Recent Advances in In-Context Learning

Guande He

2023.3.24

Transformer-based Language Models

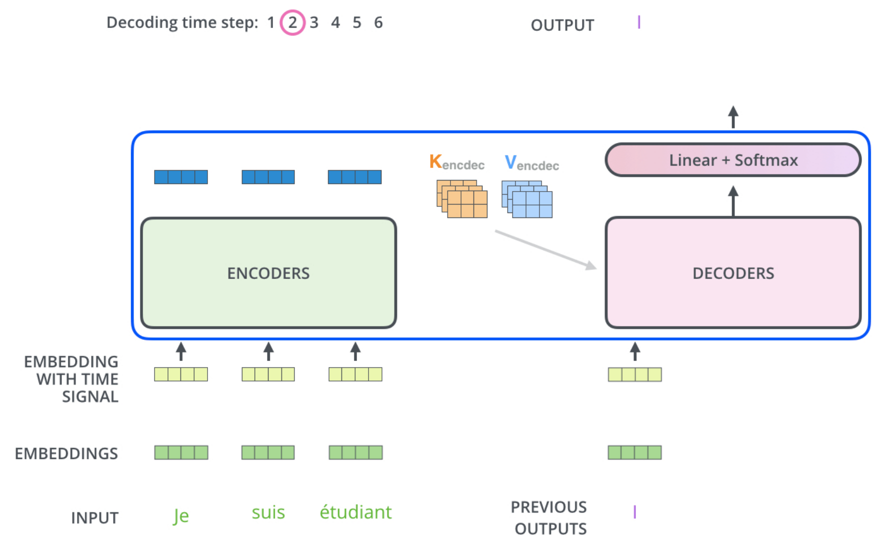
- Pre-trained generative language model: $p_\theta(\boldsymbol{x})$

Autoregressive Language Model (e.g., GPT):

$$\mathcal{L}_{\mathrm{alm}} = -\sum_{t=1}^{T} \log p_\theta(x_t | \boldsymbol{x}_{<t})$$

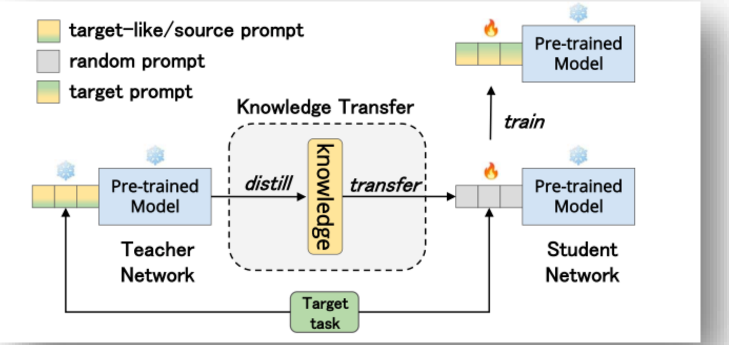
- Masked Language Model (e.g., BERT):

$$\mathcal{L}_{\mathrm{mlm}} = -\frac{1}{K}\sum_{k=1}^{K} \log p_\theta(x_{\pi_k} | \boldsymbol{x}_{-\Pi})$$



Alammar, J. " The Illustrated Transformer. " (2018).

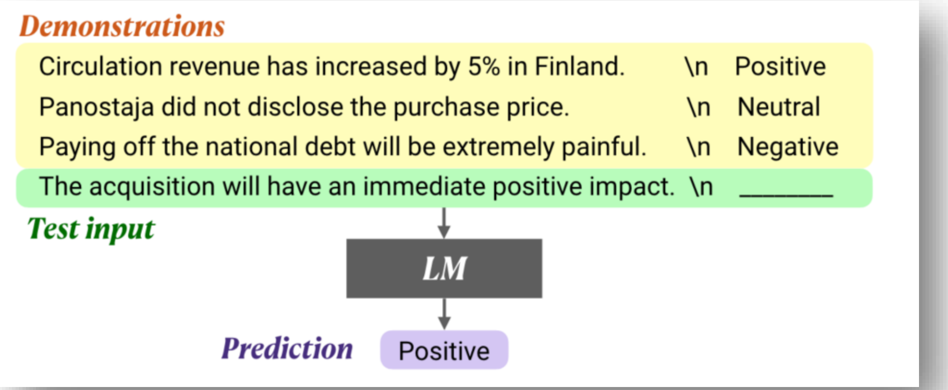
Emerging Abilities of LLMs



- In-Weight Learning:
  - Gradient-based parameter updates.
  - Learn or "remember" class information during training.

	Rank	Name	Model	URL	Score
	1	JDExplore d-team	Vega v2		91.3
+	2	Liam Fedus	ST-MoE-32B		91.2
	3	Microsoft Alexander v-team	Turing NLR v5		90.9
	4	ERNIE Team - Baidu	ERNIE 3.0		90.6
	5	Yi Tay	PaLM 540B		90.4
+	6	Zirui Wang	T5 + UDG, Single Model (Google Brain)		90.4
+	7	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4		90.3
	8	SuperGLUE Human Baselines	SuperGLUE Human Baselines		89.8

- In-Context Learning:
  - No parameter updates.
  - Learn with a concatenation of demonstrations.



In-Context Learning

In-Context Learning (ICL) was popularized in the original GPT-3 paper as a way to use language models to learn tasks given only a few examples.

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. // ________

LM

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. // ________

LM

Xie, Sang Michael, and Min, Sewon "How does in-context learning work? A framework for understanding the differences from traditional supervised learning" (2022).

The mystery of ICL

- What can ICL do?
  - On many NLP benchmarks, ICL is **competitive with supervised learning** using less labeled data.
  - ICL has enabled people to build new applications in just a few hours (prompt engineering).
- Why ICL surprising?
  - ICL **does not need any parameter updates**.
  - ICL just emerges from large PLMs, which there is a mismatch between pretraining and ICL
- What the model does when conducting ICL?
  - Indexing into a vast set of known tasks from the training data?
  - The model have developed the ability to learn new tasks from in-context examples?

Xie, Sang Michael, and Min, Sewon "How does in-context learning work? A framework for understanding the differences from traditional supervised learning" (2022).

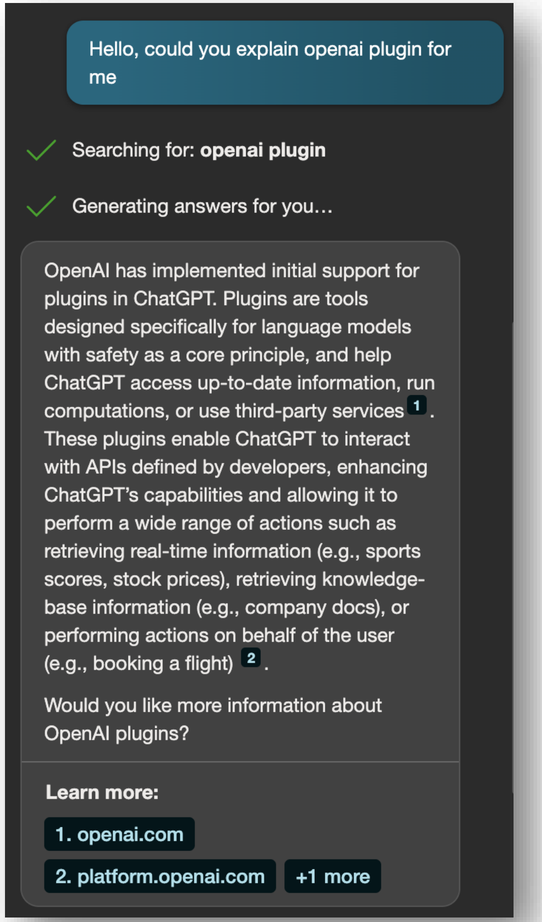
Example: OpenAI Chat API

```
Prompt: [
    {'role': 'system', 'content': 'The following are multiple choice questions (with answers) about machine_learning.'
    {'role': 'system', 'content': 'A 6-sided die is rolled 15 times and the results are: side 1 comes up 0 times; side
    {'role': 'system', 'content': 'B', 'name': 'example_assistant'},
    {'role': 'system', 'content': 'Which image data augmentation is most common for natural images?\nA. random crop an
    {'role': 'system', 'content': 'A', 'name': 'example_assistant'},
    {'role': 'system', 'content': 'You are reviewing papers for the World's Fanciest Machine Learning Conference, and
    {'role': 'system', 'content': 'C', 'name': 'example_assistant'},
    {'role': 'system', 'content': "To achieve an 0/1 loss estimate that is less than 1 percent of the true 0/1 loss (w
    {'role': 'system', 'content': 'D', 'name': 'example_assistant'},
    {'role': 'user', 'content': 'Which of the following can only be used when training data are linearly separable?\nA
Sampled: A. Linear hard-margin SVM.
```

From OpenAI/evals

Instruction as 'Zero-shot' ICL

- The ability of following instruction is obtained from instruction tuning, which allows model directly follow instructions without context examples.
- The AI model acts as an intelligent API caller. Given an API spec and a **natural-language description** of when to use the API, the model proactively calls the API to perform actions.
- Example: OpenAI Plugin
  - "description_for_model": "Plugin for searching through the user's documents (such as files, emails, and more) to find answers to questions and retrieve relevant information. Use it whenever a user asks something that might be found in their personal information."
  - Description acts as hyperparameter.



A Framework for ICL

- Pretraining distribution.
  - A latent concept (task) θ from a family of concepts Θ defines a distribution over observed tokens o from a vocabulary $\mathcal{O}$.
  - Generating document: First sample $\theta \sim p(\theta)$, then generate corresponding document by $p(o_1, \ldots, o_T \mid \theta)$, which is defined by a HMM. The concept θ determines the transition probability matrix of HMM $h_1, \ldots, h_T$ from a hidden state set $\mathcal{H}$.
  - Pretraining: $p(o_1, \ldots, o_T) = \int_{\theta \in \Theta} p(o_1, \ldots, o_T \mid \theta) p(\theta) d\theta$

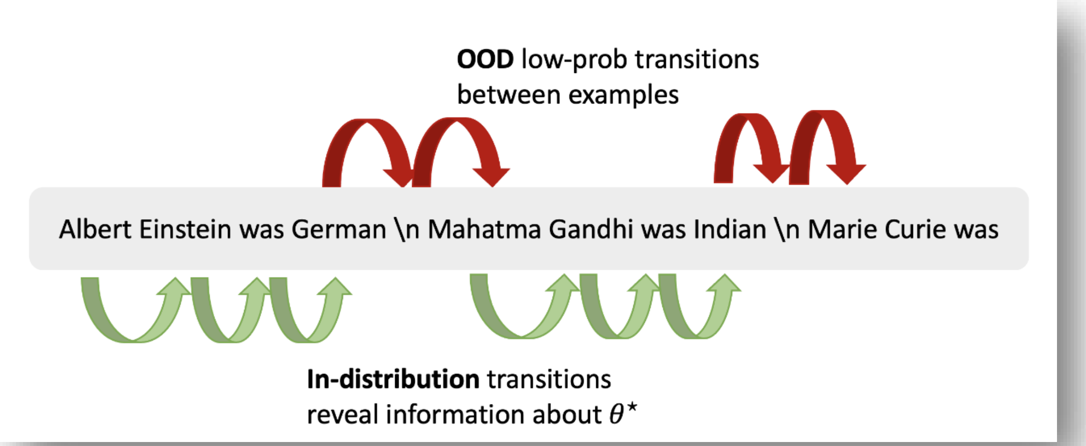
- Prompt distribution.
  - Prompt input: $[S_n, x_{\text{test}}] = [x_1, y_1, o^{\text{delim}}, x_2, y_2, o^{\text{delim}}, \ldots, x_n, y_n, o^{\text{delim}}, x_{\text{test}}] \sim p_{\text{prompt}}$
  - All exemplars $O_i = [x_i, y_i]$ are conditioned on a shared concept θ^*: $p\left(O_i \mid h_i^{\text{start}}, \theta^*\right)$
  - Test example:

$$y_{\text{test}} \sim p_{\text{prompt}}(y \mid x_{\text{test}}) = \mathbb{E}_{h_{\text{test}}^{\text{start}} \sim p_{\text{prompt}}(h_{\text{test}}^{\text{start}} \mid x_{\text{test}})}\left[p\left(y \mid x_{\text{test}}, h_{\text{test}}^{\text{start}}, \theta^*\right)\right]$$

Xie, Sang Michael, et al. “An explanation of in-context learning as implicit bayesian inference.” (ICLR 2022).

A Framework for ICL

- Mismatch between prompt and pretraining distributions:



- In-context predictor through pretraining distribution:

$$f_n(x_{\text{test}}) = \arg\max_y p(y|S_n, x_{\text{test}})$$

$$L_{0-1}(f_n) = \mathbb{E}_{x_{\text{test}}, y_{\text{test}} \sim p_{\text{prompt}}}[\mathbf{1}[f_n(x_{\text{test}}) \neq y_{\text{test}}]]$$

Xie, Sang Michael, et al. “An explanation of in-context learning as implicit bayesian inference.” (ICLR 2022).

High Level Method

- Goal: Show $\arg\max_y p\left(y|S_n, x_{\text{test}}\right) \to \arg\max_y p_{\text{prompt}}\left(y|x_{\text{test}}\right)$ as n grows.
- Expand $p\left(y|S_n, x_{\text{test}}\right)$:

$$p(y|S_n, x_{\text{test}}) \propto \int_\theta \sum_{h_{\text{test}}^{\text{start}} \in \mathcal{H}} p\left(y|x_{\text{test}}, h_{\text{test}}^{\text{start}}, \theta\right) p\left(h_{\text{test}}^{\text{start}}|S_n, x_{\text{test}}, \theta\right) \frac{p(S_n, x_{\text{test}}|\theta)}{p(S_n, x_{\text{test}}|\theta^*)} p(\theta) d\theta$$

- If $\frac{p(S_n, x_{\text{test}}|\theta)}{p(S_n, x_{\text{test}}|\theta^*)} \to 0$ for all concepts θ except the prompt concept θ^*, then the prompt θ^* is “selected” as a consequence of Bayesian inference.

Xie, Sang Michael, et al. “An explanation of in-context learning as implicit bayesian inference.” (ICLR 2022).

Formal Results (Presented Intuitively)

Condition 1 (Distinguishability). *We define θ^* to be distinguishable if for all $\theta \in \Theta, \theta \neq \theta^*$,*

$$\sum_{j=1}^{k} KL_j(\theta^* \| \theta) > \epsilon_{start}^{\theta} + \epsilon_{delim}^{\theta}.$$

Theorem 1. *Assume the assumptions in Section 2.1 hold. If Condition 1 holds, then as $n \to \infty$ the prediction according to the pretraining distribution is*

$$\arg\max_y \; p(y|S_n, x_{test}) \to \arg\max_y \; p_{prompt}(y|x_{test}). \tag{15}$$

Thus, the in-context predictor f_n achieves the optimal 0-1 risk: $\lim_{n\to\infty} L_{0\text{-}1}(f_n) = \inf_f \; L_{0\text{-}1}(f)$.

Xie, Sang Michael, et al. "An explanation of in-context learning as implicit bayesian inference." (ICLR 2022).

Prompts Provide Noisy Evidence for Bayesian Inference

- Context examples provides signal.
  - The input distribution, label distribution and input-output mapping all provide signal for Bayesian inference.

- ICL is robust to some noise.
  - With a strong signal, some forms of noise (e.g., low-prob transitions between examples, removed input-output mapping) could be tolerable.

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

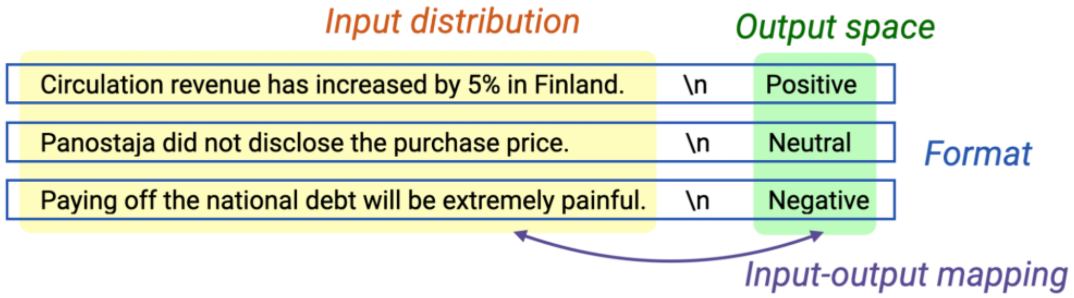
They defeated … in the NFC Championship Game. // Sports

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

Xie, Sang Michael, and Min, Sewon "How does in-context learning work? A framework for understanding the differences from traditional supervised learning" (2022).

Empirical Evidence

- Typical in-context examples consists of 4 components:

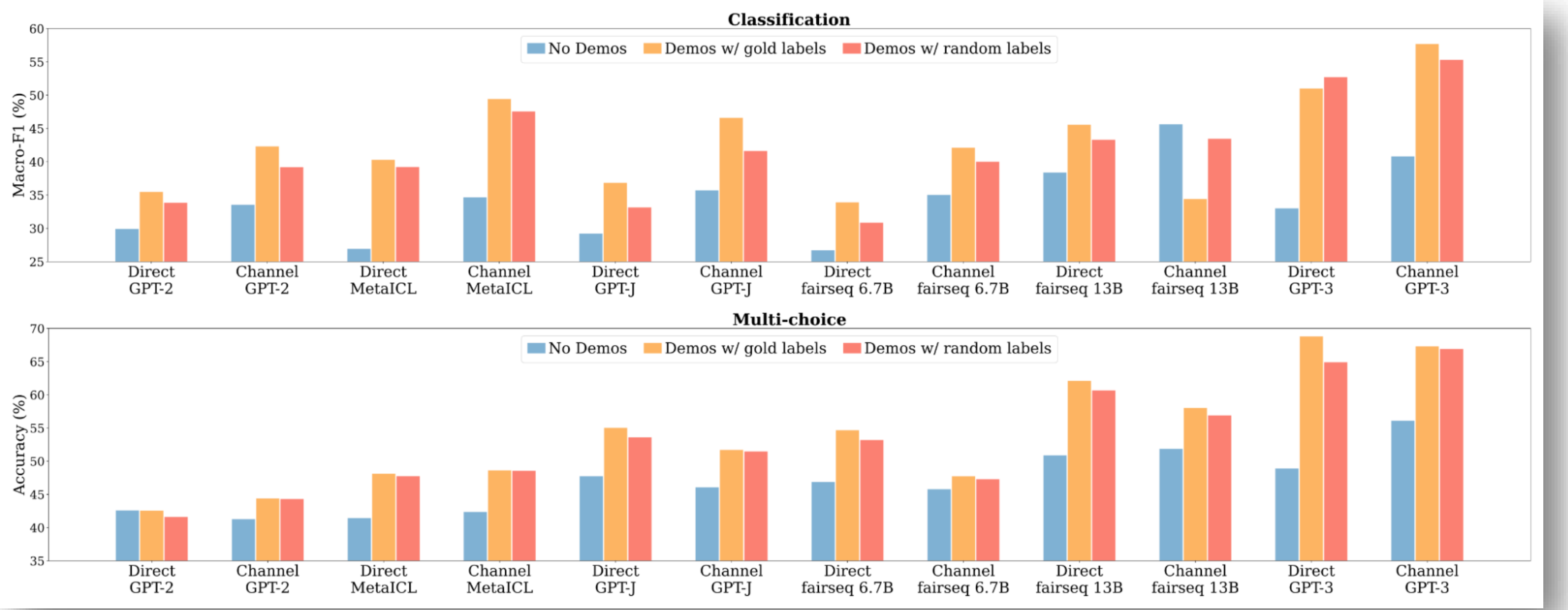


- Examine the role of input-output mapping by:
    - Zero-Shot learning
    - Examples with ground-truth outputs
    - Examples with random outputs

Min, Sewon, et al. "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?." (EMNLP 2022).

Empirical Evidence

- Results of models whose sizes range from 774M to 175B

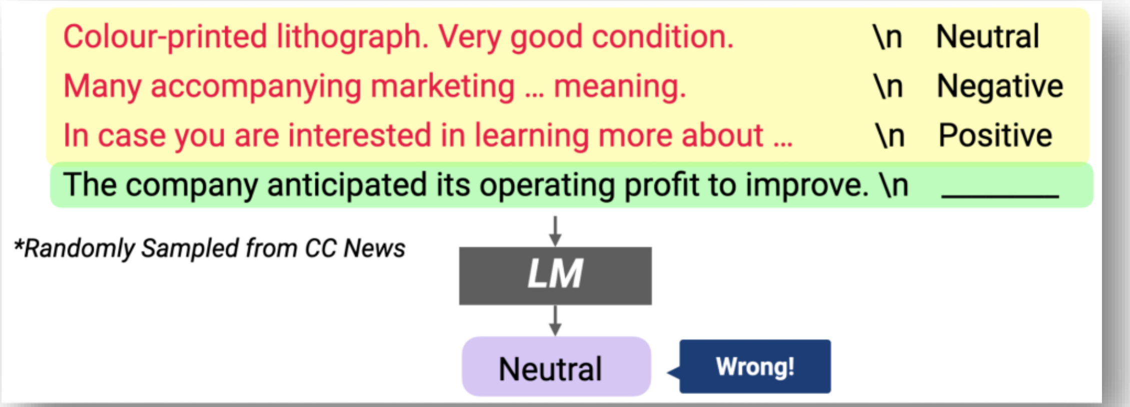


- Correct input-output mapping has a marginal effect on ICL (with implications).

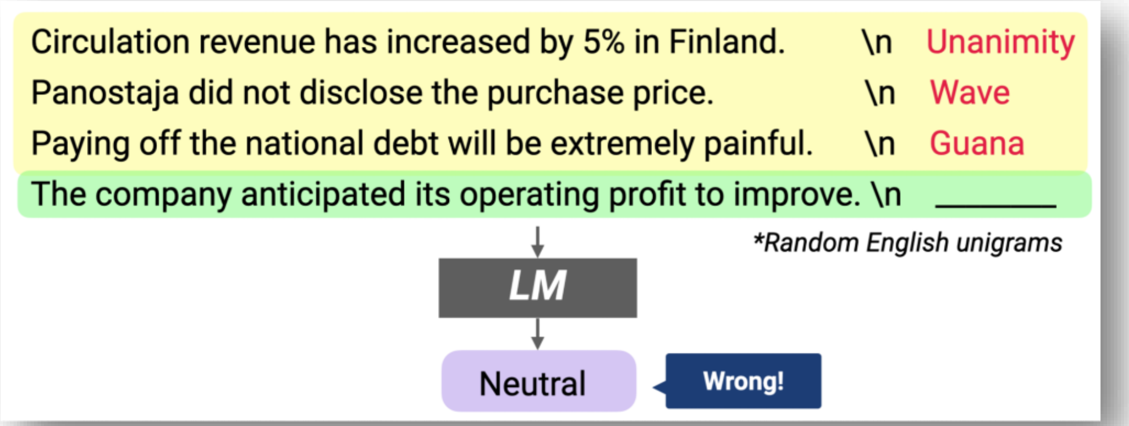
Min, Sewon, et al. "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?." (EMNLP 2022).

Effect of Input and Label Space

- The input distribution and the label space of in-context examples matter.



Replace the prompt input with random inputs from an external corpus

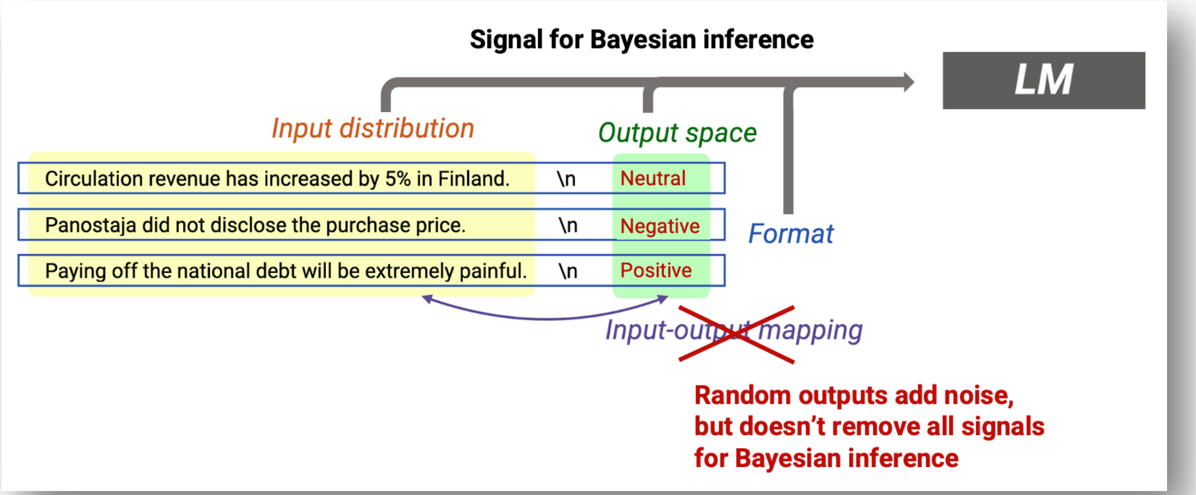


Replace the prompt label with random English unigrams

- Both change can lead to a significant performance drop.

A Framework for ICL

- Pretraining distribution: $p\left(o_1, \ldots, o_T\right)=\int_{\theta \in \Theta} p\left(o_1, \ldots, o_T \mid \theta\right) p(\theta) d \theta$

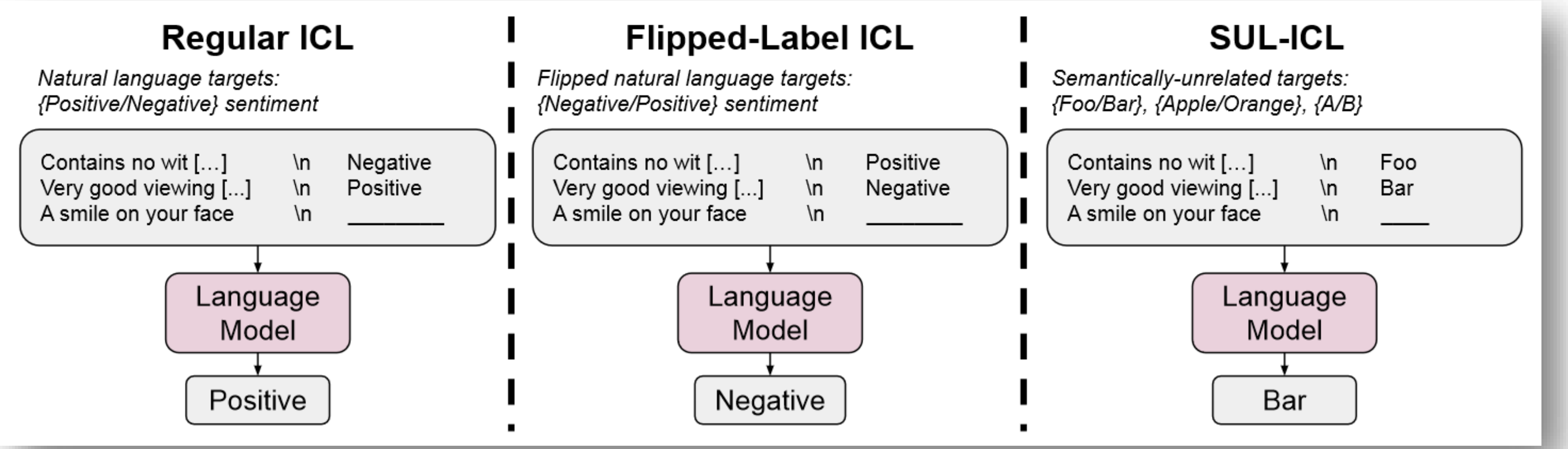


- What if we define a novel concept θ^* for ICL?

Xie, Sang Michael, and Min, Sewon "How does in-context learning work? A framework for understanding the differences from traditional supervised learning" (2022).

Different Story for **Larger** LMs

- To successfully perform ICL, models can
  - Mostly use semantic prior knowledge to predict labels while following the format of in-context examples
  - Learn the input-label mappings from examples (overriding semantic prior or only exploit the input-output mapping).
- Study how semantic priors and input-label mappings interact in several experimental settings.



Wei, Jerry, et al. "Larger language models do in-context learning differently." arXiv preprint arXiv:2303.03846 (2023).

Experiment Setting

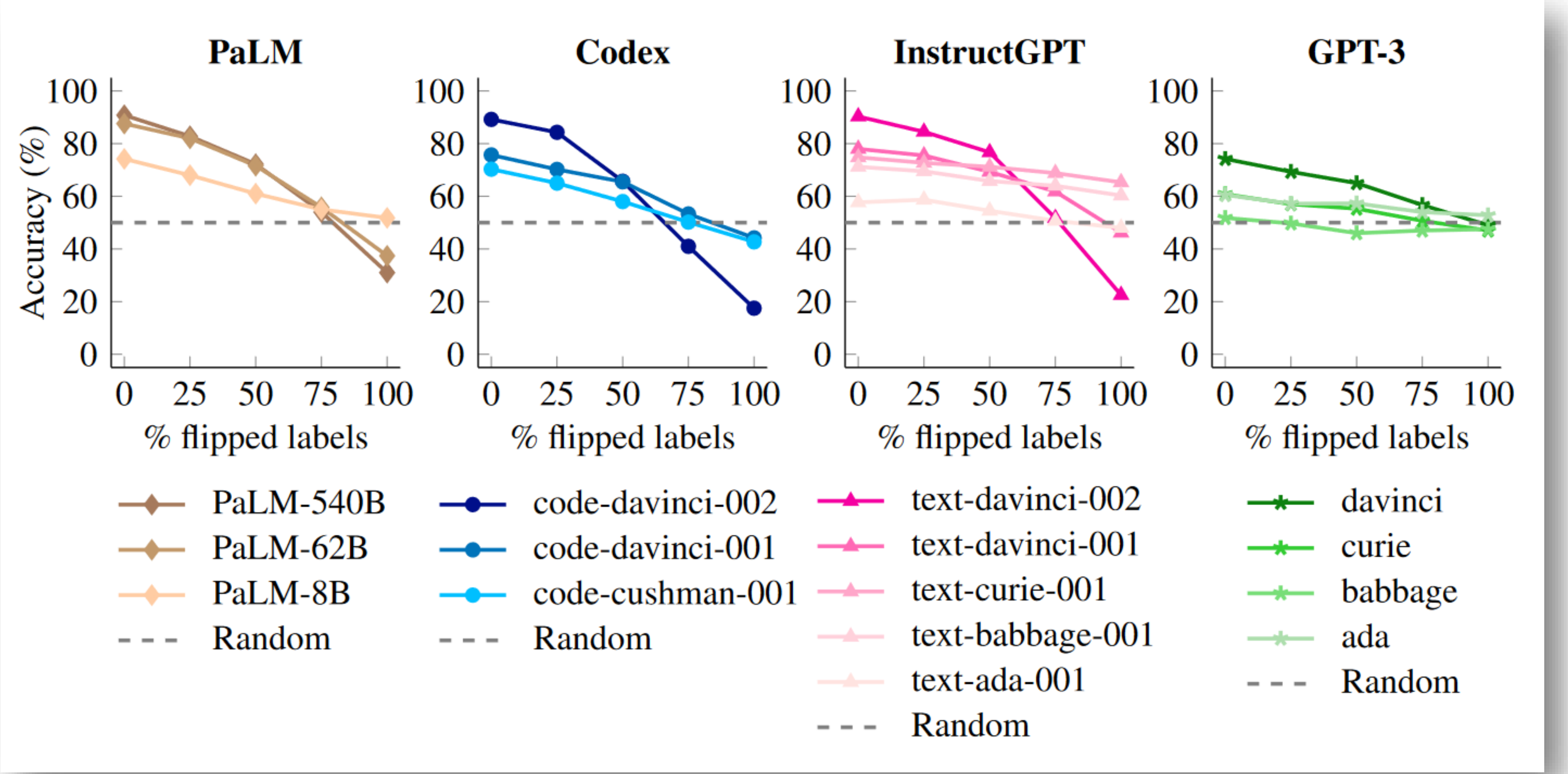
- Tasks: standard NLP classification datasets
- Models: ranging 350M ~ 540B, w/ and w/o instruct tuning.
- Use 16 context examples for each dataset.
- Use 100 randomly sampled evaluation examples per dataset.

Model Family	Model Name (Abbreviation)
GPT-3	ada (a), babbage (b), curie (c), davinci (d)
InstructGPT	text-ada-001 (a-1), text-babbage-001 (b-1), text-curie-001 (c-1), text-davinci-001 (d-1), text-davinci-002 (d-2)
Codex	code-cushman-001 (c-c-1), code-davinci-001 (c-d-1), code-davinci-002 (c-d-2)
PaLM	PaLM-8B, PaLM-62B, PaLM-540B
Flan-PaLM	Flan-PaLM-8B, Flan-PaLM-62B, Flan-PaLM-540B

Table 1: Models used in this paper.

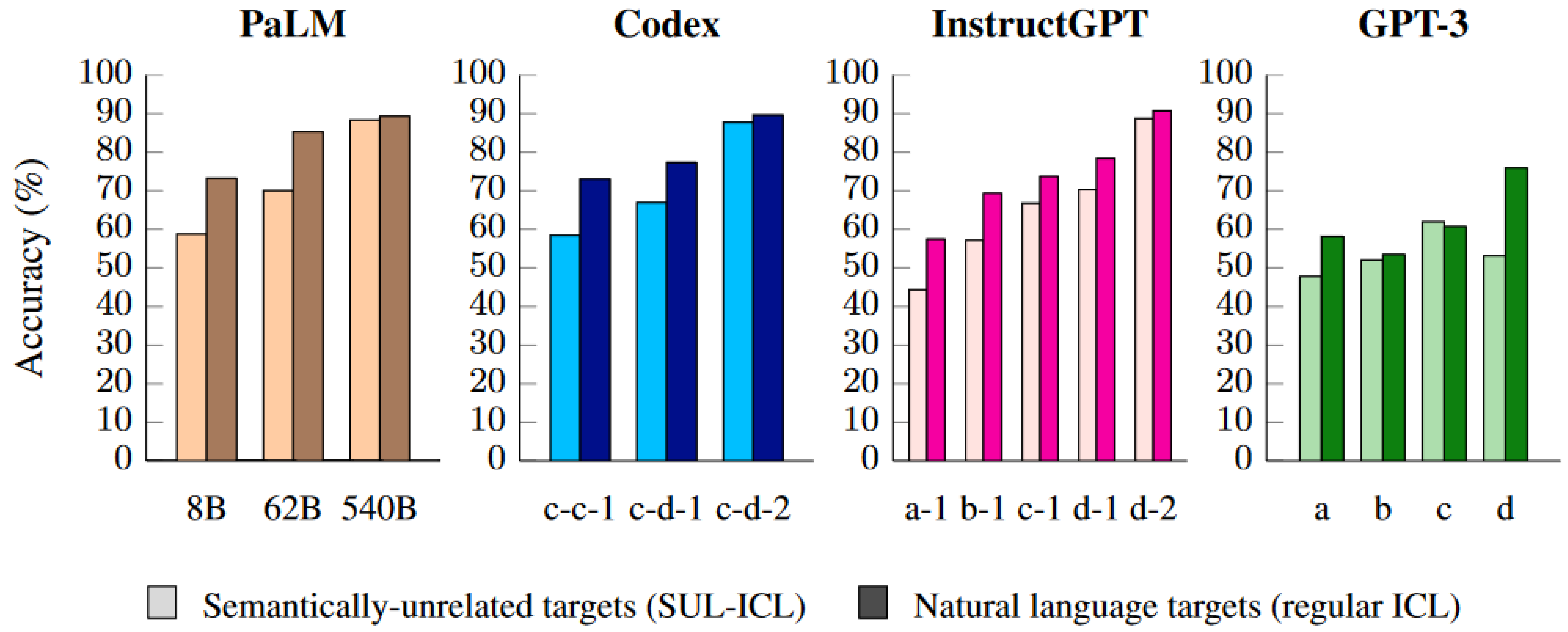
Wei, Jerry, et al. "Larger language models do in-context learning differently." arXiv preprint arXiv:2303.03846 (2023).

Input-Label Mapping Override Semantic Priors in LLMs



Wei, Jerry, et al. "Larger language models do in-context learning differently." arXiv preprint arXiv:2303.03846 (2023).

ICL with Semantically Unrelated Labels Emerges with Scale



Wei, Jerry, et al. "Larger language models do in-context learning differently." arXiv preprint arXiv:2303.03846 (2023).

ICL with Semantically Unrelated Labels Emerges with Scale

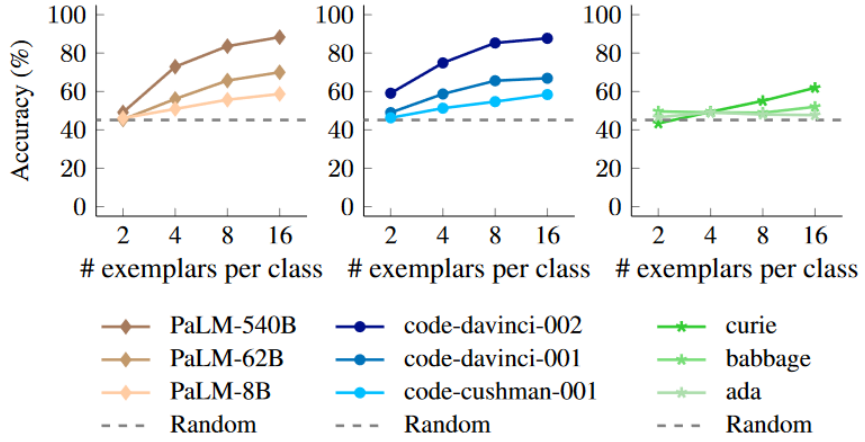


Figure 4: In the SUL-ICL setup, larger models benefit more from additional exemplars than smaller models do. Accuracy is calculated over 100 evaluation examples per dataset and averaged across all datasets. A per-dataset version of this figure is shown in Figure 18 in the Appendix.

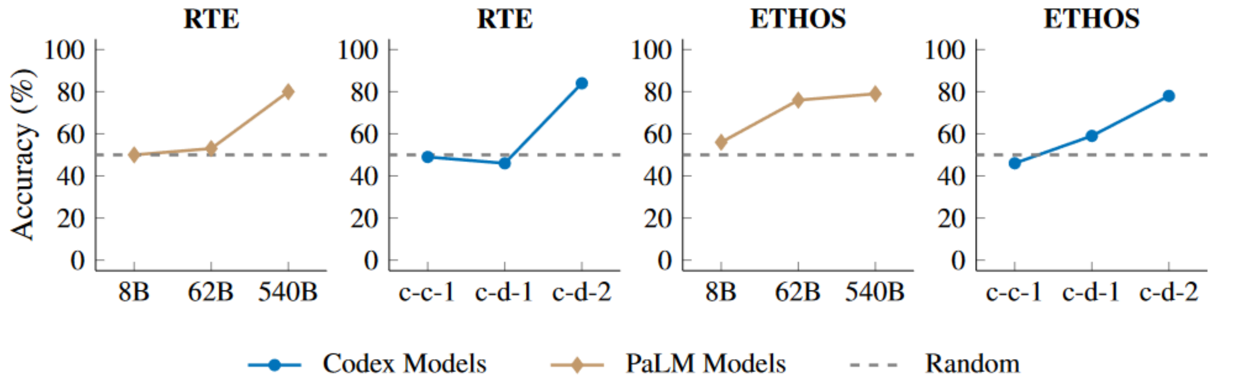
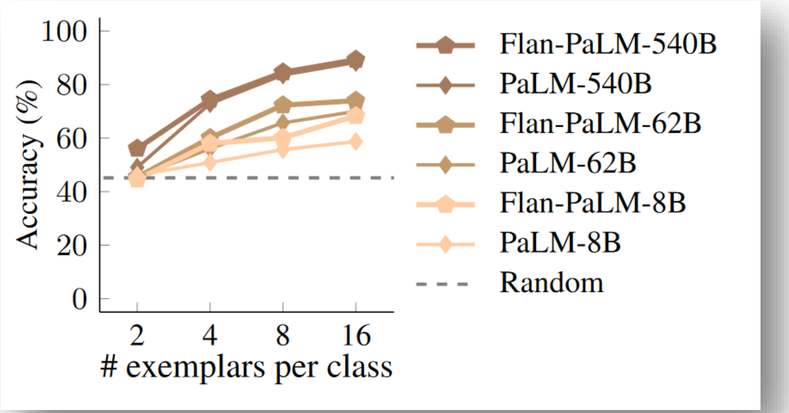


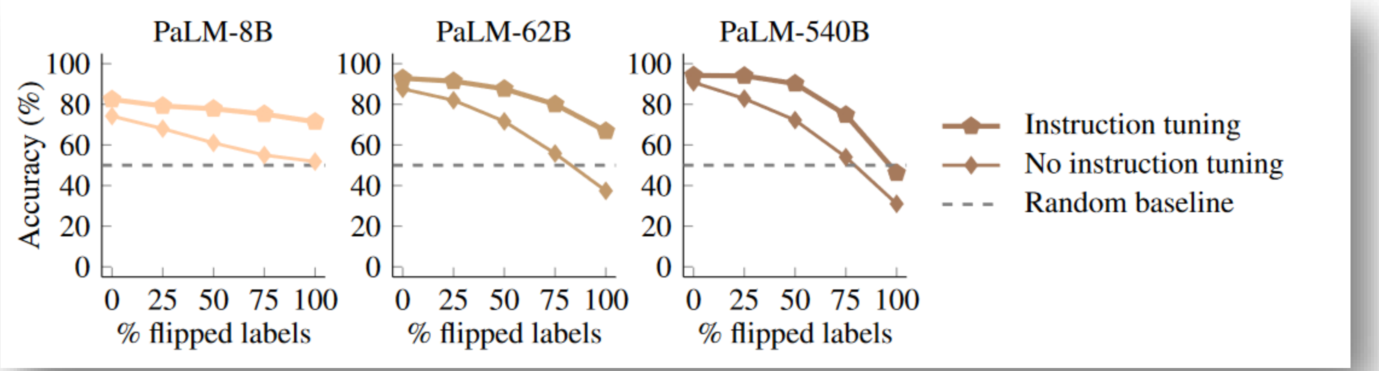
Figure 5: Some tasks in the SUL-ICL setting emerge with scale and can only be successfully performed by large-enough models. These experiments use $k = 8$ in-context exemplars per class. Accuracy is calculated over 100 evaluation examples.

Effect of Instruction Tuning

- Better at learning novel input-output label mapping



- Bad at overriding semantic prior



LLMs can Perform Linear Classification

Input: 59, 874, 536, 60, 824, 223, 555, 809, 727, 448, 20, 482, 523, 928, 331, 182

Output: Bar

Input: 669, 414, 858, 114, 509, 393, 222, 627, 579, 336, 455, 732, 799, 636, 771, 990

Output: Bar

Input: 405, 146, 99, 760, 880, 778, 922, 555, 170, 600, 843, 358, 323, 654, 501, 603

Output: Bar

Input: 839, 45, 729, 900, 235, 605, 973, 304, 558, 479, 645, 77, 345, 768, 927, 734

Output: Bar

Input: 319, 605, 921, 13, 449, 608, 157, 718, 316, 409, 558, 364, 860, 215, 740, 909

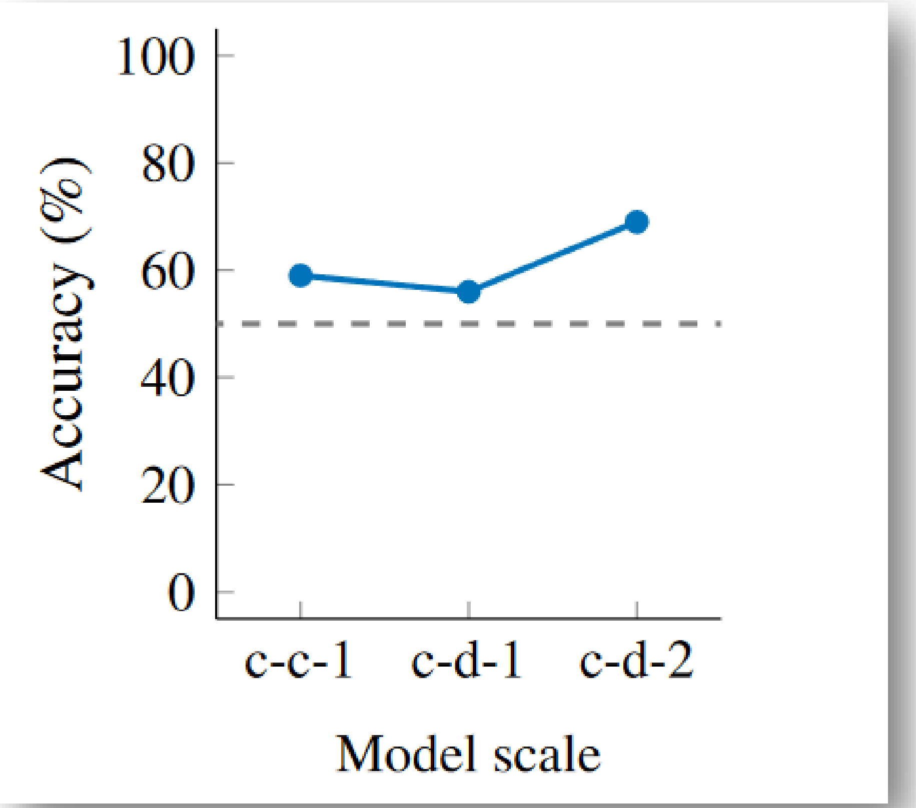
Output: Bar

Input: 101, 969, 495, 149, 394, 964, 428, 946, 542, 814, 240, 467, 435, 987, 297, 466

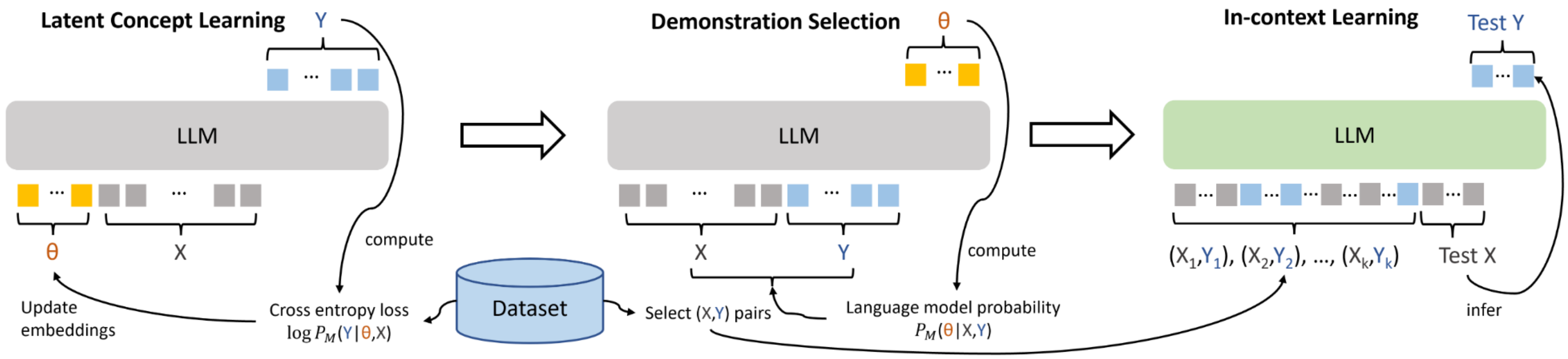
Output:

Answer:

Bar



Inspired Method for In-Context Example Selection



Wang, Xinyi, Wanrong Zhu, and William Yang Wang. "Large Language Models Are Implicitly Topic Models: Explaining and Finding Good Demonstrations for In-Context Learning." arXiv preprint arXiv:2301.11916 (2023).

Summary: ICL in LLM

- In-Context Learning as an emerging ability from LLM:
  - In-Context Learning is an empirical method that enables effective ‘learning’ happens.
  - Understanding how LLM conduct ICL, e.g., **conducting Bayesian inference to ‘locate and extract’ some pre-trained knowledge** or **learning novel tasks from context.**
- Next: In-Context Learning as a learning paradigm.
  - When can ICL happen?
  - What and How transformers learn with ICL.

Data Distribution Properties Drive ICL

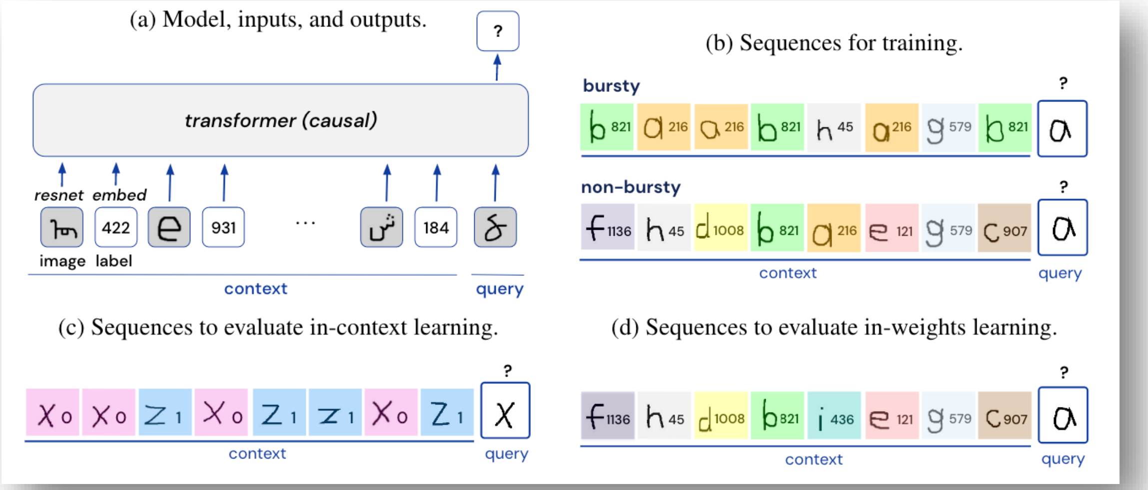
- Explore the possibility that a capacity for ICL depends on the distributional qualities of the training data.

- Properties of natural (language) data:
  - Natural data is temporally "bursty", e.g., a given entity may have a distribution that is **not uniform across time**, instead tending to appear in clusters.
  - Natural data has the property that the marginal distribution across entities is highly skewed, following a Zipfian distribution with **a long tail of infrequent items**.
  - The **semantic** of entities in natural data is often **dynamic** rather than fixed, they should be interpreted using context.

Chan, Stephanie CY, et al. "Data distributional properties drive emergent in-context learning in transformers." (NeurIPS 2022).

Experimental Design

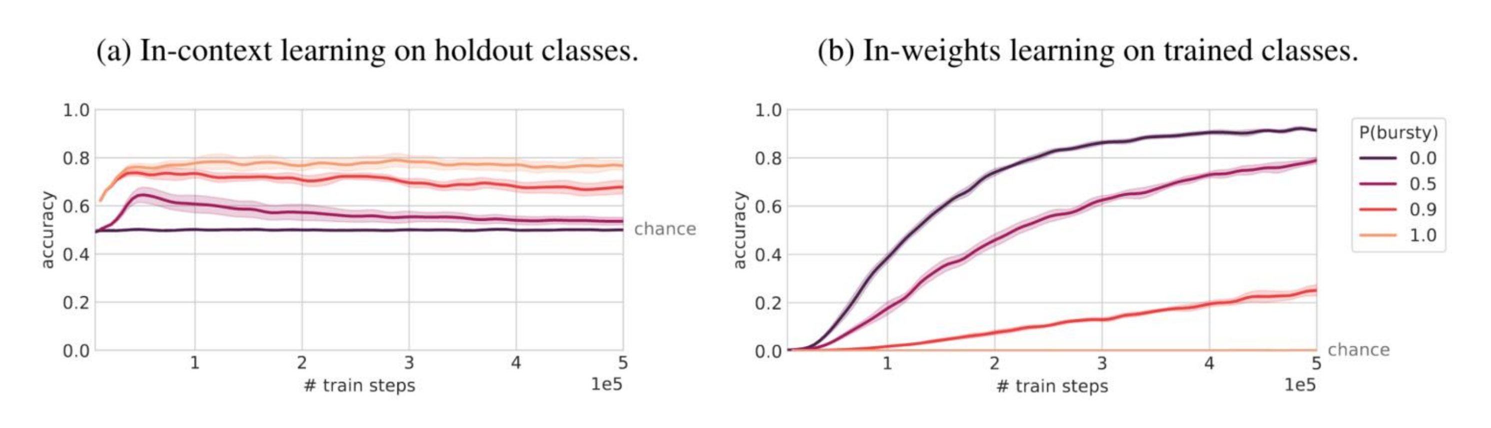
- Training data: Omniglot dataset
  - Consists of 1623 different character classes from various international alphabets.
  - Each class contains 20 handwritten examples.
  - “Bursty”: 2 classes of examples appear 3 times in a sequence.
  - “Non-Bursty”: each class appears uniformly.
- Evaluation data:
  - In-weight: the image classes were forced to be unique within each sequence.
  - In-context: a random ordering of 2 different newly assigned image classes with 4 examples.



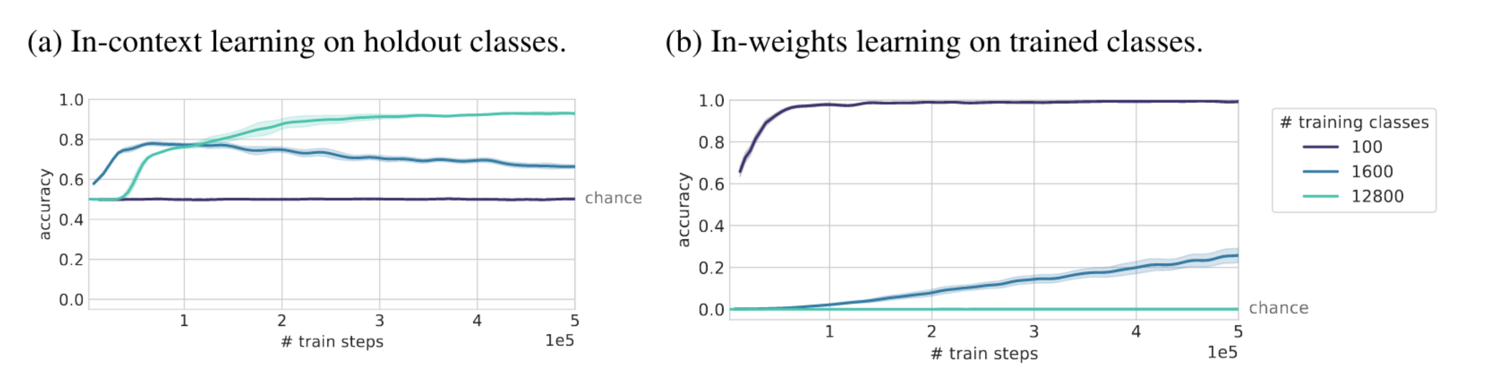
Chan, Stephanie CY, et al. "Data distributional properties drive emergent in-context learning in transformers." (NeurIPS 2022).

What Kinds of Training Data Promote ICL?

- Burstiness



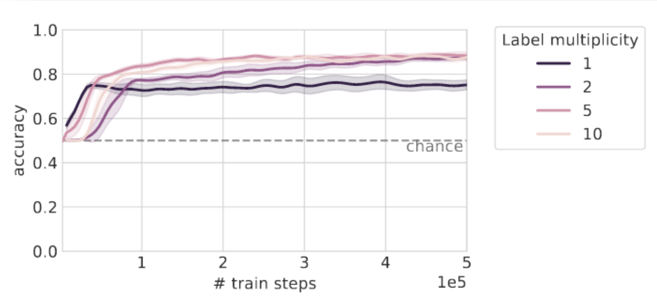
- Infrequent Classes



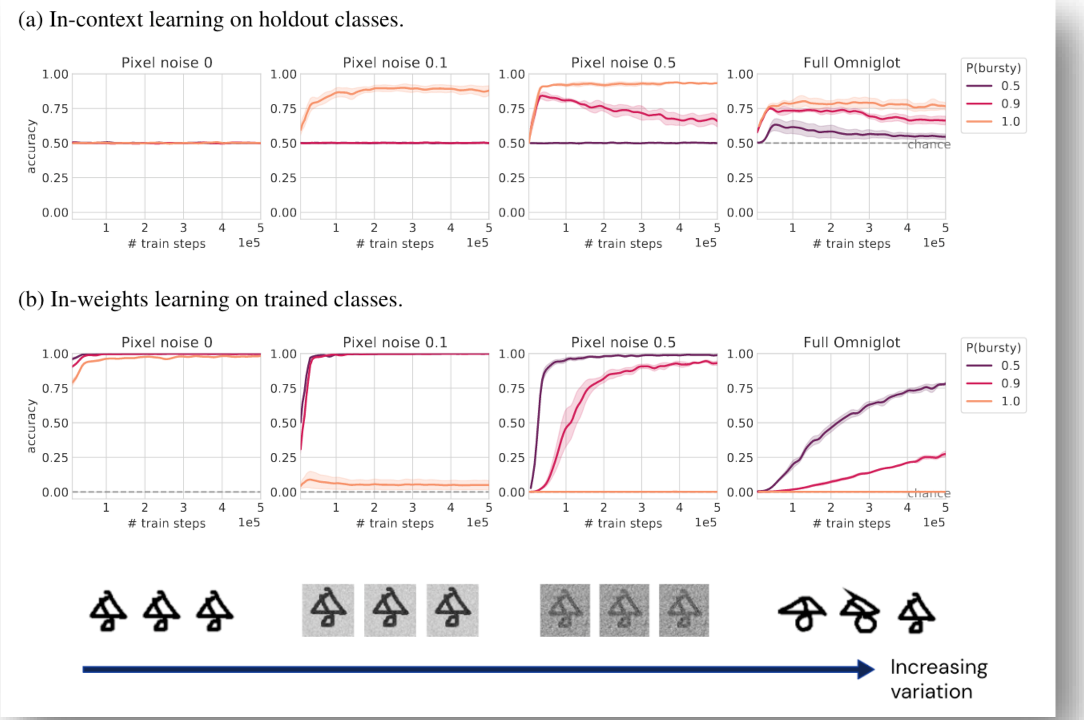
Chan, Stephanie CY, et al. "Data distributional properties drive emergent in-context learning in transformers." (NeurIPS 2022).

What Kinds of Training Data Promote ICL?

- Dynamic label



- Distribution shift

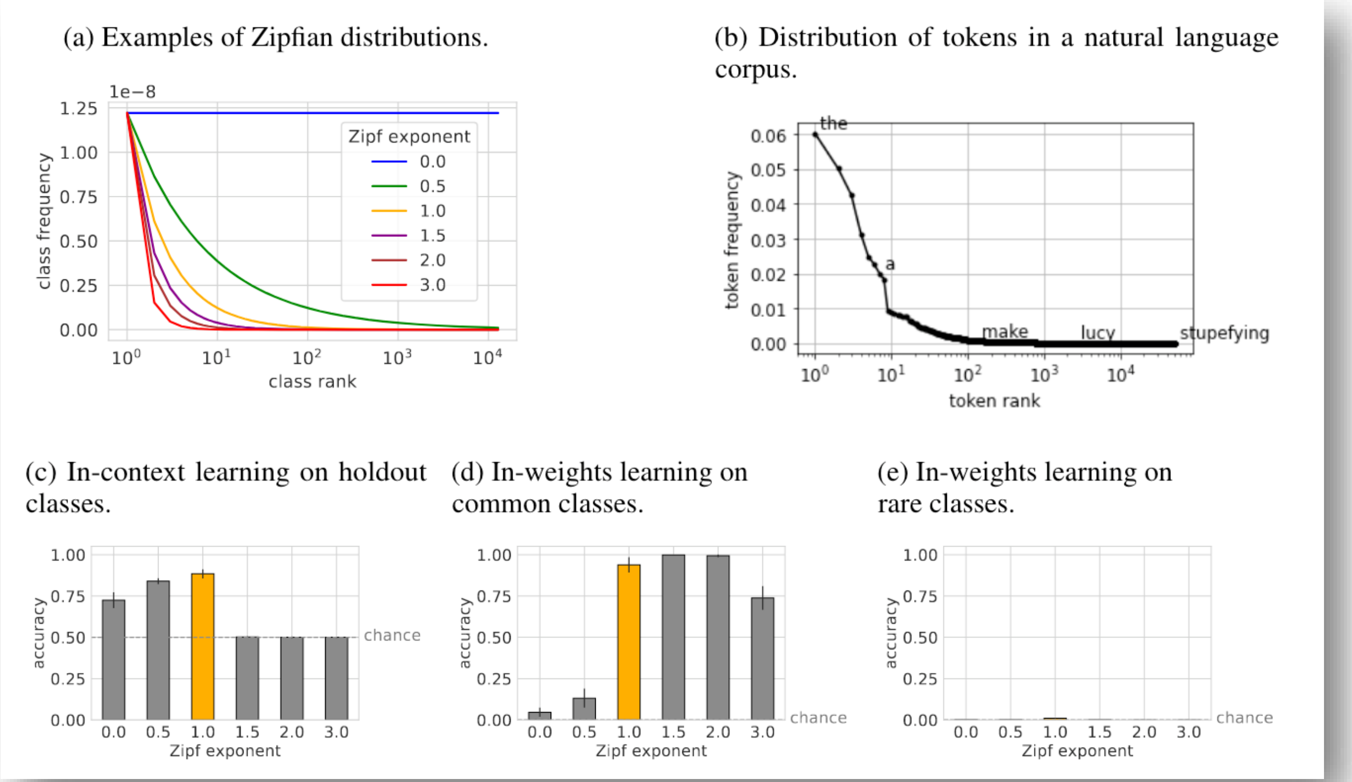


Chan, Stephanie CY, et al. "Data distributional properties drive emergent in-context learning in transformers." (NeurIPS 2022).

Does IWL and ICL Compatible?

- In-weight learning and in-context learning both exist in LLM.
- Many natural phenomena such as word distributions are described as a Zipfian (power law) distribution.

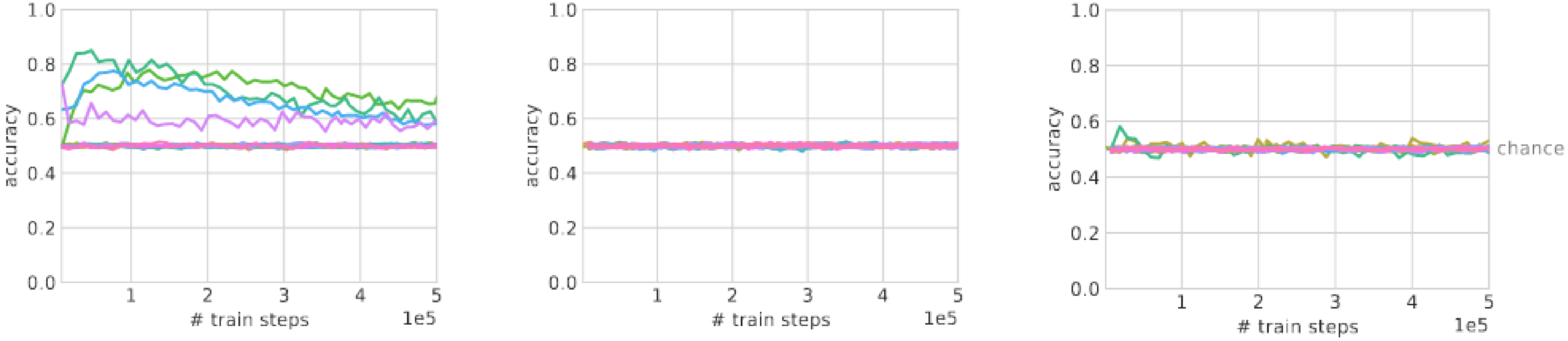
$$p(X = x) \propto \frac{1}{x^{\alpha}}$$



Chan, Stephanie CY, et al. "Data distributional properties drive emergent in-context learning in transformers." (NeurIPS 2022).

Architecture Matters Too

(a) Transformer. (b) Vanilla RNN. (c) LSTM.



Chan, Stephanie CY, et al. "Data distributional properties drive emergent in-context learning in transformers." (NeurIPS 2022).

Brief Review of Transformer Block

- Transformers are seq2seq NNs that map input vectors $\boldsymbol{x} = [x_1, \ldots, x_n]$ to a sequence of output vectors $\boldsymbol{y} = [y_1, \ldots, y_n]$.
- Each block (layer) in a transformer maps a matrix $H^{(l)} = [\boldsymbol{h}_1^l, \ldots \boldsymbol{h}_n^l]$ to $H^{(l+1)}$.
- Computation of typical autoregressive (decoder-only) transformer models:
  - Self-Attention:

$$\begin{aligned}\boldsymbol{b}_j &= \mathrm{softmax}\left(\left(W_j^Q \boldsymbol{h}_i\right)^\top \left(W_j^K H_{:i}\right)\right)\left(W_j^V H_{:i}\right) \\ \boldsymbol{a}_i &= \mathrm{Attention}\left(\boldsymbol{h}_i^{(l)}; W^F, W^Q, W^K, W^V\right) \\ &= W^F\left[\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m\right]\end{aligned}$$

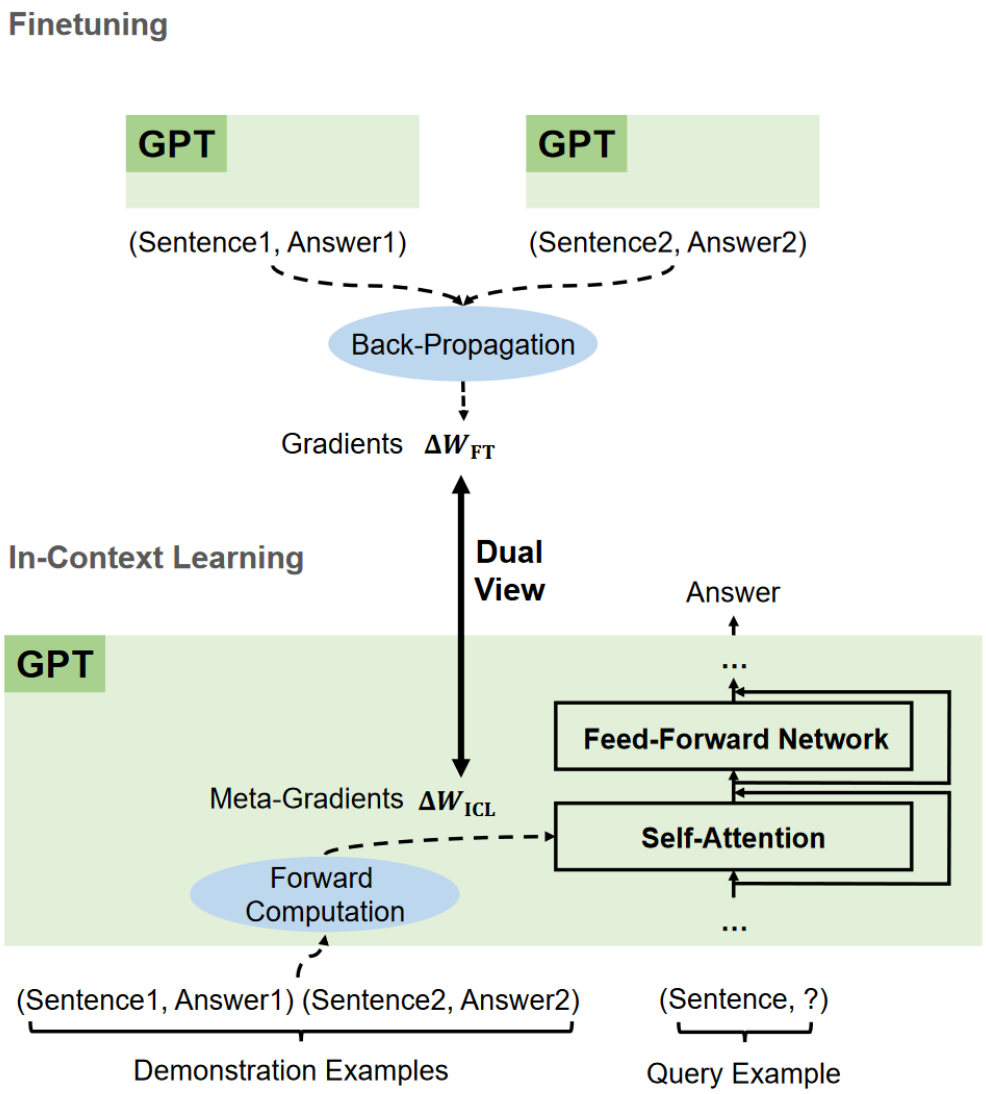
  - Feed-forward transformation:

$$\begin{aligned}\boldsymbol{h}_i^{(l+1)} &= \mathrm{FF}\left(\boldsymbol{a}_i; W_1, W_2\right) \\ &= W_1 \sigma\left(W_2 \lambda\left(\boldsymbol{a}_i + \boldsymbol{h}_i^{(l)}\right)\right) + \boldsymbol{a}_i + \boldsymbol{h}_i^{(l)}\end{aligned}$$

ICL as Implicit Fine-tuning

- The linear layers optimized by GD have a dual form of linear attention.

- ICL and explicit fine-tuning share a dual view of GD based optimization.



Dai, Damai, et al. "Why Can GPT Learn In-Context? Language Models Secretly Perform Gradient Descent as Meta Optimizers." arXiv preprint arXiv:2212.10559 (2022).

Dual Form of a Linear Layer Trained by GD

- Linear layer: $\mathcal{F}(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x}, \boldsymbol{x} \in \mathbb{R}^{d_{in}}, \boldsymbol{W} \in \mathbb{R}^{d_{out}}$
- A Linear layer in a NN trained by GD in some error function $\mathcal{L}$ using η training inputs $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ and corresponding BP error signals $(\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n)$, where $\boldsymbol{e}_i = -\eta_i(\Delta_{\boldsymbol{y}}\mathcal{L})_i$, can be represented by:

$$\begin{aligned}
\mathcal{F}(\boldsymbol{x}) &= (\boldsymbol{W}_0 + \Delta\boldsymbol{W})\,\boldsymbol{x} \\
&= \boldsymbol{W}_0\boldsymbol{x} + \Delta\boldsymbol{W}\boldsymbol{x} \\
&= \boldsymbol{W}_0\boldsymbol{x} + \sum_i \left(\boldsymbol{e}_i \otimes \boldsymbol{x}_i'^T\right)\boldsymbol{x} \\
&= \boldsymbol{W}_0\boldsymbol{x} + \sum_i \boldsymbol{e}_i \left(\boldsymbol{x}_i'^T\boldsymbol{x}\right) \\
&= \boldsymbol{W}_0\boldsymbol{x} + \mathrm{LinearAttn}\left(E, X', \boldsymbol{x}\right)
\end{aligned}$$

Irie, Kazuki, Róbert Csordás, and Jürgen Schmidhuber. "The dual form of neural networks revisited: Connecting test time predictions to training patterns via spotlights of attention." (ICML 2022).

Attention Computation in the ICL setting

- In the ICL setting, let $[X'; X]$ be the demonstrations, $\boldsymbol{x} \in \mathbb{R}^d$ be the input representation of a query token, and $\boldsymbol{q} = W^Q \boldsymbol{x}$ be the attention query vector.
- The attention result of a head is formulated as:

$$\begin{aligned}\mathcal{F}_{\mathrm{ICL}}(\mathbf{q}) &= \mathrm{Attn}(V, K, \mathbf{q}) \\ &= W^V [X'; X] \operatorname{softmax}\left(\frac{\left(W^K [X'; X]\right)^T \mathbf{q}}{\sqrt{d}}\right)\end{aligned}$$

- Approximate the standard linear attention by linear attention:

$$\begin{aligned}\mathcal{F}_{\mathrm{ICL}}(\mathbf{q}) &= \mathrm{Attn}(V, K, \mathbf{q}) \\ &\approx W^V [X'; X] \left(W^K [X'; X]\right)^T \mathbf{q} \\ &= W^V X \left(W^K X\right)^T \mathbf{q} + W^V X' \left(W^K X'\right)^T \mathbf{q} \\ &= \widetilde{\mathcal{F}}_{\mathrm{ICL}}(\mathbf{q}).\end{aligned}$$

Dai, Damai, et al. "Why Can GPT Learn In-Context? Language Models Secretly Perform Gradient Descent as Meta Optimizers." arXiv preprint arXiv:2212.10559 (2022).

Dual Form of the Transformer Attention

- Define $W_{\mathrm{ZSL}} = W^V X \left(W^K X\right)^T$, then:

$$
\begin{aligned}
\widetilde{\mathcal{F}}_{\mathrm{ICL}}(\mathbf{q}) &= W_{\mathrm{ZSL}}\mathbf{q} + W^V X' \left(W^K X'\right)^T \mathbf{q} \\
&= W_{\mathrm{ZSL}}\mathbf{q} + \mathrm{LinearAttn}\left(W^V X', W^K X', \mathbf{q}\right) \\
&= W_{\mathrm{ZSL}}\mathbf{q} + \sum_i W^V \mathbf{x}'_i \left(\left(W^K \mathbf{x}'_i\right)^T \mathbf{q}\right) \\
&= W_{\mathrm{ZSL}}\mathbf{q} + \sum_i \left(W^V \mathbf{x}'_i \otimes \left(W^K \mathbf{x}'_i\right)^T\right) \mathbf{q} \\
&= W_{\mathrm{ZSL}_{\mathbf{q}}} + \Delta W_{\mathrm{ICL}_{\mathbf{q}}} \\
&= \left(W_{\mathrm{ZSL}} + \Delta W_{\mathrm{ICL}}\right) \mathbf{q}.
\end{aligned}
$$

which is similar to fine-tuning:

$$
\begin{aligned}
\tilde{\mathcal{F}}_{\mathrm{FT}}(\mathbf{q}) &= \left(W^V + \Delta W^V\right) X X^T \left(W^K + \Delta W^K\right)^T \mathbf{q} \\
&= \left(W_{\mathrm{ZSL}} + \Delta W_{\mathrm{FT}}\right) \mathbf{q}
\end{aligned}
$$

Dai, Damai, et al. "Why Can GPT Learn In-Context? Language Models Secretly Perform Gradient Descent as Meta Optimizers." arXiv preprint arXiv:2212.10559 (2022).

Experimental Results

Model	Task	Rec2FTP	SimAOU	Random SimAOU	SimAM	ZSL SimAM
GPT 1.3B	CB	91.67	0.189	0.004	0.386	0.152
	SST2	86.32	0.128	0.003	0.608	0.555
	SST5	70.16	0.173	0.004	0.430	0.391
	Subj	84.39	0.070	0.004	0.504	0.378
	MR	92.14	0.188	0.003	0.513	0.398
	AGNews	85.41	0.155	0.003	0.536	0.152
GPT 2.7B	CB	100.00	0.184	-0.001	0.362	0.228
	SST2	93.87	0.113	0.003	0.687	0.687
	SST5	74.32	0.142	0.001	0.411	0.380
	Subj	90.46	0.100	0.004	0.375	0.346
	MR	95.44	0.120	0.001	0.346	0.314
	AGNews	87.48	0.210	-0.003	0.305	0.172

Table 3: Rec2FTP, SimAOU, and SimAM scores on six classification datasets. The demonstrated SimAOU and SimAM scores are averaged across examples and layers. For comparison, we also show two baseline metrics for SimAOU and SimAM, respectively. On all of these datasets, ICL tends to perform similar behavior to finetuning at the prediction, representation, and attention behavior levels.

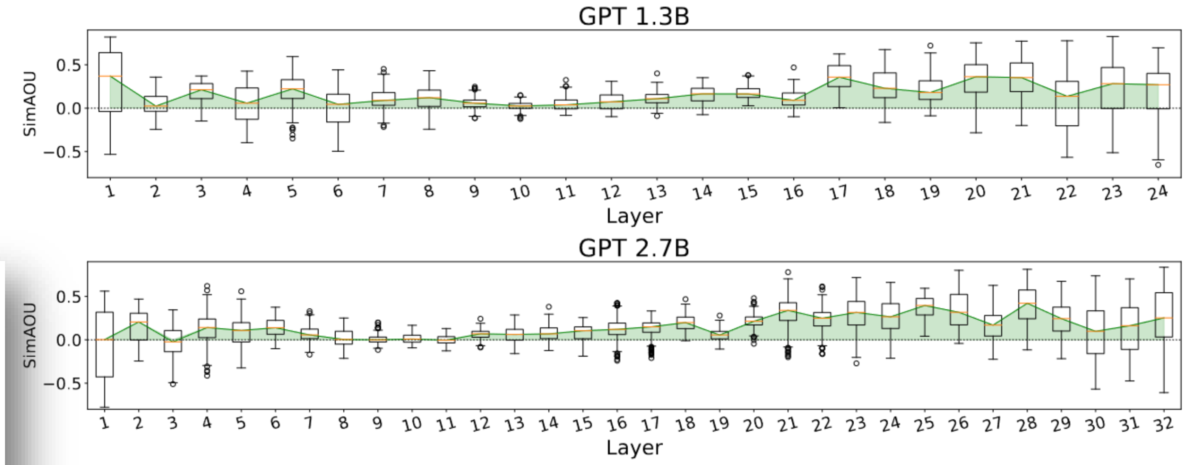


Figure 2: Statistics of the SimAOU scores at different layers. The yellow lines denote medians.

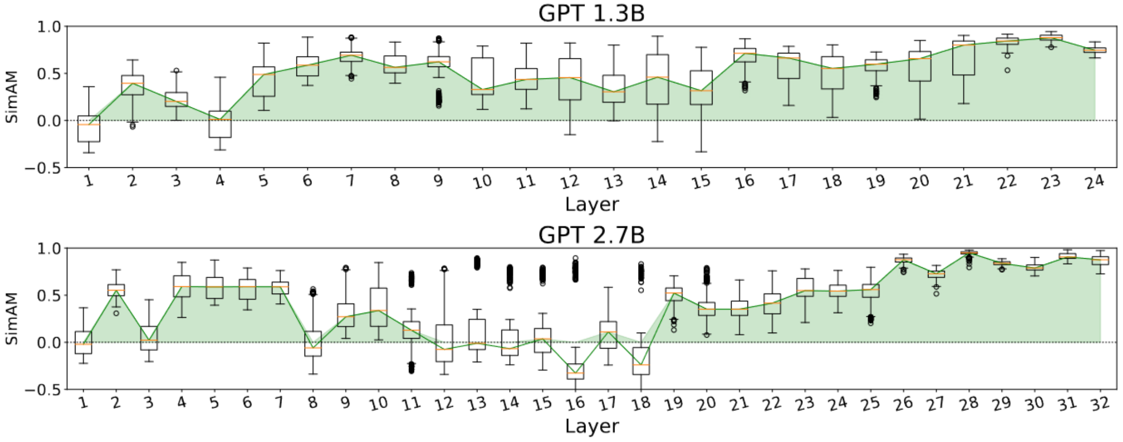


Figure 3: Statistics of the SimAM scores at different layers. The yellow lines denote medians.

Dai, Damai, et al. "Why Can GPT Learn In-Context? Language Models Secretly Perform Gradient Descent as Meta Optimizers." arXiv preprint arXiv:2212.10559 (2022).

What can Transformers Learn In-Context?

- It is unclear to what extent transformers have **developed the ability to learn new tasks from in-context examples alone** as opposed to **simply indexing into a vast set of known tasks from the training data.**
- Consider a well-defined problem to learning a *function class* from in-context examples:
  - Let $D_{\mathcal{X}}$ be a distribution over inputs and $D_{\mathcal{F}}$ be a distribution over functions in $\mathcal{F}$.
  - A prompt P is a sequence $\left(x_1, f\left(x_1\right), \ldots, x_k, f\left(x_k\right), x_{\text{query}}\right)$
  - A model M can in-context learn the function class $\mathcal{F}$ up to ϵ , w.r.t. $\left(D_{\mathcal{F}}, D_{\mathcal{X}}\right)$, if

$$\mathbb{E}_P\left[\ell\left(M(P), f\left(x_{\text{query}}\right)\right)\right] \leq \epsilon$$

- Can we train a model to in-context learn a certain function class?

Garg, Shivam, et al. "What can transformers learn in-context? a case study of simple function classes." (NeurIPS 2022).

Training Transformers for ICL

- Constructing prompts $P = \left(x_1, f\left(x_1\right), \ldots, x_{k+1}, f\left(x_{k+1}\right)\right)$
  - Denote $P^i = \left(x_1, f\left(x_1\right), x_2, f\left(x_2\right), \ldots, x_i, f\left(x_i\right), x_{i+1}\right)$
  - In the case of linear functions, $f(x) = w^\top x$, $w, x \sim \mathcal{N}(0, I_d)$
- Training objective:

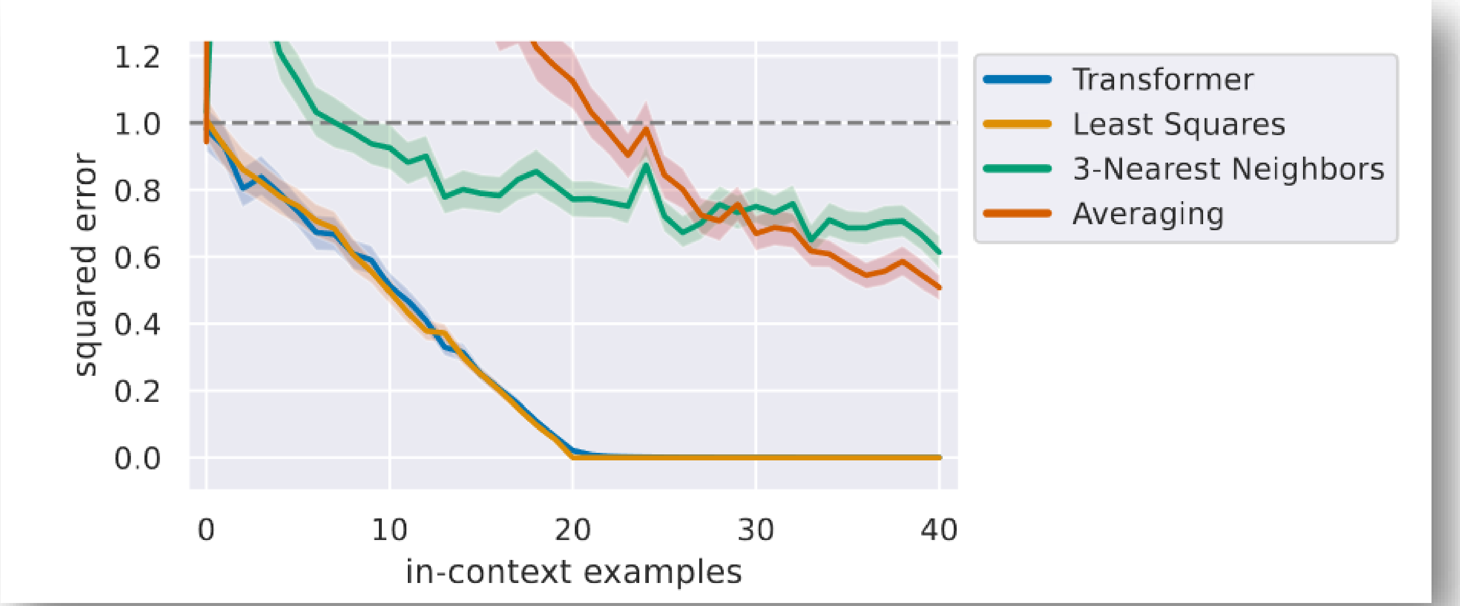
$$\min_\theta \mathbb{E}_P \left[\frac{1}{k+1} \sum_{i=0}^{k} \ell\left(M_\theta\left(P^i\right), f\left(x_{i+1}\right)\right) \right]$$

- Model structure: 12 layers, 8 attention heads, 256-dim hidden space decoder-only transformer (22.4M parameters).
- In this work, **training is done from scratch**, instead of fine-tune a pre-trained LM.

Garg, Shivam, et al. "What can transformers learn in-context? a case study of simple function classes." (NeurIPS 2022).

ICL of linear functions

- Dim $d = 20$.
- Baselines:
  - Least squares estimator
  - n-Nearest Neighbors
  - Averaging: $\hat{w} = \sum_i y_i x_i / k$

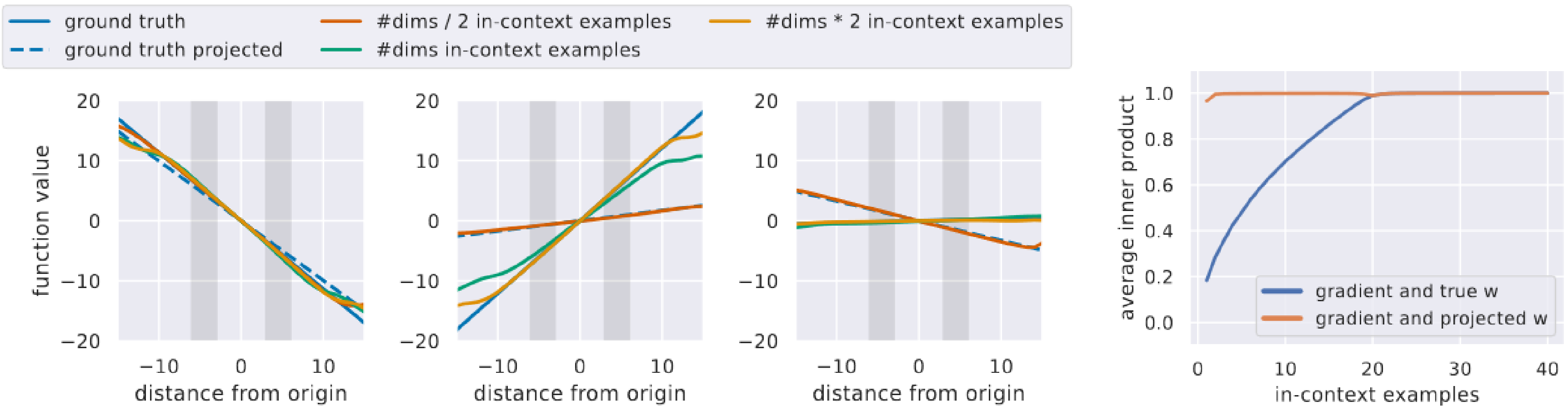


- Memorization can't explain model performance:
  - The inputs alone lie in a 800-dim space when predicting with 2d in-context examples.
  - The best weight vector in the training set can not achieve such test error.

Garg, Shivam, et al. "What can transformers learn in-context? a case study of simple function classes." (NeurIPS 2022).

What Function is the Model Learning In-Context

- Consider the case of $k < d$ (fewer in-context examples). The ideal model should approximate the projection of true w onto the subspace spanned by $x_1, \ldots, x_k$.

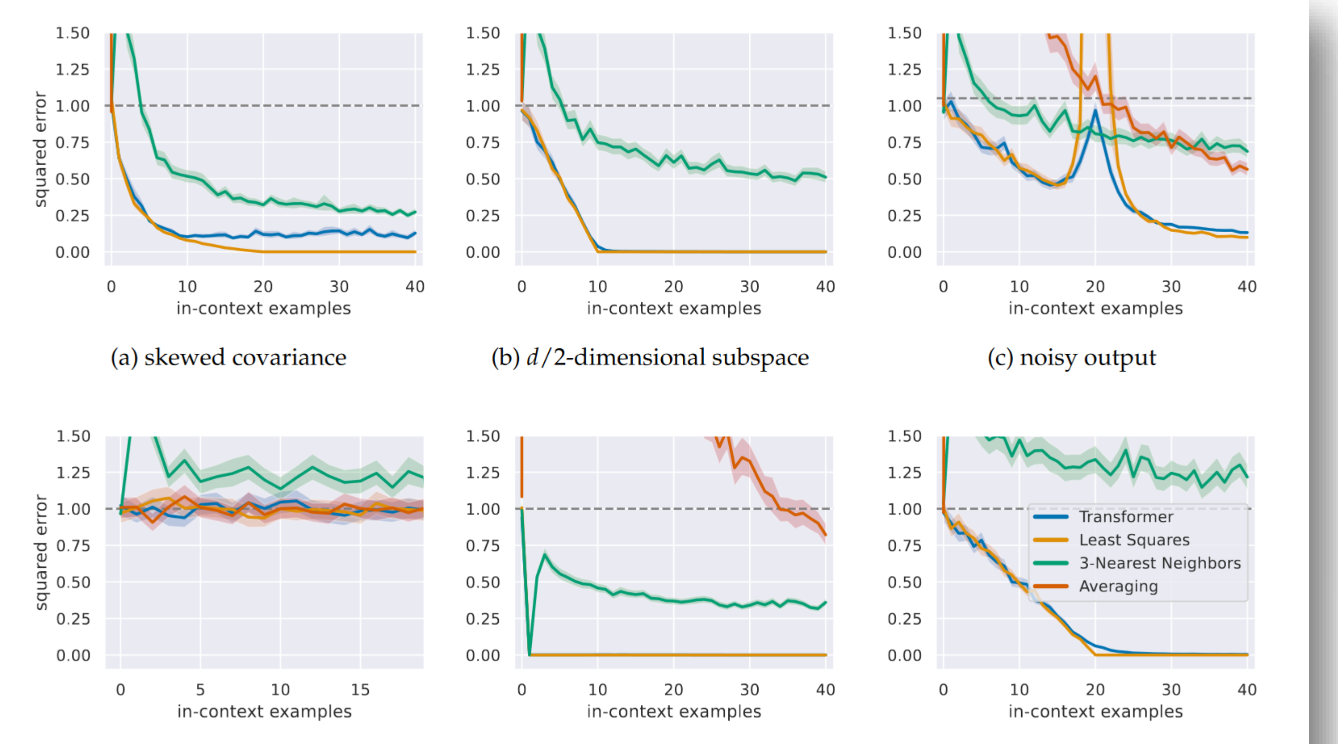


(a) function visualizations (b) gradients

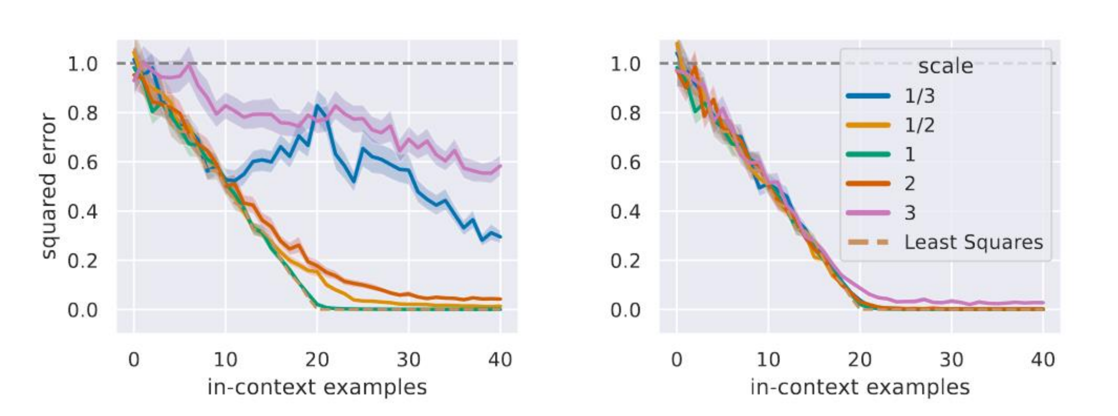
$\hat{f}_{w,x_{1:k}}(\lambda x)$

Garg, Shivam, et al. "What can transformers learn in-context? a case study of simple function classes." (NeurIPS 2022).

Extrapolating Beyond the Training Distribution



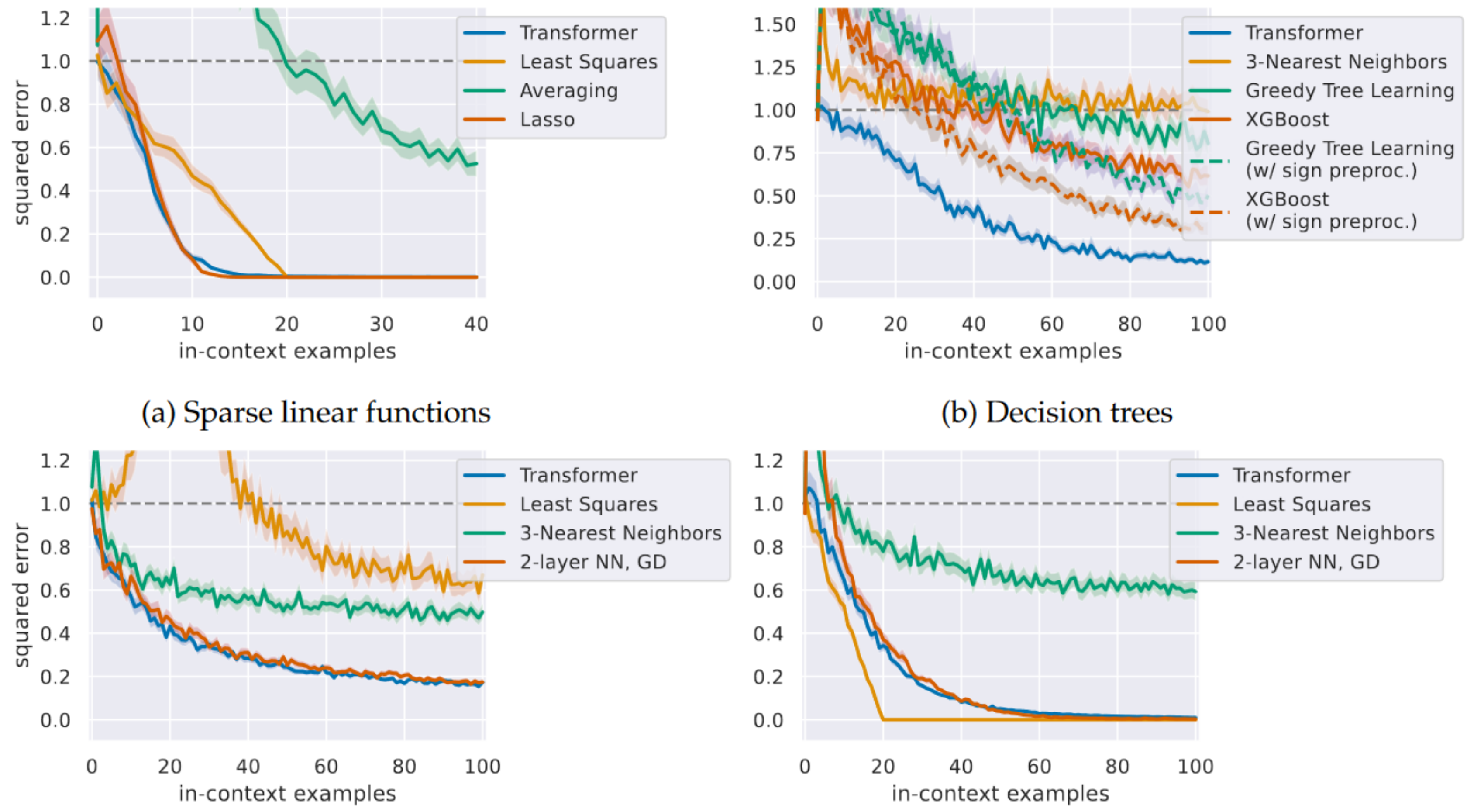
(a) skewed covariance
(b) $d/2$-dimensional subspace
(c) noisy output
(d) orthogonal query
(e) query matches in-context example
(f) different orthants



(a) scaled x, Transformer
(b) scaled w, Transfomer

Garg, Shivam, et al. "What can transformers learn in-context? a case study of simple function classes." (NeurIPS 2022).

More Complex Function Classes



(a) Sparse linear functions

(b) Decision trees

(c) 2-layer NN

(d) 2-layer NN, eval on linear functions

Garg, Shivam, et al. "What can transformers learn in-context? a case study of simple function classes." (NeurIPS 2022).

What Learning Algorithm is in ICL?

- Instead of understanding **what functions ICL can learn**, this work focuses on **how it learns these functions**.
- Theoretically, this paper proves by construction that, for d-dim regression problems, a transformer with $\mathcal{O}(d)$ hidden size and constant depth can **implement a single step of GD**; with $\mathcal{O}(d^2)$ hidden size and constant depth, a transformer can **update a ridge regression solution**.
- Empirically, this paper shows that how ICL-based models are matched by existing predictors.
- Some ICL appears to involve familiar algorithms, discovered and implemented by transformers from sequence modeling task alone.

Akyürek, Ekin, et al. "What learning algorithm is in-context learning? investigations with linear models." (ICLR 2023).

What can Transformers Do by Construction

- Consider some functions from $\mathbb{R}^{H\times T} \to \mathbb{R}^{H\times T}$:
  - $\mathbf{mov}\,(H; s, t, i, j, i', j')$: selects the entries of the s^{th} column of H between rows i and j, and copies them into the t^{th} column of H between rows i' and j', yielding the matrix:

$$\begin{bmatrix} & H_{:i-1,t} & | \\ H_{:,:t} & H_{i':j',s} & H_{:,t+1:} \\ | & H_{j,t} & | \end{bmatrix}$$

  - $\mathbf{mul}\,(H; a, b, c, (i, j), (i', j'), (i'', j''))$: $[\boldsymbol{h}_{:i''-1}, A_1 A_2, \boldsymbol{h}_{j'':}]^\top$
  - $\mathbf{div}\,(H; (i, j), i', (i'', j''))$: $[\boldsymbol{h}_{:i''-1}, \boldsymbol{h}_{i:j} / |\boldsymbol{h}_{i'}|, \boldsymbol{h}_{j'':}]^\top$
  - $\mathbf{aff}\,(H; (i, j), (i', j'), (i'', j''), W_1, W_2, b)$: $[\boldsymbol{h}_{:i''-1}, W_1 \boldsymbol{h}_{i:j} + W_2 \boldsymbol{h}_{i':j'} + b, \boldsymbol{h}_{j'':}]^\top$

Lemma 1. *Each of* `mov`, `mul`, `div` *and* `aff` *can be implemented by a single transformer decoder layer: in Eq.* (1) *and Eq.* (4), *there exist matrices* W^Q, W^K, W^V, W^F, W_1 *and* W_2 *such that, given a matrix* H *as input, the layer's output has the form of the corresponding function output above.* [1]

Akyürek, Ekin, et al. "What learning algorithm is in-context learning? investigations with linear models." (ICLR 2023).

Example: One-Step GD

The operations for 1-step SGD with single exemplar can be expressed as following chain (please see proofs for the Transformer implementation of these operations (Lemma 1) in Appendix C):

- $\texttt{mov}(;1,0,(1,1+d),(1,1+d))$ (move $\boldsymbol{x}$)
- $\texttt{aff}(;(1,1+d),(),(1+d,2+d),W_1=\boldsymbol{w})$ $(\boldsymbol{w}^\top\boldsymbol{x})$
- $\texttt{aff}(;(1+d,2+d),(0,1),(2+d,3+d),W_1=I,W_2=-I)$ $(\boldsymbol{w}^\top\boldsymbol{x}-y)$
- $\texttt{mul}(;d,1,1,(1,1+d),(2+d,3+d),(3+d,3+2d))$ $(\boldsymbol{x}(\boldsymbol{w}^\top\boldsymbol{x}-y))$
- $\texttt{aff}(;(),(),(3+2d,3+3d),b=\boldsymbol{w},)$ (write $\boldsymbol{w}$)
- $\texttt{aff}(;(3+d,3+2d),(3+2d,3+3d),(3+3d,3+4d),W_1=I,W_2=-\lambda)$ $(\boldsymbol{x}(\boldsymbol{w}^\top\boldsymbol{x}-y)-\lambda\boldsymbol{w})$
- $\texttt{aff}(;(3+2d,3+3d),(3+3d,3+4d),(3+2d,3+3d),W_1=I,W_2=-2\alpha,)$ $(\boldsymbol{w}')$
- $\texttt{mov}(;2,1,(3+2d,3+3d),(3+2d,3+3d))$ (move $\boldsymbol{w}'$)
- $\texttt{mul}(;1,d,1,(3+2d,3+3d),(1,1+d),(3+3d,4+3d))$ $(\boldsymbol{w}'^\top x_2)$

This will map:

$$\begin{bmatrix} 0 & y_1 & 0 \\ x_1 & 0 & x_2 \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \\ & & \end{bmatrix} \mapsto \begin{bmatrix} 0 & y_1 & 0 \\ x_1 & x_1 & x_2 \\ w^\top x_1 & w^\top x_1 & w^\top x_2 \\ w^\top x_1 & w^\top x_1 - y & w^\top x_2 \\ x_1 w^\top x_1 & x_1(w^\top x_1 - y) & x_2 w^\top x_1 \\ w & w & w \\ x_1 w^\top x_1 - \lambda w & x_1(w^\top x_1 - y) - \lambda w & x_2 w^\top x_1 - \lambda w \\ w - 2\alpha(x_1 w^\top x_1 - \lambda w) & w' & w - 2\alpha(x_2 w^\top x_1 - \lambda w) \\ w - 2\alpha(x_1 w^\top x_1 - \lambda w) & w' & w' \\ (w - 2\alpha(x_1 w^\top x_1 - \lambda w))^\top x1 & w'^\top x_1 & \mathbf{w}'^\top \mathbf{x_2} \end{bmatrix}$$

- Takeaway: The theoretical finding shows the implementation of a single step of an iterative algorithm can be done in practical in-context learning setting.

Akyürek, Ekin, et al. "What learning algorithm is in-context learning? investigations with linear models." (ICLR 2023).

The Behavior of Real Learners

- Empirically explain ICL at **computational level** by identifying the kind of algorithms to regression problems that transformer-based ICL implements.
- Behavioral Metrics: Quantifying the degree to which two predictors agree.
  - **Squared prediction difference.** Given any learning algorithm $\mathcal{A}$ that maps from a set of input-output pairs $D = [x_1, y_1, \ldots, x_n, y_n]$ to a predictor $f(x) = \mathcal{A}(D)(x)$, the SPD is defined as:

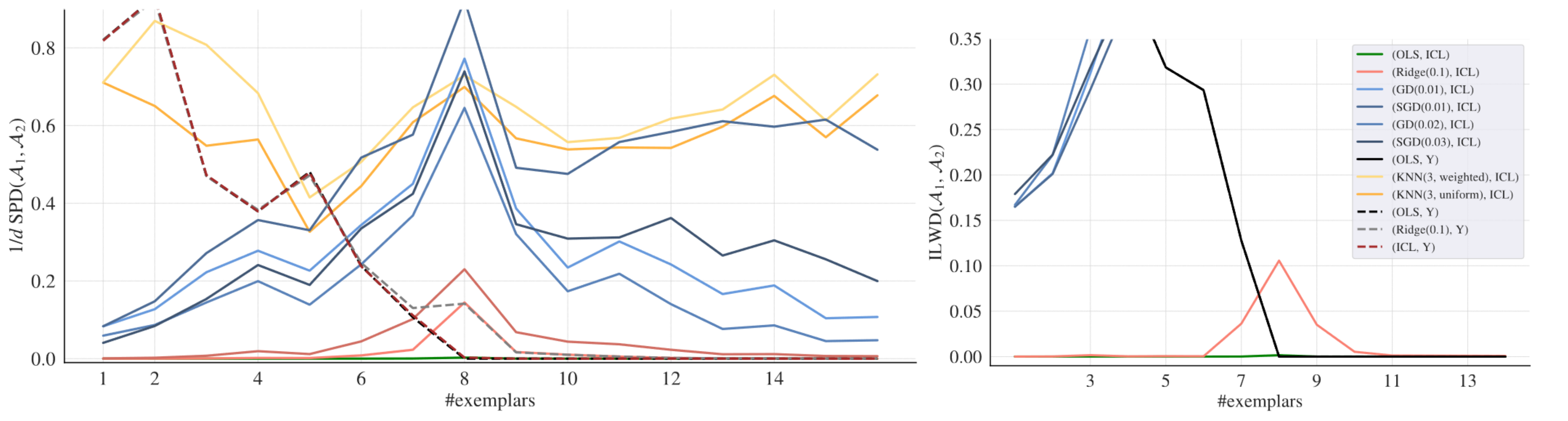
$$\operatorname{SPD}\left(\mathcal{A}_1, \mathcal{A}_2\right) = \underset{\substack{D=[\boldsymbol{x}_1, \ldots] \sim p(D) \\ \boldsymbol{x}' \sim p(\boldsymbol{x})}}{\mathbb{E}} \left(\mathcal{A}_1(D)\left(\boldsymbol{x}'\right) - \mathcal{A}_2(D)\left(\boldsymbol{x}'\right)\right)^2$$

  - **Implicit linear weight difference.** Given $\mathcal{A}$, $\mathcal{D}$, and an additional collection of unlabeled test inputs $D_{\mathcal{X}'} = \{\boldsymbol{x}_i'\}$ and compute a predictor-specific dataset $D_{\mathcal{A}} = \{(\boldsymbol{x}_i', \hat{y}_i)\} = \{(\boldsymbol{x}_i, \mathcal{A}(D)\left(\boldsymbol{x}_i'\right))\}$ then:

$$\hat{\boldsymbol{w}}_{\mathcal{A}} = \underset{\boldsymbol{w}}{\arg\min} \sum_i \left(\hat{y}_i - \boldsymbol{w}^\top \boldsymbol{x}_i'\right)^2 \qquad \operatorname{ILWD}\left(\mathcal{A}_1, \mathcal{A}_2\right) = \mathbb{E}_D \mathbb{E}_{D_{\mathcal{X}'}} \left\|\hat{\boldsymbol{w}}_{\mathcal{A}_1} - \hat{\boldsymbol{w}}_{\mathcal{A}_2}\right\|_2^2$$

Akyürek, Ekin, et al. "What learning algorithm is in-context learning? investigations with linear models." (ICLR 2023).

Results on Noiseless Datasets



(a) Predictor–ICL fit w.r.t. prediction differences. (b) Predictor–ICL fit w.r.t implicit weights.

- The agreement between the ICL and OLS is considerably high.
- When the number of in-context examples is less than the input dimension ($d = 8$), there are multiple linear models can exactly fit the under-determined linear regression problem.
- ICL behaves like OLS in this case, which selects the minimum-norm weight vector, indicating that ICL learns to output the ***minimum Bayes risk*** solution when predicting under uncertainty.

Akyürek, Ekin, et al. "What learning algorithm is in-context learning? investigations with linear models." (ICLR 2023).

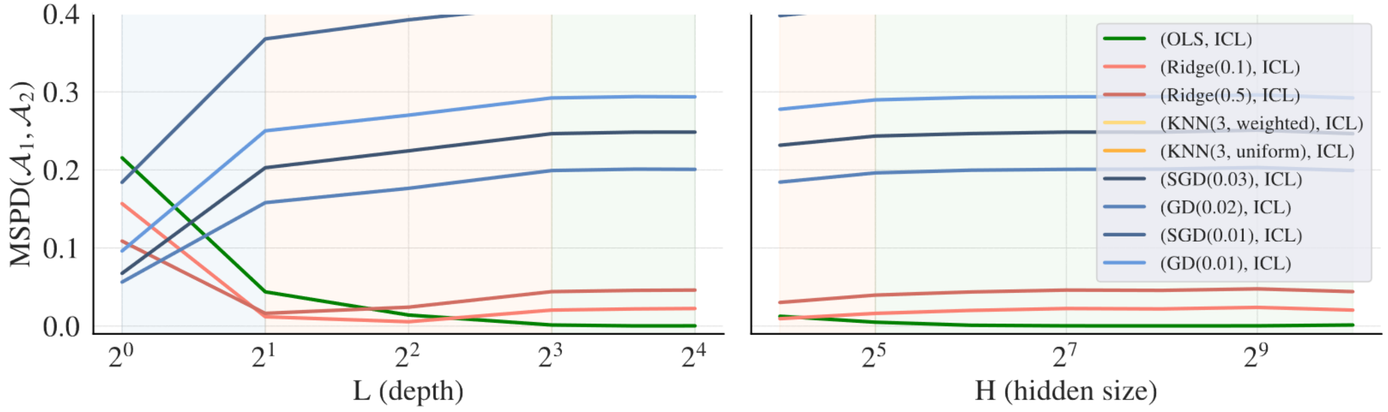
Results on Noisy Datasets

(A_1, A_2) \ σ^2/τ^2	$(0.0/1.0)^2 = 0$	$(0.25/1.0)^2 = 1/16$	$(0.5/1.5)^2 = 1/9$	$(0.5/1.0)^2 = 1/4$	$(0.5/0.75)^2 = 4/9$
(Lstsq, ICL)	1.25e-05	1.34e-04	3.96e-04	1.51e-03	4.13e-03
(Ridge(1/16), ICL)	1.10e-04	3.29e-05	1.12e-04	8.24e-04	2.92e-03
(Ridge(1/9), ICL)	3.49e-04	9.65e-05	3.86e-05	4.50e-04	2.15e-03
(Ridge(1/4), ICL)	1.69e-03	8.64e-04	4.39e-04	3.30e-05	6.81e-04
(Ridge(4/9), ICL)	4.83e-03	3.09e-03	2.21e-03	7.52e-04	6.10e-05

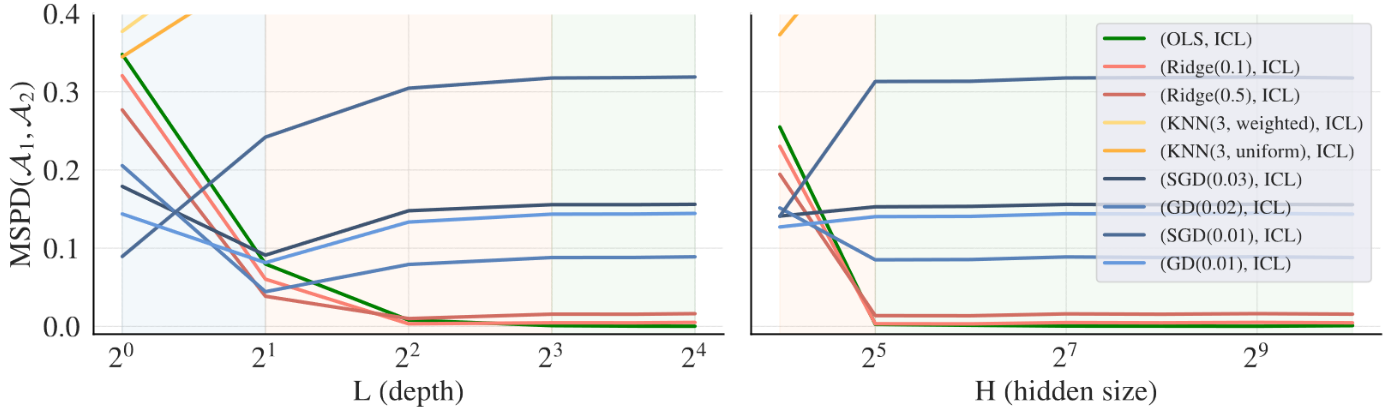
- As noise variance increases, the value of the ridge parameter that best explains ICL behavior also increases, showing that ICL in this setting **behaviorally matches minimum-Bayes-risk predictor**.

Akyürek, Ekin, et al. "What learning algorithm is in-context learning? investigations with linear models." (ICLR 2023).

Relation between Size and Implemented Algorithm



(a) Linear regression problem with $d = 8$



(b) Linear regression problem with $d = 16$

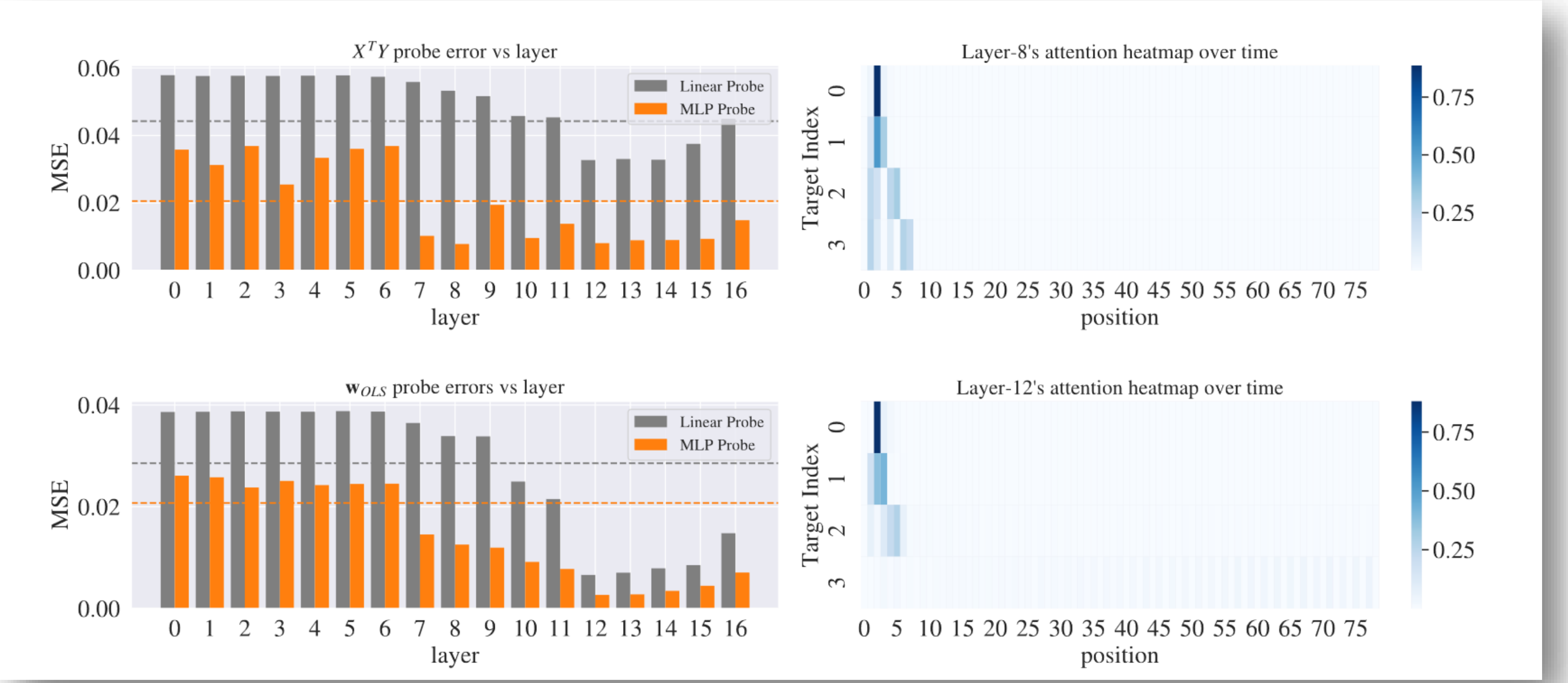
Akyürek, Ekin, et al. "What learning algorithm is in-context learning? investigations with linear models." (ICLR 2023).

Does ICL Encode Meaningful Intermediates?

- Take a trained in-context learner with frozen weights, then train an auxiliary probing model to recover some target quantities from the model's hidden representations.

$$\boldsymbol{\alpha} = \mathrm{softmax}\left(\boldsymbol{s}_v\right)$$

$$\hat{\boldsymbol{v}} = \mathrm{FF}_{\mathrm{v}}\left(\boldsymbol{\alpha}^{\top} W_v H^{(l)}\right)$$



Akyürek, Ekin, et al. "What learning algorithm is in-context learning? investigations with linear models." (ICLR 2023).

Summary

- Data distribution and transformer architecture play an important role in ICL.
  - Natural data distribution’s special properties contribute to the emerge of ICL.
  - Model’s inductive bias may help encode some powerful learning algorithm.
  - Scaling up the model may produce more complex or multi-step learning algorithm that enables learning from context (e.g., novel label space).

THANKS!