In-Context Learning with Transformers: Softmax Attention Adapts to Function Lipschitzness

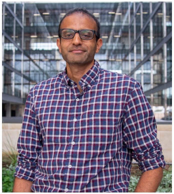
Aryan Mokhtari
ECE Department, UT Austin

ITA Workshop, San Diego
February 23rd, 2024

Collaborators

Liam Collins*

Advait Parulekar*



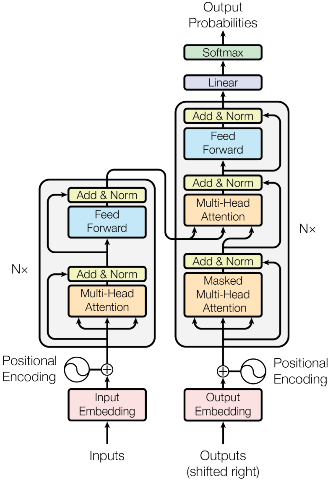
Sujay Sanghavi

Sanjay Shakkottai

*Equal Contribution

Transformer background

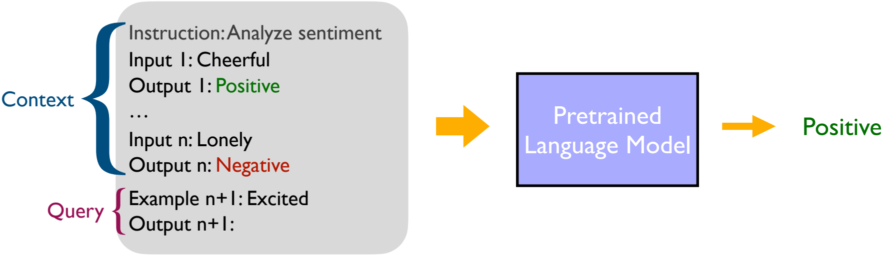
- SOTA language and vision models are ***transformers***
- **Transformer:** Neural network architecture built around ***self-attention*** units [Vaswani et al., 2017]
- **Self-attention:** Maps token sequence to sequence of convex combinations of embeddings of the other tokens, weighted by **softmax** attention score



[Vaswani et al., 2017]

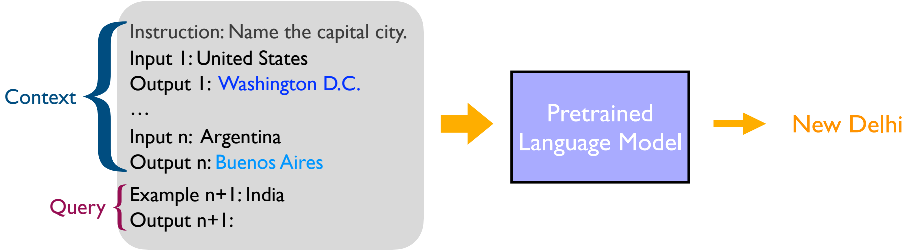
In-Context Learning

- Pretrained language models can perform ***in-context learning*** (ICL) of tasks not seen during pretraining [Brown et al., 2020]
- **ICL**: *few-shot* learning with *single forward pass* (no model updates)



In-Context Learning

- Pretrained language models can perform ***in-context learning*** (ICL) of tasks not seen during pretraining [Brown et al., 2020]
- **ICL**: *few-shot* learning with *single forward pass* (no model updates)



How to explain ICL?

We follow prior work by considering ICL as regression [Garg et al., 2022]

- **ICL task function class:** $\mathcal{F} \subseteq \{f : \mathbb{R}^d \to \mathbb{R}\}$
- **ICL task** t: For some $f_t \in \mathcal{F}$, the model takes as input
  - **context:** $(\mathbf{x}_{t,1}, f_t(\mathbf{x}_{t,1}) + \epsilon_{t,1}), \ldots, (\mathbf{x}_{t,n}, f_t(\mathbf{x}_{t,n}) + \epsilon_{t,n}), \;\; \mathbf{x}_{t,i} \in \mathbb{R}^d$
  - **query:** $\mathbf{x}_{t,n+1}$

goal: predict $f_t(\mathbf{x}_{t,n+1})$

How to explain ICL?

We follow prior work by considering ICL as regression [Garg et al., 2022]

- **ICL task function class:** $\mathcal{F} \subseteq \{f : \mathbb{R}^d \to \mathbb{R}\}$
- **ICL task** t**:** For some $f_t \in \mathcal{F}$, the model takes as input
  - **context:** $(\boldsymbol{x}_{t,1}, f_t(\boldsymbol{x}_{t,1}) + \epsilon_{t,1}), \ldots, (\boldsymbol{x}_{t,n}, f_t(\boldsymbol{x}_{t,n}) + \epsilon_{t,n}),\ \ \boldsymbol{x}_{t,i} \in \mathbb{R}^d$
  - **query:** $\boldsymbol{x}_{t,n+1}$

  goal: predict $f_t(\boldsymbol{x}_{t,n+1})$
- **Pretraining:** train transformer to minimize MSE of its prediction of $f_t(\boldsymbol{x}_{t,n+1})$ across T such tasks

How to explain ICL?

We follow prior work by considering ICL as regression [Garg et al., 2022]

- **ICL task function class:** $\mathcal{F} \subseteq \{f : \mathbb{R}^d \to \mathbb{R}\}$
- **ICL task** t**:** For some $f_t \in \mathcal{F}$, the model takes as input
  - **context:** $(\boldsymbol{x}_{t,1}, f_t(\boldsymbol{x}_{t,1}) + \epsilon_{t,1}), \ldots, (\boldsymbol{x}_{t,n}, f_t(\boldsymbol{x}_{t,n}) + \epsilon_{t,n}), \;\; \boldsymbol{x}_{t,i} \in \mathbb{R}^d$
  - **query:** $\boldsymbol{x}_{t,n+1}$

  goal: predict $f_t(\boldsymbol{x}_{t,n+1})$
- **Pretraining:** train transformer to minimize MSE of its prediction of $f_t(\boldsymbol{x}_{t,n+1})$ across T such tasks
- **Downstream evaluation:** given n labelled examples from a new task $T+1$, predict $f_{T+1}(\boldsymbol{x}_{T+1,n+1})$

ICL as Regression

- Each ICL task is written as a token sequence $\boldsymbol{Z}_t \in \mathbb{R}^{(d+1)\times(n+1)}$:

$$\boldsymbol{Z}_t := \begin{bmatrix} \boldsymbol{x}_{t,1} & \boldsymbol{x}_{t,2} & \dots & \boldsymbol{x}_{t,n} & \boldsymbol{x}_{t,n+1} \\ f_t(\boldsymbol{x}_{t,1}) + \epsilon_{t,1} & f_t(\boldsymbol{x}_{t,2}) + \epsilon_{t,2} & \dots & f_t(\boldsymbol{x}_{t,n}) + \epsilon_{t,n} & 0 \end{bmatrix}$$

- Denote $\boldsymbol{Z}_t = [\boldsymbol{z}_{t,1}, \dots, \boldsymbol{z}_{t,n}]$, where $\boldsymbol{z}_{t,i} = \begin{bmatrix} \boldsymbol{x}_{t,i} \\ f_t(\boldsymbol{x}_{t,i}) + \epsilon_{t,i} \end{bmatrix}$
- Each column $\boldsymbol{z}_{t,i}$ of $\boldsymbol{Z}_t$ is an *embedded token*
- **Context:** $\boldsymbol{z}_{t,1}, \dots, \boldsymbol{z}_{t,n}$
- **Query:** $\boldsymbol{z}_{t,n+1}$ – Goal is to predict its missing label

What is known about ICL in this setting?

Popular idea: ICL can be interpreted as gradient descent (GD)

- [von Oswald et al., 2023, Akyürek et al., 2022, Bai et al., 2023, Fu et al., 2023]: *Existence* of transformers that implement GD and other gradient-based algorithms during ICL on linear regression tasks
  - Unclear whether *pretraining* leads to such transformers
- [Zhang et al., 2022, Ahn et al., 2023, Mahankali et al., 2023]: Solving pretraining loss yields transformers that execute preconditioned GD during ICL
  - Only holds for *linear* attention and *linear* tasks
- [Cheng et al., 2023] Extension to nonlinear attn and tasks: ICL is functional GD
  - Requires activation to be a kernel (does not include softmax)
  - For accurate predictions, requires activation kernel to align with kernel that generates task labels (implicitly assumed in linear analysis)

Key Question

How does **softmax self-attention** learn to perform ICL?

What is known about ICL in this setting?

Popular idea: ICL can be interpreted as gradient descent (GD)

- [von Oswald et al., 2023, Akyürek et al., 2022, Bai et al., 2023, Fu et al., 2023]: *Existence* of transformers that implement GD and other gradient-based algorithms during ICL on linear regression tasks
  - Unclear whether *pretraining* leads to such transformers
- [Zhang et al., 2022, Ahn et al., 2023, Mahankali et al., 2023]: Solving pretraining loss yields transformers that execute preconditioned GD during ICL
  - Only holds for *linear* attention and *linear* tasks
- [Cheng et al., 2023] Extension to nonlinear attn and tasks: ICL is functional GD
  - Requires activation to be a kernel (does not include softmax)
  - For accurate predictions, requires activation kernel to align with kernel that generates task labels (implicitly assumed in linear analysis)

Key Question

How does **softmax self-attention** learn to perform ICL?

What is known about ICL in this setting?

Popular idea: ICL can be interpreted as gradient descent (GD)

- [von Oswald et al., 2023, Akyürek et al., 2022, Bai et al., 2023, Fu et al., 2023]: *Existence* of transformers that implement GD and other gradient-based algorithms during ICL on linear regression tasks
  - Unclear whether *pretraining* leads to such transformers
- [Zhang et al., 2022, Ahn et al., 2023, Mahankali et al., 2023]: Solving pretraining loss yields transformers that execute preconditioned GD during ICL
  - Only holds for *linear* attention and *linear* tasks
- [Cheng et al., 2023] Extension to nonlinear attn and tasks: ICL is functional GD
  - Requires activation to be a kernel (does not include softmax)
  - For accurate predictions, requires activation kernel to align with kernel that generates task labels (implicitly assumed in linear analysis)

Key Question

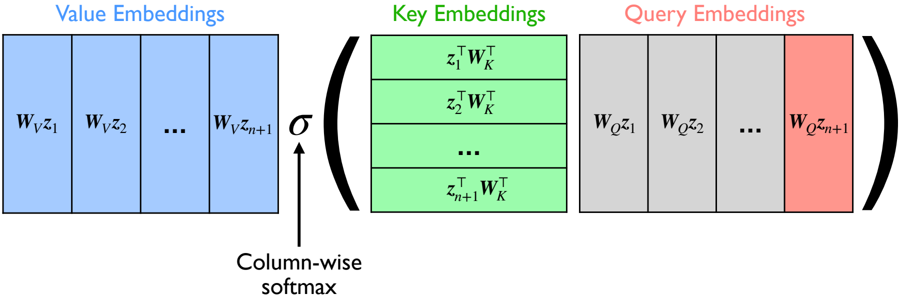
How does **softmax self-attention** learn to perform ICL?

Outline

- **Intuition:** Softmax Attention allows for adapting an *attention window*
- **Results Part I:** Attention window adapts to *function Lipschitzness*
- **Results Part II:** Attention window adapts to *direction-wise function Lipschitzness*

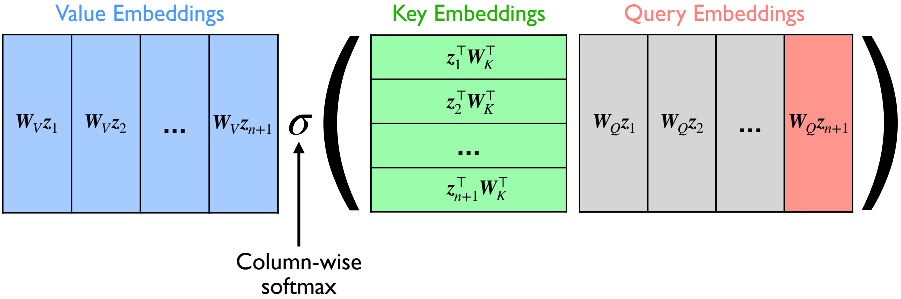
Learning Model: One Layer of Softmax Self-Attention

- **Self-Attention:** Maps each token in $\boldsymbol{Z}$ to another token of same dimension (drop subscript t ease of notation)
- **Parameters:** $\boldsymbol{\theta} := \{\boldsymbol{W}_V, \boldsymbol{W}_K, \boldsymbol{W}_Q\} \in (\mathbb{R}^{(d+1)\times(d+1)})^3$



Learning Model: One Layer of Softmax Self-Attention

- **Self-Attention:** Maps each token in $\boldsymbol{Z}$ to another token of same dimension (drop subscript t ease of notation)
- **Parameters:** $\boldsymbol{\theta} := \{\boldsymbol{W}_V, \boldsymbol{W}_K, \boldsymbol{W}_Q\} \in (\mathbb{R}^{(d+1)\times(d+1)})^3$



$$h_{SA}(\boldsymbol{z}_i, \boldsymbol{Z}; \boldsymbol{\theta}) := \sum_{j=1}^{n} \underbrace{(\boldsymbol{W}_V \boldsymbol{z}_j)}_{\boldsymbol{z}_j \text{ value embedding}} \underbrace{\frac{e^{(\boldsymbol{W}_K \boldsymbol{z}_j)^\top (\boldsymbol{W}_Q \boldsymbol{z}_i)}}{\sum_{j'=1}^{n} e^{(\boldsymbol{W}_K \boldsymbol{z}'_j)^\top (\boldsymbol{W}_Q \boldsymbol{z}_i)}}}_{\text{Attention } \boldsymbol{z}_i \text{ pays to } \boldsymbol{z}_j}. \tag{1}$$

Pretraining Loss

Recall

$$\boldsymbol{Z} := \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_n & \boldsymbol{x}_{n+1} \\ f(\boldsymbol{x}_1) + \epsilon_1 & f(\boldsymbol{x}_2) + \epsilon_1 & \dots & f(\boldsymbol{x}_n) + \epsilon_n & \color{red}{0} \end{bmatrix}$$

For simplicity, we consider

$$\boldsymbol{W}_V = \begin{bmatrix} \boldsymbol{0}_{d\times d} & \boldsymbol{0}_d \\ \boldsymbol{0}_d & 1 \end{bmatrix}, \quad \boldsymbol{W}_K = \begin{bmatrix} \boldsymbol{M}_K & \boldsymbol{0}_d \\ \boldsymbol{0}_d & 0 \end{bmatrix}, \quad \boldsymbol{W}_Q = \begin{bmatrix} \boldsymbol{M}_Q & \boldsymbol{0}_d \\ \boldsymbol{0}_d & 0 \end{bmatrix}$$

and define $\boldsymbol{M} := \boldsymbol{M}_K^\top \boldsymbol{M}_Q \in \mathbb{R}^{d\times d}$, thus

$$h_{SA}(\boldsymbol{z}_{n+1}, \boldsymbol{Z}; \boldsymbol{M})_{\color{blue}{d+1}} = \sum_{i=1}^{n} (f(\boldsymbol{x}_i) + \epsilon_i) \frac{e^{\boldsymbol{x}_i^\top \boldsymbol{M} \boldsymbol{x}_{n+1}}}{\sum_{i=1}^{n} e^{\boldsymbol{x}_i^\top \boldsymbol{M} \boldsymbol{x}_{n+1}}} \tag{2}$$

- (d+1)-th element of $h_{SA}(\boldsymbol{z}_{n+1}, \boldsymbol{Z}; \boldsymbol{M})$: prediction of $f(\boldsymbol{x}_{n+1})$

Pretraining Loss

- Let $\boldsymbol{y}_{t,i} := f_t(\boldsymbol{x}_{t,i}) + \epsilon_{t,i}$. Empirical loss on T contexts:

$$\hat{\mathcal{L}}(\boldsymbol{M}) := \frac{1}{T}\sum_{t=1}^{T}\left(\sum_{i=1}^{n}\boldsymbol{y}_{t,i}\frac{e^{\boldsymbol{x}_{t,i}^\top \boldsymbol{M}\,\boldsymbol{x}_{t,n+1}}}{\sum_{i=1}^{n} e^{\boldsymbol{x}_{t,i}^\top \boldsymbol{M}\,\boldsymbol{x}_{t,n+1}}} - \boldsymbol{y}_{t,n+1}\right)^2$$

Pretraining Loss

- Let $\boldsymbol{y}_{t,i} := f_t(\boldsymbol{x}_{t,i}) + \epsilon_{t,i}$. Empirical loss on T contexts:

$$\hat{\mathcal{L}}(\boldsymbol{M}) := \frac{1}{T}\sum_{t=1}^{T}\left(\sum_{i=1}^{n}\boldsymbol{y}_{t,i}\frac{e^{\boldsymbol{x}_{t,i}^\top \boldsymbol{M}\,\boldsymbol{x}_{t,n+1}}}{\sum_{i=1}^{n} e^{\boldsymbol{x}_{t,i}^\top \boldsymbol{M}\,\boldsymbol{x}_{t,n+1}}} - \boldsymbol{y}_{t,n+1}\right)^2$$

- For simplicity, we consider the population version:

$$\mathcal{L}(\boldsymbol{M}) := \mathbb{E}_{f,\{\boldsymbol{x}_i\}_i,\{\epsilon_i\}_i}\left(\sum_{i=1}^{n}(f(\boldsymbol{x}_i)+\epsilon_i)\frac{e^{\boldsymbol{x}_i^\top \boldsymbol{M}\,\boldsymbol{x}_{n+1}}}{\sum_{i=1}^{n} e^{\boldsymbol{x}_i^\top \boldsymbol{M}\,\boldsymbol{x}_{n+1}}} - f(\boldsymbol{x}_{n+1})\right)^2$$

where $f \sim D(\mathcal{F})$, $\boldsymbol{x}_i \overset{\text{i.i.d.}}{\sim} D_{\boldsymbol{x}}$, $\epsilon_i \overset{\text{i.i.d.}}{\sim} D_\epsilon$

How do minimizers of (ICL) adapt to $D(\mathcal{F}), D_{\boldsymbol{x}}, D_\epsilon$?

ICL estimator intuition

Let us return to the ICL estimator:

$$\hat{f}(\boldsymbol{x}_{n+1}) = \sum_{i=1}^{n} (f(\boldsymbol{x}_i) + \epsilon_i) \; \frac{e^{\boldsymbol{x}_i^\top \boldsymbol{M} \boldsymbol{x}_{n+1}}}{\sum_{j=1}^{n} e^{\boldsymbol{x}_j^\top \boldsymbol{M} \boldsymbol{x}_{n+1}}}$$

↑
some type of distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_{n+1}$

Lemma 1: Inverting the data covariance

Under natural symmetry conditions:

$$\boldsymbol{x}_i \sim \boldsymbol{\Sigma}^{1/2}\mathcal{U}^d \;\; \forall\, i \;\; \Longrightarrow \;\; \boldsymbol{M}^* = w_{KQ}\boldsymbol{\Sigma}^{-1}$$

for any $\boldsymbol{M}^* \in \arg\min \mathcal{L}(\boldsymbol{M})$ and some $w_{KQ} \geq 0$, where $\mathcal{U}^d$ is the uniform distribution over the d-dimensional hypersphere.

For the remainder of the talk, $\boldsymbol{\Sigma} = \boldsymbol{I}_d$ WLOG.

Does not explain ICL completely. $\rightarrow$ What is w_{KQ}?

ICL Estimator Intuition

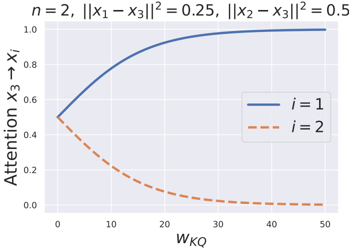
$$\text{Attention } \boldsymbol{x}_{n+1} \to \boldsymbol{x}_i\text{: } \frac{e^{-\frac{w_{KQ}}{2}\| \boldsymbol{x}_i - \boldsymbol{x}_{n+1} \|^2}}{\sum_{j=1}^{n} e^{-\frac{w_{KQ}}{2}\| \boldsymbol{x}_j - \boldsymbol{x}_{n+1} \|^2}}$$

1. **Observation 1:** Attention is *larger for points closer to* $\boldsymbol{x}_{n+1}$
   - convenient if $\| \boldsymbol{x}_i - \boldsymbol{x}_{n+1} \|$ is a proxy for $|f(\boldsymbol{x}_i) - f(\boldsymbol{x}_{n+1})|$

ICL Estimator Intuition

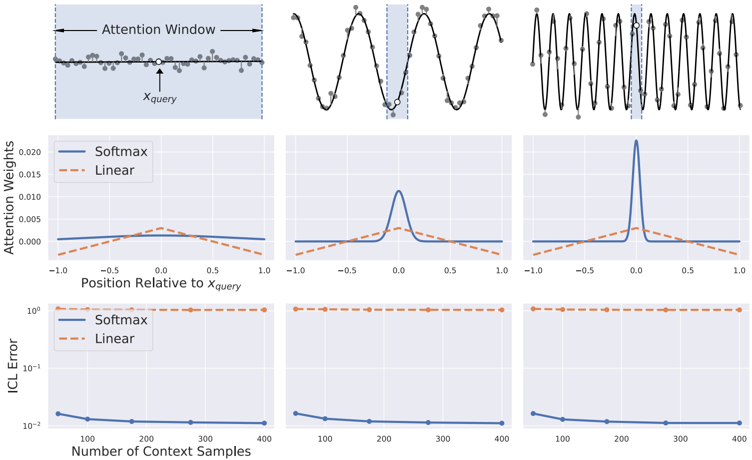
$$\text{Attention } \boldsymbol{x}_{n+1} \to \boldsymbol{x}_i\text{: } \frac{e^{-\frac{w_{KQ}}{2}\| \boldsymbol{x}_i - \boldsymbol{x}_{n+1} \|^2}}{\sum_{j=1}^{n} e^{-\frac{w_{KQ}}{2}\| \boldsymbol{x}_j - \boldsymbol{x}_{n+1} \|^2}}$$

1. **Observation 1:** Attention is *larger for points closer to* $\boldsymbol{x}_{n+1}$
   - convenient if $\| \boldsymbol{x}_i - \boldsymbol{x}_{n+1} \|$ is a proxy for $|f(\boldsymbol{x}_i) - f(\boldsymbol{x}_{n+1})|$
2. **Observation 2:** How much larger? Controlled by w_{KQ}



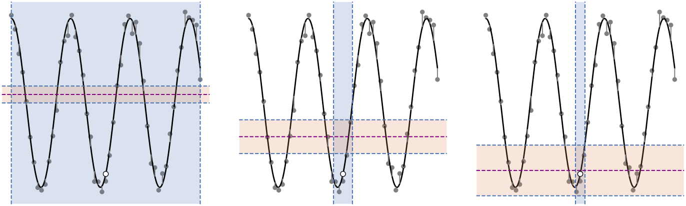
- Larger w_{KQ} → attention concentrates on the closest point(s)

ICL estimator intuition



Top: Varying Lipschitzness in the ground truth results in different optimal attention windows. **Middle:** Attention unit can adapt to these attention windows using softmax. **Bottom:** ICL error with varying number of context tokens.

ICL estimator intuition



How $\|\boldsymbol{M}\|$ changes the prediction.

Tradeoff	$\boldsymbol{M}$ large	$\boldsymbol{M}$ small
Attention decays	quickly with distance	slowly with distance
Number of points that are attended to	fewer	more
Estimator bias	Low	High
Noise Variance	High	Low

Intuition Summary

Function change	Desired (attention window, w_{KQ})
Rapid	(Small, Large)
Slow	(Large, Small)

To formally capture these intuitions, we consider the following ReLU-based function class (see paper for additional classes):

$$\mathcal{F}_L^+ := \{f : f(\boldsymbol{x}) = \ell_1(\boldsymbol{q}^\top \boldsymbol{x})_+ + \ell_2(-\boldsymbol{q}^\top \boldsymbol{x})_+ + b, \;\; \ell_1, \ell_2, b \in [-L, L]^3\}$$

where $D(\mathcal{F}_L^+)$ is induced by $\ell_1, \ell_2, b \sim \mathrm{Unif}([-L, L])$, and $(z)_+ := \max(z, 0)$.

Definition (Lipschitzness)

The Lipschitzness of a function f is defined as:

$$\mathrm{Lip}(f) := \inf_{L \in \mathbb{R}} \{L : \|f(\boldsymbol{x}) - f(\boldsymbol{x}')\| \leq L\|\mathrm{x} - \mathrm{x}'\| \quad \forall \; \boldsymbol{x}, \boldsymbol{x}'\}$$

- For any $f \in \mathcal{F}$, $\mathrm{Lip}(f) = \max(|\ell_1|, |\ell_2|)$.

Results Part I: Softmax Attention Adapts Attention Window to Lipschitzness

Theorem 2

For sufficiently large n, the minimizer of the of the pretraining population loss induced by $D(\mathcal{F}_L^+)$ is $\boldsymbol{M}^* = w_{KQ}\boldsymbol{I}_d$ where

$$\Omega\left(\left(\frac{nL^2}{\sigma^2}\right)^{\frac{1}{d}}\right) < w_{KQ} < \mathcal{O}\left(\left(\frac{nL^2}{\sigma^2}\right)^{\frac{2}{d}}\right)$$

Results Part I: Softmax Attention Adapts Attention Window to Lipschitzness

Theorem 2

For sufficiently large n, the minimizer of the of the pretraining population loss induced by $D(\mathcal{F}_L^+)$ is $\boldsymbol{M}^* = w_{KQ}\boldsymbol{I}_d$ where

$$\Omega\left(\left(\frac{nL^2}{\sigma^2}\right)^{\frac{1}{d}}\right) < w_{KQ} < \mathcal{O}\left(\left(\frac{nL^2}{\sigma^2}\right)^{\frac{2}{d}}\right)$$

Some extreme examples

- $\sigma \to \infty$: no signal, average the noise, $w_{KQ} \to 0$.
- $L \gg \sigma$: no point aggregating noise, pick the nearest neighbour, $w_{KQ} \to \infty$.
- $n \to \infty$: the query token is in the context! Choose the nearest neighbour, $w_{KQ} \to \infty$.
- To our knowledge, *first result* showing how pretrained softmax attention facilitates ICL.

Pretraining Loss

- Pretraining population loss, where $f \sim D(\mathcal{F})$, $\boldsymbol{x}_i \overset{\text{i.i.d.}}{\sim} D_{\boldsymbol{x}}$, $\epsilon_i \overset{\text{i.i.d.}}{\sim} D_\epsilon$:

$$\mathcal{L}(\boldsymbol{M}) := \mathbb{E}_{f,\{\boldsymbol{x}_i\},\{\epsilon_i\}} \left(\frac{\sum_{i=1}^{n} (f(\boldsymbol{x}_i) + \epsilon_i)\, e^{\boldsymbol{x}_i^\top \boldsymbol{M} \boldsymbol{x}_{n+1}}}{\sum_{i=1}^{n} e^{\boldsymbol{x}_i^\top \boldsymbol{M} \boldsymbol{x}_{n+1}}} - f(\boldsymbol{x}_{n+1}) \right)^2$$

Pretraining Loss

- Pretraining population loss, where $f \sim D(\mathcal{F})$, $\boldsymbol{x}_i \overset{\text{i.i.d.}}{\sim} D_{\boldsymbol{x}}$, $\epsilon_i \overset{\text{i.i.d.}}{\sim} D_\epsilon$:

$$\mathcal{L}(\boldsymbol{M}) := \mathbb{E}_{f,\{\boldsymbol{x}_i\},\{\epsilon_i\}} \left(\frac{\sum_{i=1}^n (f(\boldsymbol{x}_i) + \epsilon_i)\, e^{\boldsymbol{x}_i^\top \boldsymbol{M} \boldsymbol{x}_{n+1}}}{\sum_{i=1}^n e^{\boldsymbol{x}_i^\top \boldsymbol{M} \boldsymbol{x}_{n+1}}} - f(\boldsymbol{x}_{n+1}) \right)^2$$

- Decomposition:

$$\mathcal{L}(\boldsymbol{M}) := \underbrace{\mathbb{E}_{f,\{\boldsymbol{x}_i\},} \left(\sum_{i=1}^n f(\boldsymbol{x}_i) \frac{e^{\boldsymbol{x}_i^\top \boldsymbol{M} \boldsymbol{x}_{n+1}}}{\sum_{i=1}^n e^{\boldsymbol{x}_i^\top \boldsymbol{M} \boldsymbol{x}_{n+1}}} - f(\boldsymbol{x}_{n+1}) \right)^2}_{\mathcal{L}_{\text{signal}}(\boldsymbol{M})}$$

$$+ \underbrace{\mathbb{E}_{\{\boldsymbol{x}_i\},\{\epsilon_i\}} \left(\frac{\sum_{i=1}^n \epsilon_i \, e^{\boldsymbol{x}_i^\top \boldsymbol{M} \boldsymbol{x}_{n+1}}}{\sum_{i=1}^n e^{\boldsymbol{x}_i^\top \boldsymbol{M} \boldsymbol{x}_{n+1}}} \right)^2}_{\mathcal{L}_{\text{noise}}(\boldsymbol{M})} \qquad \text{(ICL)}$$

Proof Sketch

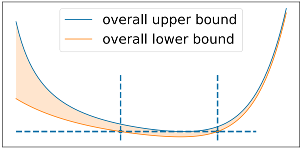
- Concentrations of various functionals of the empirical distribution of tokens on the hypersphere:
  - Recall $\mathcal{L}_{\text{noise}}(w_{KQ}) = \sigma^2 \sum_i \frac{e^{-2w_{KQ}\| x_i - x_{n+1} \|^2}}{\left(\sum_j e^{-w_{KQ}\| x_j - x_{n+1} \|^2}\right)^2}$. For the relevant range of w_{KQ}, $\mathcal{L}_{\text{noise}}(w_{KQ}) = \Theta\left(\sigma^2 \frac{w_{KQ}^{\frac{d}{2}}+1}{n}\right)$.
  - Recall $\mathcal{L}_{\text{bias}}(w_{KQ}) = \left(\sum_i f(\boldsymbol{x}_i) \frac{e^{-w_{KQ}\| x_i - x_{n+1} \|^2}}{\sum_j e^{-w_{KQ}\| x_j - x_{n+1} \|^2}} - f(\boldsymbol{x}_{n+1})\right)^2$.

$$\Omega\left(\frac{L^2}{w_{KQ}^2}\right) < \mathcal{L}_{\text{bias}}(w_{KQ}) < \Theta\left(\frac{L^2}{w_{KQ}} + \frac{L^2}{n}\right).$$

- Concentrations of various functionals of the empirical distribution of tokens on the hypersphere:
  - Recall $\mathcal{L}_{\text{noise}}(w_{KQ}) = \sigma^2 \sum_i \frac{e^{-2w_{KQ}\| \mathbf{x}_i - \mathbf{x}_{n+1} \|^2}}{\left(\sum_j e^{-w_{KQ}\| \mathbf{x}_j - \mathbf{x}_{n+1} \|^2}\right)^2}$. For the relevant range of w_{KQ}, $\mathcal{L}_{\text{noise}}(w_{KQ}) = \Theta\left(\sigma^2 \frac{w_{KQ}^{\frac{d}{2}}+1}{n}\right)$.
  - Recall $\mathcal{L}_{\text{bias}}(w_{KQ}) = \left(\sum_i f(\mathbf{x}_i) \frac{e^{-w_{KQ}\| \mathbf{x}_i - \mathbf{x}_{n+1} \|^2}}{\sum_j e^{-w_{KQ}\| \mathbf{x}_j - \mathbf{x}_{n+1} \|^2}} - f(\mathbf{x}_{n+1})\right)^2$.

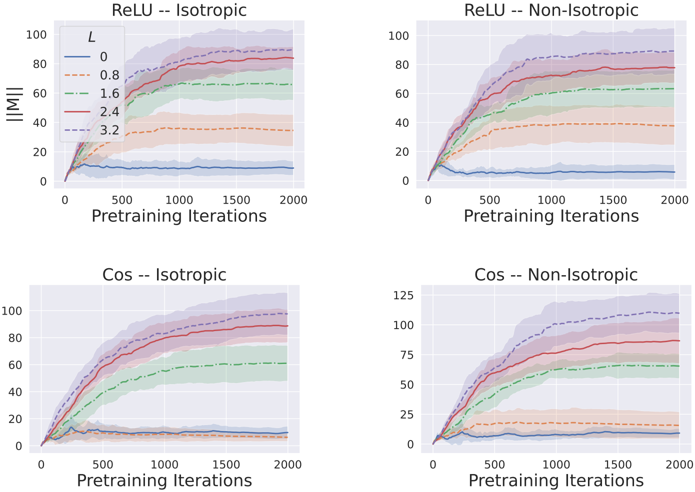
$$\Omega\left(\frac{L^2}{w_{KQ}^2}\right) < \mathcal{L}_{\text{bias}}(w_{KQ}) < \Theta\left(\frac{L^2}{w_{KQ}} + \frac{L^2}{n}\right).$$

- Use these to get a range for w_{KQ}.



Empirical Scaling of $\|M\|$ with L

- $\|\boldsymbol{M}\|$ grows faster during pretraining with larger L, on both ReLU and Cosine tasks, with both isotropic and non-isotropic $\boldsymbol{x}_i$.



Shared Lipschitzness across (train, test) is both **necessary** and **sufficient** for ICL.

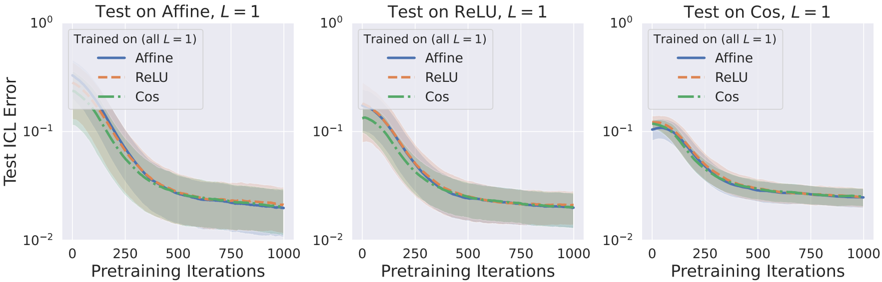
Theorem 3: Generalization

Suppose $\boldsymbol{M}$ is pretrained on the loss induced by $D(\mathcal{F}_L^+)$. Suppose it is tested on $D(\mathcal{F}_L^+)$ then for large enough n, $\mathcal{L}(\boldsymbol{M}) \leq L^{2-\frac{2}{d+2}} \left(\frac{\sigma^2}{n}\right)^{\frac{1}{d+2}}$. Meanwhile, if it is tested on $D(\mathcal{F}_{L'}^+)$,

$$\mathcal{L}(\boldsymbol{M}) \geq \begin{cases} L'^2 \left(\frac{\sigma^2}{nL^2}\right)^{\frac{2}{d+2}} & L' > L \quad \text{(worse dependence on } L'\text{)} \\ \frac{\left(\frac{nL^2}{\sigma^2}\right)^{\frac{d}{2(d+2)}}}{n} & L' \leq L \quad \text{(no benefit of } L'\text{)} \end{cases}$$

Empirical Generalization

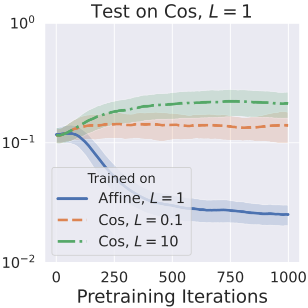
Shared Lipschitzness across (train, test) is **sufficient** for ICL.



- 3 attention units, pretrained on Affine, ReLU, and Cosine tasks with $L = 1$
- Tested on **(Left)** Affine, **(Middle)** ReLU, **(Right)** Cosine, with $L = 1$
- **All three attention units generalize to all test distributions since L is the same**

Empirical Generalization

Shared Lipschitzness across (train, test) is **necessary** for ICL.



- 3 attention units, pretrained on Affine tasks with $L = 1$, Cosine with $L = 0.1$, Cosine with $L = 10$
- Test on Cosine tasks with $L = 1$
- **Only pretraining on the same Lipschitzness generalizes**
  - Please see paper for formal results (Theorem 3.5)

Results Part II: Softmax Attn Learns Attn Window *Direction*

- Now suppose labels depend only on a *low-dimensional* component of the input
- $\exists\, \boldsymbol{B} \in \mathbb{R}^{d \times k}$ such that all $f_t \in \mathcal{F}$ satisfy $f_t(\boldsymbol{x}) = g_t(\boldsymbol{B}^\top \boldsymbol{x})$ for some $g_t : \mathbb{R}^k \to \mathbb{R}$
  - Let $\boldsymbol{B}$ have orthonormal columns WLOG
- Interesting case: $k \ll d$, then learning $\mathrm{col}(\boldsymbol{B})$ drastically reduces ICL problem dimension
  - Attention window of softmax attention should depend only on projections of the input onto $\mathrm{col}(\boldsymbol{B})$

Key Question

Does softmax attention recover $\mathrm{col}(\boldsymbol{B})$?
$\iff$ Does attention window depend only on projections onto $\mathrm{col}(\boldsymbol{B})$?

Function Class and Connection with Lipschitzness

- Function class: $\mathcal{F}_{\boldsymbol{B}}^{\text{lin}} := \{f : f(\boldsymbol{x}) = \boldsymbol{a}^\top \boldsymbol{B}^\top \boldsymbol{x},\ \boldsymbol{a} \in \mathbb{S}^{k-1}\}$, and $D(\mathcal{F}_{\boldsymbol{B}}^{\text{lin}})$ is induced by drawing $\boldsymbol{a} \sim \text{Unif}(\mathbb{S}^{k-1})$.

Function Class and Connection with Lipschitzness

- Function class: $\mathcal{F}_{\boldsymbol{B}}^{\text{lin}} := \{f : f(\boldsymbol{x}) = \boldsymbol{a}^\top \boldsymbol{B}^\top \boldsymbol{x},\ \boldsymbol{a} \in \mathbb{S}^{k-1}\}$, and $D(\mathcal{F}_{\boldsymbol{B}}^{\text{lin}})$ is induced by drawing $\boldsymbol{a} \sim \text{Unif}(\mathbb{S}^{k-1})$.

Definition (Direction-wise Lipschitzness - Informal)

For any direction $\boldsymbol{s} \in \mathbb{S}^{d-1}$:

$$\text{Lip}_{\boldsymbol{s}}(f) := \inf_{L \in \mathbb{R}} \{L :\ f(\boldsymbol{s}\boldsymbol{s}^\top \boldsymbol{x}) - f(\boldsymbol{s}\boldsymbol{s}^\top \boldsymbol{x}') \leq L|\boldsymbol{s}^\top \boldsymbol{x} - \boldsymbol{s}^\top \boldsymbol{x}'|\ \ \forall\ \boldsymbol{x}, \boldsymbol{x}'\}$$

Function Class and Connection with Lipschitzness

- Function class: $\mathcal{F}_{\boldsymbol{B}}^{\text{lin}} := \{f : f(\boldsymbol{x}) = \boldsymbol{a}^\top \boldsymbol{B}^\top \boldsymbol{x},\ \boldsymbol{a} \in \mathbb{S}^{k-1}\}$, and $D(\mathcal{F}_{\boldsymbol{B}}^{\text{lin}})$ is induced by drawing $\boldsymbol{a} \sim \text{Unif}(\mathbb{S}^{k-1})$.

Definition (Direction-wise Lipschitzness - Informal)

For any direction $\boldsymbol{s} \in \mathbb{S}^{d-1}$:

$$\text{Lip}_{\boldsymbol{s}}(f) := \inf_{L \in \mathbb{R}} \{L :\ f(\boldsymbol{s}\boldsymbol{s}^\top \boldsymbol{x}) - f(\boldsymbol{s}\boldsymbol{s}^\top \boldsymbol{x}') \leq L|\boldsymbol{s}^\top \boldsymbol{x} - \boldsymbol{s}^\top \boldsymbol{x}'|\ \ \forall\ \boldsymbol{x}, \boldsymbol{x}'\}$$

- If $\boldsymbol{s} \in \text{col}(\boldsymbol{B})$ (label-relevant direction)
  - $\max_{f \in \mathcal{F}_{\boldsymbol{B}}^{\text{lin}}} \text{Lip}_{\boldsymbol{s}}(f) = 1$, Recovering $\text{col}(\boldsymbol{B}) \implies \boldsymbol{s}^\top \boldsymbol{M} \boldsymbol{s} > 0$

Function Class and Connection with Lipschitzness

- Function class: $\mathcal{F}_{\boldsymbol{B}}^{\text{lin}} := \{f : f(\boldsymbol{x}) = \boldsymbol{a}^\top \boldsymbol{B}^\top \boldsymbol{x},\ \boldsymbol{a} \in \mathbb{S}^{k-1}\}$, and $D(\mathcal{F}_{\boldsymbol{B}}^{\text{lin}})$ is induced by drawing $\boldsymbol{a} \sim \text{Unif}(\mathbb{S}^{k-1})$.

Definition (Direction-wise Lipschitzness - Informal)

For any direction $\boldsymbol{s} \in \mathbb{S}^{d-1}$:

$$\text{Lip}_{\boldsymbol{s}}(f) := \inf_{L \in \mathbb{R}} \{L :\ f(\boldsymbol{s}\boldsymbol{s}^\top \boldsymbol{x}) - f(\boldsymbol{s}\boldsymbol{s}^\top \boldsymbol{x}') \leq L|\boldsymbol{s}^\top \boldsymbol{x} - \boldsymbol{s}^\top \boldsymbol{x}'|\ \forall\ \boldsymbol{x}, \boldsymbol{x}'\}$$

- If $\boldsymbol{s} \in \text{col}(\boldsymbol{B})$ (label-relevant direction)
  - $\max_{f \in \mathcal{F}_{\boldsymbol{B}}^{\text{lin}}} \text{Lip}_{\boldsymbol{s}}(f) = 1$, Recovering $\text{col}(\boldsymbol{B}) \implies \boldsymbol{s}^\top \boldsymbol{M}\boldsymbol{s} > 0$
- If $\boldsymbol{s} \in \text{col}(\boldsymbol{B})^\perp$ (spurious direction)
  - $\max_{f \in \mathcal{F}_{\boldsymbol{B}}^{\text{lin}}} \text{Lip}_{\boldsymbol{s}}(f) = 0$, Recovering $\text{col}(\boldsymbol{B}) \implies \boldsymbol{s}^\top \boldsymbol{M}\boldsymbol{s} = 0$

Recovering $\text{col}(\boldsymbol{B}) \equiv$ Learning direction-wise Lipschitzness of $\mathcal{F}_{\boldsymbol{B}}^{\text{lin}}$

Main Result

Theorem 4 (Informal)

Suppose $n = 2$ *or* $\sigma = 0$. *Then for some* $C = \Omega(1)$, *any optimal solution* $\boldsymbol{M}^*$ *of the ICL pretraining loss induced by the task distribution* $D(\mathcal{F}_{\boldsymbol{B}}^{lin})$ *optimized over the set* $\mathcal{M} := \{\boldsymbol{M} : \boldsymbol{M} = \boldsymbol{M}^\top, \|\boldsymbol{B}^\top \boldsymbol{M} \boldsymbol{B}\|_2 \leq C\}$ *satisfies, for* $c \in (0, C]$,

$$\boldsymbol{M}^* = c\boldsymbol{B}\boldsymbol{B}^\top.$$

- If $\boldsymbol{s} \in \text{col}(\boldsymbol{B})$ then $\boldsymbol{s}^\top \boldsymbol{M}^* \boldsymbol{s} > 0$
- If $\boldsymbol{s} \in \text{col}(\boldsymbol{B})^\perp$ then $\boldsymbol{s}^\top \boldsymbol{M}^* \boldsymbol{s} = 0$

  $\implies$ $\boldsymbol{M}^*$ learns direction-wise Lipschitzness, recovers $\text{col}(\boldsymbol{B})$

  $\iff$ softmax attn window depends only on projections onto $\text{col}(\boldsymbol{B})$

- To our knowledge, *first result* showing softmax attention learns low-dimensional structure among ICL pretraining tasks.

Conclusion

- Softmax attention learns shared Lipschitzness – both scale and direction – among pretraining tasks that facilitates downstream ICL.
- Future work:
  - Moving beyond ICL-as-regression framework.
    - Auto-regressive pretraining
    - Sequences in which position is relevant to prediction
    - Discrete data
  - Role of MLPs.
  - Multiple attention layers and parallel heads.

References I

Ahn, K., Cheng, X., Daneshmand, H., and Sra, S. (2023).
Transformers learn to implement preconditioned gradient descent for in-context learning.

Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., and Zhou, D. (2022).
What learning algorithm is in-context learning? investigations with linear models.
arXiv preprint arXiv:2211.15661.

Bai, Y., Chen, F., Wang, H., Xiong, C., and Mei, S. (2023).
Transformers as statisticians: Provable in-context learning with in-context algorithm selection.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020).
Language models are few-shot learners.
Advances in neural information processing systems, 33:1877–1901.

Cheng, X., Chen, Y., and Sra, S. (2023).
Transformers implement functional gradient descent to learn non-linear functions in context.
arXiv preprint arXiv:2312.06528.

Fu, D., Chen, T.-Q., Jia, R., and Sharan, V. (2023).
Transformers learn higher-order optimization methods for in-context learning: A study with linear models.
arXiv preprint arXiv:2310.17086.

Garg, S., Tsipras, D., Liang, P. S., and Valiant, G. (2022).
What can transformers learn in-context? a case study of simple function classes.
Advances in Neural Information Processing Systems, 35:30583–30598.

References II

Mahankali, A., Hashimoto, T. B., and Ma, T. (2023).
One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention.
arXiv preprint arXiv:2307.03576.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017).
Attention is all you need.
Advances in neural information processing systems, 30.

von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. (2023).
Transformers learn in-context by gradient descent.
In *International Conference on Machine Learning*, pages 35151–35174. PMLR.

Zhang, Y., Liu, B., Cai, Q., Wang, L., and Wang, Z. (2022).
An analysis of attention via the lens of exchangeability and latent variable models.

Softmax Attention Learns Low-Dimensional Structure in Nonlinear Functions

- We consider generalized linear models (GLMs) with affine, quadratic, and cosine link functions
  - Affine: $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + 2$
  - Quadratic: $f(\boldsymbol{x}) = (\boldsymbol{w}^\top \boldsymbol{x})^2$
  - Cosine: $f(\boldsymbol{x}) = \cos(4\boldsymbol{w}^\top \boldsymbol{x})$