Why Larger Language Models Do In-context Learning Differently?

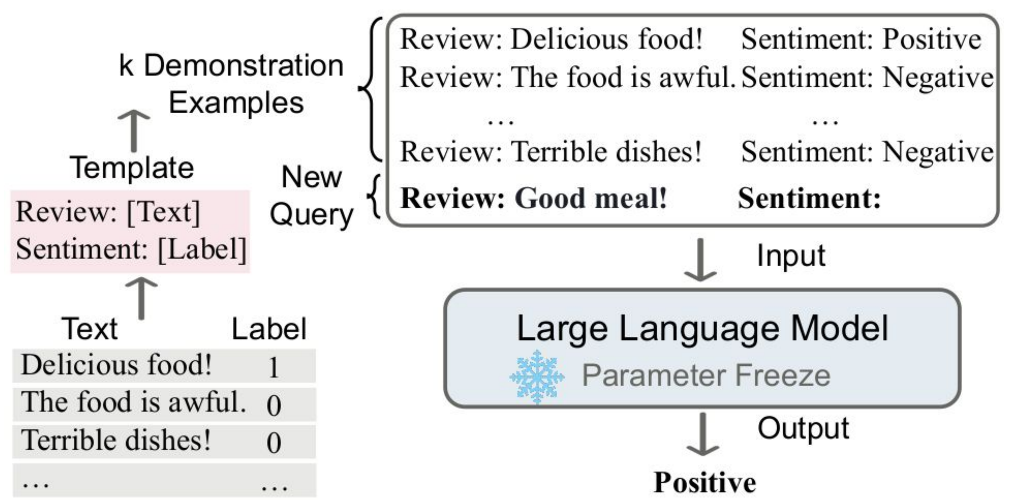
Zhenmei Shi, Junyi Wei, Zhuoyan Xu, Yingyu Liang

WISCONSIN UNIVERSITY OF WISCONSIN–MADISON

香 港 大 學 THE UNIVERSITY OF HONG KONG

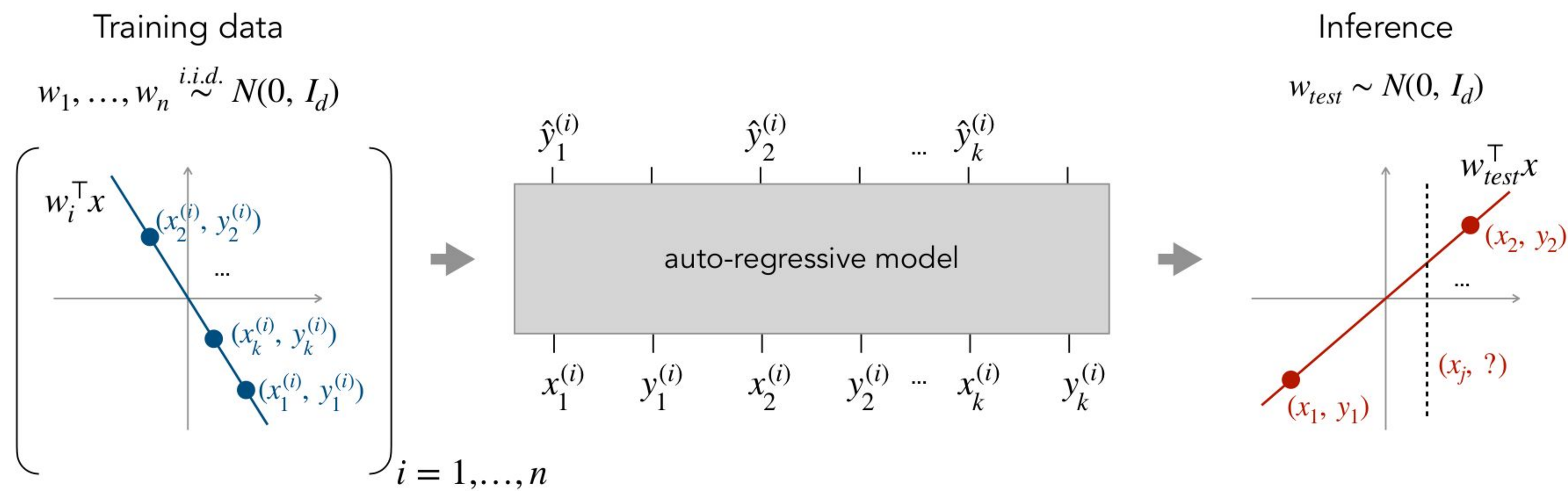
ICML International Conference On Machine Learning

Background



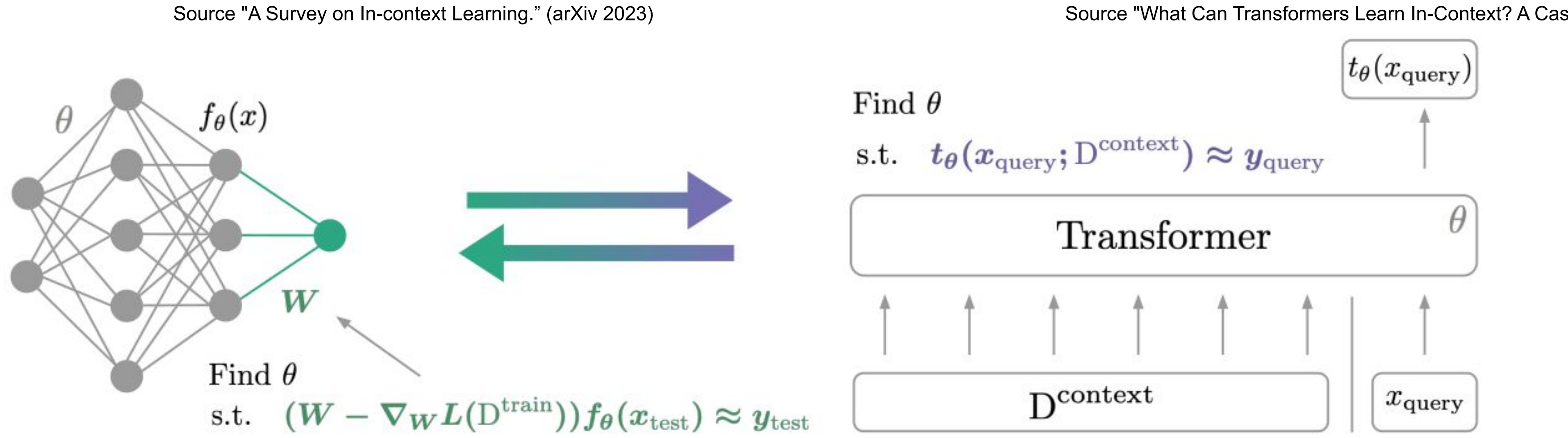
In-context Learning

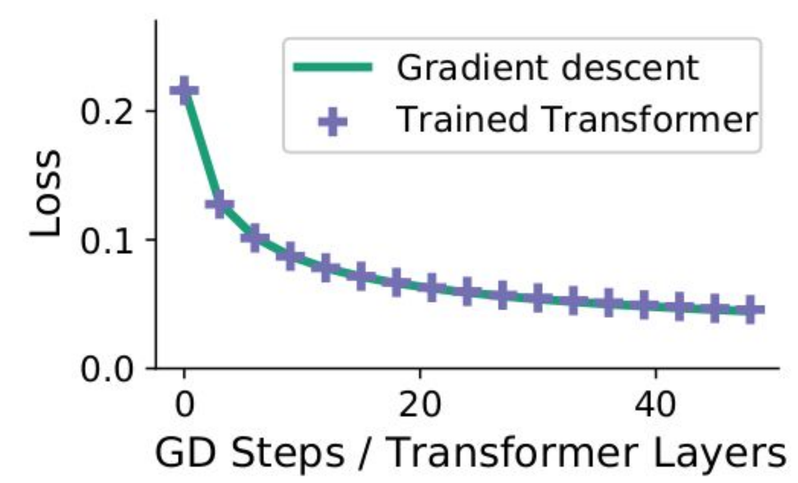
Source "A Survey on In-context Learning." (arXiv 2023)



Linear Regression In-context Learning Data Model

Source "What Can Transformers Learn In-Context? A Case Study of Simple Function Classes." (NeurIPS 2022)

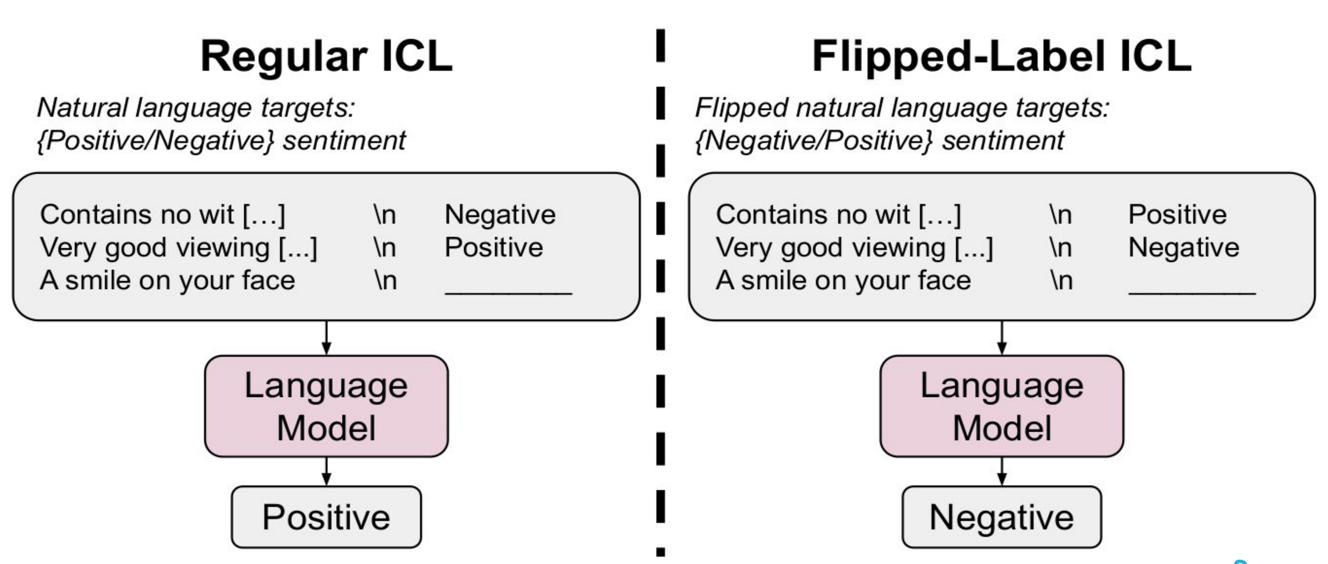




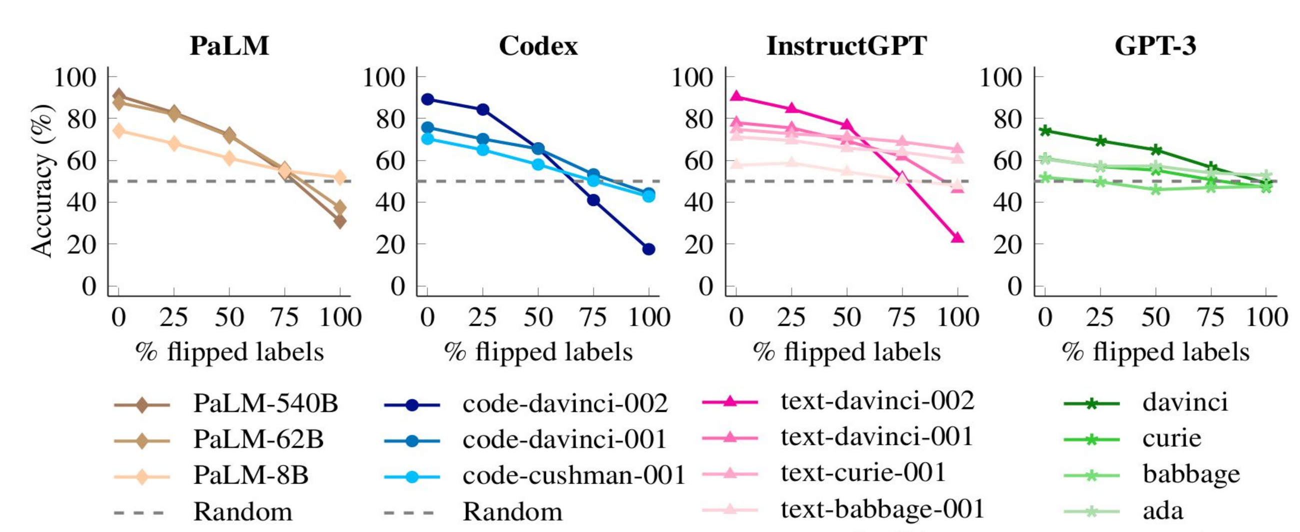
Gradient-based Optimization and Attention-based In-context Learning

Source "Transformers Learn In-Context by Gradient Descent." (ICML 2023)

Motivation

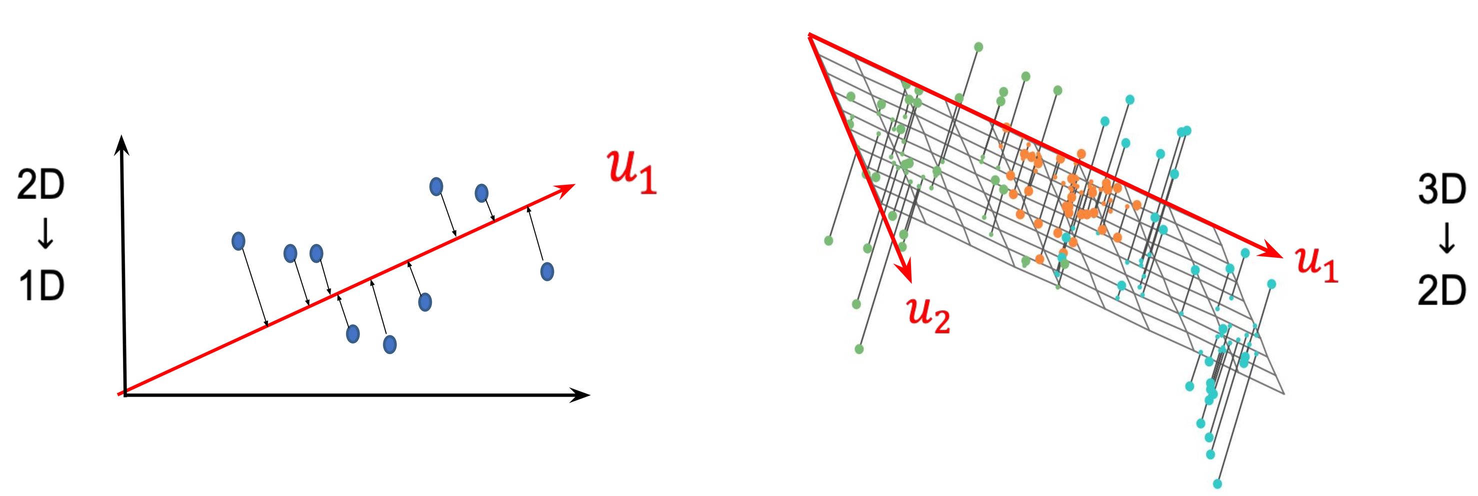


Why are larger language models more sensitive to noise in in-context learning?

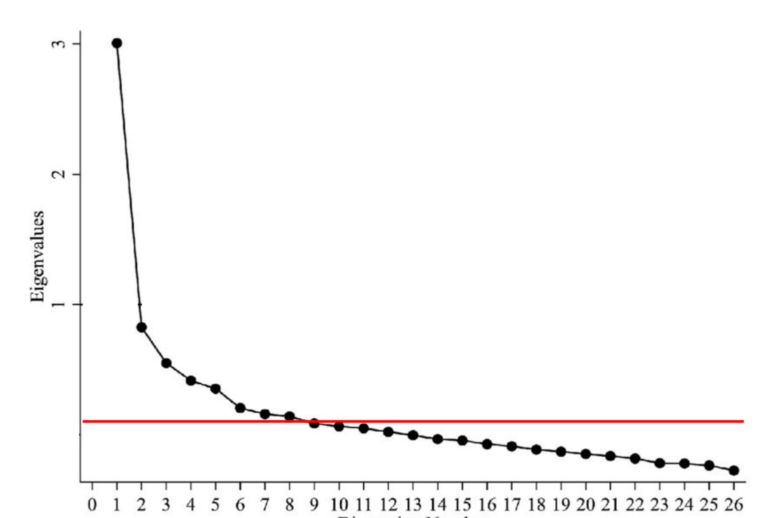


Larger Language Models Do In-context Learning Differently

Source "Larger Language Models Do In-context Learning Differently." (arXiv 2023)



Intuition from Principal Components Analysis (**PCA**)



Signals+Noise across the dimensions

Setup: In-context Learning and Linear Self-Attention

- Formally, a prompt from a task τ is formed by N examples $(x_{\tau,1}, y_{\tau,1}), \dots, (x_{\tau,N}, y_{\tau,N})$ and a query input $x_{\tau,q}$ for prediction,
- $y_{\tau,q}$ is the target label

$$\mathbf{E}_\tau := \begin{pmatrix} \mathbf{x}_{\tau,1} & \mathbf{x}_{\tau,2} & \dots & \mathbf{x}_{\tau,N} & \mathbf{x}_{\tau,q} \\ y_{\tau,1} & y_{\tau,2} & \dots & y_{\tau,N} & 0 \end{pmatrix} \quad y_{\tau,q}$$

Attention Mechanism in LLMs

- Compute projections (key, value, query)

$$\mathbf{Q} = \mathbf{X}\mathbf{W}^Q, \mathbf{K} = \mathbf{X}\mathbf{W}^K, \mathbf{V} = \mathbf{X}\mathbf{W}^V$$

- Compute attention scores using softmax on inner products
- Compute the weighted sum of values weighted by attention

$$\mathbf{Y} = \mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}$$

- We consider the Linear Self-attention Networks (only one-layer attention, and remove softmax):

Model size: rank r of the weight matrice

$$f_{\mathrm{LSA},\theta}(\mathbf{E}) = \left[\mathbf{E} + \mathbf{W}^{PV}\mathbf{E} \cdot \frac{\mathbf{E}^\top \mathbf{W}^{KQ}\mathbf{E}}{\rho}\right]$$

Main Results: ICL Linear Regression

- Task distribution: $\mathbf{w}_\tau \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I_{d\times d})$
- $\mathbf{x}_{\tau,i}, \mathbf{x}_{\tau,q} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Lambda)$
- $y_{\tau,i} = \langle \mathbf{w}_\tau, \mathbf{x}_{\tau,i}\rangle + \epsilon_i$ where $\epsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$

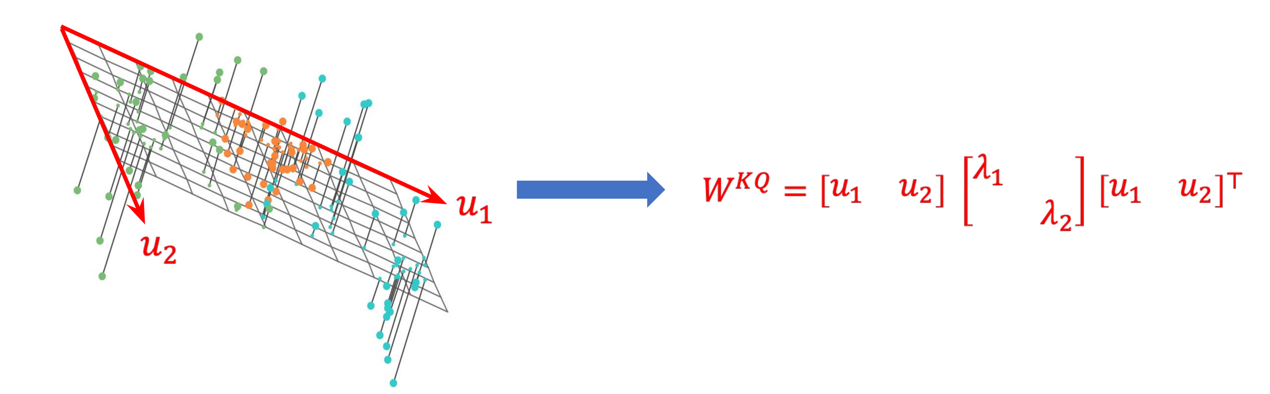
Theorem (Behavior difference, special case and informal)

For $r_1 < r_2$, let f_1 be the optimal rank-r_1 model, and f_2 be the optimal rank-r_2 model. Assume the true regression weight lies in S_{r1}. Let s be the projection of the true weight in S_{r1}.

$$\mathcal{L}(f_2) - \mathcal{L}(f_1) \approx \frac{r_2 - r_1}{M}\|s\|_D^2 + \frac{r_2 - r_1}{M}\sigma^2$$

Theorem (Optimal rank-r solution, informal)

The optimal rank-r weight matrix W^{KQ} lies in S_r, the span of the top-r eigenvectors of the data covariance. It is the truncated version (low rank approximation) of the optimal full-rank solution.



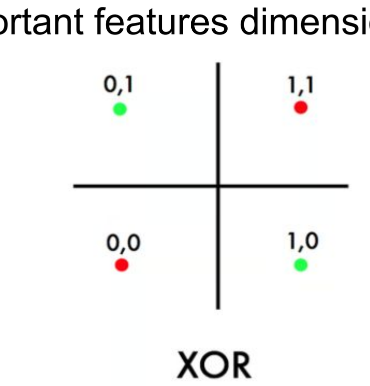
Main Results: Sparse Parity Classification

Setup

- Consider random pick 2 dimension as XOR input
- Important features dimensions and less-important features dimensions
- Linear self-attention with ReLU MLP

$$g(\mathbf{X}, \mathbf{y}, \mathbf{x}_q) = \sum_{i\in[m]} \mathbf{a}_i \sigma\left[\frac{\mathbf{y}^\top \mathbf{X}}{N}\mathbf{W}^{(i)}\mathbf{x}_q\right]$$

- Consider Hinge Loss
- Number of heads to measure mode size



Theorem (Optimal solution for parity)

The optimal solution of smaller number of heads model will mainly encode the **important features**, while larger number of heads model will encode **all features**.

Theorem (Behavior difference for parity)

During evaluation, we can decompose the input into two parts: signal and noise. Both the larger model and smaller model can capture the signal part well. However, the smaller model has a much smaller influence from noise than the larger model.

Take Home Messages

- Larger models are more sensitive to noise injected in-context
- We provide theoretical insights by analyzing two stylized settings
- Main insight: larger models cover more feature directions, but also covers more injected noise, thus are more sensitive to noise

Further thoughts:

- How to ``regularize'' the model to be more robust in-context?
- The same insight can explain the scaling law of LLMs?