Trained Transformers Learn Linear Models In-Context (2024 JMLR)

Presenter: Gao Zhuo

HSBC Business School, Peking University

2025/08/17

1. Introduction
2. Preliminaries
3. Main results

In-context Learning (ICL)

Prompt

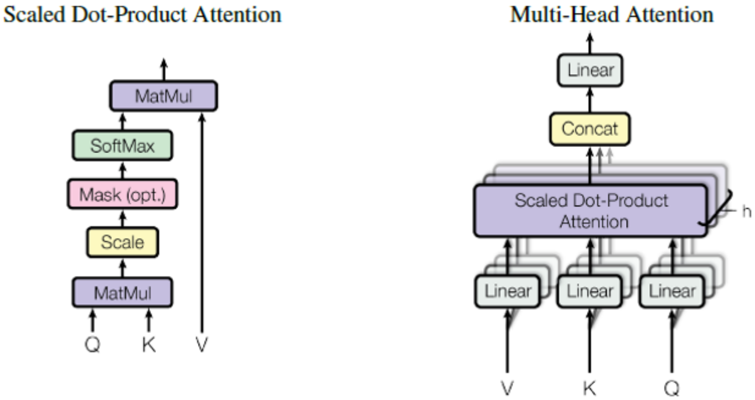
$$P = (x_1, h(x_1), \ldots, x_N, h(x_N), x_{query})$$

where:

- x_i, x_{query} is sampled i.i.d. from a distribution D_x
- h is sampled independently from a distribution over functions in a function class H
- $h(x_i)$ is the output for each input sample data x_i in the prompt

- **ICL:** when given a short sequence of input-output pairs (called a prompt) from a particular task as input, the model can formulate predictions on test examples without having to make any updates to the parameters in the model.

Transformer



- The core part of transformer: self-attention module is shown above. Can we just use this module to perform ICL procedure?

Transformer

Attention + Residual Connection

$$f_{Attn} = E + W^P W^V E \cdot softmax(\frac{(W^K E)^T W^Q E}{\rho})$$

where:

- $E \in R^{d_e, d_N}$ is an embedding matrix formed using a prompt $(x_1, y_1, ..., x_N, y_N, x_{query})$
- One natural way to form E is to stack $(x_i, y_i) \in R^{d+1}$ as the first N columns of E and to let the final column be $(x_{query}, 0) \in R^{d+1}$
- W^K, W^Q, W^V is the key, query, and value weight matrices
- W^P is the projection matrix
- $\rho > 0$ is a normalization factor
- Softmax is applied column-wise and, given a vector input of v, the i-th entry of $softmax(v)$ is given by $exp(v_i) / \sum_s exp(v_s)$

In-context Learning (ICL)

Prompt

$$P = (x_1, h(x_1), \ldots, x_N, h(x_N), x_{query})$$

where:

- x_i, x_{query} is sampled i.i.d. from a distribution D_x
- h is sampled independently from a distribution over functions in a function class H
- $h(x_i)$ is the output for each input sample data x_i in the prompt

- The behavior of the trained transformers can mimic those of familiar learning algorithms like ordinary least squares
- What about the case when the mapping function h is linear? $h(x) = < w, x >$

In-context Learning (ICL)

Prompt in Linear Form

$$P = (x_1, < w, x_1 >, ..., x_N, < w, x_N >, x_{query})$$

where:

- x_i, x_{query} is sampled i.i.d. from a distribution D_x
- w is weighted vector of standard Gaussian distribution

- Transformers based neural network architectures which are capable of achieving small prediction error for query examples
- However,how transformer architectures produce models which are capable of in-context learning?

In-context Learning

Definition 1 (Trained on in-context examples) *Let $\mathcal{D}_x$ be a distribution over an input space $\mathcal{X}$, $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ a set of functions $\mathcal{X} \to \mathcal{Y}$, and $\mathcal{D}_{\mathcal{H}}$ a distribution over functions in $\mathcal{H}$. Let $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a loss function. Let $\mathcal{S} = \cup_{n \in \mathbb{N}} \{(x_1, y_1, \ldots, x_n, y_n) : x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$ be the set of finite-length sequences of (x, y) pairs and let*

$$\mathcal{F}_\Theta = \{f_\theta : \mathcal{S} \times \mathcal{X} \to \mathcal{Y},\ \theta \in \Theta\}$$

be a class of functions parameterized by θ in some set Θ. For $N > 0$, we say that a model $f : \mathcal{S} \times \mathcal{X} \to \mathcal{Y}$ is trained on in-context examples of functions in $\mathcal{H}$ under loss ℓ w.r.t. *$(\mathcal{D}_{\mathcal{H}}, \mathcal{D}_x)$ if $f = f_{\theta^*}$ where $\theta^* \in \Theta$ satisfies*

$$\theta^* \in \operatorname{argmin}_{\theta \in \Theta} \mathbb{E}_{P=(x_1, h(x_1), \ldots, x_N, h(x_N), x_{\mathsf{query}})} \left[\ell \left(f_\theta(P), h(x_{\mathsf{query}}) \right) \right], \tag{1}$$

where $x_i, x_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_x$ and $h \sim \mathcal{D}_{\mathcal{H}}$ are independent. We call N the length of the prompts seen during training.

- After training, we hope that the predictor $f_\theta(P)$ is as close as the true value $h(x_{query})$

In-context Learning

Definition 2 (In-context learning of a hypothesis class) *Let $\mathcal{D}_x$ be a distribution on an input space $\mathcal{X}$, $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ a class of functions $\mathcal{X} \to \mathcal{Y}$, and $\mathcal{D}_{\mathcal{H}}$ a distribution on functions in $\mathcal{H}$. Let $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a loss function. Let $\mathcal{S} = \cup_{n\in\mathbb{N}}\{(x_1, y_1, \ldots, x_n, y_n) : x_i \in \mathcal{X}, y_i \in \mathcal{Y}\}$ be the set of finite-length sequences of (x, y) pairs. We say that a model $f : \mathcal{S} \times \mathcal{X} \to \mathcal{Y}$ defined on prompts of the form $P = (x_1, h(x_1), \ldots, x_M, h(x_M), x_{\mathsf{query}})$* in-context learns a hypothesis class $\mathcal{H}$ under loss ℓ with respect to $(\mathcal{D}_{\mathcal{H}}, \mathcal{D}_x)$ up to error $\eta \in \mathbb{R}$ *if there exists a function $M_{\mathcal{D}_{\mathcal{H}},\mathcal{D}_x}(\varepsilon) : (0, 1) \to \mathbb{N}$ such that for every $\varepsilon \in (0, 1)$, and for every prompt P of length $M \geq M_{\mathcal{D}_{\mathcal{H}},\mathcal{D}_x}(\varepsilon)$,*

$$\mathbb{E}_{P=(x_1,h(x_1),\ldots,x_M,h(x_M),x_{\mathsf{query}})}\left[\ell\Big(f(P), h\left(x_{\mathsf{query}}\right)\Big)\right] \leq \eta + \varepsilon, \tag{2}$$

where the expectation is over the randomness in $x_i, x_{\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathcal{D}_x$ and $h \sim \mathcal{D}_{\mathcal{H}}$.

- i.e., for any level of precision we desire, there can exist a test prompt length that meets our requirements

Single-layer Linear Self-attention

Linear Self-attention (LSA)

$$f_{LSA} = E + W^{PV}E \cdot \frac{E^T W^{KQ} E}{\rho}$$

where:

- $E \in R^{d_e, d_N}$ is an embedding matrix formed using a prompt $(x_1, y_1, ..., x_N, y_N, x_{query})$
- One natural way to form E is to stack $(x_i, y_i) \in R^{d+1}$ as the first N columns of E and to let the final column be $(x_{query}, 0) \in R^{d+1}$
- $\rho > 0$ is a normalization factor
- $W^{KQ} = (W^K)^T W^Q$, $W^{PV} = W^P W^V$
- Softmax is removed

$$E = E(P) = \begin{pmatrix} x_1 & x_2 & \cdots & x_N & x_{\text{query}} \\ y_1 & y_2 & \cdots & y_N & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(N+1)}.$$

Single-layer Linear Self-attention

by the LSA layer, actually only part of W^{PV} and W^{KQ} affect the prediction. To see how, let us denote

$$W^{PV} = \begin{pmatrix} W_{11}^{PV} & w_{12}^{PV} \\ (w_{21}^{PV})^\top & w_{22}^{PV} \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)}, \quad W^{KQ} = \begin{pmatrix} W_{11}^{KQ} & w_{12}^{KQ} \\ (w_{21}^{KQ})^\top & w_{22}^{KQ} \end{pmatrix} \in \mathbb{R}^{(d+1)\times(d+1)}, \tag{5}$$

where $W_{11}^{PV} \in \mathbb{R}^{d\times d}$; $w_{12}^{PV}, w_{21}^{PV} \in \mathbb{R}^d$; $w_{22}^{PV} \in \mathbb{R}$; and $W_{11}^{KQ} \in \mathbb{R}^{d\times d}$; $w_{12}^{KQ}, w_{21}^{KQ} \in \mathbb{R}^d$; $w_{22}^{KQ} \in \mathbb{R}$. Then, the prediction $\widehat{y}_{\mathsf{query}}$ is

$$\widehat{y}_{\mathsf{query}} = \begin{pmatrix} (w_{21}^{PV})^\top & w_{22}^{PV} \end{pmatrix} \cdot \left(\frac{EE^\top}{N}\right) \begin{pmatrix} W_{11}^{KQ} \\ (w_{21}^{KQ})^\top \end{pmatrix} x_{\mathsf{query}}, \tag{6}$$

since only the last row of W^{PV} and the first d columns of W^{KQ} affects the prediction, which means we can simply take all other entries zero in the following sections.

- Since the prediction takes only the right-bottom entry of the token matrix output, we can write $\hat{y}_{query}$ into the above form

Training Procedure

In this work, we will consider the task of in-context learning linear predictors. We will assume training prompts are sampled as follows. Let Λ be a positive definite covariance matrix. Each training prompt, indexed by $\tau \in \mathbb{N}$, takes the form

$$P_\tau = (x_{\tau,1}, h_\tau(x_{\tau_1}), \ldots, x_{\tau,N}, h_\tau(x_{\tau,N}), x_{\tau,\mathsf{query}}),$$

where task weights $w_\tau \overset{\text{i.i.d.}}{\sim} \mathsf{N}(0, I_d)$, inputs $x_{\tau,i}, x_{\tau,\mathsf{query}} \overset{\text{i.i.d.}}{\sim} \mathsf{N}(0, \Lambda)$, and labels $h_\tau(x) = \langle w_\tau, x \rangle$.

Each prompt corresponds to an embedding matrix E_τ, formed using the transformation (4):

$$E_\tau := \begin{pmatrix} x_{\tau,1} & x_{\tau,2} & \cdots & x_{\tau,N} & x_{\tau,\mathsf{query}} \\ \langle w_\tau, x_{\tau,1} \rangle & \langle w_\tau, x_{\tau,2} \rangle & \cdots & \langle w_\tau, x_{\tau,N} \rangle & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(N+1)}.$$

We denote the prediction of the LSA model on the query label in the task τ as $\widehat{y}_{\tau,\mathsf{query}}$, which is the bottom-right element of $f_{\mathsf{LSA}}(E_\tau)$, where f_{LSA} is the linear self-attention model defined in (3). The empirical risk over B independent prompts is defined as

$$\widehat{L}(\theta) = \frac{1}{2B} \sum_{\tau=1}^{B} \left(\widehat{y}_{\tau,\mathsf{query}} - \langle w_\tau, x_{\tau,\mathsf{query}} \rangle \right)^2. \tag{7}$$

We shall consider the behavior of gradient flow-trained networks over the population loss induced by the limit of infinite training tasks/prompts $B \to \infty$:

$$L(\theta) = \lim_{B\to\infty} \widehat{L}(\theta) = \frac{1}{2} \mathbb{E}_{w_\tau, x_{\tau,1}, \cdots, x_{\tau,N}, x_{\tau,\mathsf{query}}} \left[\left(\widehat{y}_{\tau,\mathsf{query}} - \langle w_\tau, x_{\tau,\mathsf{query}} \rangle \right)^2 \right] \tag{8}$$

Convergence

Assumption 3 (Initialization) *Let $\sigma > 0$ be a parameter, and let $\Theta \in \mathbb{R}^{d\times d}$ be any matrix satisfying $\|\Theta\Theta^\top\|_F = 1$ and $\Theta\Lambda \neq 0_{d\times d}$. We assume*

$$W^{PV}(0) = \sigma \begin{pmatrix} 0_{d\times d} & 0_d \\ 0_d^\top & 1 \end{pmatrix}, \quad W^{KQ}(0) = \sigma \begin{pmatrix} \Theta\Theta^\top & 0_d \\ 0_d^\top & 0 \end{pmatrix}. \tag{10}$$

Theorem 4 (Convergence and limits) *Consider gradient flow of a linear self-attention network f_{LSA} defined in (3) over the population loss (8). Suppose the initialization satisfies Assumption 3 with initialization scale $\sigma > 0$ satisfying $\sigma^2 \|\Gamma\|_{op}\sqrt{d} < 2$ where we have defined*

$$\Gamma := \left(1 + \frac{1}{N}\right)\Lambda + \frac{1}{N}\operatorname{tr}(\Lambda) I_d \in \mathbb{R}^{d\times d}.$$

Then gradient flow converges to a global minimum of the population loss (8). Moreover, W^{PV} and W^{KQ} converge to W_*^{PV} and W_*^{KQ} respectively, where

$$W_*^{KQ} = \left[\operatorname{tr}\left(\Gamma^{-2}\right)\right]^{-\frac{1}{4}} \begin{pmatrix} \Gamma^{-1} & 0_d \\ 0_d^\top & 0 \end{pmatrix}, \qquad W_*^{PV} = \left[\operatorname{tr}\left(\Gamma^{-2}\right)\right]^{\frac{1}{4}} \begin{pmatrix} 0_{d\times d} & 0_d \\ 0_d^\top & 1 \end{pmatrix}. \tag{11}$$

Convergence

$$\widehat{y}_{\mathrm{query}} = \begin{pmatrix} 0_d^\top & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{M}\sum_{i=1}^{M} x_i x_i^\top + \frac{1}{M} x_{\mathrm{query}} x_{\mathrm{query}}^\top & \frac{1}{M}\sum_{i=1}^{M} x_i x_i^\top w \\ \frac{1}{M}\sum_{i=1}^{M} w^\top x_i x_i^\top & \frac{1}{M}\sum_{i=1}^{M} w^\top x_i x_i^\top w \end{pmatrix} \begin{pmatrix} \Gamma^{-1} & 0_d \\ 0_d^\top & 0 \end{pmatrix} \begin{pmatrix} x_{\mathrm{query}} \\ 0 \end{pmatrix}$$

$$= x_{\mathrm{query}}^\top \Gamma^{-1} \left(\frac{1}{M} \sum_{i=1}^{M} x_i x_i^\top \right) w. \qquad (12)$$

- Obviously, when N is large enough, $\Gamma \to \Lambda$
- $\sum_{i=1}^{M} \frac{1}{M} x_i x_i^T \to \Lambda$
- $\hat{y}_{query} \to x_{query}^T w$

Convergence

$$\widehat{y}_{\text{query}} = \begin{pmatrix} 0_d^\top & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{M}\sum_{i=1}^M x_i x_i^\top + \frac{1}{M} x_{\text{query}} x_{\text{query}}^\top & \frac{1}{M}\sum_{i=1}^M x_i y_i \\ \frac{1}{M}\sum_{i=1}^M x_i^\top y_i & \frac{1}{M}\sum_{i=1}^M y_i^2 \end{pmatrix} \begin{pmatrix} \Gamma^{-1} & 0_d \\ 0_d^\top & 0 \end{pmatrix} \begin{pmatrix} x_{\text{query}} \\ 0 \end{pmatrix}$$
$$= x_{\text{query}}^\top \Gamma^{-1} \left(\frac{1}{M} \sum_{i=1}^{M} y_i x_i \right). \tag{13}$$

- When h in test prompt is not linear, how's the thing going?
- Linear optimal as well!
- Suppose $(x_i, y_i) \sim D$ and $x_i \sim N(0, \Lambda)$
- $\Lambda^{-1} E_{(x,y \sim D}[xy] = \arg\min_w E[(y - w^T x)^2]$
- Consider why?

Convergence

Theorem 5 *Let $\mathcal{D}$ be a distribution over $(x, y) \in \mathbb{R}^d \times \mathbb{R}$, whose marginal distribution on x is $\mathcal{D}_x = \mathsf{N}(0, \Lambda)$. Assume $\mathbb{E}_{\mathcal{D}}[y], \mathbb{E}_{\mathcal{D}}[xy], \mathbb{E}_{\mathcal{D}}[y^2 xx^\top]$ exist and are finite. Assume the test prompt is of the form $P = (x_1, y_1, \ldots, x_M, y_M, x_{\mathsf{query}})$, where $(x_i, y_i), (x_{\mathsf{query}}, y_{\mathsf{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$. Let f^*_{LSA} be the LSA model with parameters W^{PV}_* and W^{KQ}_* in (11), and $\widehat{y}_{\mathsf{query}}$ is the prediction for x_{query} given the prompt. If we define*

$$a := \Lambda^{-1}\mathbb{E}_{(x,y)\sim\mathcal{D}}[xy], \qquad \Sigma := \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\left(xy - \mathbb{E}(xy)\right)\left(xy - \mathbb{E}(xy)\right)^\top\right], \tag{15}$$

then, for $\Gamma = \Lambda + \frac{1}{N}\Lambda + \frac{1}{N}\operatorname{tr}(\Lambda) I_d$. we have,

$$\mathbb{E}\left(\widehat{y}_{\mathsf{query}} - y_{\mathsf{query}}\right)^2 = \underbrace{\min_{w\in\mathbb{R}^d}\mathbb{E}\left(\langle w, x_{\mathsf{query}}\rangle - y_{\mathsf{query}}\right)^2}_{\textit{Error of best linear predictor}}$$
$$+\boxed{\frac{1}{M}}\operatorname{tr}\left[\Sigma\Gamma^{-2}\Lambda\right] + \boxed{\frac{1}{N^2}}\left[\|a\|^2_{\Gamma^{-2}\Lambda^3} + 2\operatorname{tr}(\Lambda)\|a\|^2_{\Gamma^{-2}\Lambda^2} + \operatorname{tr}(\Lambda)^2\|a\|^2_{\Gamma^{-2}\Lambda}\right], \tag{16}$$

where the expectation is over $(x_i, y_i), (x_{\mathsf{query}}, y_{\mathsf{query}}) \overset{\text{i.i.d.}}{\sim} \mathcal{D}$.

- The prediction error is at most $O(1/M + 1/N^2)$

Convergence

Let us now consider when $\mathcal{D}$ corresponds to noiseless linear models, so that for some $w \in \mathbb{R}^d$, we have $(x, y) = (x, \langle w, x \rangle)$, in which case the prediction of the trained transformer is given by (12). Moreover, a simple calculation shows that the Σ from Theorem 5 takes the form $\Sigma = \|w\|_\Lambda^2 \Lambda + \Lambda w w^\top \Lambda$. Hence Theorem 5 implies the prediction error for the prompt $P = (x_1, \langle w, x_1 \rangle, \ldots, x_M, \langle w, x_M \rangle, x_{\mathsf{query}})$ is

$$\begin{aligned}
&\mathbb{E}_{x_1,\ldots,x_M,x_{\mathsf{query}}} \left(\widehat{y}_{\mathsf{query}} - \langle w, x_{\mathsf{query}} \rangle\right)^2 \\
&= \frac{1}{M} \left\{ \|w\|_{\Gamma^{-2}\Lambda^3}^2 + \mathrm{tr}(\Gamma^{-2}\Lambda^2) \|w\|_\Lambda^2 \right\} \\
&\quad + \frac{1}{N^2} \left\{ \|w\|_{\Gamma^{-2}\Lambda^3}^2 + 2 \|w\|_{\Gamma^{-2}\Lambda^2}^2 \mathrm{tr}(\Lambda) + \|w\|_{\Gamma^{-2}\Lambda}^2 \mathrm{tr}(\Lambda)^2 \right\} \\
&\le \frac{d+1}{M} \|w\|_\Lambda^2 + \frac{1}{N^2} \left[\|w\|_\Lambda^2 + 2 \|w\|_2^2 \mathrm{tr}(\Lambda) + \|w\|_{\Lambda^{-1}}^2 \mathrm{tr}(\Lambda)^2 \right].
\end{aligned}$$

The inequality above uses that $\Gamma \succ \Lambda$. Finally, if we assume that $w \sim \mathsf{N}(0, I_d)$ and denote κ as the condition number of Λ, then by taking expectations over w we get the following:

$$\begin{aligned}
&\mathbb{E}_{x_1,\ldots,x_M,x_{\mathsf{query}},w} \left(\widehat{y}_{\mathsf{query}} - \langle w, x_{\mathsf{query}} \rangle\right)^2 \\
&\le \frac{(d+1)\,\mathrm{tr}(\Lambda)}{M} + \frac{1}{N^2} \left[\mathrm{tr}(\Lambda) + 2d\,\mathrm{tr}(\Lambda) + \mathrm{tr}(\Lambda^{-1})\,\mathrm{tr}(\Lambda)^2 \right] \\
&\le \frac{(d+1)\,\mathrm{tr}(\Lambda)}{M} + \frac{(1 + 2d + d^2\kappa)\,\mathrm{tr}(\Lambda)}{N^2},
\end{aligned}$$

- $tr(\Lambda)$, condition number($\kappa = \frac{\lambda_{max}}{\lambda_{min}}$), covariate dimension d can also affect the precision of prediction

Prompts with Random Covariate Distributions

Covariate Shifts

When $D_x^{train} \neq D_x^{test}$, the approximation does not hold when M and N are large. For instance, if we consider test prompts where the covariates are scaled by a constant $c \neq 1$:

$$\hat{y}_{query} \to x_{query}^T \Lambda^{-1} c^2 \Lambda w \neq x_{query}^T w$$

- To further research on this problem, we set the covariate matrix Λ to random
- Then, for each task τ and coordinate $i \in [d]$, we sample $\lambda_{\tau,i}$ independently such that the distribution of each $\lambda_{\tau,i}$ is fixed and has finite third moments and is strictly positive almost surely. We then form a diagonal matrix:

$$\Lambda_\tau = diag(\lambda_{\tau,1}, ..., \lambda_{\tau,d})$$

Prompts with Random Covariate Distributions

Theorem 8 (Global convergence with random covariance) *Consider gradient flow of the linear self-attention network f_{LSA} defined in (3) over the population loss (20), where Λ_τ are diagonal with independent diagonal entries which are strictly positive a.s. and have finite third moments. Suppose the initialization satisfies Assumption 3, $\|\mathbb{E}\Lambda_\tau\Theta\|_F \neq 0$, with initialization scale $\sigma > 0$ satisfying*

$$\sigma^2 < \frac{2\left\|\mathbb{E}\Lambda_\tau\Theta\right\|_F^2}{\sqrt{d}\left[\mathbb{E}\left\|\Gamma_\tau\right\|_{op}\left\|\Lambda_\tau\right\|_F^2\right]}. \tag{21}$$

Then gradient flow converges to a global minimum of the population loss (20). Moreover, W^{PV} and W^{KQ} converge to W_*^{PV} and W_*^{KQ} respectively, where

$$\begin{aligned} W_*^{KQ} &= \left\|\left[\mathbb{E}\Gamma_\tau\Lambda_\tau^2\right]^{-1}\mathbb{E}\left[\Lambda_\tau^2\right]\right\|_F^{-\frac{1}{2}} \cdot \begin{pmatrix} \left[\mathbb{E}\Gamma_\tau\Lambda_\tau^2\right]^{-1}\left[\mathbb{E}\Lambda_\tau^2\right] & 0_d \\ 0_d^\top & 0 \end{pmatrix}, \\ W_*^{PV} &= \left\|\left[\mathbb{E}\Gamma_\tau\Lambda_\tau^2\right]^{-1}\mathbb{E}\left[\Lambda_\tau^2\right]\right\|_F^{\frac{1}{2}} \cdot \begin{pmatrix} 0_{d\times d} & 0_d \\ 0_d^\top & 1 \end{pmatrix}, \end{aligned} \tag{22}$$

where $\Gamma_\tau = \frac{N+1}{N}\Lambda_\tau + \frac{1}{N}\operatorname{tr}(\Lambda_\tau)I_d \in \mathbb{R}^{d\times d}$ and the expectations above are over the distribution of Λ_τ.

Prompts with Random Covariate Distributions

$$
\begin{aligned}
&\widehat{y}_{\mathsf{query}} \\
&= \begin{pmatrix} 0_d^\top & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{M}\sum_{i=1}^{M} x_i x_i^\top + \frac{1}{M} x_{\mathsf{query}} x_{\mathsf{query}}^\top & \frac{1}{M}\sum_{i=1}^{M} x_i y_i \\ \frac{1}{M}\sum_{i=1}^{M} x_i^\top y_i & \frac{1}{M}\sum_{i=1}^{M} y_i^2 \end{pmatrix} \begin{pmatrix} \left[\mathbb{E}\Gamma_\tau \Lambda_\tau^2\right]^{-1} \mathbb{E}\Lambda_\tau^2 & 0_d \\ 0_d^\top & 0 \end{pmatrix} \begin{pmatrix} x_{\mathsf{query}} \\ 0 \end{pmatrix} \\
&= x_{\mathsf{query}}^\top \cdot \left[\mathbb{E}\Lambda_\tau^2\right] \left[\mathbb{E}\Gamma_\tau \Lambda_\tau^2\right]^{-1} \cdot \left[\frac{1}{M}\sum_{i=1}^{M} x_i x_i^\top\right] w \\
&\to x_{\mathsf{query}}^\top \cdot \left[\mathbb{E}\Lambda_\tau^2\right] \left[\mathbb{E}\Gamma_\tau \Lambda_\tau^2\right]^{-1} \cdot \Lambda_{\mathsf{new}} w \quad \text{almost surely when } M \to \infty. \qquad (23)
\end{aligned}
$$

- One clear example is: considering $\lambda_{\tau,i} \sim Exp(1)$ and $\Gamma_\tau \to \Lambda_\tau$, then $E(\Lambda_\tau) = 1$, $E(\Lambda_\tau^2) = 2$, $E(\Lambda_\tau^3) = 6$, as a result:

$$E[\hat{y}_{query} | x_{query}, w] \to \frac{1}{3} w^T x_{query}$$

Experiments with Large, Nonlinear Transformers