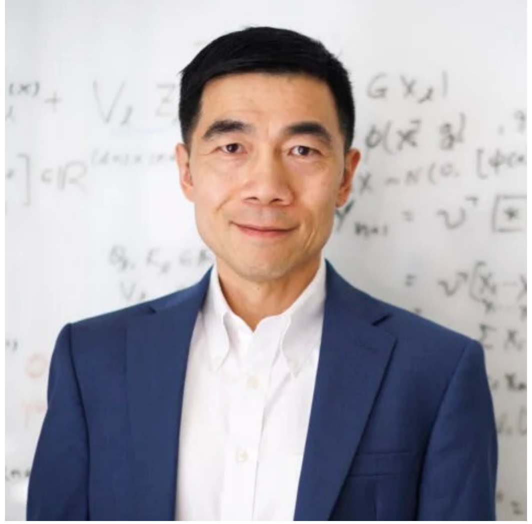
Asymptotic Theory of In-Context Learning by Linear Attention

Mary Letey

Kempner Institute – Spring into Science

May 2024

with …



Yue Lu

Jacob Zavatone-Veth

Anindita Maiti

Cengiz Pehlevan

… and

Kavli Institute for Theoretical Physics

VE RI TAS

Kempner INSTITUTE

Learning In-Context

Neural networks, particularly attention-based architectures, exhibit ability to learn and execute tasks based only on examples seen in input, without needing explicit training.

e.g. translation with example input-output texts provided.*

Hello	nuqneH
Help	QaH
Thank you	tlho’
How much	ar
Warp	vlH
Federation	Dlvl’

input-output examples

query token Earth ??? (tera) **test label**

*Brown, Tom, et al. "Language models are few-shot learners."

Learning In-Context

Neural networks, particularly attention-based architectures, exhibit ability to learn and execute tasks based only on examples seen in input, without needing explicit training.

When does such an ability emerge?

What **algorithm** is learned ICL for solving a task?

What **size** must the **model** have for ICL to emerge?

What **properties** of **data** affect ICL in transformers?

Setup

Model sees context of ℓ input-output pairs*.

$$\{(x_1, f(x_1)), \ldots, (x_\ell, f(x_\ell)), (\underbrace{x_{\ell+1}}_{\text{query token}}, \underbrace{???}_{\text{test label}})\}$$

Predict: test label $= f(x_{\ell+1})$

f changes from context to context

*Srivastava, Aarohi, et al. "Beyond the imitation game: Quantifying and extrapolating the capabilities of language models."
Wei, Jason, et al. "Emergent abilities of large language models."
Olsson, Catherine, et al. "In-context learning and induction heads."
Chan, Stephanie, et al. "Data distributional properties drive emergent in-context learning in transformers."
Reddy, Gautam. "The mechanistic basis of data dependence and abrupt learning in an in-context classification task."
Bai, Yu, et al. "Transformers as statisticians: Provable in-context learning with in-context algorithm selection."
Akyürek, Ekin, et al. "What learning algorithm is in-context learning? investigations with linear models."

Linear Regression

Simplest choice of f for theory = **Linear** function of input tokens!

Model sees context of ℓ input-output pairs $\{(x_1, y_1), \ldots, (x_\ell, y_\ell), (x_{\ell+1}, y_{\ell+1})\}$

query token — test label

context dependent **task vector** $\in \mathbb{R}^d$

where label $y_i = \langle x_i, w \rangle + \epsilon_i$ label noise

token $\in \mathbb{R}^d$

Model

Want to study algorithm learned by attention to solve ICL task.

Simplest model: **linear attention***

$$A(Z) = Z + \frac{1}{\ell}(VZ)(KZ)^\top(QZ)$$

where $Z \in \mathbb{R}^{\text{token size} \times \text{sequence size}}$ holds the input context.

*Wang, Sinong, et al. "Linformer: Self-attention with linear complexity."

Predictor for $y_{\ell+1}$

Chose an embedding* of input context

$$Z = \begin{bmatrix} x_1 & \cdots & x_\ell & x_{\ell+1} \\ y_1 & \cdots & y_\ell & 0 \end{bmatrix} \in \mathbb{R}^{(d+1)\times(\ell+1)}$$

Predicted value of interest is
$\hat{y} = A(Z)_{d+1,\ell+1}$

*Zhang, Ruiqi, Spencer Frei, and Peter L. Bartlett. "Trained transformers learn linear models in-context."
Wu, Jingfeng, et al. "How Many Pretraining Tasks Are Needed for In-Context Learning of Linear Regression?."
Wang, Sinong, et al. "Linformer: Self-attention with linear complexity."

Predictor for $y_{\ell+1}$

Can argue that predictor

$$A(Z)_{d+1,\ell+1} = \hat{y} = \langle \Gamma, H_Z \rangle$$

for **parameters** $\Gamma \in \mathbb{R}^{d\times(d+1)}$ $\qquad \Gamma := v_{22}\left[\frac{1}{d} M_{11}^\top \quad m_{21}\right]$

and **features** $H_Z \in \mathbb{R}^{d\times(d+1)}$ $\qquad H_Z := x_{\ell+1}\left[\frac{d}{\ell}\sum_{i=1}^{\ell} y_i x_i^\top \quad \frac{1}{\ell}\sum_{i=1}^{\ell} y_i^2\right]$

where $V = \begin{bmatrix} V_{11} & v_{12} \\ v_{21}^\top & v_{22} \end{bmatrix}, \quad M = \begin{bmatrix} M_{11} & m_{12} \\ m_{21}^\top & m_{22} \end{bmatrix} := K^\top Q$

Intuition for **learning algorithm**

Recall $\hat{y} = \langle \Gamma, H_Z \rangle$ for parameters $\Gamma := v_{22} \begin{bmatrix} \frac{1}{d} M_{11}^\top & m_{21} \end{bmatrix}$

features $H_Z := x_{\ell+1} \begin{bmatrix} \frac{d}{\ell} \sum_{i=1}^{\ell} y_i x_i^\top & \frac{1}{\ell} \sum_{i=1}^{\ell} y_i^2 \end{bmatrix}$

Approximate features as

$$H_Z \sim x_{\ell+1} w^\top \widehat{C_x}$$

Γ needs to learn to invert covariance of tokens

where $\widehat{C_x}$ is the ℓ-sample estimator for the true covariance of the input tokens.

Pretraining data

Want multiple sample contexts, not just one. $\longrightarrow$ $\{(x_1^\mu, y_1^\mu), \dots, (x_\ell^\mu, y_\ell^\mu), (x_{\ell+1}^\mu, y_{\ell+1}^\mu)\}$

for sample index $\mu = 1, \cdots, n$

Tokens $x_i^\mu \sim \mathcal{N}(0, \frac{1}{d} I_d)$ i.i.d.

Noise $\epsilon_i^\mu \sim \mathcal{N}(0, \rho)$ i.i.d.

Labels $y_i^\mu = \langle x_i^\mu, w^\mu \rangle + \epsilon_i^\mu$

Tasks Each w^μ chosen uniformly from options $\{w_1, \dots, w_k\}$

where $w_j \sim \mathcal{N}(0, I_d)$ i.i.d. for $j = 1, \dots, k$.

Testing data

Want multiple sample contexts, not just one. $\longrightarrow$ $\{(x_1^\mu, y_1^\mu), \ldots, (x_\ell^\mu, y_\ell^\mu), (x_{\ell+1}^\mu, y_{\ell+1}^\mu)\}$

for sample index $\mu = 1, \cdots, n$

Tokens $x_i^\mu \sim \mathcal{N}(0, \frac{1}{d} I_d)$ i.i.d.

Noise $\epsilon_i^\mu \sim \mathcal{N}(0, \rho)$ i.i.d.

Labels $y_i^\mu = \langle x_i^\mu, w^\mu \rangle + \epsilon_i^\mu$

Tasks At **testing** time, resample task for each context fresh from full $\mathcal{N}(0, I_d)$ task distribution.

Result: Asymptotic Learning Curve

Joint **Scaling**:

$$\alpha := \frac{\ell}{d} \qquad \kappa := \frac{k}{d} \qquad \tau := \frac{n}{d^2}$$

Result 1 (ICL generalization error in the ridgeless limit). *Let*

$$q^* := \frac{1+\rho}{\alpha}, \qquad m^* := \mathcal{M}_\kappa\left(q^*\right), \qquad \text{and} \qquad \mu^* := q^* \mathcal{M}_{\kappa/\tau}(q^*),$$

where $\mathcal{M}_\kappa(\cdot)$, defined in (B.3), is a function related to the Stieltjes transform of the Marchenko-Pastur law. Then

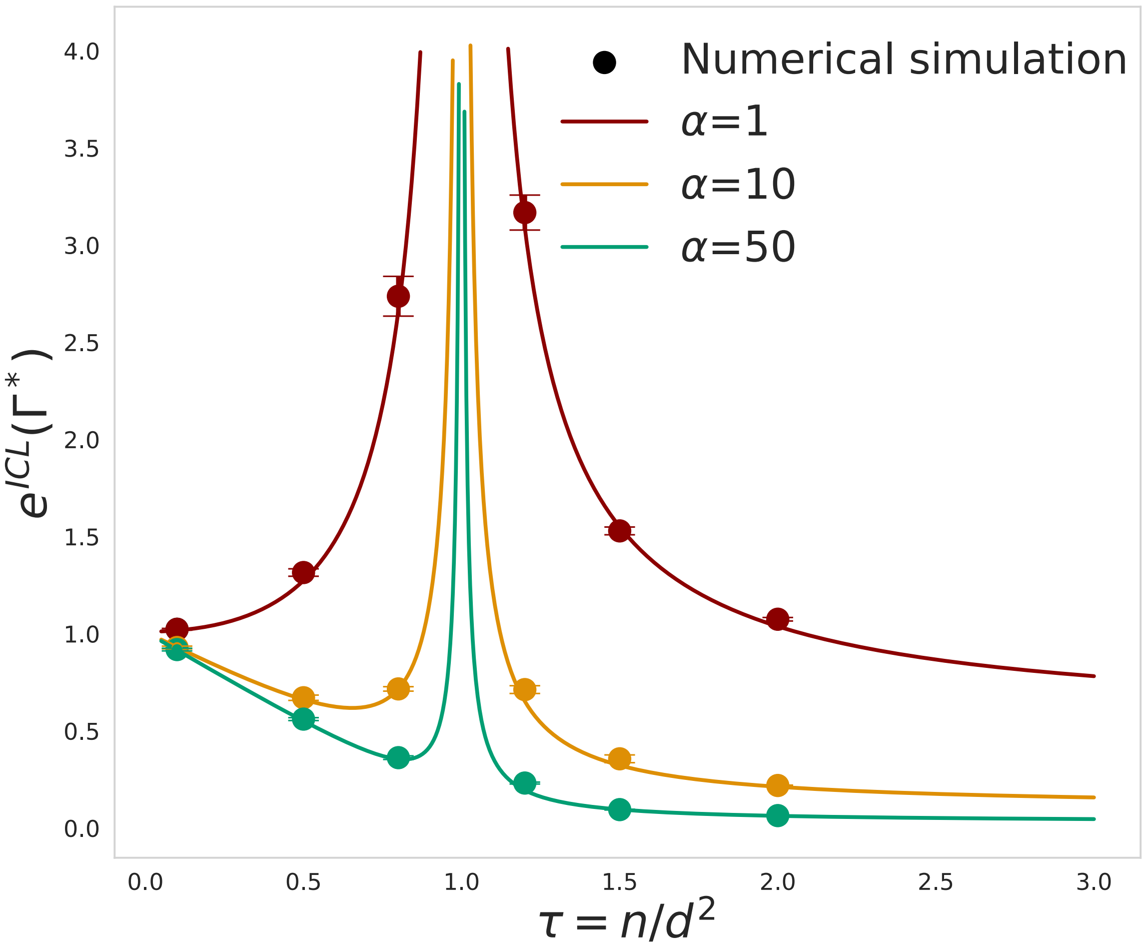
$$e^{\mathrm{ICL}}_{\mathrm{ridgeless}} := \lim_{\lambda \to 0^+} e^{\mathrm{ICL}}(\tau, \alpha, \kappa, \rho, \lambda)$$
$$= \begin{cases} \frac{\tau(1+q^*)}{1-\tau}\left[1 - \tau(1-\mu^*)^2 + \mu^*(\rho/q^* - 1)\right] - 2\tau(1-\mu^*) + (1+\rho) & \tau < 1 \\ (q^*+1)\left(1 - 2q^*m^* - (q^*)^2 \mathcal{M}_\kappa'(q^*) + \frac{(\rho + q^* - (q^*)^2 m^*)m^*}{\tau - 1}\right) - 2(1 - q^*m^*) + (1+\rho) & \tau > 1 \end{cases},$$

where $\mathcal{M}_\kappa'(\cdot)$ denotes the derivative of $\mathcal{M}_\kappa(q)$ with respect to q.

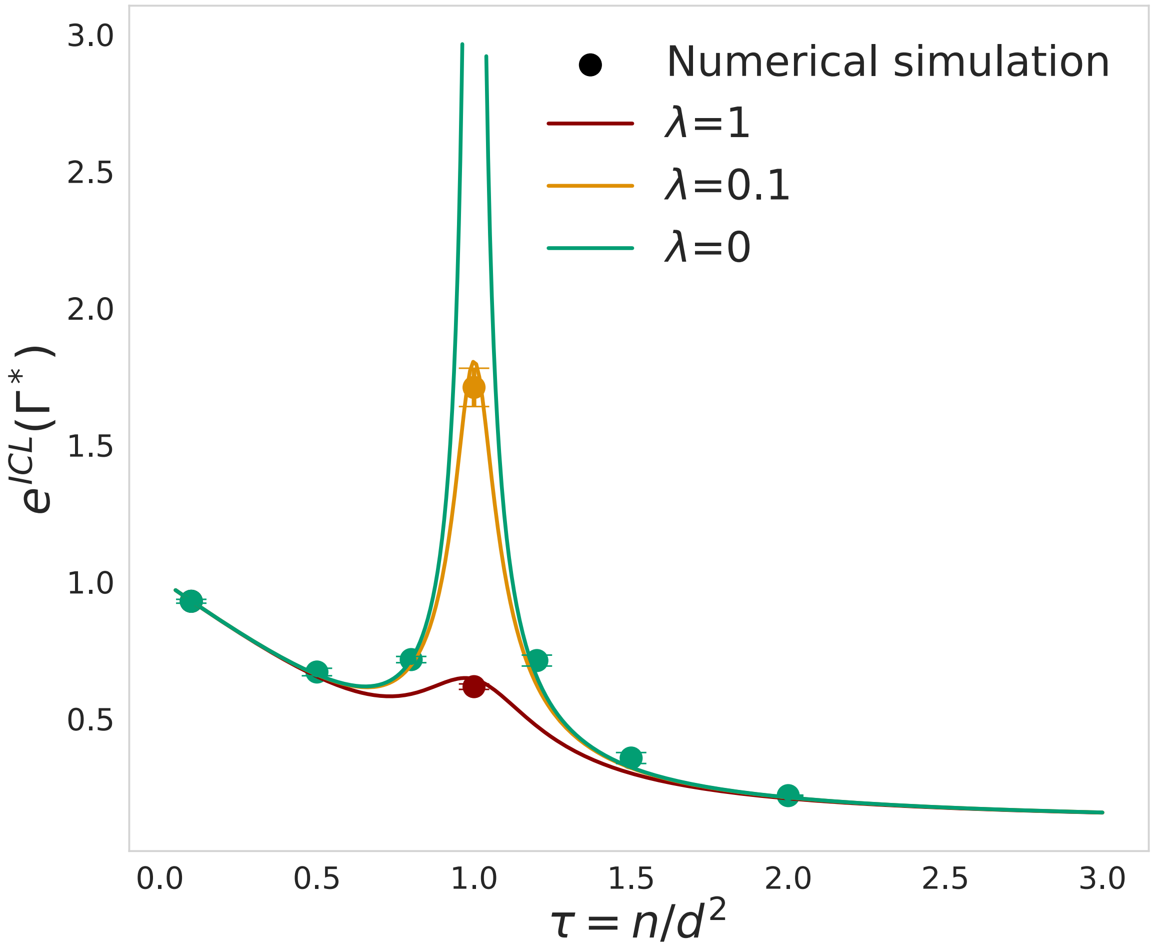
Deterministic formula valid as $d, \ell, k, n \to \infty$ when holding $\alpha, \kappa, \tau = \mathcal{O}(1)$

Result: Sample-wise Double Descent

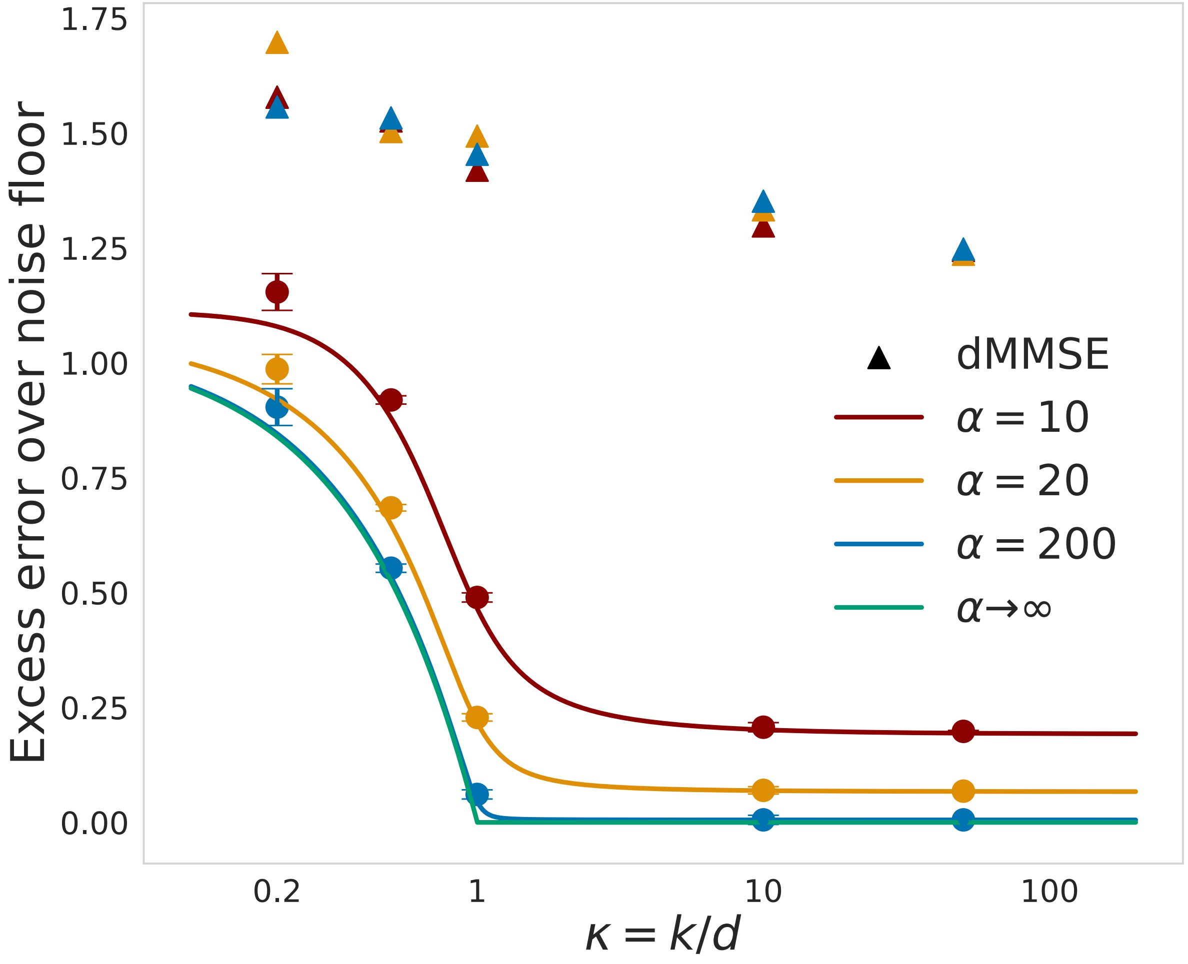
Ridgeless ICL Generalization Error against τ



Finite Ridge ICL Generalization Error against τ



Transition from Memorization to ICL



dMMSE = **'memorisation prior'**
model assumes tasks can only be the ones it has inferred over the training set, i.e. $w_1, \cdots, w_k$.

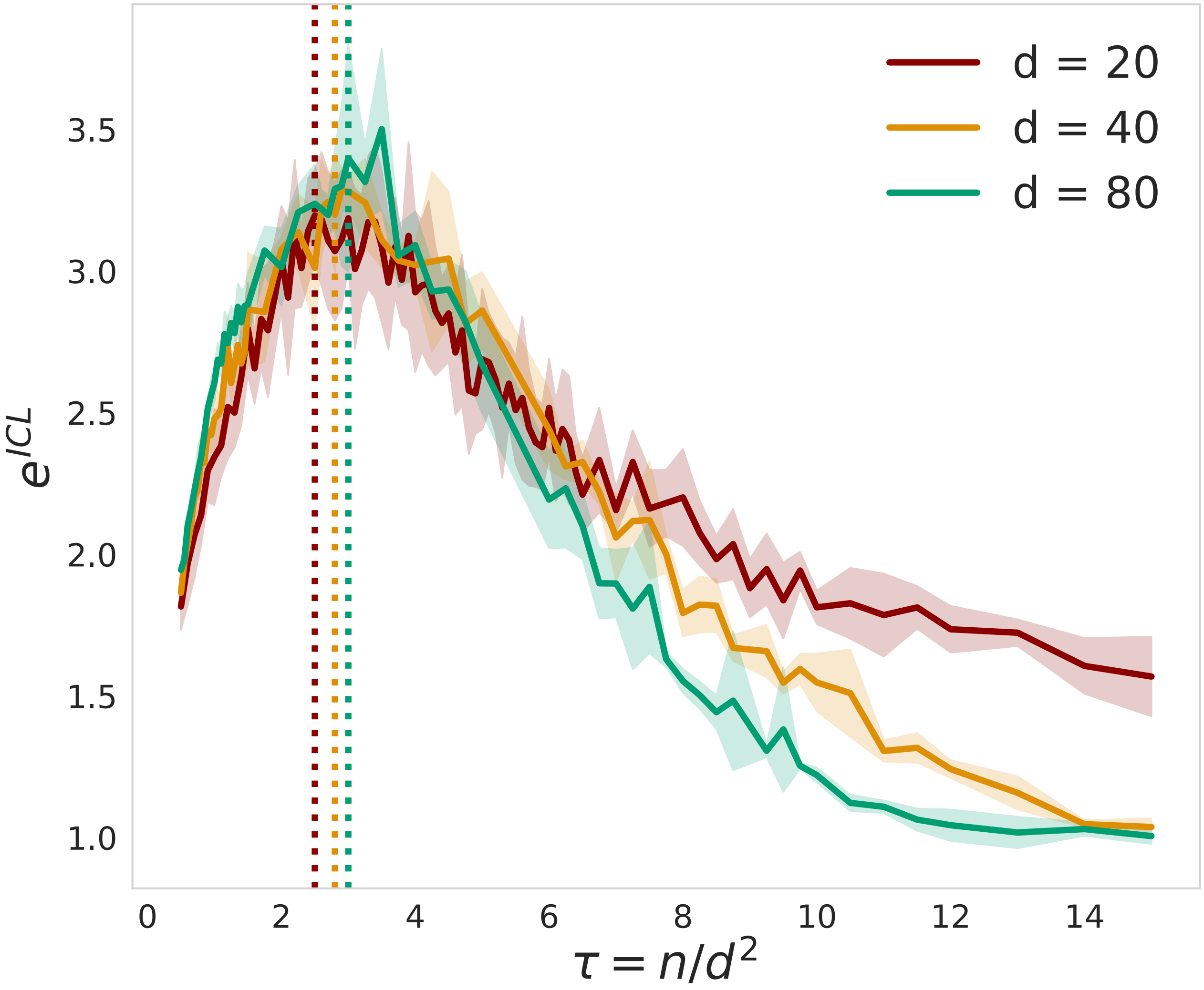
Theory predicts **transition** at $\kappa = 1$

$$\lim_{\alpha \to \infty} e^{ICL} = \begin{cases} \rho + (1-\kappa)\left(1 + \dfrac{\rho}{1+\rho}\dfrac{\tau}{\alpha}\right) & \kappa < 1 \\ \rho & \kappa > 1 \end{cases}$$

Full Transformer: sample-wise double descent

From theory we expect double-descent in number of context samples …

… with scaling of n controlled by τ

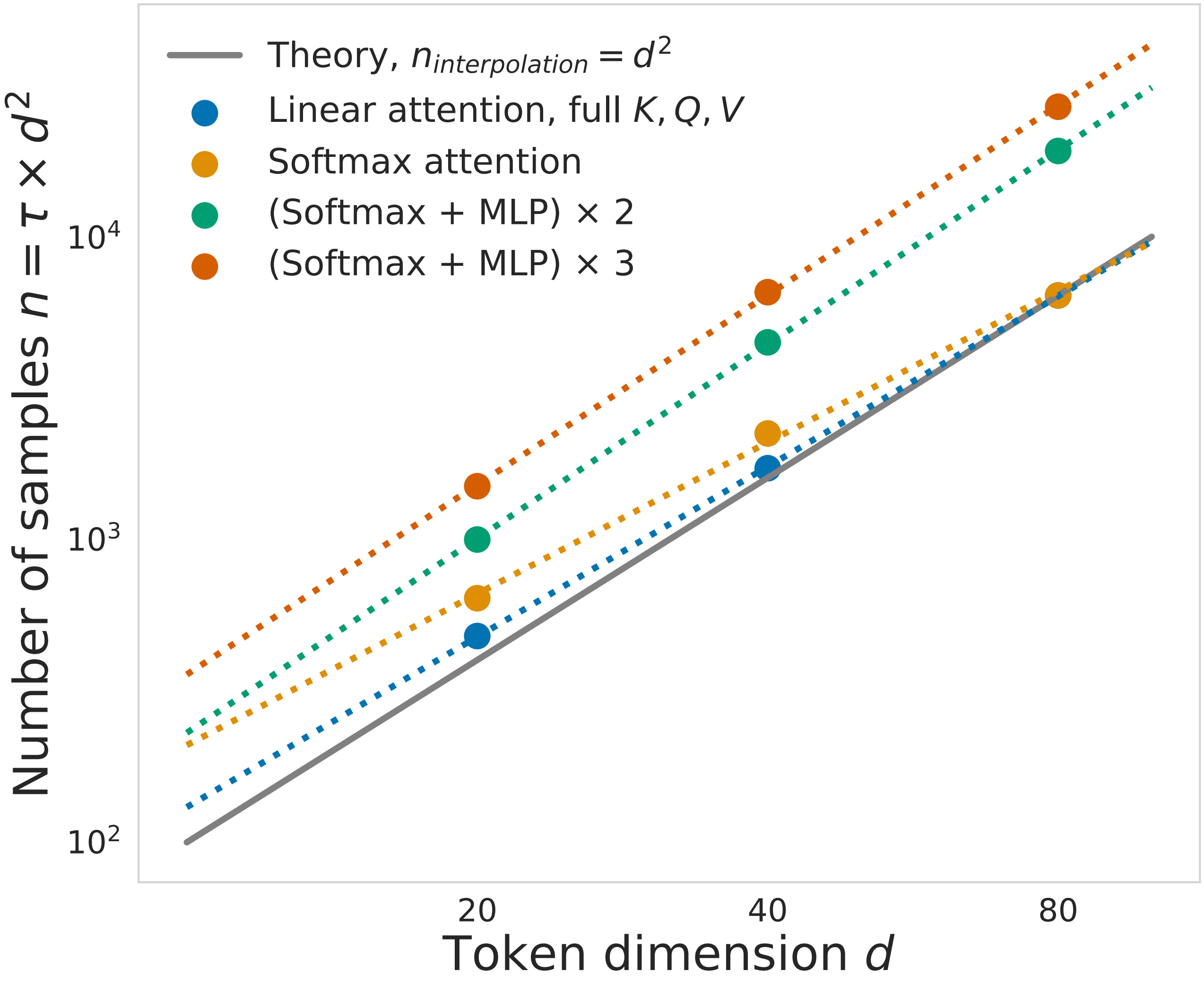


Full Transformer: correct sample scaling

Double descent: where does it happen?

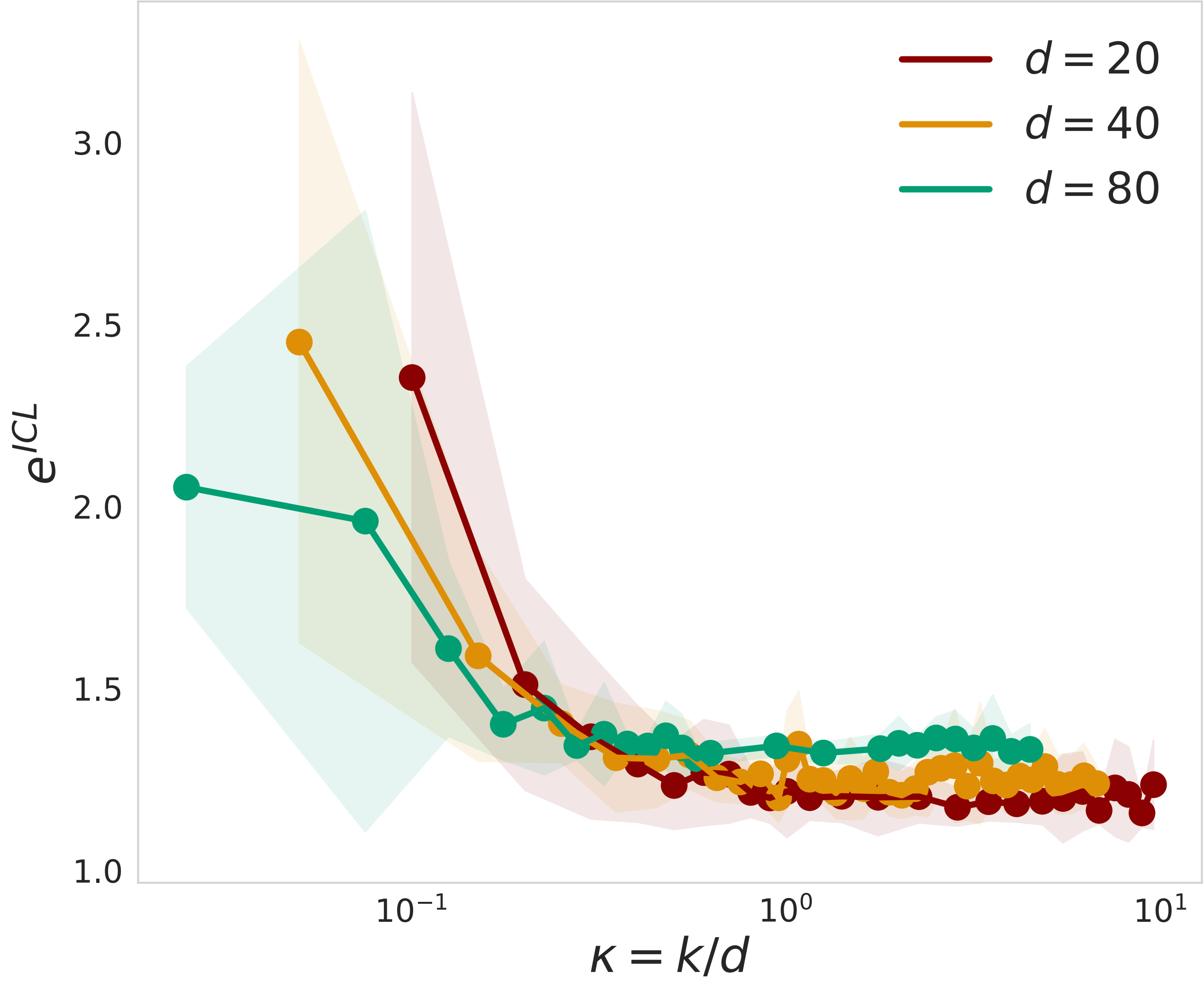
From theory we expect:

$$n_{\text{peak}} = c \cdot d^2$$



Full Transformer: transition in κ

From theory we expect sharp **transition** from memorization to generalization.



Thank You!

Preprint on arxiv

2405.11751