Understanding In-Context Learning using Simple Models

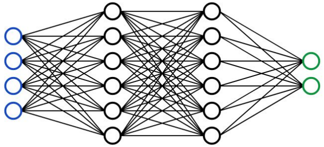
Percy Liang

Center for Research on Foundation Models

A language model called GPT-3

Dataset	Quantity (tokens)
Common Crawl (filtered)	410 billion
WebText2	19 billion
Books1	12 billion
Books2	55 billion
Wikipedia	3 billion

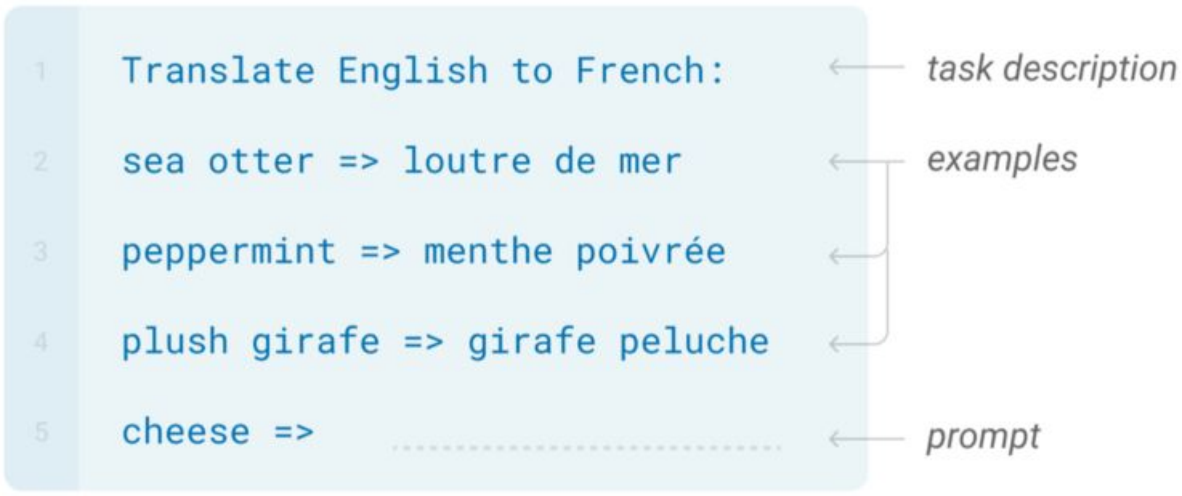
In 1885, Stanford University was ____



175 billion parameters

3640 petaFLOPS-days

In-context learning in GPT-3 (Brown et al. 2020)



GPT-3 can perform “unnatural” tasks (Rong, 2021)

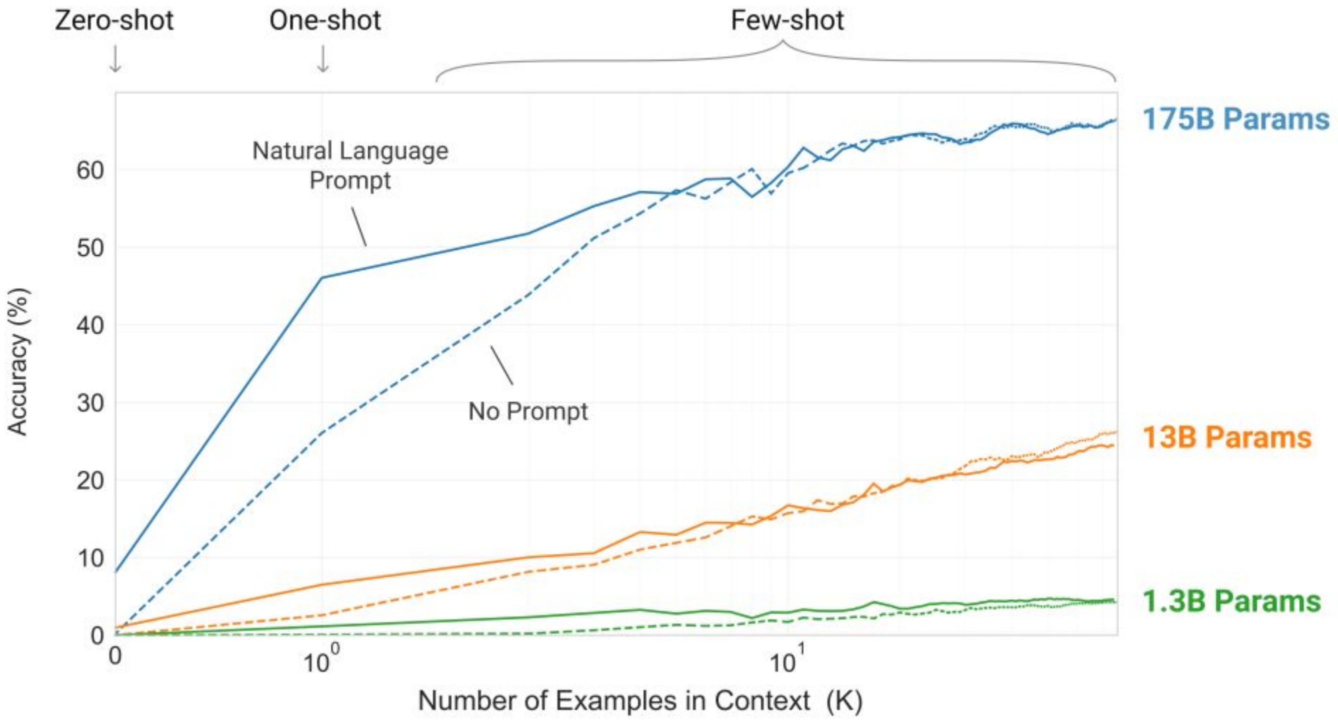
```
Input: 2014-06-01
Output: !06!01!2014!
Input: 2007-12-13
Output: !12!13!2007!
Input: 2010-09-23
Output: !09!23!2010!
Input: 2005-07-23
Output: !07!23!2005!
```

in-context examples

test example

model completion

Scale matters



Why in-context learning matters

Scientific: emergent phenomena

- GPT-3 was **not built** to explicitly do in-context learning
- Train (predict next word) ≠ test (wide range of downstream tasks)
- What **else** is there? Chain of thought (Wei et al. 2022), etc.

Practical: paradigm shift in how we build ML systems

- Can **prototype** new tasks in an afternoon rather than setup a data collection
- In real world, we don't start with a dataset, but with a **vague idea** of what we want to do

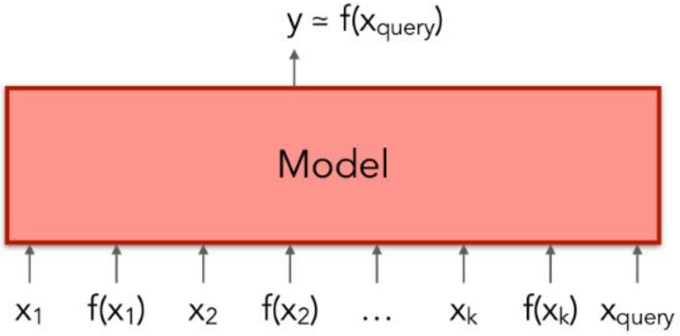
Two types of learning

Standard learning: **gradient**

$$w \leftarrow w - \nabla \, \mathrm{loss}(x_i, y_i, w)$$

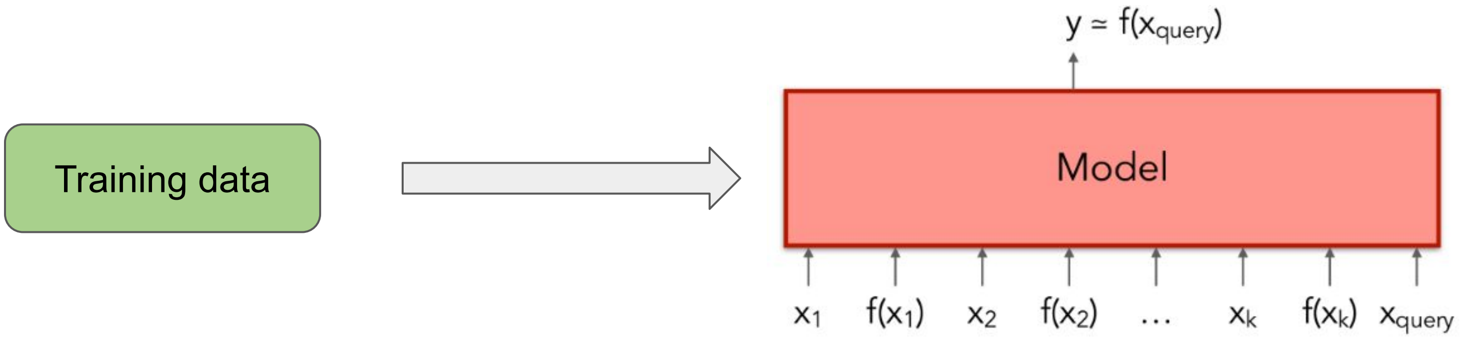
In-context learning: **conditioning**

$$p(y_{query} \mid x_1, y_1, \ldots, x_k, y_k, x_{query})$$



Relationship to meta-learning

Black-box meta learning is a paradigm for (meta-)training…



…a model that can perform **in-context learning**.

Questions

1. How can a **fixed** model (e.g., a Transformer) perform in-context learning?

2. How does such a model arise from **training** (e.g., on next word prediction)?

How can we understand in-context learning?

Components

- **Data:** Synthetic? CommonCrawl? The Pile?
- **Model architecture**: Transformer? RNNs? Mixture of experts?
- Training objective / algorithm: autoregressive? contrastive learning?

How can we study this?

- **Theoretical:** develop toy model, analytically understand why
- **Synthetic experiments:** develop simple model, draw clean conclusions
- Real-world experiments: realistic, but messy and expensive

Outline

1. What Can Transformers Learn In-Context? (NeurIPS 2022)

    architecture, synthetic experiments

2. In-context Learning as Implicit Bayesian Inference (ICLR 2022)

    data, synthetic experiments + theory

What Can Transformers Learn In-Context?

(NeurIPS 2022)

Shivam Garg
(on the job market!)

Dimitris Tsipras
(on the job market!)

Percy Liang

Greg Valiant

Language models can perform in-context learning

night -> Nacht
book -> Buch
table -> Tisch
chair -> Stuhl

Review: Pretty bad movie.
Sentiment: Red

Review: I loved this film!
Sentiment: Green

2 # 2 = 4
3 # 8 = 11
8 # 2 = 10
2 # 4 = 6

(purely at inference-time)

Are models actually doing in-context **learning**?

night -> Nacht
book -> Buch
table -> Tisch
chair -> Stuhl

German translation of 'book'

book

[bʊk]

Word Frequency

NOUN

1. Buch *nt*; *(= exercise book)* Heft *nt*; *(= division: in Bible, poem etc)* Buch *nt*

the (good) Book das Buch der Bücher

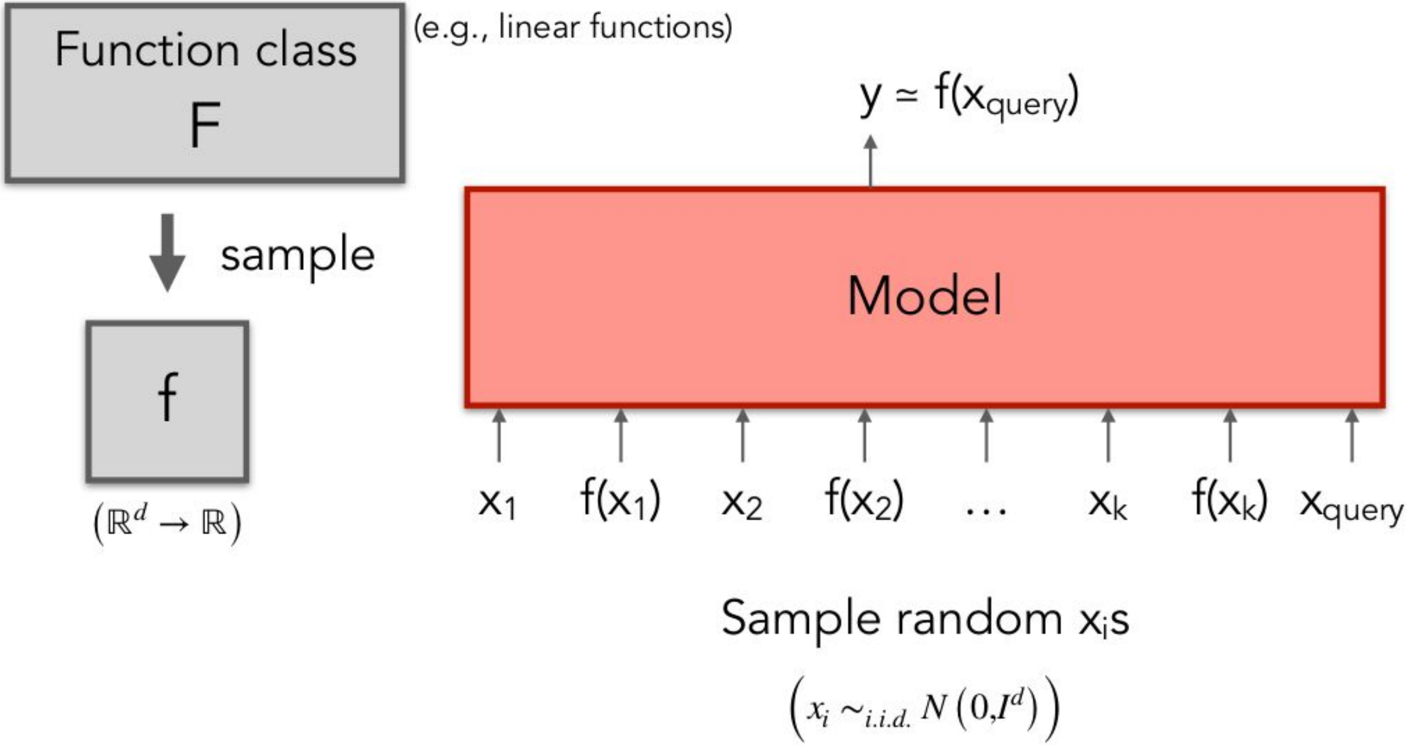
https://www.collinsdictionary.com/dictionary/english-german/book

Here's how to enhance your confidence by starting with some basic words and phrases to build your German word bank:

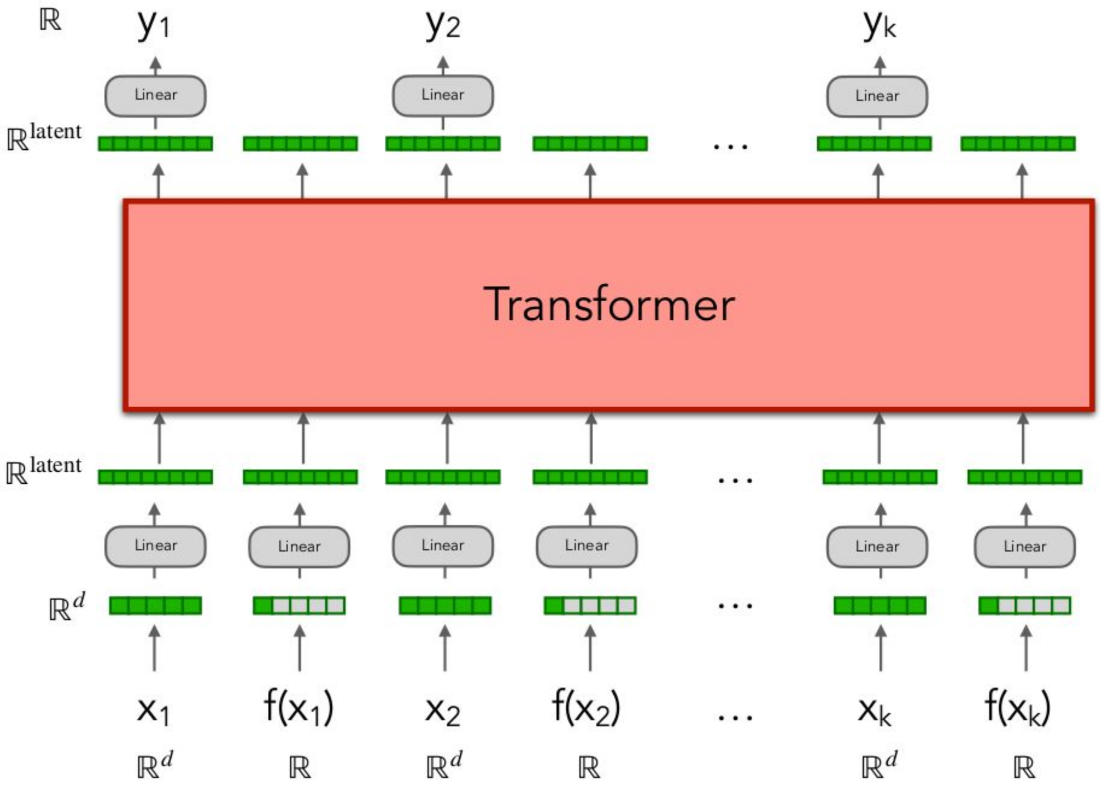
- Guten Tag = Good day
- Hallo = Hello
- Auf Wiedersehen = Goodbye
- Bitte = Please
- Danke = Thanks, Thank you
- Entschuldigung = Sorry
- Gesundheit = Bless you (after someone sneezes)
- Ja = Yes
- Nein = No

https://www.rosettastone.com/languages/german-words

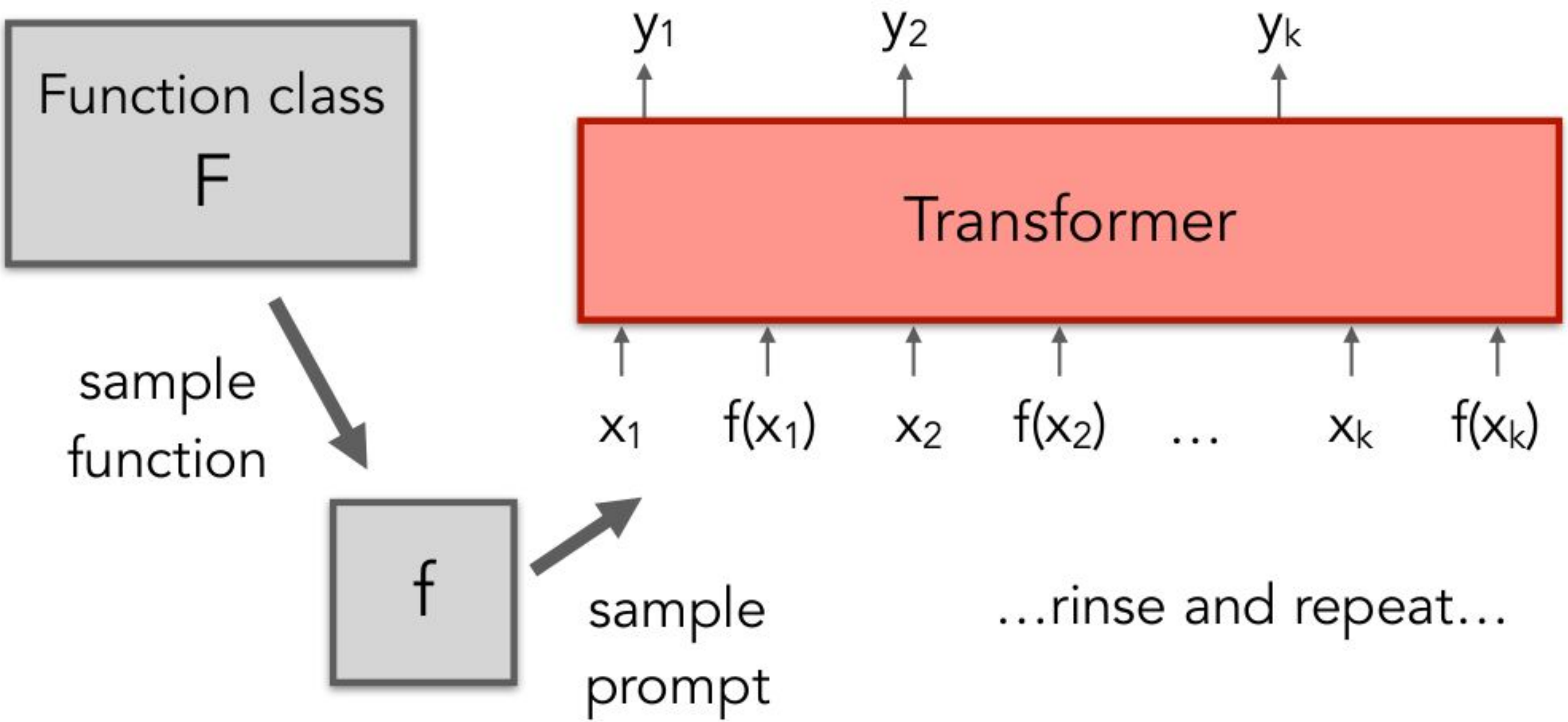
Definition: in-context learning a **function class**



Model architecture



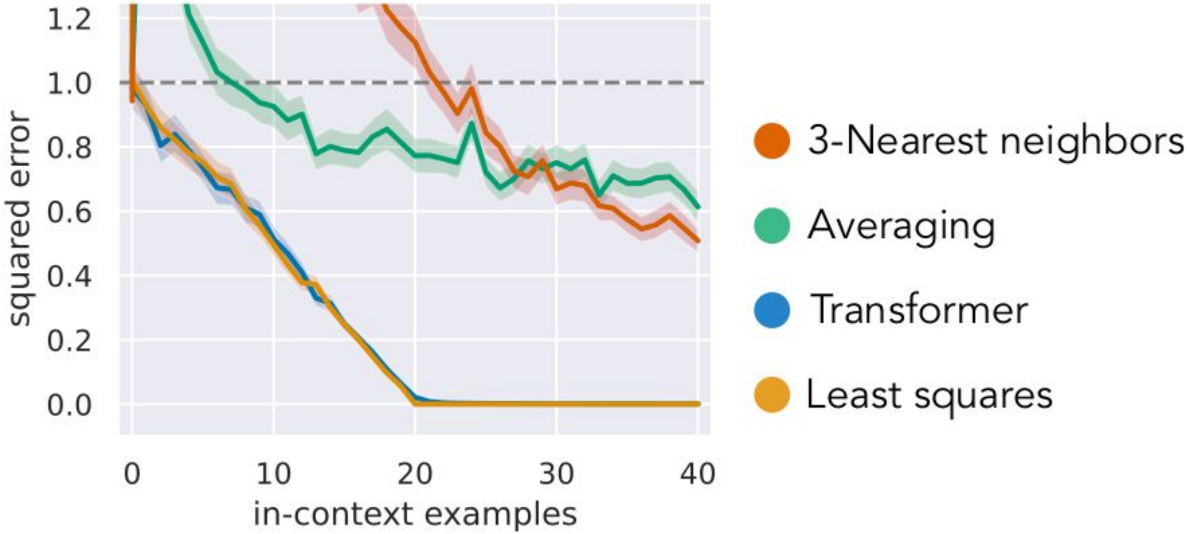
Training for in-context learning



(from scratch, no text)

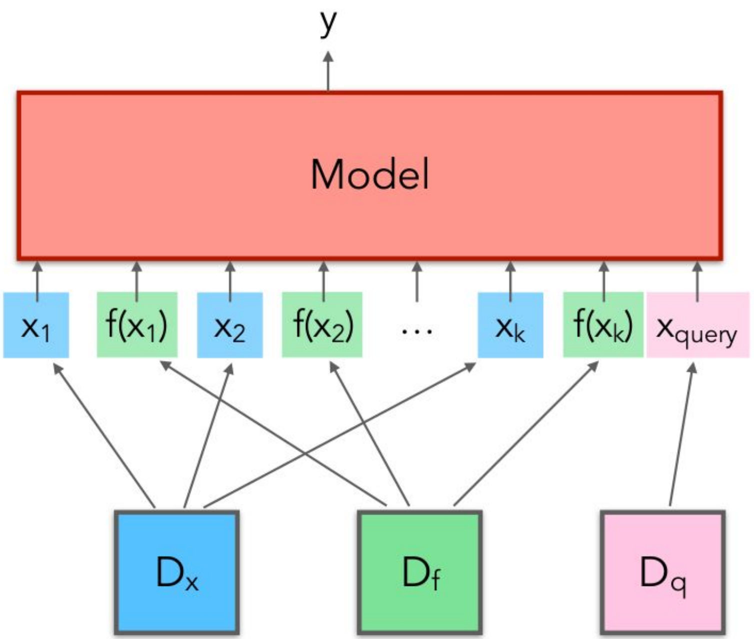
What can the trained Transformer in-context learn?

In-context learning linear functions (20 dimensions)



Transformer implements an algorithm like least squares!

Has the Transformer **really** learned to do least squares?

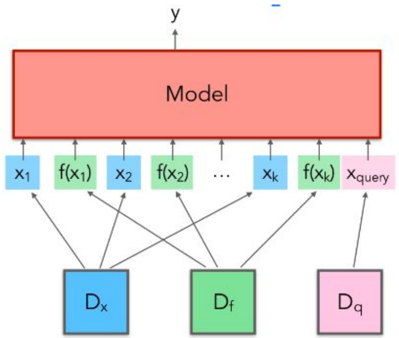


So far,

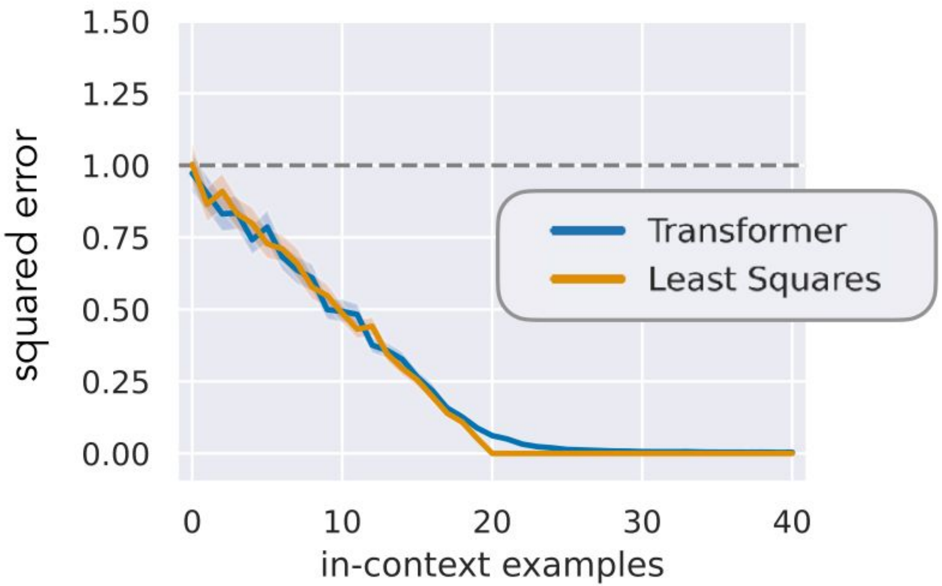
$D^{train} = D^{test}$

Can the Transformer extrapolate to different distributions?

Train: x ~ N(0, I)



Test: x and q from different orthants



Transformer matches least squares!

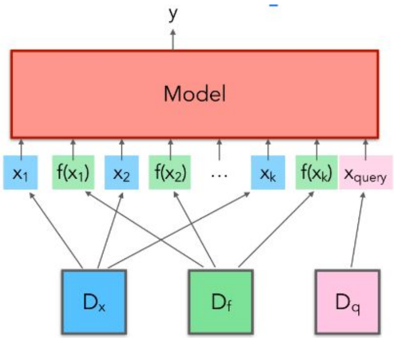
Train: identity covariance

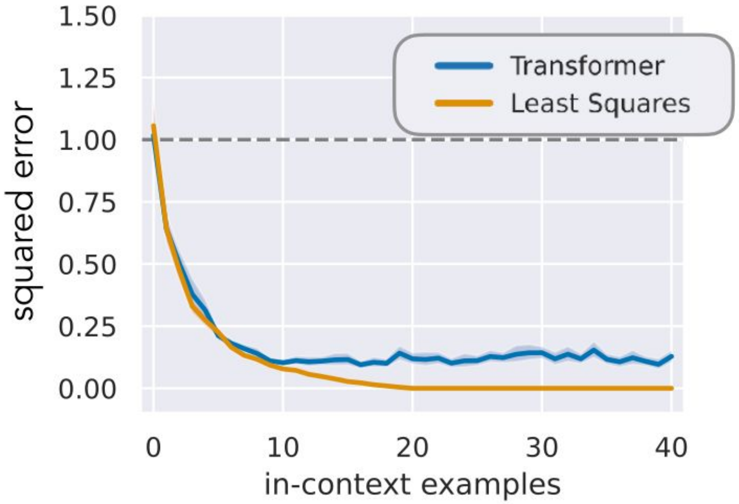
Test: skewed covariance

Skewed covariance: $x \sim N(0, \Sigma^2)$

(the i-th eigenvalue is proportional to $1/i^2$)

$(D_x^{train} \mathrel{!=} D_x^{test})$





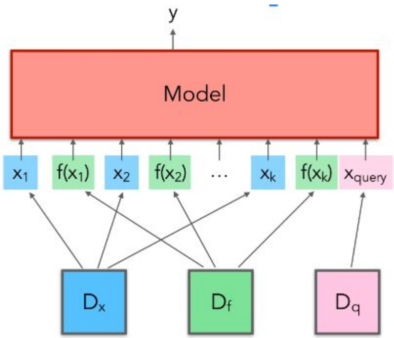
Transformer degrades a bit…

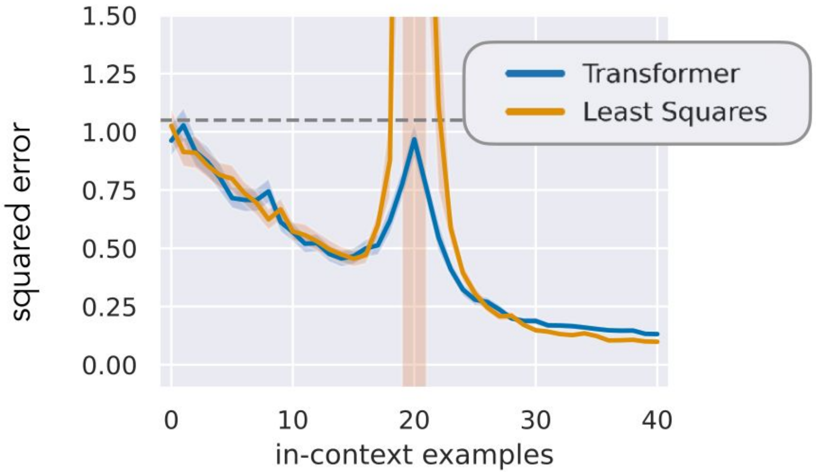
Train: no noise

Test: label noise

Label noise: $y = w^T x + N(0, 1)$

(D_f^{train} != D_f^{test})

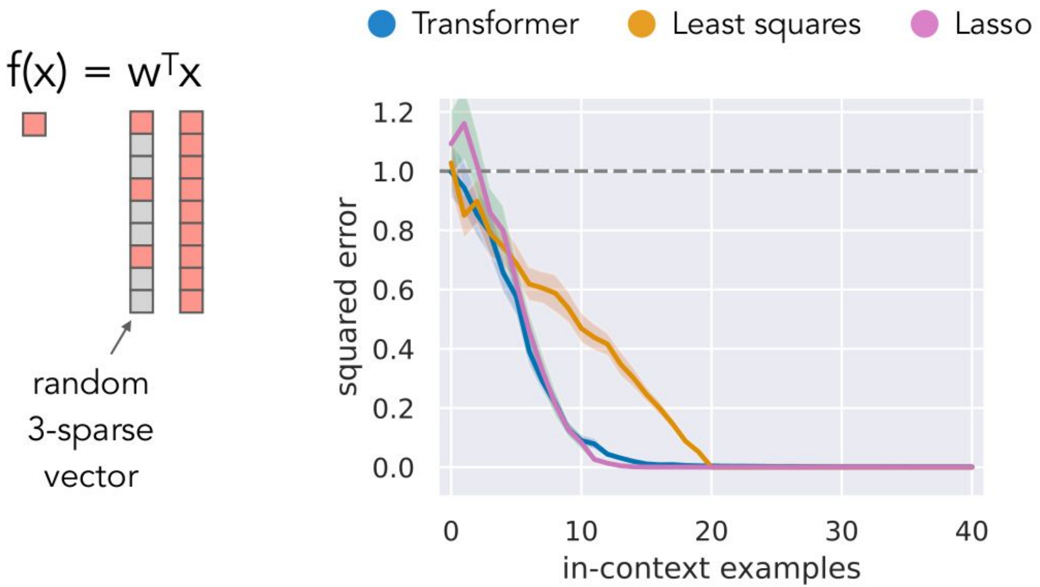




Transformer and least squares both exhibit double descent!

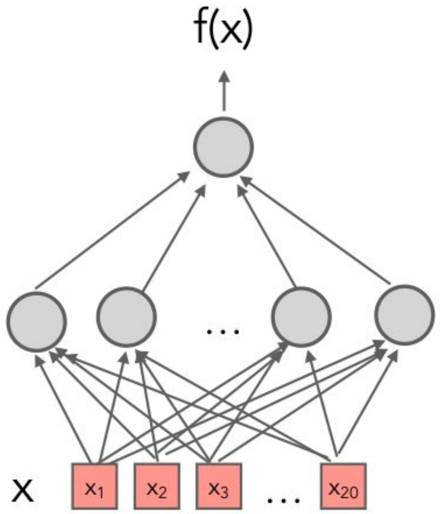
Can we train a Transformer to in-context learn function classes beyond linear functions?

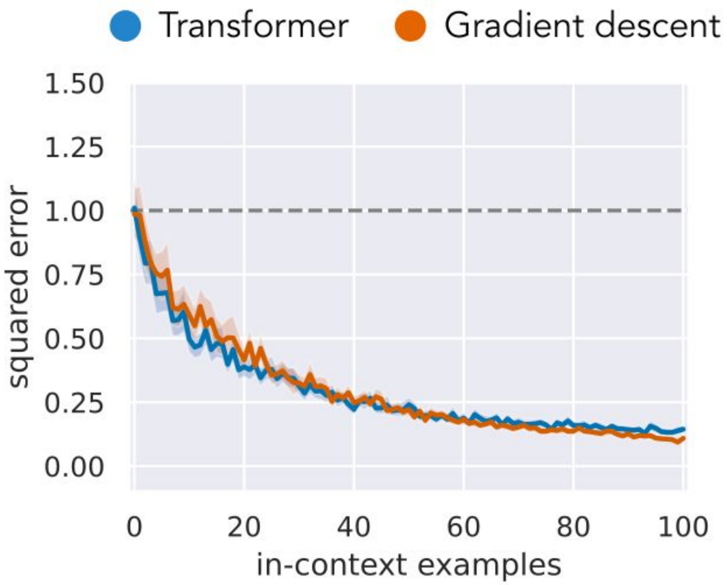
In-context learning sparse linear functions



Transformer matches lasso

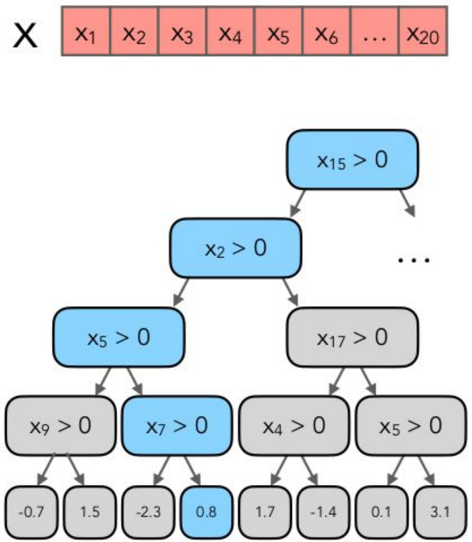
In-context learning 2-layer ReLU networks

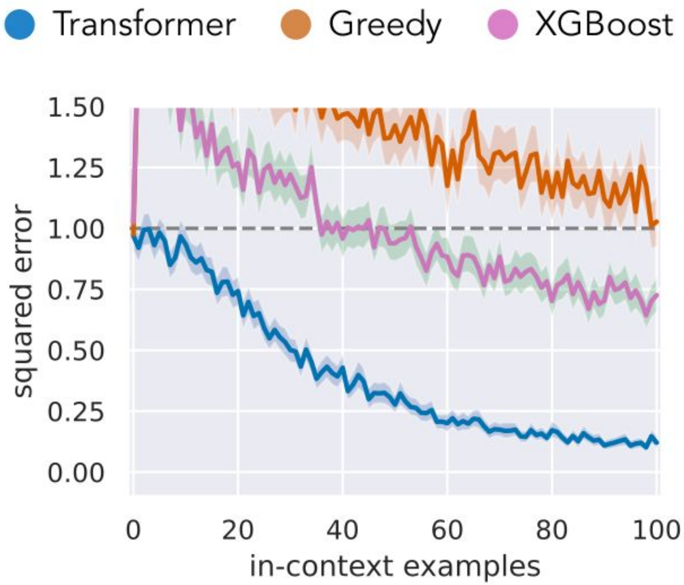




Transformer matches gradient descent

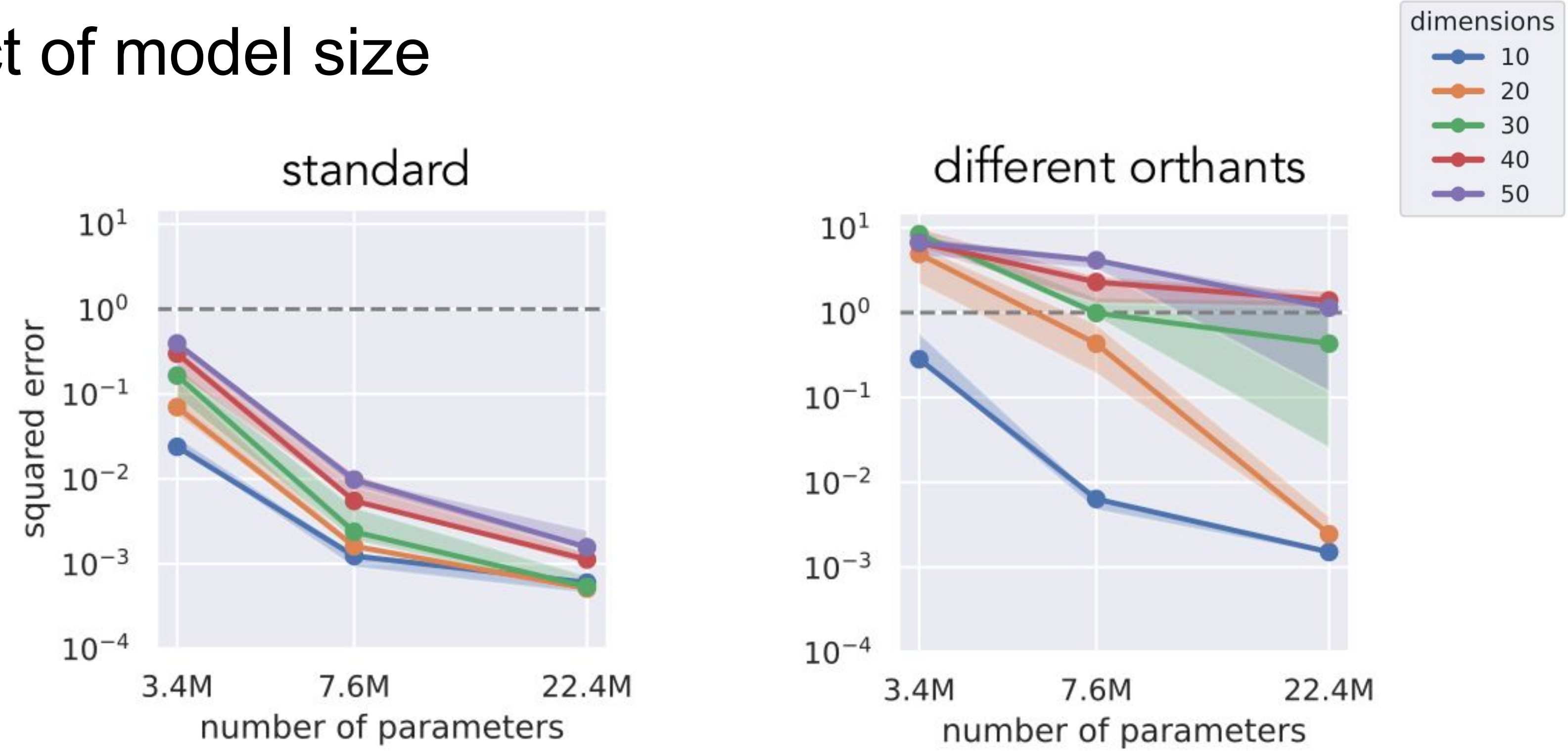
In-context learning decision trees





Transformer outperforms XGBoost!

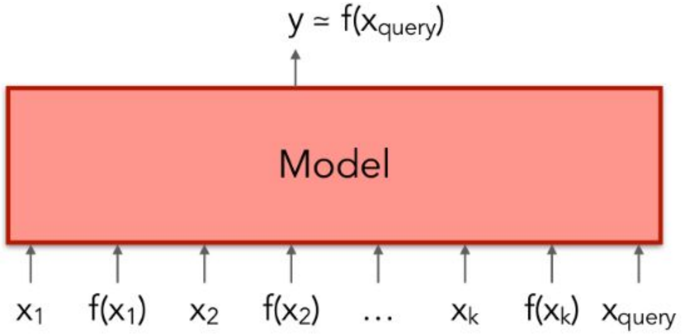
Impact of model size



Model size is especially important for extrapolation

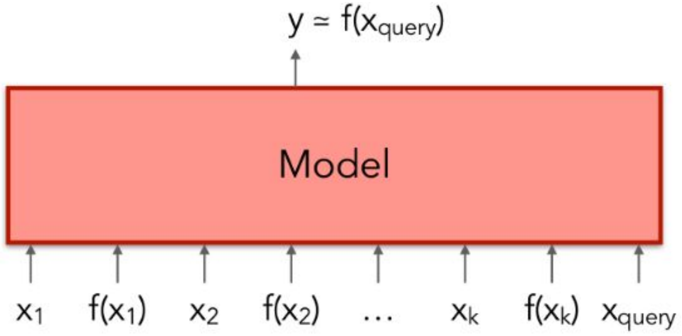
Summary

- Define in-context learning of function class (property of a model)
- Can train Transformer to in-context learn linear functions, sparse linear functions, neural networks, decision trees
- Evaluate Transformer on robustness to **out-of-distribution** prompts
- Can match qualitative behavior of lasso, double descent, etc. - Transformers are representing **learning algorithms**?



Open questions

- What are the properties of the Transformer's in-context learned **function**?
- Can other **architectures** (e.g., RNNs, S4, etc.) perform in-context learning?
- How can we understand the Transformer's algorithm **mechanistically**?
  - Construction of Transformer that does linear regression (Akyürek et al. 2022)
- Can we gain new **algorithmic insights**?
- How do we tie this back to **real tasks** with prior knowledge?



Outline

1. What Can Transformers Learn In-Context? (NeurIPS 2022)

    architecture, synthetic experiments

2. In-context Learning as Implicit Bayesian Inference (ICLR 2022)

    data, synthetic experiments + theory

In-context Learning as Implicit Bayesian Inference
(ICLR 2022)

Michael Xie Aditi Raghunathan Percy Liang Tengyu Ma

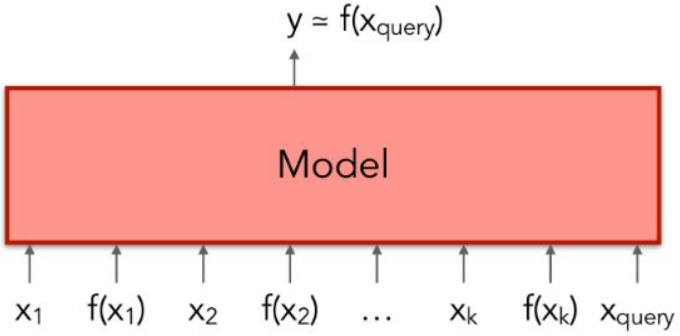
Two types of learning

Standard learning: **gradient**

$$w \leftarrow w - \nabla \, \text{loss}(x_i, y_i, w)$$

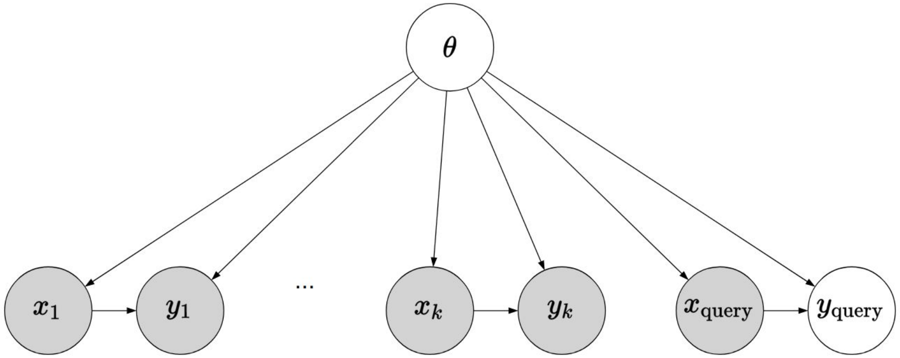
In-context learning: **conditioning**

$$p(y_{query} \mid x_1, y_1, \ldots, x_k, y_k, x_{query})$$



Bayesian inference

Posit a latent concept θ (e.g., task)



$$\underbrace{p(y_{\text{query}} \mid x_1, y_1, \ldots, x_k, y_k, x_{\text{query}})}_{\text{posterior predictive}} = \int p(y_{\text{query}} \mid x_{\text{query}}, \theta) \underbrace{p(\theta \mid x_1, y_1, \ldots, x_k, y_k)}_{\text{posterior}} d\theta$$

Transformer directly fits the posterior predictive distribution!

Questions

1. How can a **fixed** model (e.g., a Transformer) perform in-context learning?

    First part of talk: showed Transformer can learn linear regression

2. How does such a model arise from **training** (e.g., on next word prediction)?

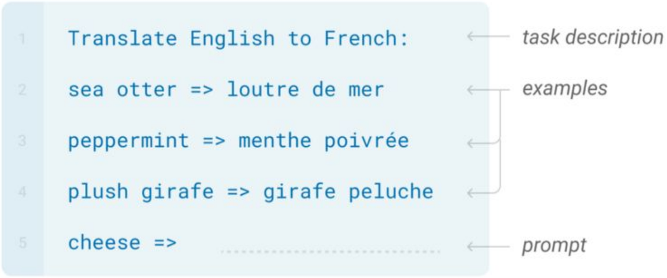
    This is the harder question…

Main challenge: distribution shift

Pretraining distribution

Dataset	Quantity (tokens)
Common Crawl (filtered)	410 billion
WebText2	19 billion
Books1	12 billion
Books2	55 billion
Wikipedia	3 billion

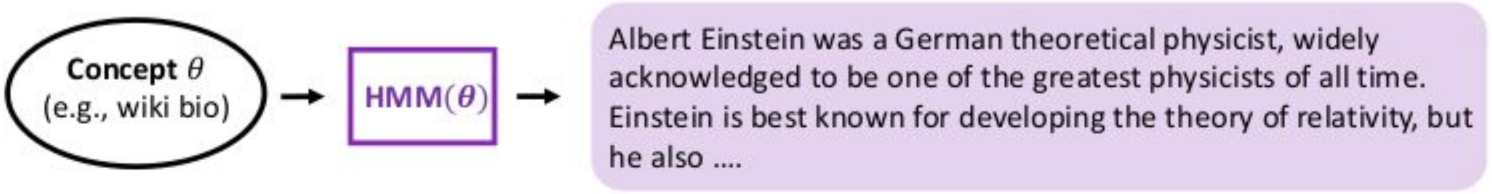
Prompting distribution



pretraining distribution ≠ prompting distribution

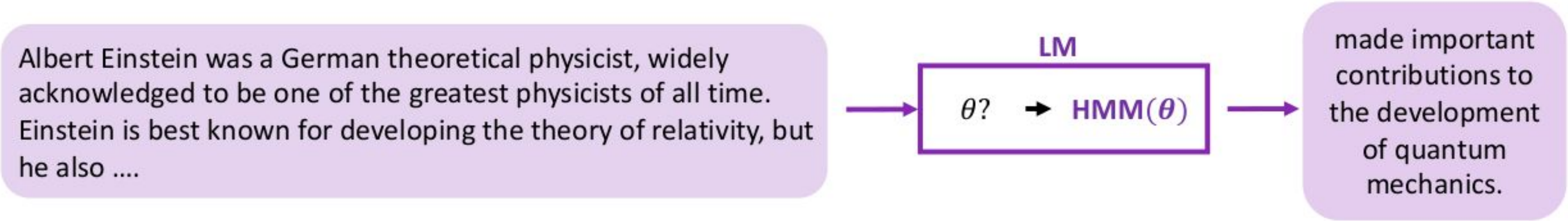
Pretraining distribution: mixture of HMMs

Concept θ encodes transitions in HMM



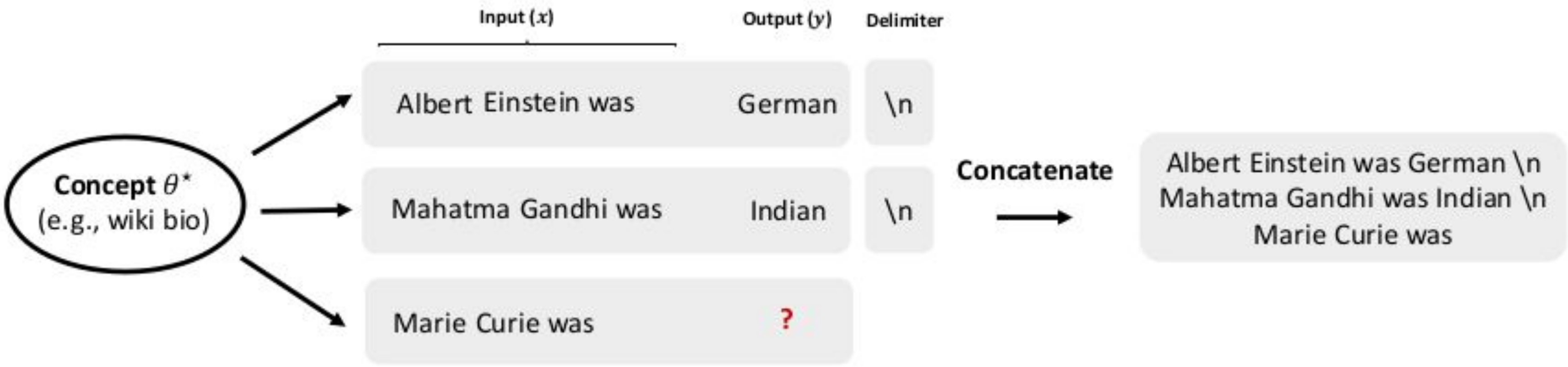
Intuition

Language model **implicitly** infers latent concept θ



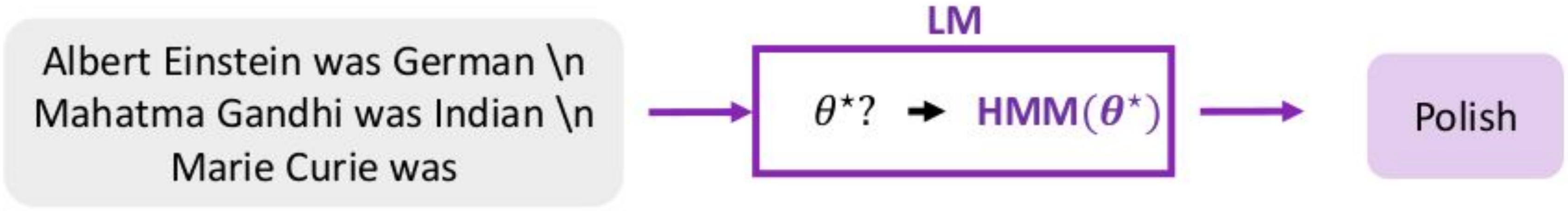
Prompting distribution: one HMM, independent pieces

Generate in-context examples independently from HMM(θ^*)



Intuition

Can the language model infer concept θ*?



Difficulty: condition on samples from prompt distribution, not training distribution!

Prompting distribution

Low-prob transitions
between examples

In-distribution transitions
reveal information about $\theta^\star$

Main theoretical result (sketch)

If

Token length of one example

$$\sum_{j=1}^{k} KL_j(\theta^* \| \theta) > \epsilon^{\theta}_{start} + \epsilon^{\theta}_{delim} \quad \text{for all } \theta \neq \theta^\star$$

Signal about $\theta^\star$

Error from low-prob transitions

Then as $k \to \infty$, in-context learning is asymptotically consistent:

$$\text{argmax}_y\, p_{train}(y \mid x_1, y_1, \ldots, x_k, y_k, x_{query}) \to \text{argmax}_y\, p_{prompt}(y \mid x_{query}, \theta^*)$$

Takeways

- Try to make prompting distribution close to training distribution
    - e.g., Berlin **is the capital of** Germany
- Use neural delimiters that don’t increase probability of wrong concept
    - e.g., \n, #

Next: run experiments on a synthetic dataset to test theory…

GINC: **g**enerative **in-c**ontext dataset

- A small-scale dataset for studying in-context learning
- Pretraining: 1000 documents, each doc is one long sequence from some HMM(θ)
- Prompting: 2500 prompts with concatenated independent examples

Pretraining document

```
f / h x ax o a k au ap /
a o u au ae f ao an / ah
u y as a k au j w ax l
aw r ae au g au ap / / u
aj ae d a h x af u aj i
r j w j as y x n i ap
```

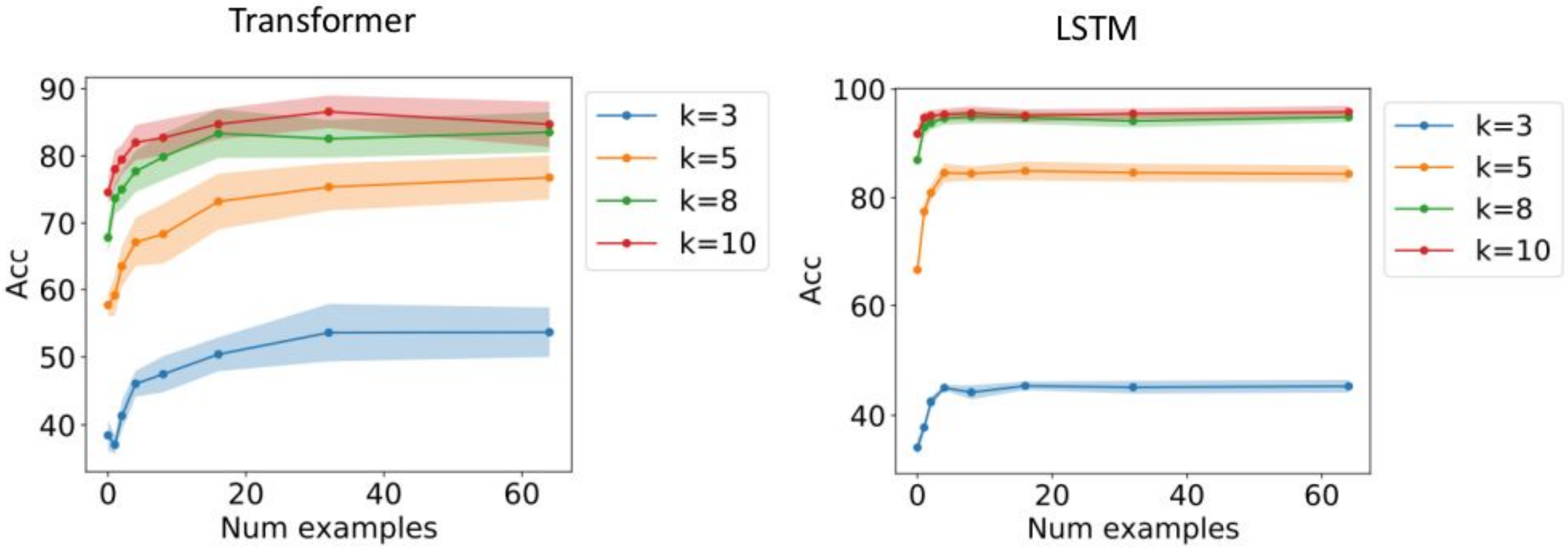
...

In-context Prompt

```
l aw ac / ax aj ae / ac j
```

Results

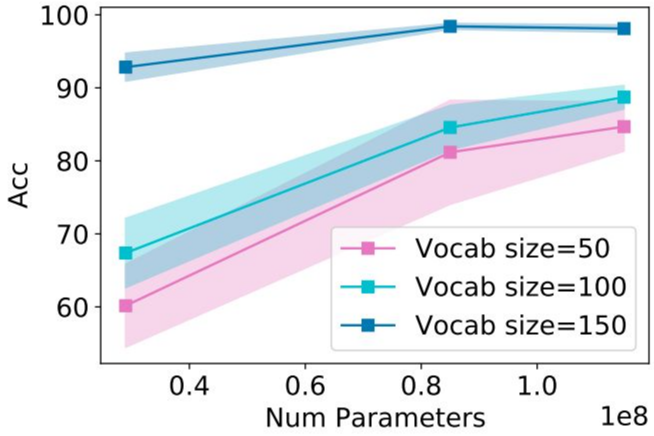
k: length of an example



Trained Transformers and LSTMs can do in-context learning

Effect of model scaling

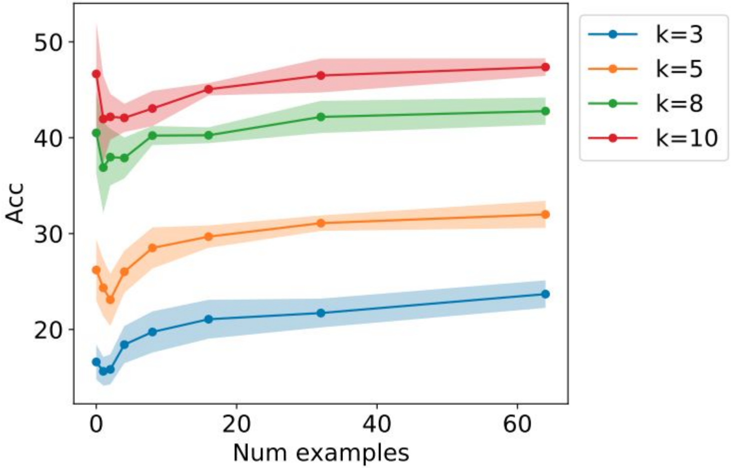
- In GINC: in-context accuracy improves with model size (no surprise)
- But it improves **even if pretraining loss is the same**
- Inductive bias for in-context learning improves with model size?



Transformer # layers	GINC Vocab size	Pretrain Val loss	In-context Acc
12 layer	50	1.33	81.2
16 layer	50	1.33	**84.7**

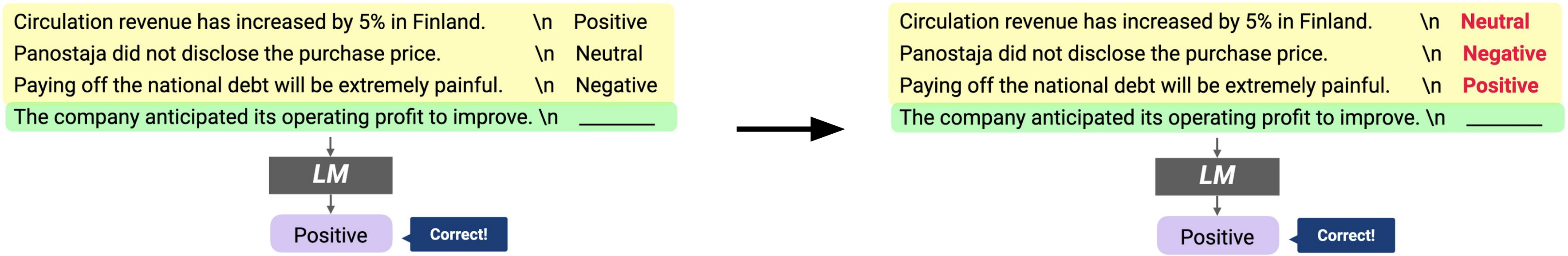
0-shot can be better than 1-shot

- GPT-3: 0-shot is better than 1-shot for some datasets (e.g., LAMBADA, HellaSwag, PhysicalQA, RACE-m)
- Sometimes the same thing happens in GINC:



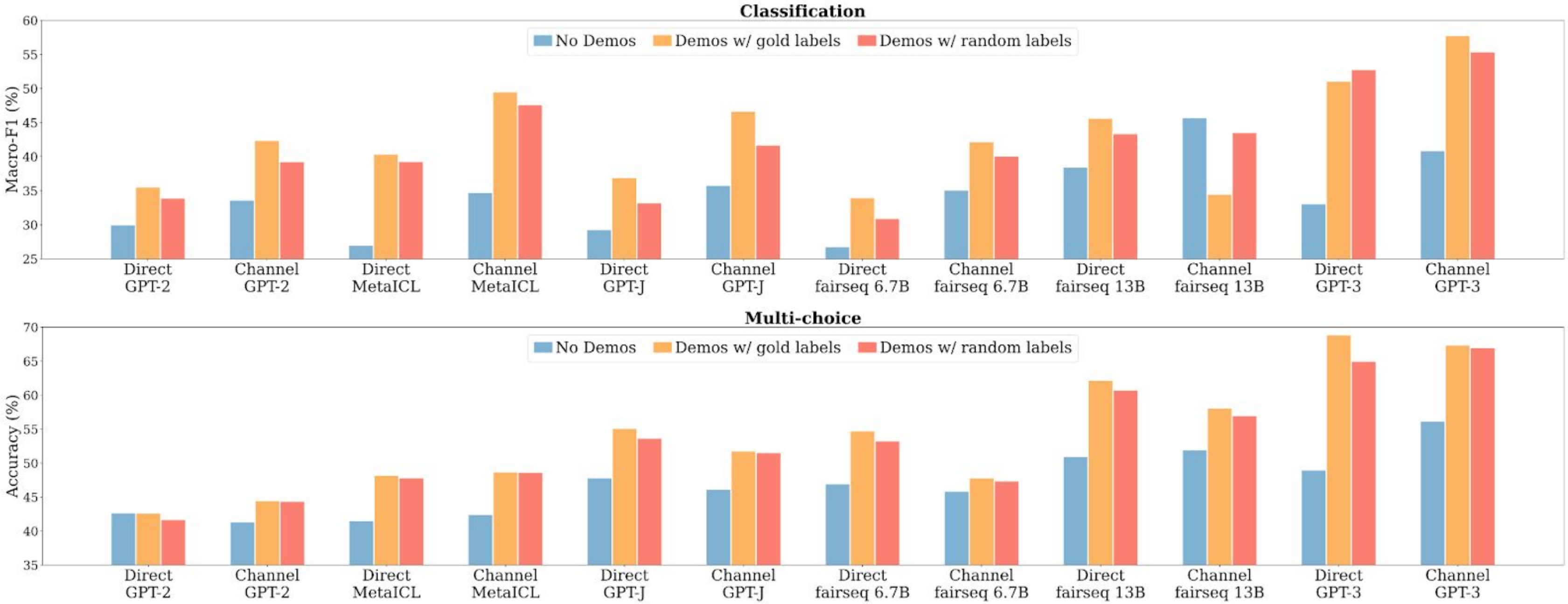
Prompting using random labels (Min et al. 2022)

If randomize labels:



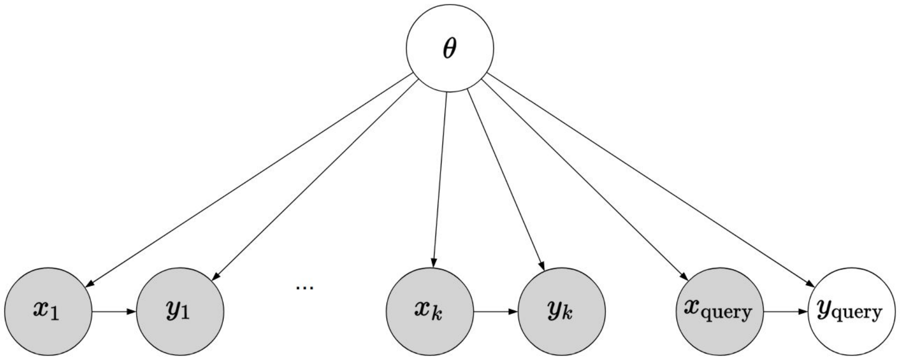
Prompting using random labels (Min et al. 2022)

Accuracy doesn't drop much (would destroy traditional supervised learning):



Explanation using Bayesian inference

In-context inputs $x_1, ..., x_k$ help us nail down the concept despite noisy labels!



Inintuitive relabelings (Rong 2021)

Training examples (truncated)

```
beet: sport
golf: animal
horse: plant/vegetable
corn: sport
football: animal
```

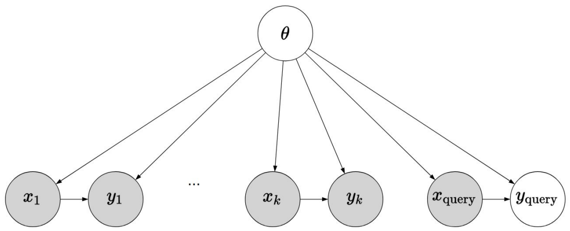
Test input and predictions

```
monkey: plant/vegetable ✓
panda: plant/vegetable ✓
cucumber: sport ✓
peas: sport ✓
baseball: animal ✓
tennis: animal ✓
```

Not captured in our framework…

Summary

- Bayesian inference: useful way to think about in-context learning
  - Learning = conditioning
  - Approximate the posterior predictive $p(y_{query} \mid x_1, y_1, \ldots, x_k, y_k, x_{query})$
- Main challenge: pretraining distribution ≠ prompting distribution
  - Can bound the errors due to transitions
- All theoretical results are independent of architecture, all about the data distributions!
- GINC: small synthetic dataset provides testbed for learning



Outline

1. What Can Transformers Learn In-Context? (NeurIPS 2022)

    architecture, synthetic experiments

2. In-context Learning as Implicit Bayesian Inference (ICLR 2022)

    data, synthetic experiments + theory

Final remarks

- In-context learning is one of the great **mysteries** in modern AI
- It is becoming the **foundation** for many AI applications
- **Understanding** is key to scientific progress and engineering better systems
- This talk: **synthetic** setups can help us more **rigorously** explore
  - …the role of the model architecture
  - …the role of the data distributions
- Open question: connect this with real settings
- Other **emergent phenomena** (chain-of-thought)
  - Scrap the idea of a task!