

# “In-context” Learning: What and Why?

Noah Amsel

# This talk is...

- Rough outline of four different papers
  - Not my work!
  - Experiments, not theory
- 
- But hopefully suggests a interesting theory project

“Language Models are Few-Shot Learners”  
Brown et al.

# Few shot learning with fine-tuning (RIP)

Traditional fine-tuning (not used for GPT-3)

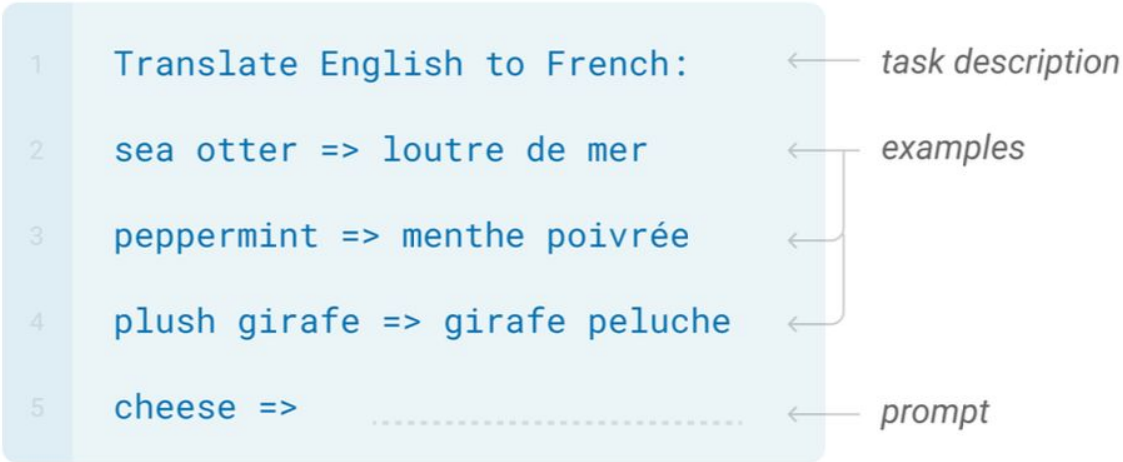
## Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.



# In-context learning

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



The diagram shows a light blue rounded rectangle containing a prompt. To the right of the rectangle, three labels with arrows point to specific parts of the prompt: 'task description' points to line 1, 'examples' points to lines 2-4, and 'prompt' points to line 5.

```
1  Translate English to French:
2  sea otter => loutre de mer
3  peppermint => menthe poivrée
4  plush girafe => girafe peluche
5  cheese => .....
```


← *task description*

← *examples*

← *prompt*

# In-context learning

1	$5 + 8 = 13$
2	$7 + 2 = 9$
3	$1 + 0 = 1$
4	$3 + 4 = 7$
5	$5 + 9 = 14$
6	$9 + 8 = 17$




↑  
sequence #1

1	gaot => goat
2	sakne => snake
3	brid => bird
4	fsih => fish
5	dcuk => duck
6	cmihp => chimp



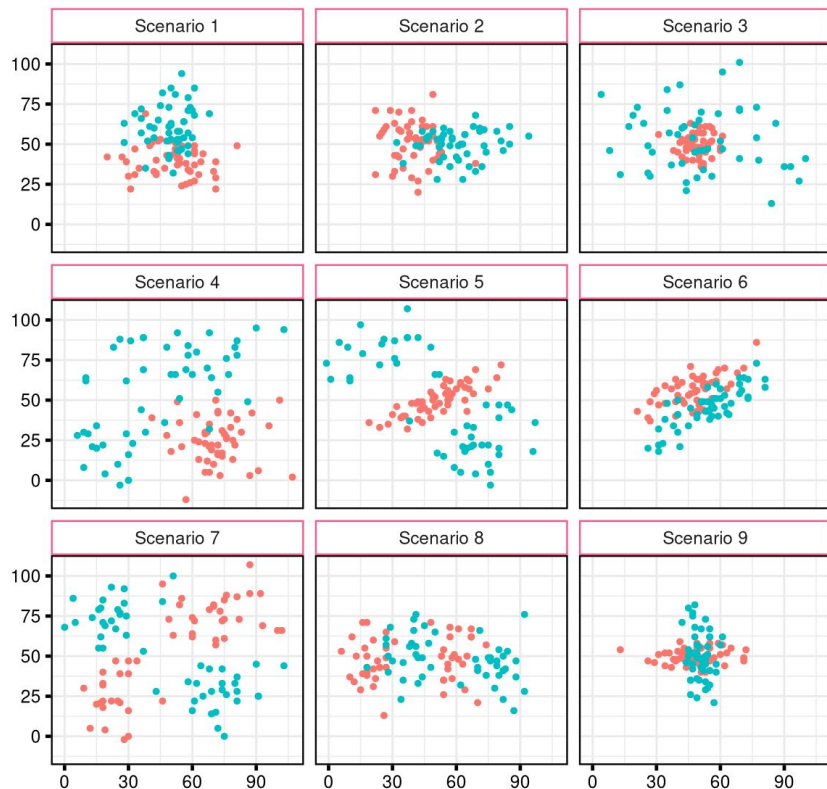
↑  
sequence #2

1	thanks => merci
2	hello => bonjour
3	mint => menthe
4	wall => mur
5	otter => loutre
6	bread => pain



↑  
sequence #3

# In-context learning



Model	Average acc.
kNN	81.78%
Logistic regr.	62.34%
Custom text	67.03%
Ada	73.70%
Babbage	72.10%
Curie	75.68%
Davinci	75.93%

“Rethinking the Role of Demonstrations:  
What Makes In-Context Learning Work?”  
Min et al.



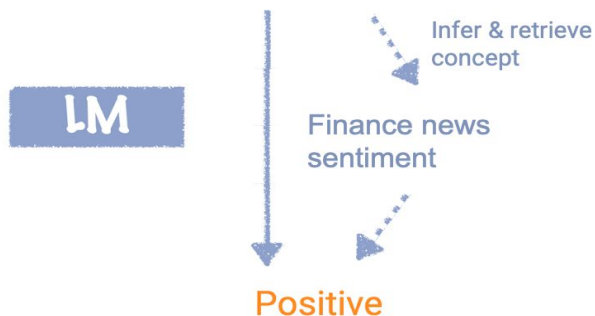
# Classification Task

Circulation revenue has increased by 5% in Finland. // Positive

Panostaja did not disclose the purchase price. // Neutral

Paying off the national debt will be extremely painful. // Negative

The company anticipated its operating profit to improve. // \_\_\_\_\_

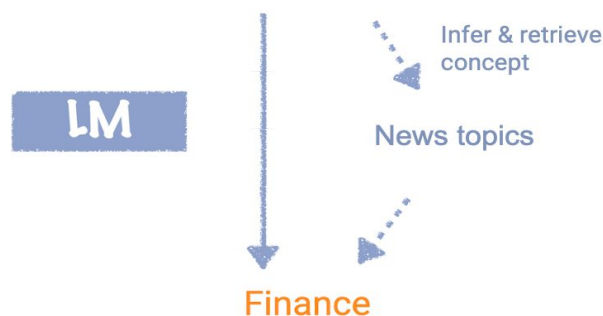


Circulation revenue has increased by 5% in Finland. // Finance

They defeated ... in the NFC Championship Game. // Sports

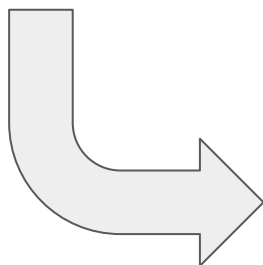
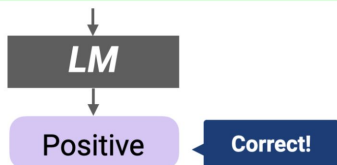
Apple ... development of in-house chips. // Tech

The company anticipated its operating profit to improve. // \_\_\_\_\_

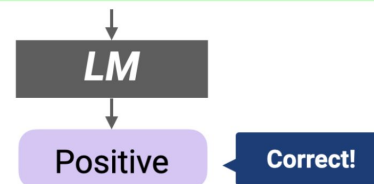


# Classification with Randomized Labels Task

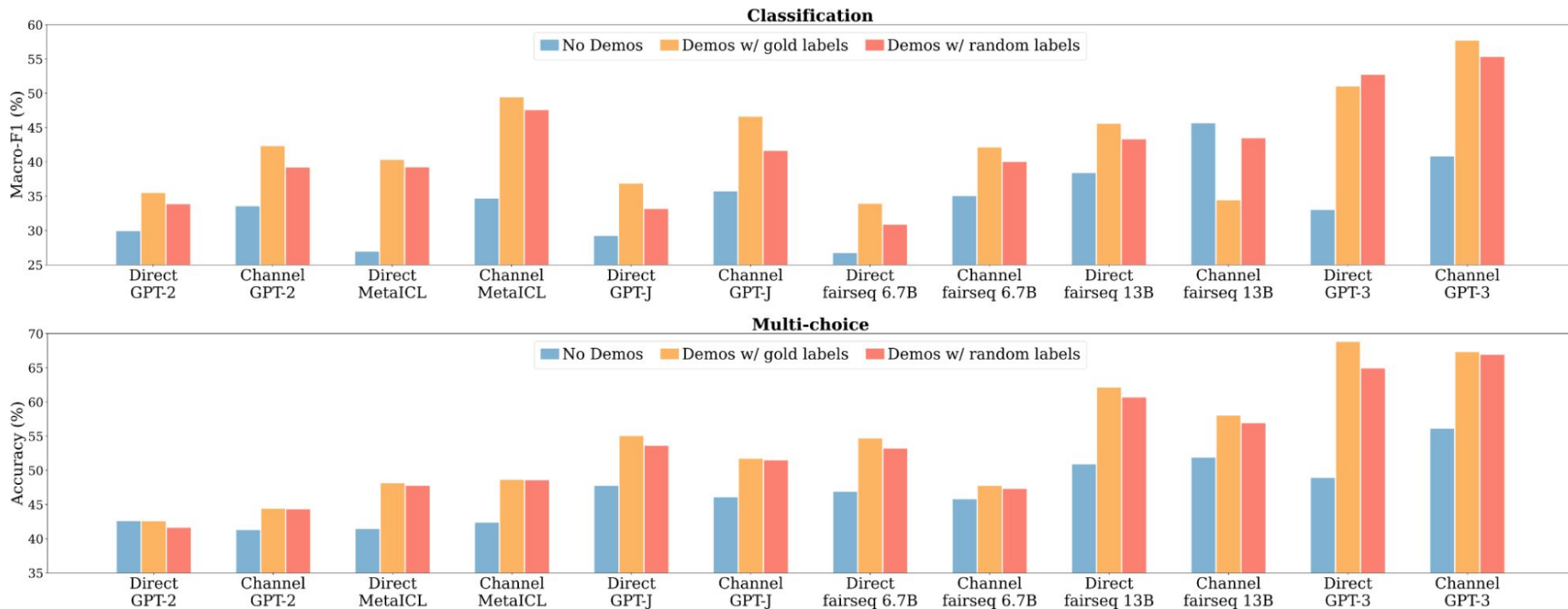
Circulation revenue has increased by 5% in Finland. \n Positive  
Panostaja did not disclose the purchase price. \n Neutral  
Paying off the national debt will be extremely painful. \n Negative  
The company anticipated its operating profit to improve. \n \_\_\_\_\_



Circulation revenue has increased by 5% in Finland. \n **Neutral**  
Panostaja did not disclose the purchase price. \n **Negative**  
Paying off the national debt will be extremely painful. \n **Positive**  
The company anticipated its operating profit to improve. \n \_\_\_\_\_



# Classification with Randomized Labels Results



# What is needed for in-context learning to work?

- ~~Input-label mapping~~
- Distribution of inputs
- Label space
- Format of input / output pairs

Colour-printed lithograph. Very good condition. \n Neutral  
Many accompanying marketing ... meaning. \n Negative  
In case you are interested in learning more about ... \n Positive

The company anticipated its operating profit to improve. \n \_\_\_\_\_

Circulation revenue has increased by 5% in Finland. \n Unanimity  
Panostaja did not disclose the purchase price. \n Wave  
Paying off the national debt will be extremely painful. \n Guana

The company anticipated its operating profit to improve. \n \_\_\_\_\_



*\*Random English unigrams*

“

If we take a strict definition of learning: capturing the input-label correspondence given in the training data, then our findings suggest that LMs do not learn new tasks at test time.

Our analysis shows that the model may ignore the task defined by the demonstrations and instead use prior from pretraining.

”

“Extrapolating to Unnatural Language  
Processing with GPT-3's In-context Learning”  
Frieda Rong

Is in-context “learning” just locating a previously learned concept?

# Simplifications

- No fancy prompts, task descriptions. Just input / output pairs
  - Structured inputs. Synthetic, algorithmic tasks
  - Still studying pretrained GPT-3
- 
- e.g. copying 5 letters: 100% accuracy

## An example prompt:

```
Input: g, c, b, h, d
Output: g, c, b, h, d
Input: b, g, d, h, a
Output: b, g, d, h, a
Input: f, c, d, e, h
Output: f, c, d, e, h
Input: c, f, g, h, d
Output: c, f, g, h, d
Input: e, f, b, g, d
Output: e, f, b, g, d
Input: a, b, c, d, e
Output:
```

## The expected completion:

```
a, b, c, d, e
```



# Permuted Classification: Task

volleyball: animal

onions: sport

broccoli: sport

hockey: animal

kale: sport

beet: sport

golf: animal

horse: plant/vegetable

corn: sport

football: animal

luge: animal

bowling: animal

beans: sport

archery: animal

sheep: plant/vegetable

zucchini: sport

goldfish: plant/vegetable

duck: plant/vegetable

leopard: plant/vegetable

lacrosse: animal

badminton: animal

lion: plant/vegetable

celery: sport

porcupine: plant/vegetable

wolf: plant/vegetable

lettuce: sport

camel: plant/vegetable

billiards: animal

zebra: plant/vegetable

radish: \_\_\_\_\_

# Permuted Classification: Results

- 93% accuracy
  - Also works if the labels are symbols [“^\*”, “#@#”, and “!~”]
- Definitely not “locating” this mapping
  - Composing two maps in a novel way
  - The permutation comes entirely from the context
  - Learned to override the natural mapping from noun to category

# Unnatural Addition: Task

- GPT-3 can add / subtract two digit numbers ~100% accurately
- It “knows” what subtraction is
  - It has seen “ $30 - 20 = 10$ ” in training data
- Let’s force a contradiction between the training corpus and the prompt

$$30 - 20 = 50$$

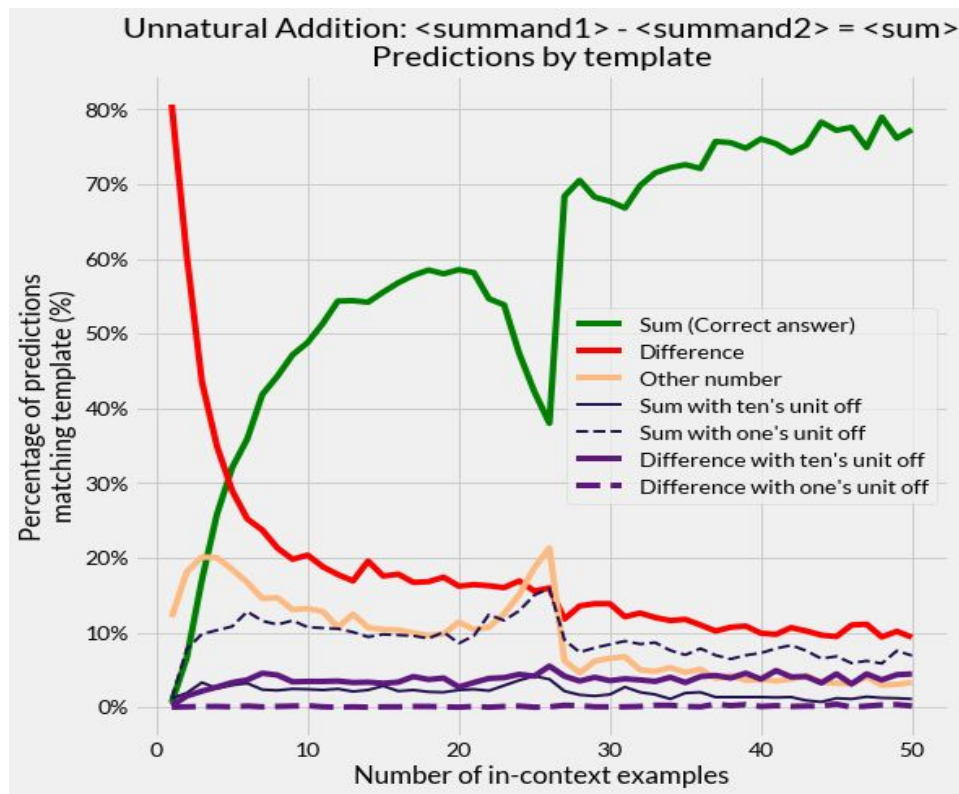
$$60 - 15 = 75$$

$$53 - 26 = 79$$

...

$$42 - 11 = ??$$

# Unnatural Addition: Results



# “Data Distributional Properties Drive Emergent In-Context Learning in Transformers”

Chan et al.

## “In-weights” learning

- Slow training with gradient descent, then it's ready-to-use (“saved” in weights)
- Requires many examples
- Many models can do this
- Objective function promotes this

## “In-context” learning

- “Training” happens at inference time from prompting. Weights are oblivious to the task
- Only needs a few examples (“few-shot”)
- Transformer-based large language models can do this
- Emergent — training doesn't explicitly promote this
  - But it happens anyway. **Why?**

# Experimental Design: Goal

Separately measure a model's in-weights knowledge from its capacity for in-context learning

No interference!

# Experimental Design: Task

## Classify “Omniglot” images

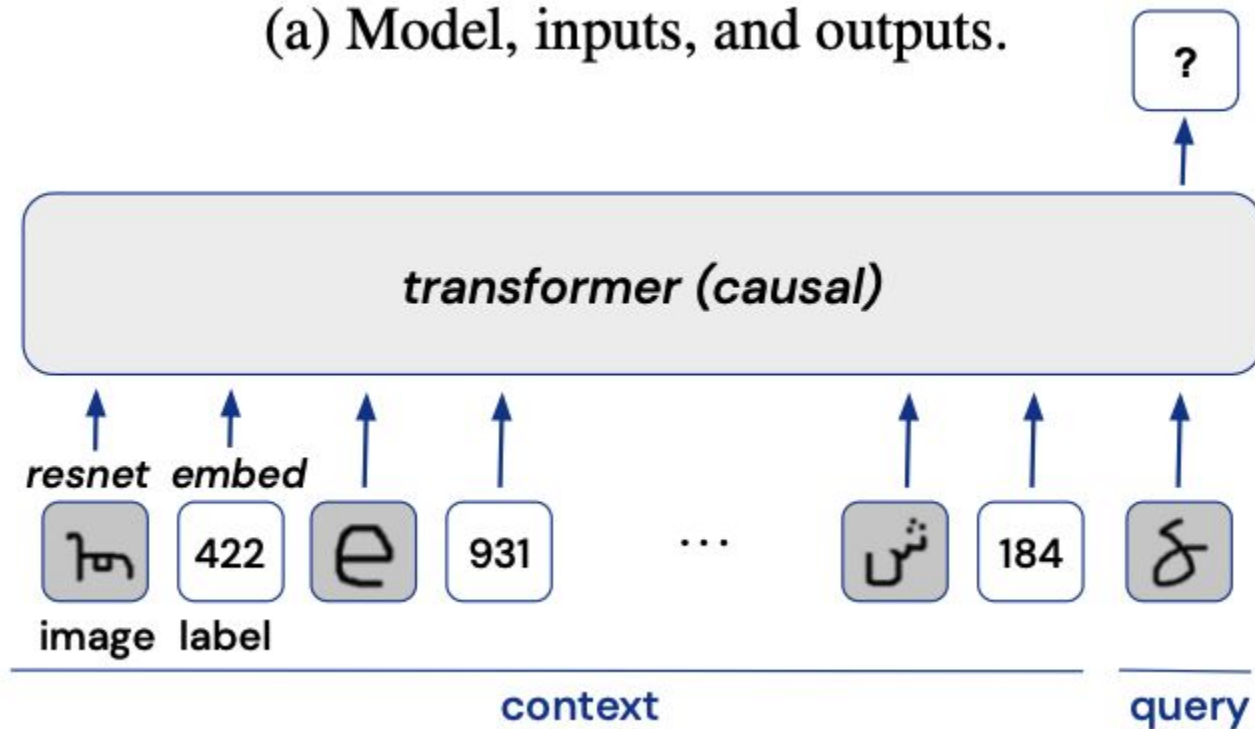
- 1623 classes (characters)
- 20 examples per class
- An “example” is  $(x, y)$ 
  - $x$  is an image of a character
  - $y$  is its integer label
- A sequence is  $x_1, y_1, x_2, y_2, \dots$   
 $x_8, y_8, x_9, \_$ 
  - Task: predict label of  $x_9$





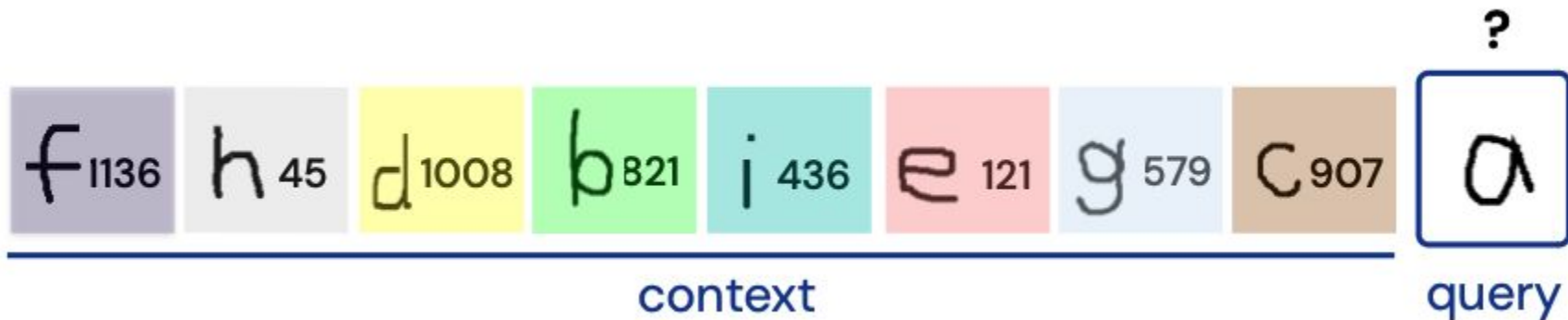
# Experimental Design: Model

(a) Model, inputs, and outputs.



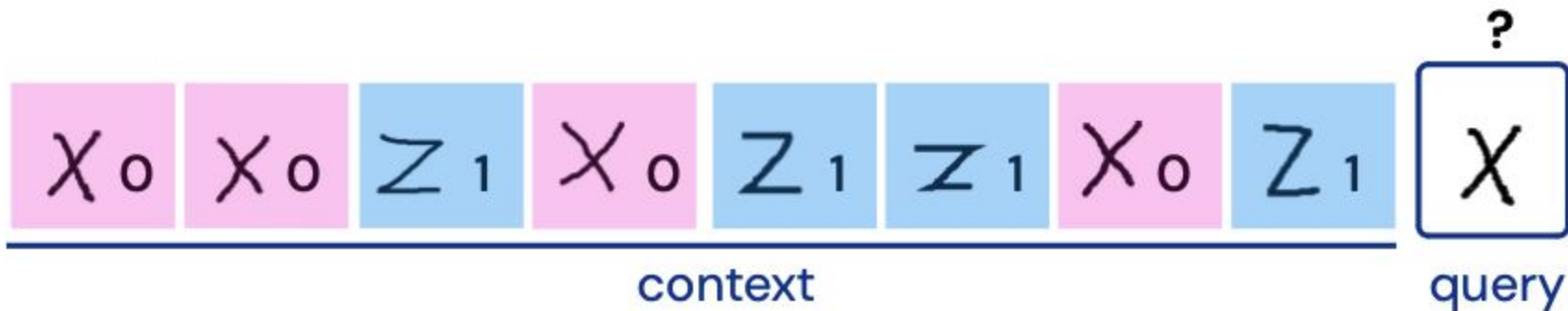
# Experimental Design: In-weights Evaluation

- The context doesn't matter
- If the model learned the training data, it should know the label of the query
  - (weights already “know” what letter “a” looks like before seeing the context)



# Experimental Design: In-context Evaluation

- The query never appeared in the training dataset, but does appear 3 times in the context
- Must learn to distinguish between two new characters from just three examples of each
  - Model can quickly “learn” a new concept from the context



Q: Why does in-context learning emerge?

A: Properties of training distribution

# Supervised learning data

- Medium number of classes
- Classes are balanced
- $X \rightarrow Y$  mapping is fixed
- i.i.d. Samples

Model gradually encodes information in its weights during training

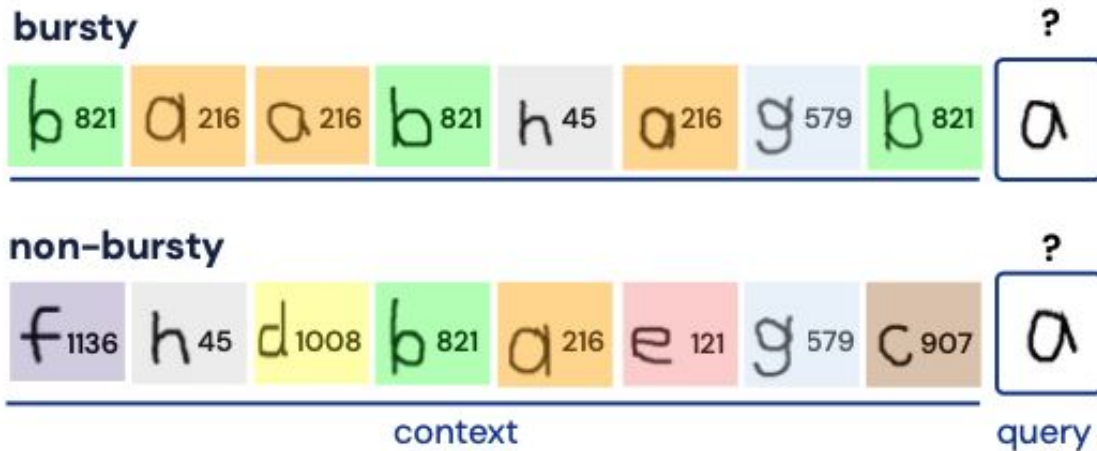
# Natural language data

- Large number of classes
- Long tailed distribution (Zipf's Law)
- Meaning is context-dependent
- “Burstiness”: if a word appears once in a context, it's more likely to occur again
  - E.g., “Zipfian” is a rare word but this paper uses it 17 times

Model learns to use the context / prompt

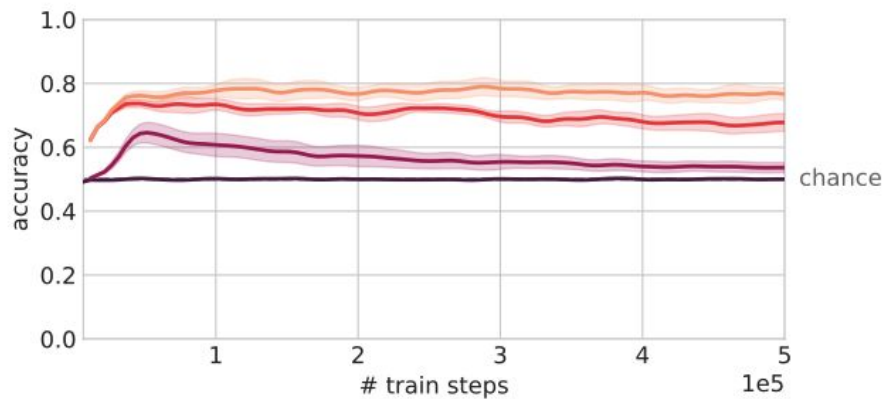
# Experimental Design: Training Data

- Bursty: 3 A's, 3 B's, 3 other. Query is either A or B
  - Like natural language document
  - Either in-context or in-weights learning will work
- Non-bursty; 8 random context symbols and a random query
  - Only in-weights training will work
- Training data is a mix of bursty and non-bursty sequences

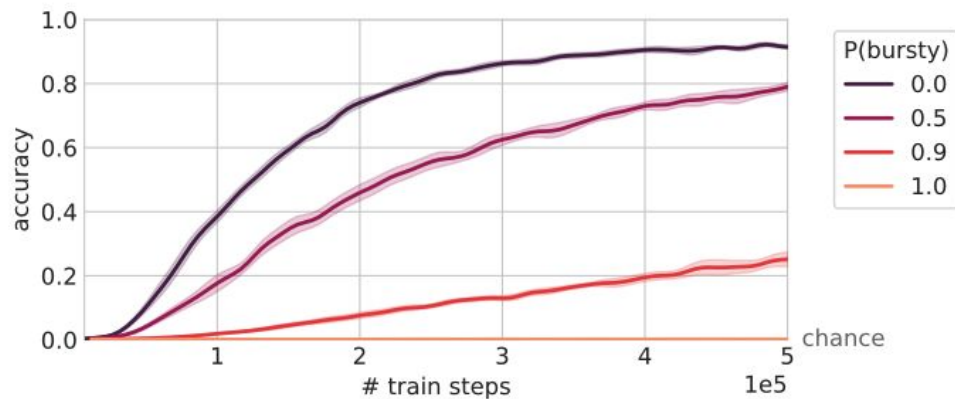


# Results: Burstiness => in-context learning

(a) In-context learning on holdout classes.

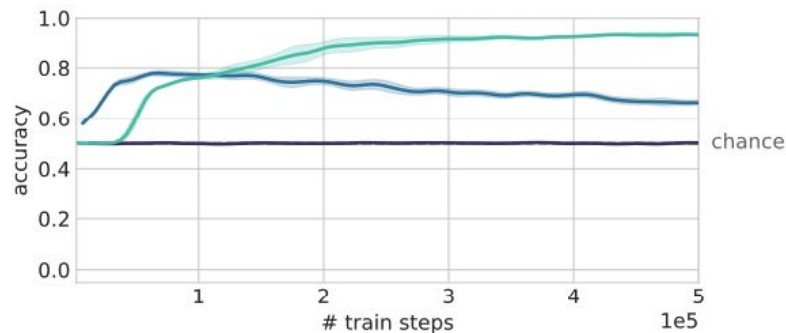


(b) In-weights learning on trained classes.

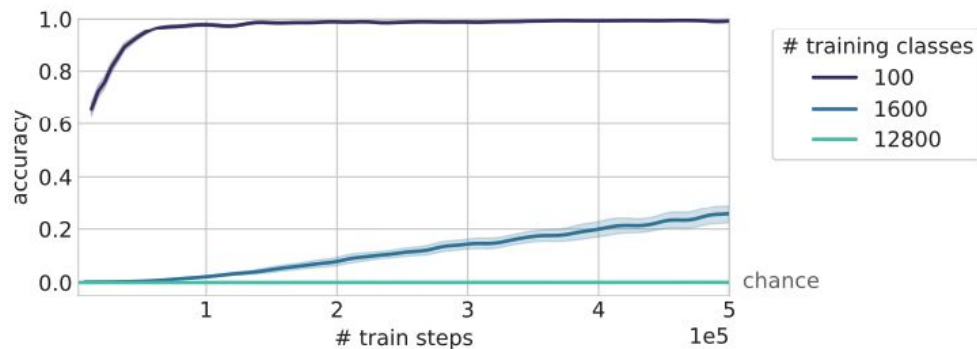


# Results: Large # classes => in-context learning

(a) In-context learning on holdout classes.



(b) In-weights learning on trained classes.





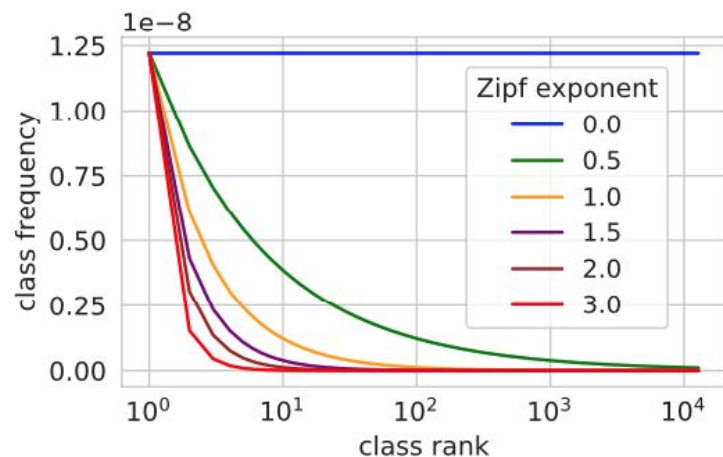
There's a tradeoff between in-context learning and in-weights learning

Can you get both?

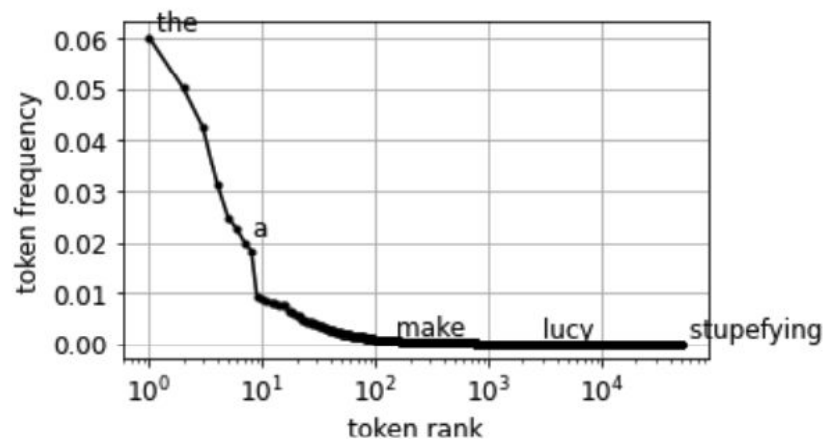
# Experimental Design: Zipfian distribution

$$p(x) \propto \frac{1}{\text{rank}(x)^\alpha}$$

(a) Examples of Zipfian distributions.



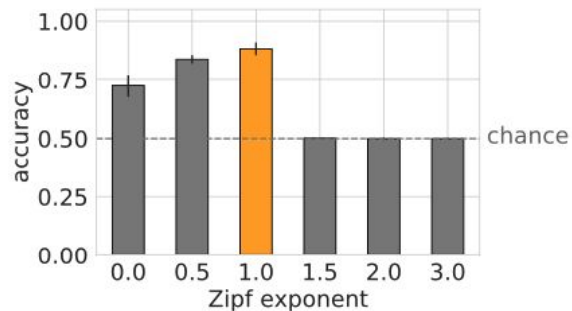
(b) Distribution of tokens in a natural language corpus.



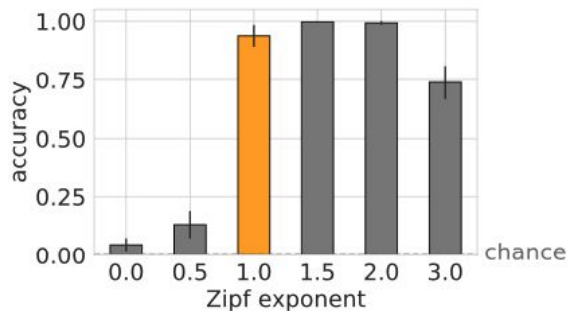
# Results: Zipfian Distribution

“Common” classes means the top 10. “Rare” is all others.

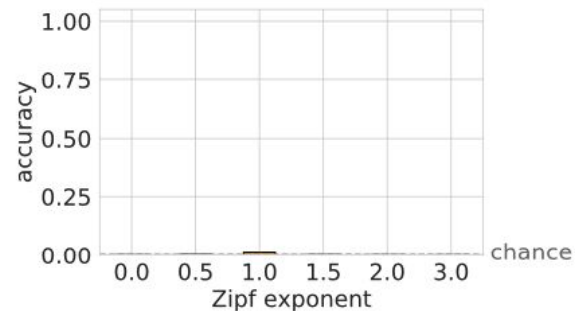
(c) In-context learning on holdout classes.



(d) In-weights learning on common classes.



(e) In-weights learning on rare classes.



# Questions

- How should we define in-context learning mathematically?
  - Formalize the distinction between “locating” a concept from the training distribution vs. learning it in context
  - Is it all just about composing concepts learned from training in novel ways?
- What’s the simplest toy problem / model for which we can prove in-context learning will emerge
  - Chan et al.’s task is designed so you can always fit training data perfectly *without* it
  - Want to capture Rong’s intuition of Bayesian updating with more in-context examples
- How does architecture influence the emergence of in-context learning
  - Chan et al. found: LSTMs can’t do it
  - Expressive enough to represent pretty general string transformations
  - Still has a bias against memorizing

# A dumb model

- Finite learning “budget” of 1
  - Cost to memorize a character’s label is  $c$
  - Cost to learn the pattern of bursty examples is  $d$
- Number of characters  $n$
- Marginal distribution of characters follows power law
  - Sampled character is one of the top  $k$  with probability  $(k/n)^{1/\alpha}$
- Training sample is bursty w.p.  $p$

=> Training performance...

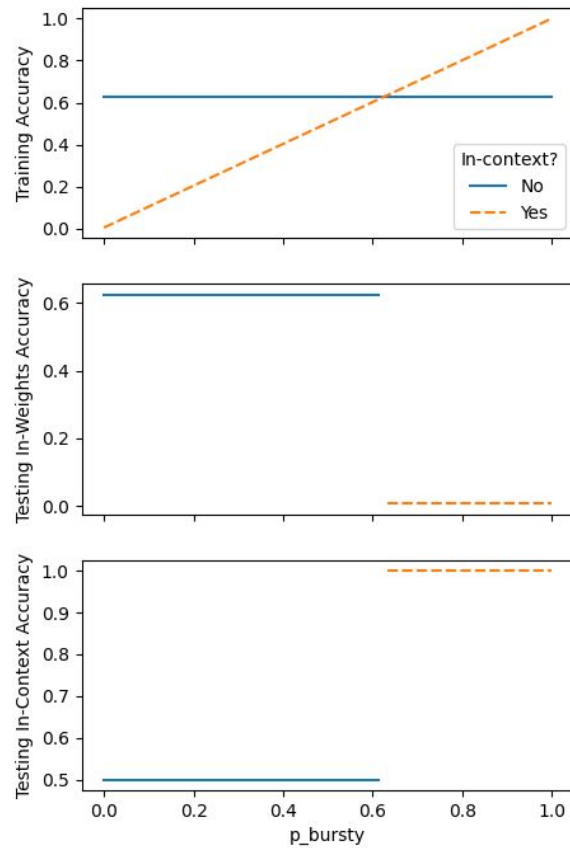
- Without in-context learning

$$\left(\frac{1/c}{n}\right)^{1/\alpha}$$

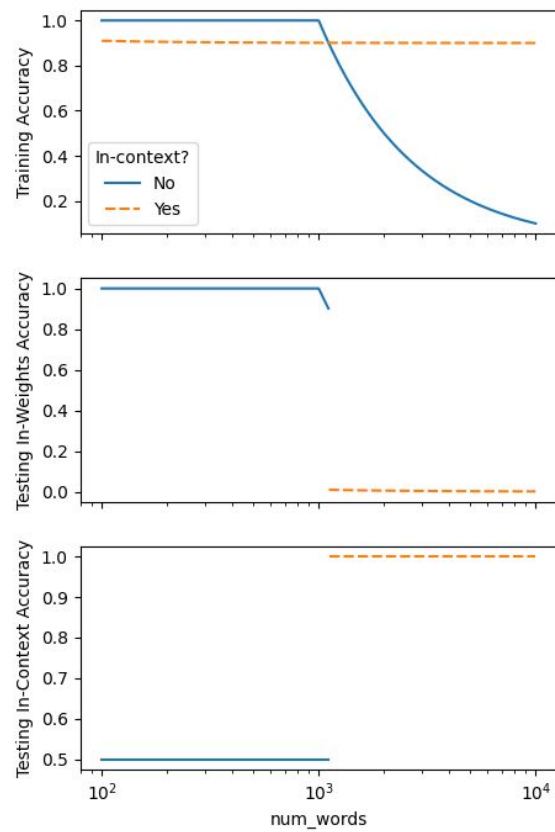
With in-context learning

$$p + (1 - p) \left(\frac{1 - d}{cn}\right)^{1/\alpha}$$

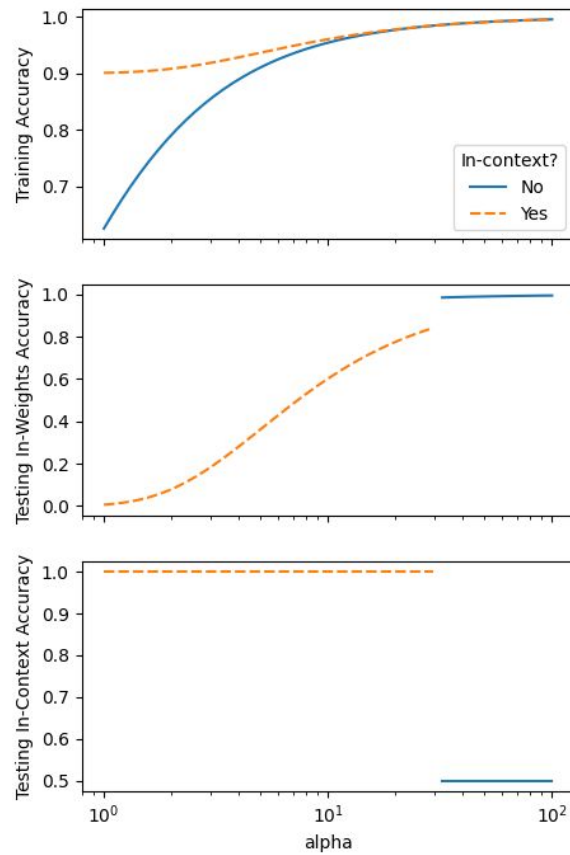
Effect of p\_bursty



Effect of num\_words



Effect of alpha





Effect of budget

