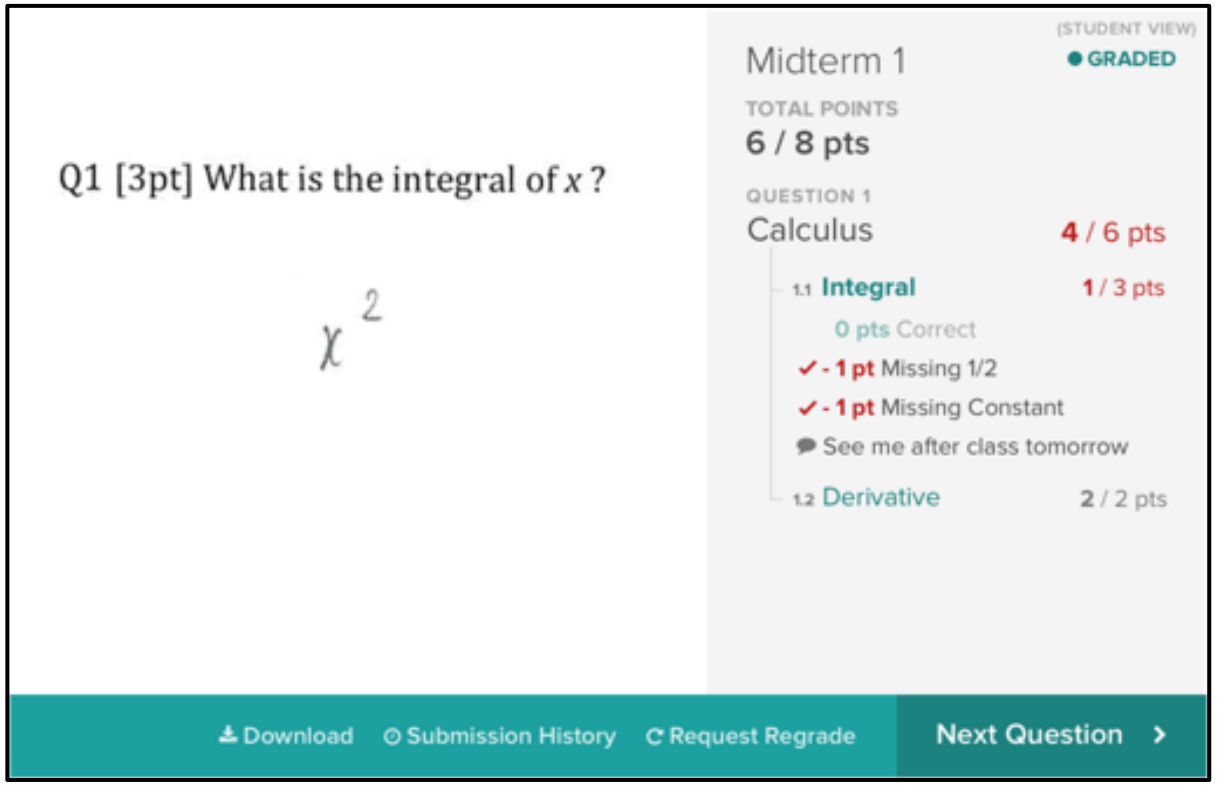
Legend

-task aug
-naming
-architecture
-side info
-pretraining
-meta algo
-supervised
-best

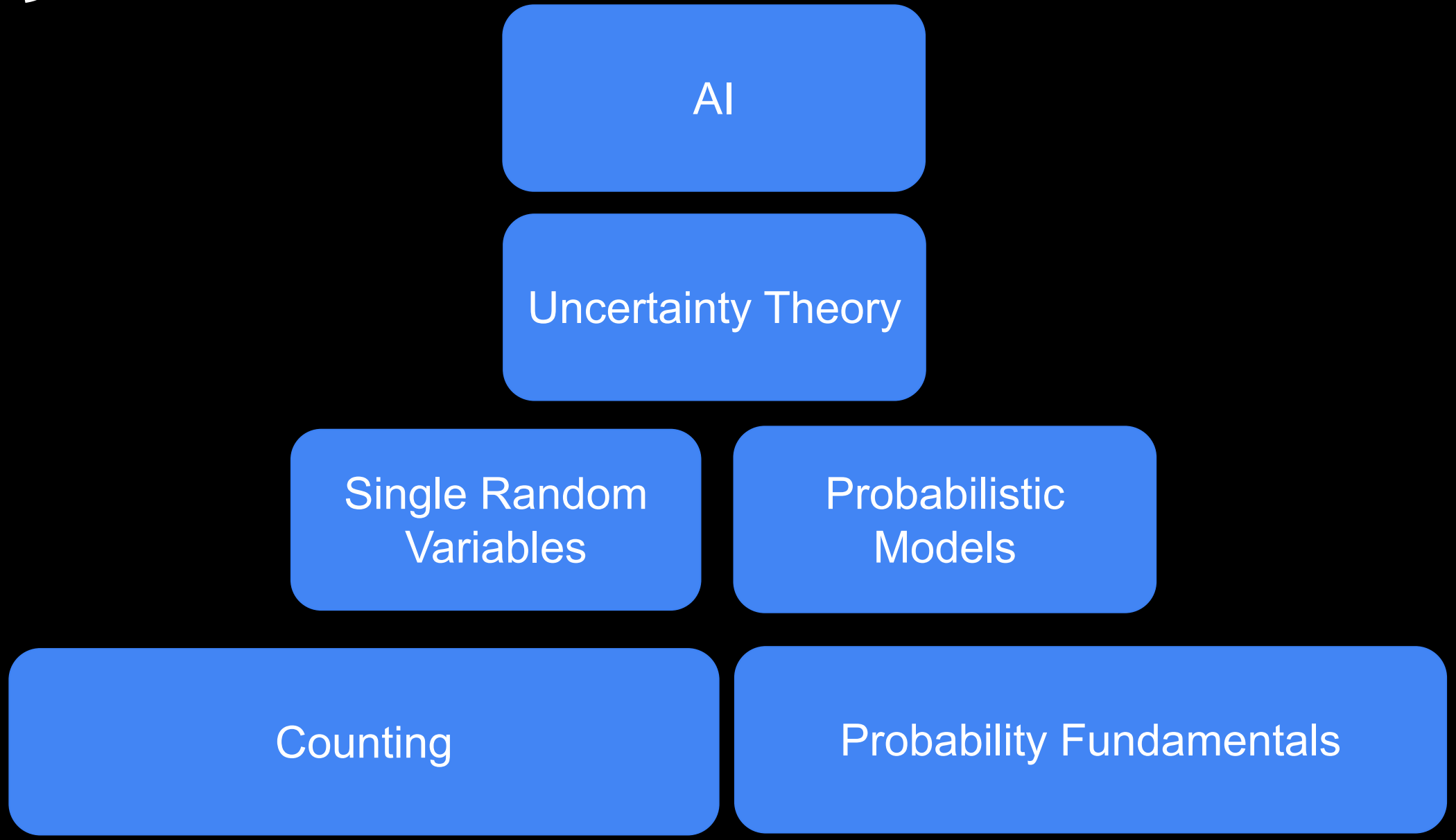


What's Next?

- **Do this at scale** (more data, more modalities, more diverse assignments). Collaboration with Gradescope?



CS109


AI
Uncertainty Theory
Single Random Variables
Probabilistic Models
Counting
Probability Fundamentals

Abundance of important problems