

Few-Shot Learning in the Real World

Meta-Learning for Giving Feedback to Students

Chelsea Finn

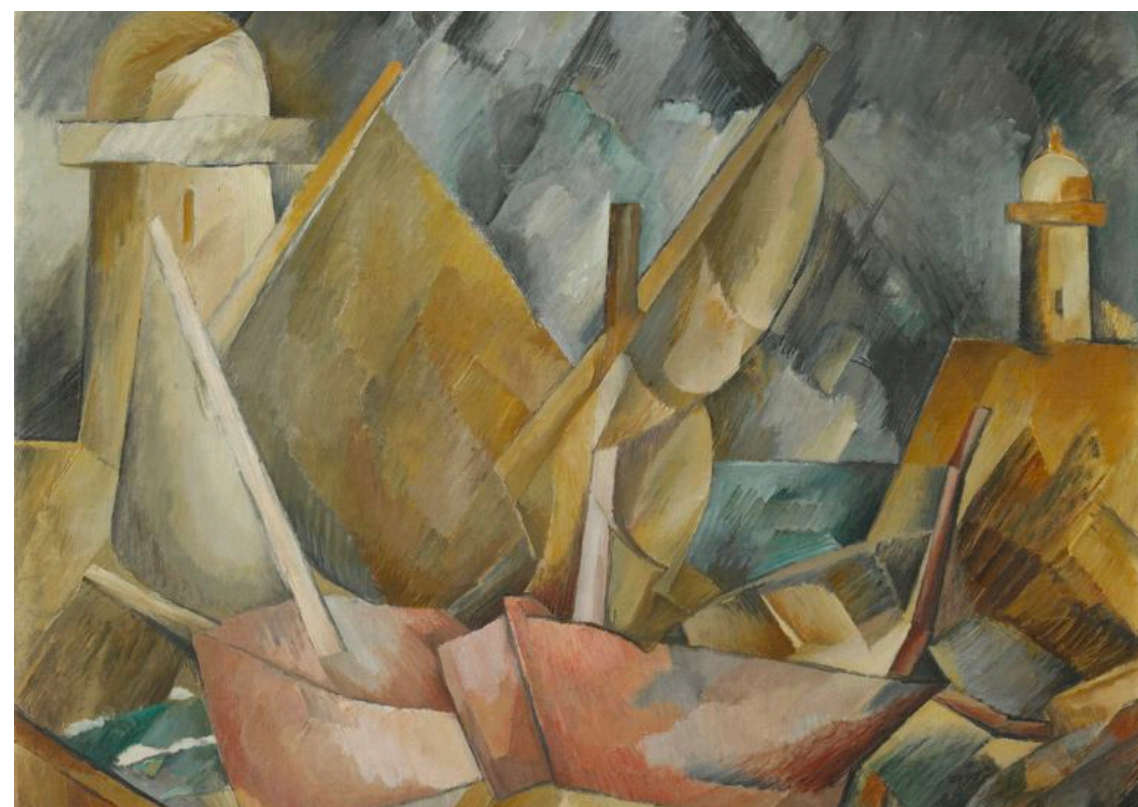


Stanford

training data

Braque

Cezanne



test datapoint



By Braque or Cezanne?

How did you accomplish this?

Through previous experience.

How might you get a machine to accomplish this task?

Modeling image formation

Geometry

SIFT features, HOG features + SVM

Fine-tuning from ImageNet features

Domain adaptation from other painters

???

Fewer human priors,
more data-driven priors

Greater success.

Can we explicitly **learn priors from previous experience**
that lead to efficient downstream learning?

Can we learn to learn?

What can meta-learning enable?

Adapting to new **objects**

provided demo



resulting policy



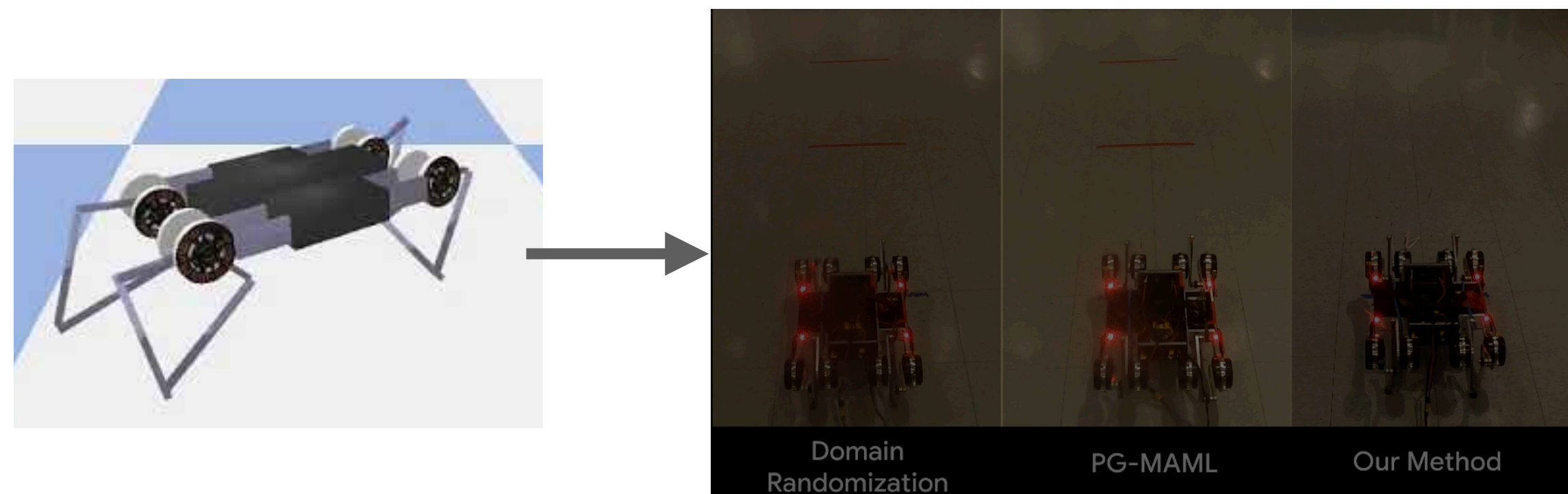
Yu*, Finn*, Xie, Dasari, Zhang, Abbeel, Levine. *One-Shot Imitation from Observing Humans*. RSS 2018

Adapting to new **molecules**

CHEMBL ID	k-NN	FINETUNE-ALL	FINETUNE-TOP	FO-MAML	ANIL	MAML
2363236	0.316 ± 0.007	0.328 ± 0.028	0.329 ± 0.023	0.337 ± 0.019	0.325 ± 0.008	0.332 ± 0.013
1614469	0.438 ± 0.023	0.470 ± 0.034	0.490 ± 0.033	0.489 ± 0.019	0.446 ± 0.044	0.507 ± 0.030
2363146	0.559 ± 0.026	0.626 ± 0.037	0.653 ± 0.029	0.555 ± 0.017	0.506 ± 0.034	0.595 ± 0.051
2363366	0.511 ± 0.050	0.567 ± 0.039	0.551 ± 0.048	0.546 ± 0.037	0.570 ± 0.031	0.598 ± 0.041
2363553	0.739 ± 0.007	0.724 ± 0.015	0.737 ± 0.023	0.694 ± 0.011	0.686 ± 0.020	0.691 ± 0.013
1963818	0.607 ± 0.041	0.708 ± 0.036	0.595 ± 0.142	0.677 ± 0.026	0.692 ± 0.081	0.745 ± 0.048
1963945	0.805 ± 0.031	0.848 ± 0.034	0.835 ± 0.036	0.779 ± 0.039	0.753 ± 0.033	0.836 ± 0.023
1614423	0.503 ± 0.044	0.628 ± 0.058	0.642 ± 0.063	0.760 ± 0.024	0.730 ± 0.077	0.837 ± 0.036*
2114825	0.679 ± 0.027	0.739 ± 0.050	0.732 ± 0.051	0.837 ± 0.042	0.759 ± 0.078	0.885 ± 0.014*
1964116	0.709 ± 0.042	0.758 ± 0.044	0.769 ± 0.048	0.895 ± 0.023	0.903 ± 0.016	0.912 ± 0.013
2155446	0.471 ± 0.008	0.473 ± 0.017	0.476 ± 0.013	0.497 ± 0.024	0.478 ± 0.020	0.500 ± 0.017
1909204	0.538 ± 0.023	0.589 ± 0.031	0.577 ± 0.039	0.592 ± 0.043	0.547 ± 0.029	0.601 ± 0.027
1909213	0.694 ± 0.009	0.742 ± 0.015	0.759 ± 0.012	0.698 ± 0.024	0.694 ± 0.025	0.729 ± 0.013
3111197	0.617 ± 0.028	0.663 ± 0.066	0.673 ± 0.071	0.636 ± 0.036	0.737 ± 0.035	0.746 ± 0.045
3215171	0.480 ± 0.042	0.552 ± 0.043	0.551 ± 0.045	0.729 ± 0.031	0.700 ± 0.050	0.764 ± 0.019
3215034	0.474 ± 0.072	0.540 ± 0.156	0.455 ± 0.189	0.819 ± 0.048	0.681 ± 0.042	0.805 ± 0.046
1909103	0.881 ± 0.026	0.936 ± 0.013	0.921 ± 0.020	0.877 ± 0.046	0.730 ± 0.055	0.900 ± 0.032
3215092	0.696 ± 0.038	0.777 ± 0.039	0.791 ± 0.042	0.877 ± 0.028	0.834 ± 0.026	0.907 ± 0.017
1738253	0.710 ± 0.048	0.860 ± 0.029	0.861 ± 0.025	0.885 ± 0.033	0.758 ± 0.111	0.908 ± 0.011
1614549	0.710 ± 0.035	0.850 ± 0.041	0.860 ± 0.051	0.930 ± 0.022	0.860 ± 0.034	0.947 ± 0.014
AVG. RANK	5.4	3.5	3.5	3.1	4.0	1.7

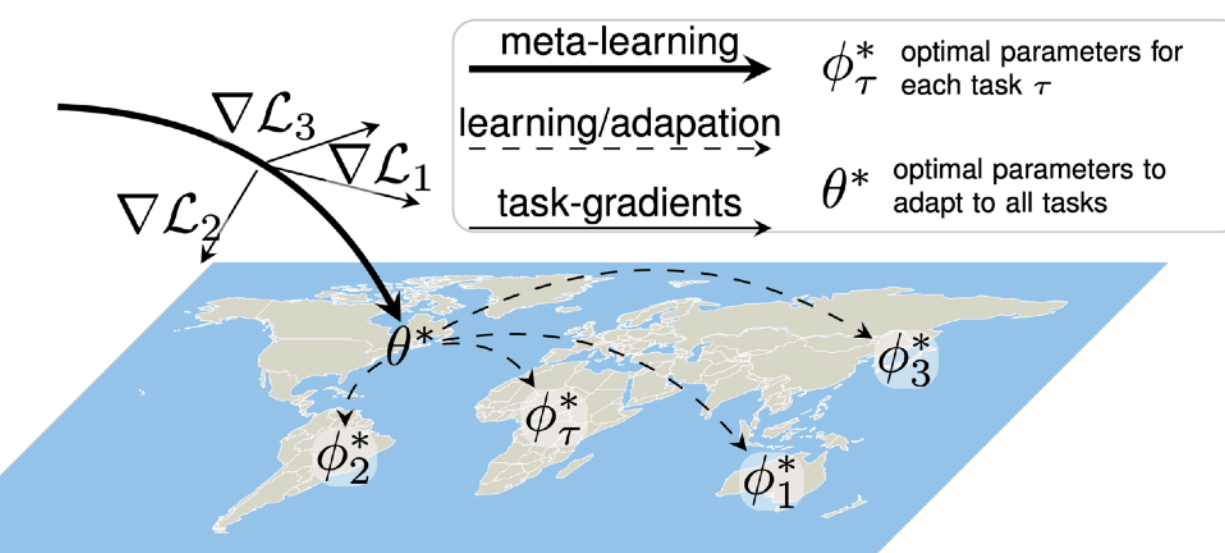
Nguyen et al. *Meta-Learning GNN Initializations for Low-Resource Molecular Property Prediction*. 2020

Adapt from **simulation to real**



Song, Yang, Choromanski, Caluwaerts, Gao, Finn, Tan. *Rapidly Adaptable Legged Robots via Evolutionary Meta-Learning*. IROS 2020

Adapting to new **regions of the world**



Rußwurm, Wang, Körner, Lobell. *Meta-Learning for Few-Shot Land Cover Classification*. CVPR 2020 EarthVision Workshop

Can we deploy few-shot learning algorithms in the real world?

- the **feedback problem**
- can we approach the problem **using meta-learning**?
- can **deploy** the approach to **real students**?
- **reflections** on real-world meta-learning

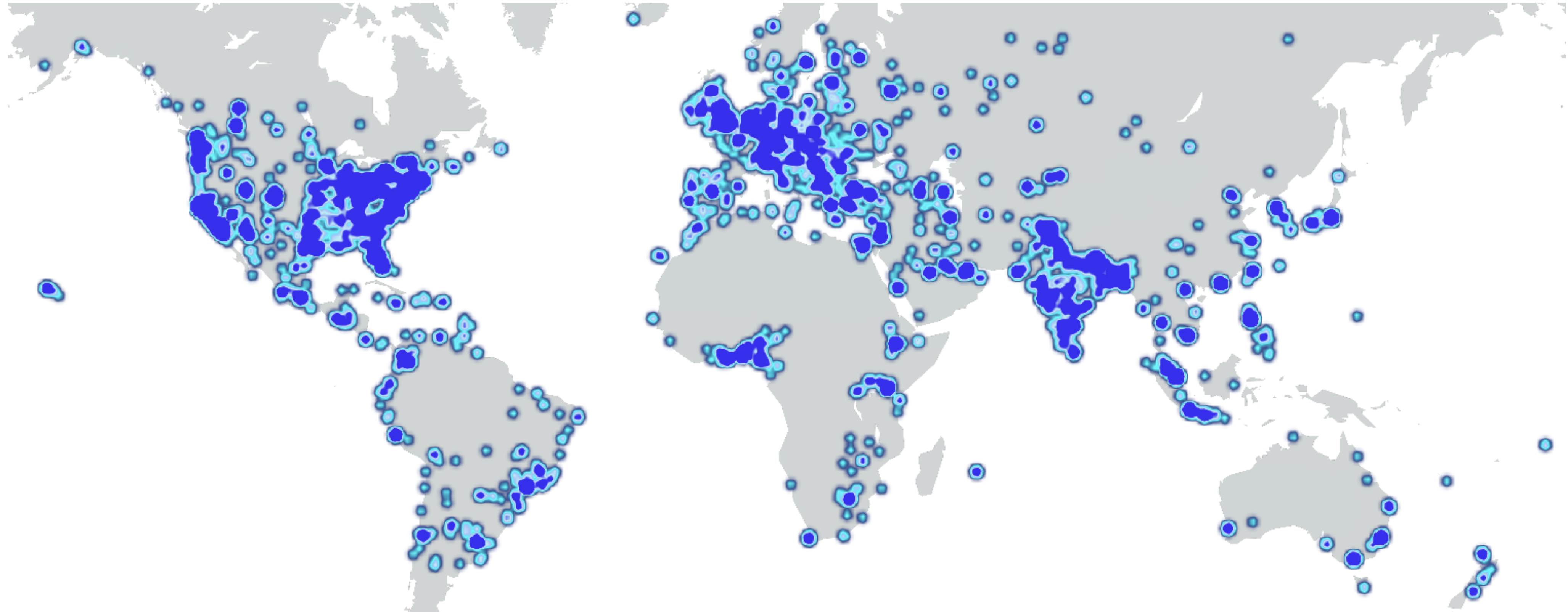
Few-shot learning to give feedback to student code

Mike Wu, Chris Piech, Noah Goodman, Chelsea Finn



The Feedback Problem

Code-in-Place 2021: Free intro to CS course, 12,000+ students from 150+ countries



How can we give feedback on a diagnostic?

Submissions: [open-ended Python code](#) snippets

Estimated [8+ months](#) of human labor



This problem isn't unique to Code-in-Place.

What does feedback look like in MOOCs?

Learn JavaScript

main.js

```
1  getReminder();
2
3  ▼ function getReminder() {
4    console.log(Forgot my string markers);
5  }
```

```
/home/ccuser/workspace/learn-javascript-functions-functions-
function-declarationV3/main.js:4
  console.log(Forgot my string markers);
                ^^^^^^
SyntaxError: missing ) after argument list
    at createScript (vm.js:53:10)
    at Object.runInThisContext (vm.js:95:10)
    at Module._compile (module.js:543:28)
    at Object.Module._extensions..js (module.js:580:10)
    at Module.load (module.js:488:32)
    at tryModuleLoad (module.js:447:12)
    at Function.Module._load (module.js:439:3)
    at Module.runMain (module.js:605:10)
    at run (bootstrap_node.js:427:7)
    at startup (bootstrap_node.js:151:9)
```

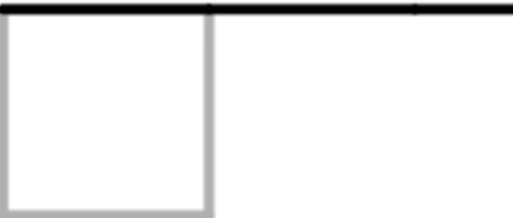
Run

View Solution

C O
D E

Artist 1 I finished!

Sign in ?



Reset

Instructions

Welcome to Artist. First off, let's try to make a simple square using the turn right block and move forward block. Each side should be 100 pixels long.

Not quite. You have to use a block you aren't using yet.

Blocks Workspace: 4 / 8 blocks Start Over Show Code

move forward by 100 pixels

turn right by 90 degrees

turn left by 90 degrees

when run

move forward by 100 pixels

move forward by 100 pixels

move forward by 100 pixels

English

Module 1 Review Quiz

✘ 3/10 points earned (30%)

[Review Related Lesson](#)

You haven't passed yet. You need at least 80% to pass.
Review the material and try again! You have 3 attempts every 8 hours.



0 / 1
points

1. Part of motivation is a feeling of competence. Both Stephen Krashen and Leo Vygotsky believe students work best just a little above their performance level. Stephen Krashen calls this...



0 / 1
points

2. Vygotsky's theory of the Zone of Proximal Development has students working slightly above their level so they feel comfortable yet challenged. To assist students in this zone, teachers offer support - scaffolding - as they master a skill. Which of the following scenarios is an example of scaffolding?



0 / 1
points

3. In order to scaffold correctly, a teacher needs to break down difficult concepts by...

The Feedback Challenge

- Train a model to infer student misconceptions, y , from the student solution, x .

```
# print 1 to n w/ loop
def my_solution(n)
    print(1)
    print(2)
    print(3)
```

[x] Incorrect Syntax

[x] Did not loop

[] Uses "print" fn

Predict!



The Feedback Challenge

Why is this a hard problem for ML?

- **Limited annotation:** grading student work takes expertise and is very time consuming.

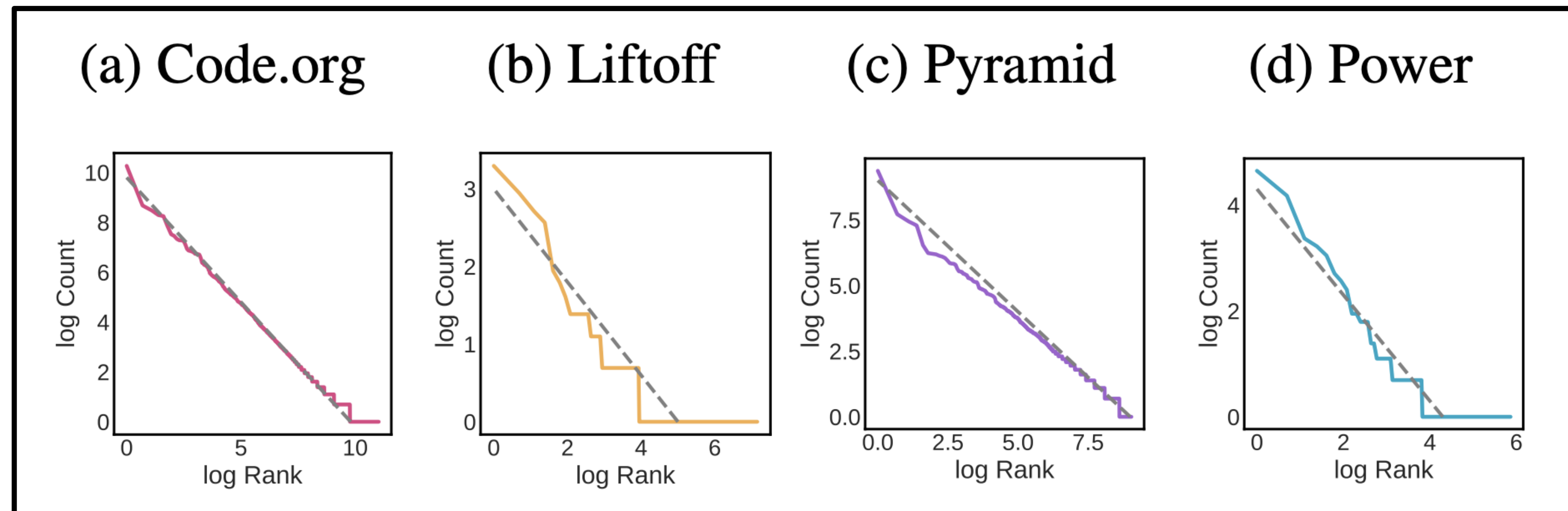
Example: annotating 800 Blockly codes took **25 hrs**



The Feedback Challenge

Why is this a hard problem for ML?

- **Limited annotation:** grading student work takes expertise and is very time consuming.
- **Long tailed distribution:** students solve the same problem in many *many* ways.



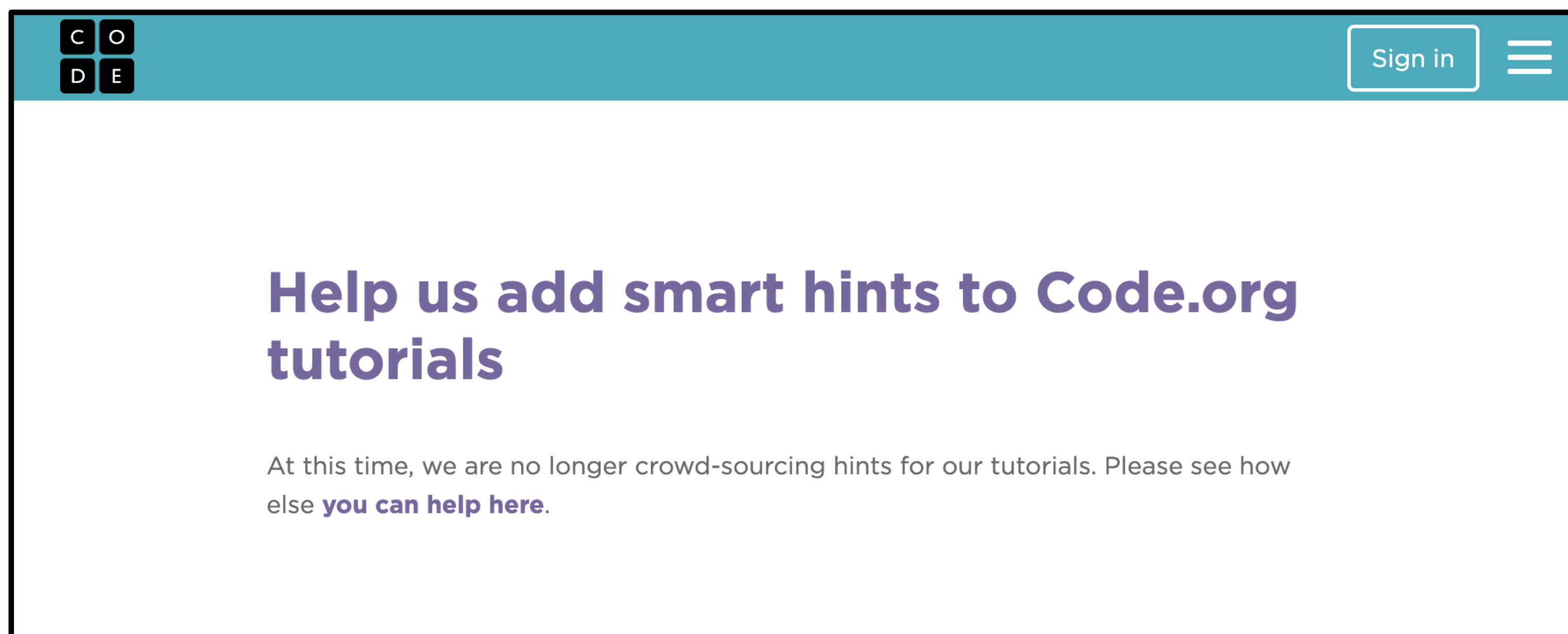
The Feedback Challenge

Why is this a hard problem for ML?

- **Limited annotation:** grading student work takes expertise and is very time consuming.
- **Long tailed distribution:** students solve the same problem in many *many* ways.
- **Changing curriculums:** instructors constantly edit assignments and exams. Student solutions and instructor feedback look different year to year.

Naive methods don't work

- **Crowdsourcing human labor:** in 2014, Code.org got 1000s of instructors to label 55k student solutions to “artist” problems. But this barely covered the distribution and new solutions were frequent.



Naive methods don't work

- **Crowdsourcing human labor:** in 2014, Code.org got 1000s of instructors to label 55k student solutions to “artist” problems.
- **Supervised learning:** dataset of a few 1000 examples (at best) + long tail make this really hard.

Model	Body F1	Tail F1
Output CNN [26]	0.10	0.10
Human	0.68	0.69

Block-based programming

Model	Body Acc	Tail Acc
NeuralNet [28]	0.20	0.21
Human	0.81	0.80

CS106A Graphics programming

Model	Avg F1	Tail F1
Handcrafted [6]	0.58	-
T&N Best [17]	0.55	-
Human	0.97	0.90

free response

Can we use prior data & formulate this as a
meta-learning problem?

Prior experience

10 years of feedback from
Stanford midterms and finals

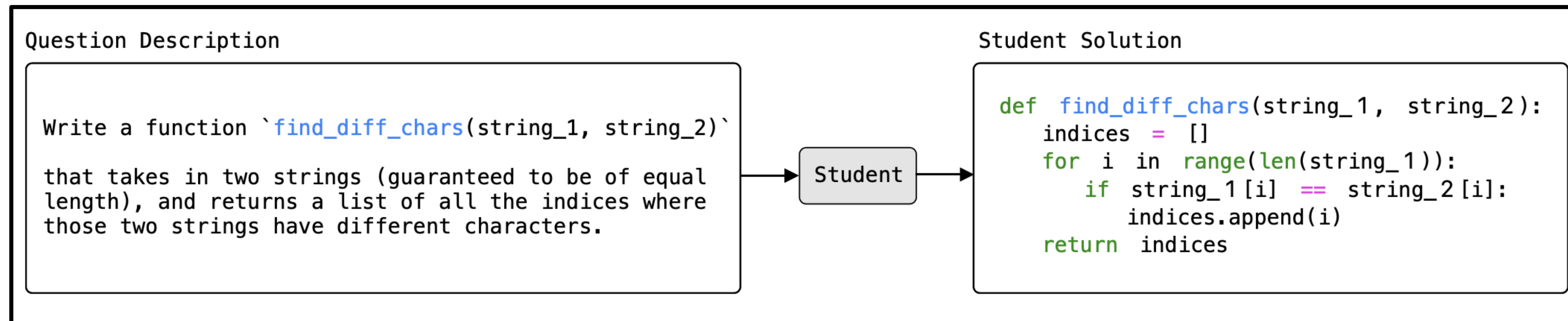
Meta-test task

Give feedback on new
problems with small amount
of labeled examples

CS106A Dataset

Contains 4 final exams and 4 midterm exams from CS106.

- Total of 63 questions and 24.8k student solutions.
- Every student solution has feedback via a rubric.
- 10% of questions were annotated by more than 1 TA, which we use to compute human accuracy.



CS106A Dataset

A rubric has several items, each describes a misconception. Each item has several options that an grader may pick to be true.

- More than one option can be true.
- Every problem has its own (possibly unique) rubric items and options.

Rubric Item: Problem Setup

- Perfect
- Minor issue
- Major issue
- No attempt

Rubric Item: General Deductions

- Perfect
- 1 syntax error
- 2 syntax errors
- >2 syntax errors
- Variable scoping issue
- Null pointer exception

Rubric Item: String Insertion

- Perfect
- Incorrectly gets character to insert
- Incorrectly assumes one digit
- Doesn't insert character at correct place

CS106A Dataset

We treat every rubric option as a task.

- Every task is a binary classification problem!
- Total of 259 tasks ($K = 10$, $Q = 10$)

Rubric Item: String Insertion

- Perfect
- Incorrectly gets character to insert
- Incorrectly assumes one digit
- Doesn't insert character at correct place

Task

Also task

Yet another task

another task?!

ProtoTransformer

Support and query sets:

$$S = \{(x_1, y_1), (x_2, y_2), \dots, (x_{K \times N}, y_{K \times N})\}$$

$$Q = \{(x_1^*, y_1^*), (x_2^*, y_2^*), \dots, (x_{Q \times N}^*, y_{Q \times N}^*)\}$$

ProtoTransformer

Support and query sets:

$$S = \{(x_1, y_1), (x_2, y_2), \dots, (x_{K \times N}, y_{K \times N})\}$$

$$Q = \{(x_1^*, y_1^*), (x_2^*, y_2^*), \dots, (x_{Q \times N}^*, y_{Q \times N}^*)\}$$

Use the support set S to derive a prototype embedding for each class. Try to classify each example in query set Q by distance to each prototype.

ProtoTransformer

Support and query sets:

$$S = \{(x_1, y_1), (x_2, y_2), \dots, (x_{K \times N}, y_{K \times N})\}$$

$$Q = \{(x_1^*, y_1^*), (x_2^*, y_2^*), \dots, (x_{Q \times N}^*, y_{Q \times N}^*)\}$$

Use the support set S to derive a prototype embedding for each class. Try to classify each example in query set Q by distance to each prototype.

p_c is the average embedding over examples in the support set with label c .

$$\mathcal{L}(x^*, y^*) = -\log \frac{\exp\{-\text{dist}(f_\theta(x^*), p_{y^*})/\tau\}}{\sum_{c=1}^N \exp\{-\text{dist}(f_\theta(x^*), p_c)/\tau\}}$$

temperature

L_2 norm

ProtoTransformer

$$\mathcal{L}(x^*, y^*) = -\log \frac{\exp\{-\text{dist}(f_\theta(x^*), p_{y^*})/\tau\}}{\sum_{c=1}^N \exp\{-\text{dist}(f_\theta(x^*), p_c)/\tau\}}$$

- We assume $x = (x_1, x_2, \dots, x_T)$ a sequence of discrete tokens (e.g. code, language).
- The embedding $f_\theta: X \rightarrow \mathbb{R}^d$ is a RoBERTa model (stacked transformers) where non-padded token embeddings are averaged (single vector).

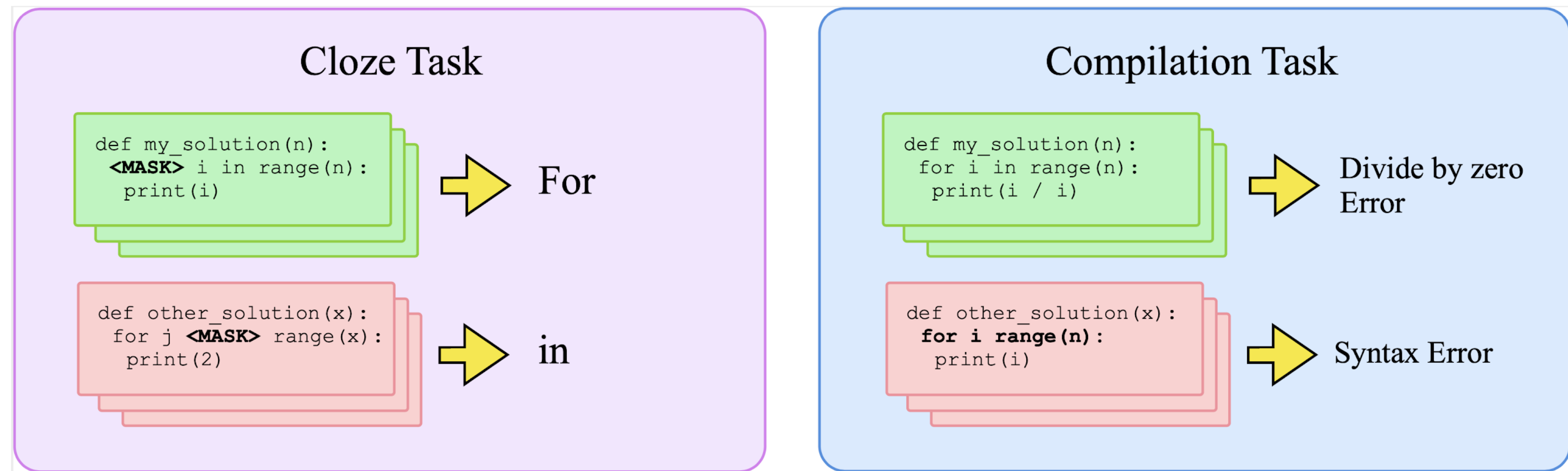
ProtoTransformer

$$\mathcal{L}(x^*, y^*) = -\log \frac{\exp\{-\text{dist}(f_\theta(x^*), p_{y^*})/\tau\}}{\sum_{c=1}^N \exp\{-\text{dist}(f_\theta(x^*), p_c)/\tau\}}$$

- We assume $x = (x_1, x_2, \dots, x_T)$ a sequence of discrete tokens (e
- Attention is **not** all you need. 😲
- The embedding $f_\theta: X \rightarrow \mathbb{R}^d$ is a RoBERTa model (stacked transformers) where non-padded token embeddings are averaged (single vector).
- Applying this “out-of-the-box” **fails**. We needed several “tricks” to get past the small data size.

Trick #1: Task Augmentation

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



Trick #2: Side Information

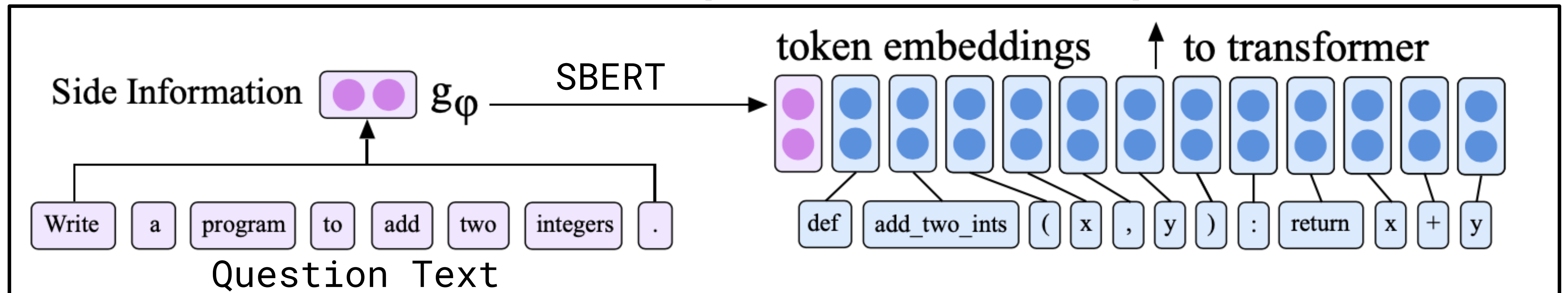
A task is only composed of 10 or 20 examples, leaving a lot of ambiguity.

Suppose we have “side information” $z = (z_1, z_2, \dots, z_T)$ about each task: **rubric option name** and **question text**. How do we add this side information into our embedding function f_θ ?

Trick #2: Side Information

A task is only composed of 10 or 20 examples, leaving a lot of ambiguity.

Suppose we have “side information” $z = (z_1, z_2, \dots, z_T)$ about each task: **rubric option name** and **question text**.

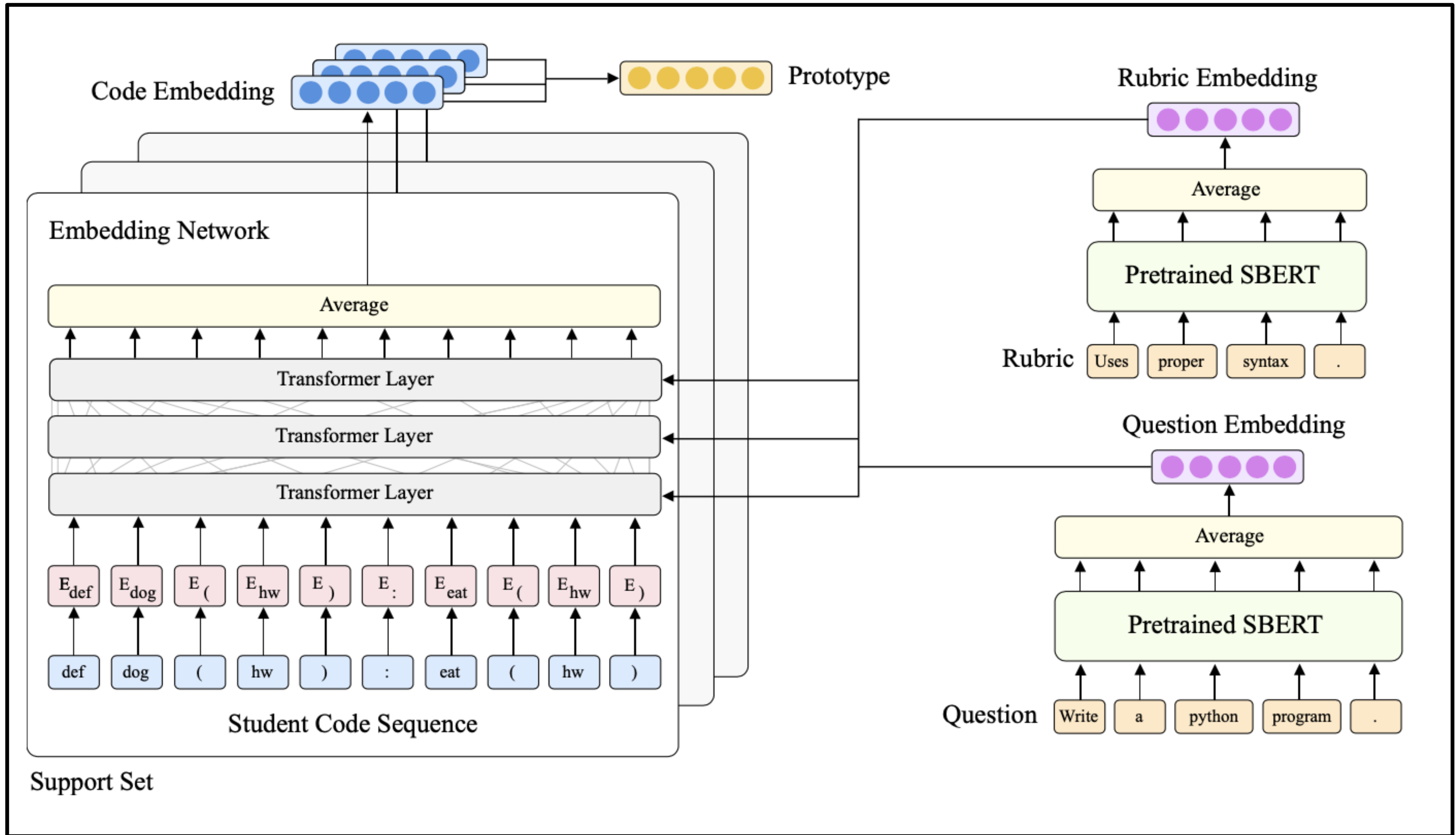


Prepend side information as a first token.

Trick #3: Code Pre-training

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

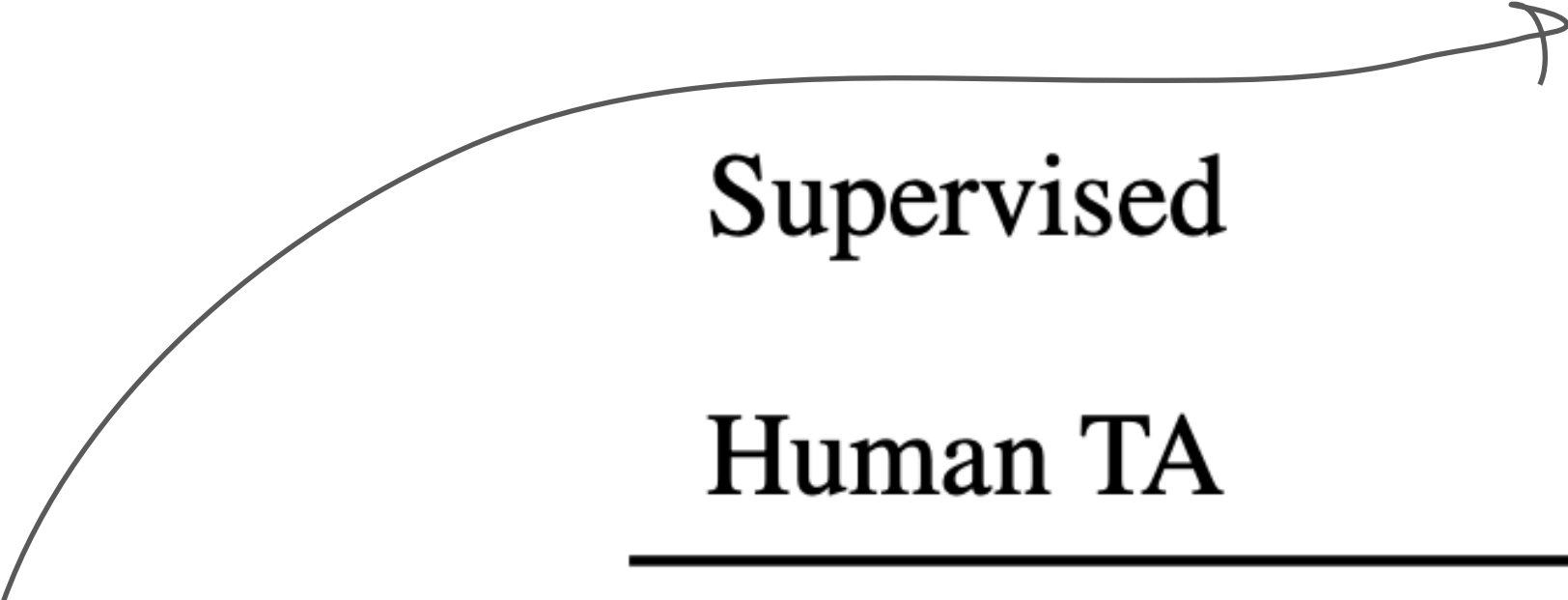
In practice, we initialize the embedding network from pretrain weights and finetune top M layers.



Results

Model	Held-out rubric			
	AP	P@50	P@75	ROC-AUC
ProtoTransformer	84.2 (± 1.7)	85.2 (± 3.8)	74.2 (± 1.4)	82.9 (± 1.3)
Supervised	66.9 (± 2.2)	59.1 (± 1.7)	53.9 (± 1.5)	61.0 (± 2.1)
Human TA	82.5	–	–	–

Model	Held-out exam			
	AP	P@50	P@75	ROC-AUC
ProtoTransformer	74.2 (± 1.6)	77.3 (± 2.7)	67.3 (± 2.0)	77.0 (± 1.4)
Supervised	65.8 (± 2.1)	60.1 (± 3.0)	54.3 (± 1.8)	60.7 (± 1.6)
Human TA	82.5	–	–	–

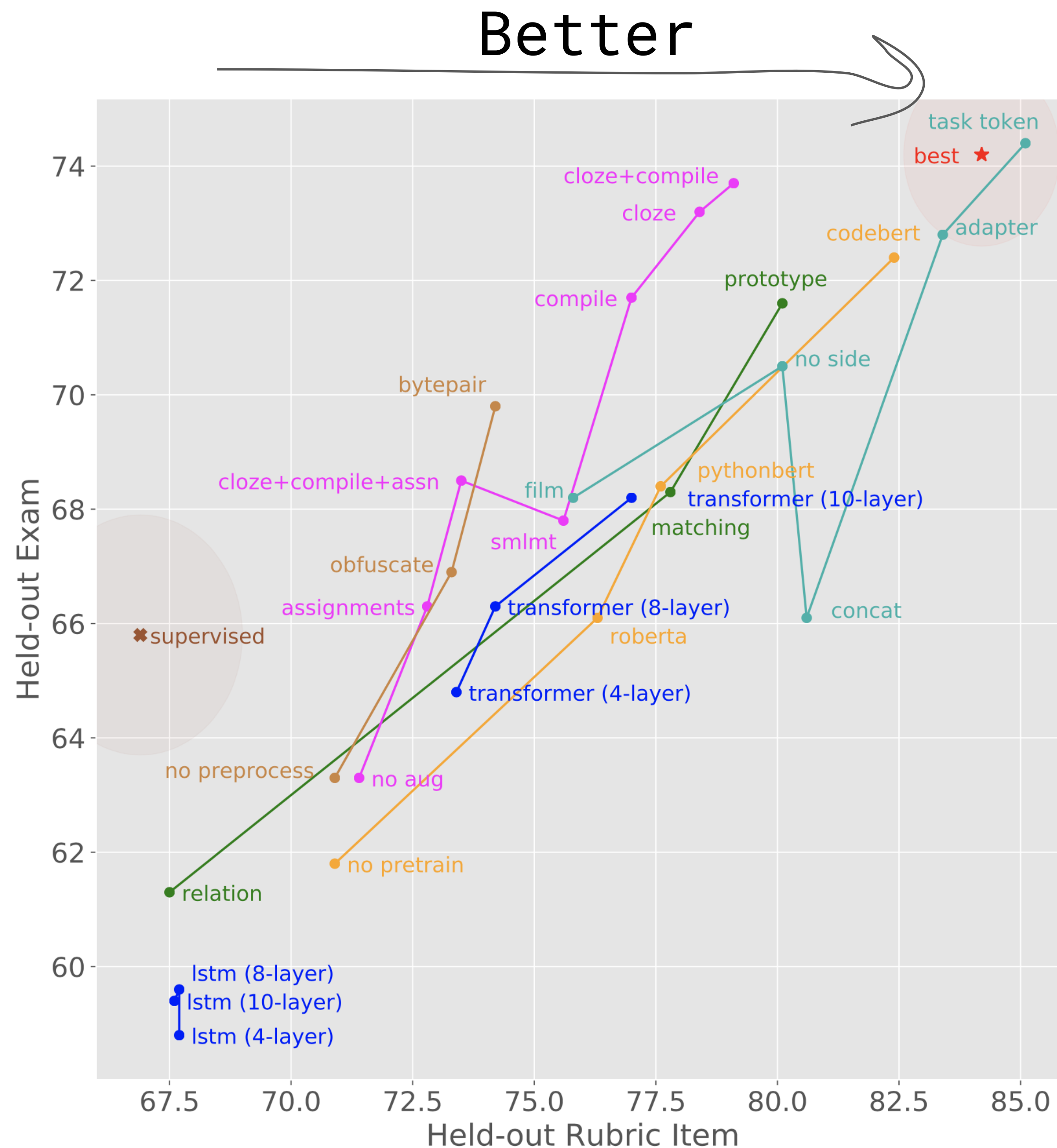


Room to grow!

Ablations

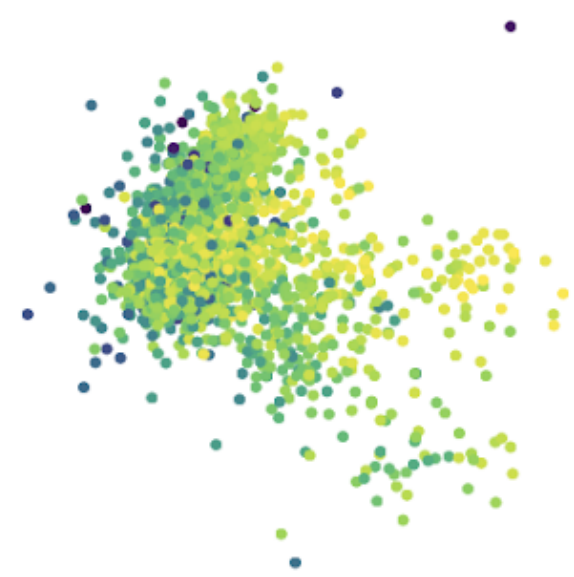
Legend

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



Embeddings

Visualize “prototype” embeddings to interpret student ability and question quality.



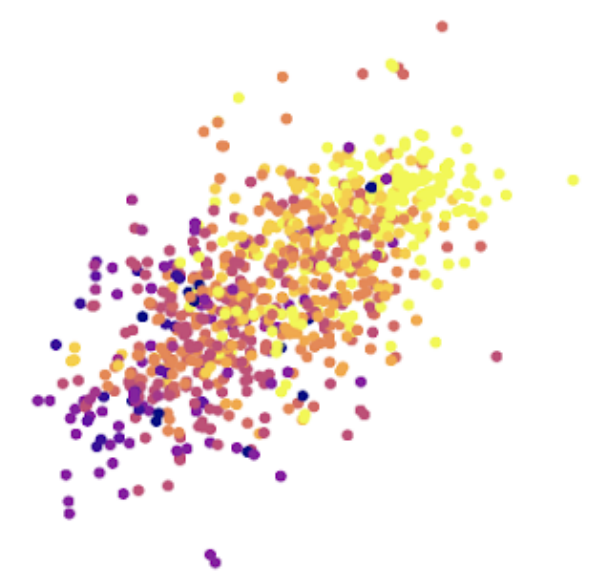
(a) All Students



(b) Random Student



(c) Random Question



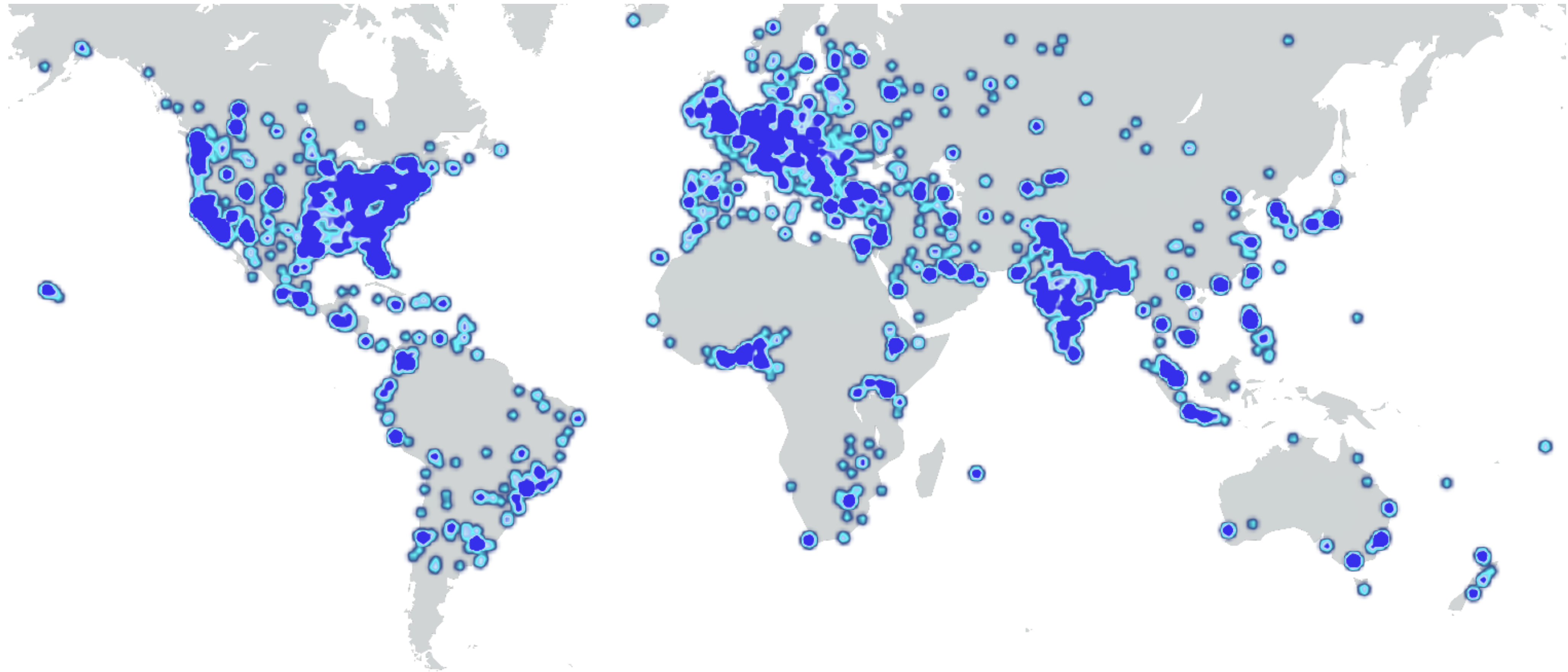
(d) Random Question

Color shows the numeric grade (not used by model ever) given to student (darker is lower).

Can we deploy few-shot learning algorithms in the real world?

- the **feedback problem**
- can we approach the problem **using meta-learning**?
- can **deploy** the approach to **real students**?
- **reflections** on real-world meta-learning

Can we deploy this to Code-in-Place?



May 10th, 2021: Students took diagnostic.



Code in Place Feedback

codeinplace.stanford.edu/diagnostic/feedback

Overview **Question 1** Question 2 Question 3 Question 4 Question 5 Wrap-Up



Back Feedback Next

GETTING INPUT FROM USER

This question requires you to get input from the user, convert it to a number, and save it as a variable. Did you correctly do all of these steps?

Close. There is a minor error with your logic to get input from user. This could be something like forgetting to convert user input to a float

Do you agree with the feedback in the purple box?

Please explain (optional):

Your Solution

```
def main():
    # TODO write your solution here
    height=input("Enter your height in meters: ")
    if height < 1.6:
        print("Below minimum astronaut height")
    if height > 1.9:
        print("Above maximum astronaut height")
    if height >= 1.6 and height <= 1.9:
        print("Correct height to be an astronaut")

if __name__ == "__main__":
    main()
```

AI generated feedback



Students evaluate the feedback



Algorithm uses attention to highlight where in the code the error comes from



Syntax error here would prevent unit tests from being useful



designed by Alan Cheng & Chris Piech

Blind, randomized trial *evaluated by real students*

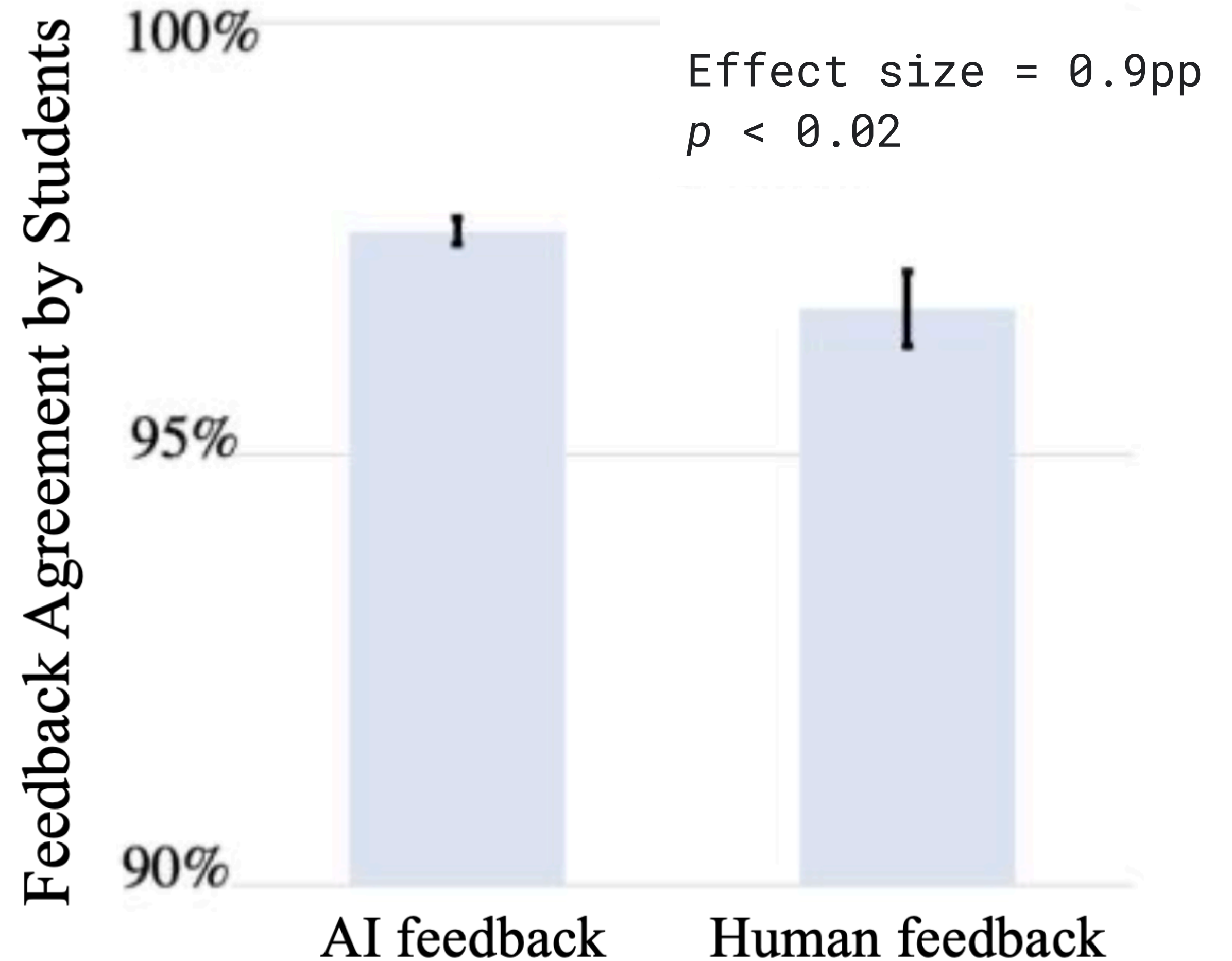
Humans gave feedback ~1k answers.

AI gave feedback on the remaining ~15k.

~2k could be auto-graded and were not included in analysis.

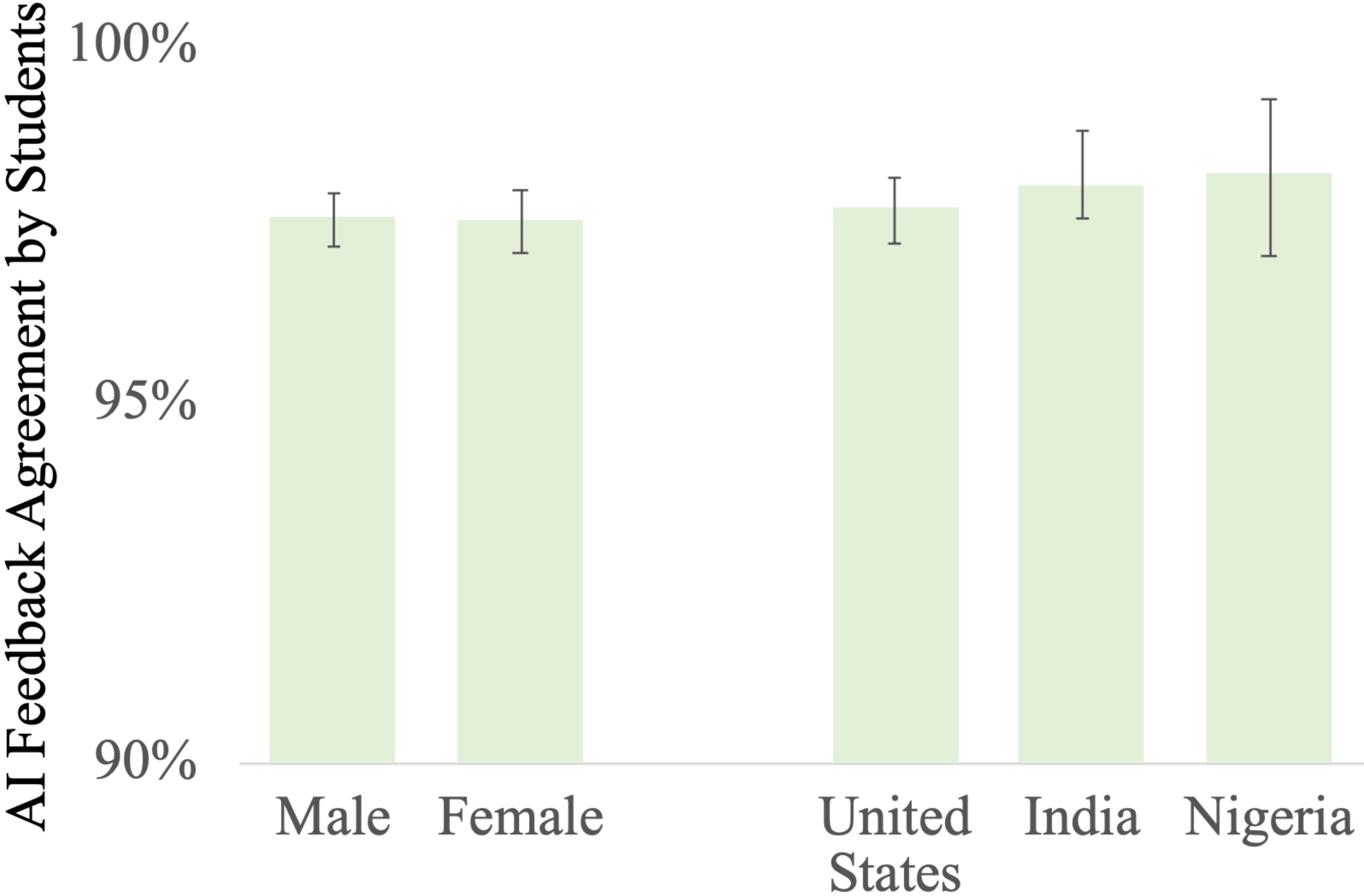
Humans gave good feedback.

ML model gave slightly better feedback.



Average holistic rating of usefulness by students was **4.6 ± 0.018 out of 5.**

No signs of bias by demographics



Can we deploy few-shot learning algorithms in the real world?

- the **feedback problem**
- can we approach the problem **using meta-learning**?
- can **deploy** the approach to **real students**?
- **reflections** on real-world meta-learning

A first for education



First successful deployment of ML-driven **feedback** to open ended student work

* to the best of our knowledge

A first for ML



First successful deployment of **prototypical networks** in live application.

What was hard and different?

1. **Limited** meta-training tasks.

- > task augmentation can help
- > regularization may help

Also see:

Bansal et al. SMLMT '20

Murty et al. DRECA '21

Also see: Yao et al. MLTI '21

2. Where does the **support set** come from?

- > active learning? expert-designed support sets?

3. Can the model **defer harder examples** for the instructor?

- > calibration, selective classification

4. **Domain shift** between meta-training & deployment.

Also see: Koh*, Sagawa* WILDS '21

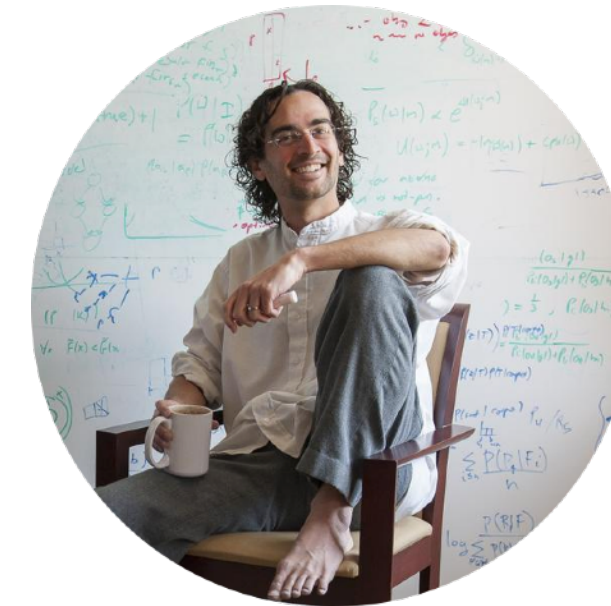
Mike Wu



Alan Cheng



Noah Goodman



Chris Piech



Want to learn more about meta-learning?

Stanford CS330: Deep Multi-Task and Meta Learning

cs330.stanford.edu

All lecture videos online!