How Transformers Perform PCA and Sparse Recovery

Qinyan Liu

July 6, 2025

Overview

Encoder-based vs Decoder-based Transformers

Aspect	**Encoder-based Transformer**	**Decoder-based Transformer**
Primary Use	Understanding	Generation
Attention Type	Bidirectional	Causal (unidirectional)
Masking	No mask (full context)	Causal mask (only past tokens visible)
Input	Full sequence at once	Left-to-right, token by token
Training Objective	Masked Language Modeling (MLM)	Autoregressive Language Modeling
Example Models	BERT, RoBERTa	GPT, ChatGPT
Output	Contextual embeddings	Predicted next tokens

Attention layer

Definition 1

(Attention layer). A *(self-)attention layer with M heads is denoted as* $\mathrm{Attn}_{\boldsymbol{\theta}}(\cdot)$ *with parameters* $\boldsymbol{\theta} = \{(\mathbf{V}_m, \mathbf{Q}_m, \mathbf{K}_m)\}_{m\in[M]} \subset \mathbb{R}^{D\times D}$. On any input sequence $\mathbf{H} \in \mathbb{R}^{D\times N}$,

$$\widetilde{\mathbf{H}} = \mathrm{Attn}_{\boldsymbol{\theta}}(\mathbf{H}) := \mathbf{H} + \frac{1}{N}\sum_{m=1}^{M}\Big(\mathbf{V}_m\mathbf{H}\cdot\sigma\Big((\mathbf{Q}_m\mathbf{H})^\top(\mathbf{K}_m\mathbf{H})\Big)\Big) \in \mathbb{R}^{D\times N},$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is the ReLU function. In vector form,

$$\widetilde{\mathbf{h}}_i = [\mathrm{Attn}_{\boldsymbol{\theta}}(\mathbf{H})]_i = \mathbf{h}_i + \sum_{m=1}^{M}\frac{1}{N}\sum_{j=1}^{N}\sigma\left(\langle\mathbf{Q}_m\mathbf{h}_i, \mathbf{K}_m\mathbf{h}_j\rangle\right)\cdot\mathbf{V}_m\mathbf{h}_j.$$

MLP Layer

Definition 2

(MLP layer). *A (token-wise) MLP layer with hidden dimension D' is denoted as*

$$\mathrm{MLP}_{\boldsymbol{\theta}}(\cdot) \text{ with parameters } \boldsymbol{\theta} = (\mathbf{W}_1, \mathbf{W}_2) \in \mathbb{R}^{D' \times D} \times \mathbb{R}^{D \times D'}.$$

On any input sequence $\mathbf{H} \in \mathbb{R}^{D \times N}$,

$$\widetilde{\mathbf{H}} = \mathrm{MLP}_{\boldsymbol{\theta}}(\mathbf{H}) := \mathbf{H} + \mathbf{W}_2 \sigma(\mathbf{W}_1 \mathbf{H}),$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is the ReLU function. In vector form, we have

$$\widetilde{\mathbf{h}}_i = \mathbf{h}_i + \mathbf{W}_2 \sigma(\mathbf{W}_1 \mathbf{h}_i).$$

Definition 3

(Transformer). *An L-layer Transformer, denoted as $\mathrm{TF}_{\boldsymbol{\theta}}(\cdot)$, is a composition of L self-attention layers each followed by an MLP layer:*

$$\mathbf{H}^{(L)} = \mathrm{TF}_{\boldsymbol{\theta}}(\mathbf{H}^{(0)}), \quad \text{where } \mathbf{H} = \mathbf{H}^{(0)} \in \mathbb{R}^{D\times N}$$

is the input sequence, and

$$\mathbf{H}^{(\ell)} = \mathrm{MLP}_{\boldsymbol{\theta}^{(\ell)}_{\mathrm{mlp}}}\left(\mathrm{Attn}_{\boldsymbol{\theta}^{(\ell)}_{\mathrm{attn}}}(\mathbf{H}^{(\ell-1)})\right), \quad \ell \in \{1,\dots,L\}.$$

Above, the parameter $\boldsymbol{\theta} = (\boldsymbol{\theta}^{(1:L)}_{\mathrm{attn}}, \boldsymbol{\theta}^{(1:L)}_{\mathrm{mlp}})$ consists of the attention layers $\boldsymbol{\theta}^{(\ell)}_{\mathrm{attn}} = \{(\mathbf{V}^{(\ell)}_m, \mathbf{Q}^{(\ell)}_m, \mathbf{K}^{(\ell)}_m)\}_{m\in[M]} \subset \mathbb{R}^{D\times D}$ and the MLP layers $\boldsymbol{\theta}^{(\ell)}_{\mathrm{mlp}} = (\mathbf{W}^{(\ell)}_1, \mathbf{W}^{(\ell)}_2) \in \mathbb{R}^{D\times D'} \times \mathbb{R}^{D'\times D}$.

Optional: Dimensional Adjustment

$$TF_{\boldsymbol{\theta}}(\mathbf{H}) := \widetilde{\mathbf{W_0}} \times \mathrm{MLP}_{\boldsymbol{\theta}_{\mathrm{mlp}}^{(\ell)}} \left(\mathrm{Attn}_{\boldsymbol{\theta}_{\mathrm{attn}}^{(\ell)}} (\mathbf{H}^{(\ell-1)}) \right) \times \widetilde{\mathbf{W_1}}, \quad \ell \in \{1, \ldots, L\}$$

The two additional matrices $\widetilde{\mathbf{W_0}} \in \mathbb{R}^{d_1 \times D}$ and $\widetilde{\mathbf{W_1}} \in \mathbb{R}^{N \times d_2}$ serve for the dimension adjustment purpose such that the output of $TF_{\boldsymbol{\theta}}()$ will be of dimension $\mathbb{R}^{d_1 \times d_2}$.

Norm of Parameters

We additionally define the following norm of a Transformer $\mathrm{TF}_{\boldsymbol{\theta}}$:

$$\|\boldsymbol{\theta}\|_{op} :=$$

$$\max_{\ell\in[L]}\left\{\max_{m\in[M]}\left\{\|\mathbf{Q}_m^{(\ell)}\|_{\mathrm{op}}, \|\mathbf{K}_m^{(\ell)}\|_{\mathrm{op}}\right\} + \sum_{m=1}^{M}\|\mathbf{V}_m^{(\ell)}\|_{\mathrm{op}} + \|\mathbf{W}_1^{(\ell)}\|_{\mathrm{op}} + \|\mathbf{W}_2^{(\ell)}\|_{\mathrm{op}}\right\}$$

We can prove that Transformer is Lipschitz continous to this norm when inputs are bounded.

Input Matrix

Input:

$$\mathbf{H} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_N & \mathbf{x}_{N+1} \\ y_1 & y_2 & \cdots & y_N & 0 \\ \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_N & \mathbf{p}_{N+1} \end{bmatrix} \in \mathbb{R}^{D\times(N+1)}, \quad \mathbf{p}_i := \begin{bmatrix} \mathbf{0}_{D-(d+3)} \\ 1 \\ 1\{i < N+1\} \end{bmatrix} \in$$

$\{\mathbf{p}_i\}$ are fixed vectors consisting of ones, zeros, and indicator for being the trained token (similar to a positional encoding vector). We assume

$$\|\mathbf{x}_i\|_2 \leq B_x, |y_i| \leq B_y, a.s.$$

Input Matrix

Input:

$$\mathbf{H} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_N & \mathbf{x}_{N+1} \\ y_1 & y_2 & \cdots & y_N & 0 \\ \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_N & \mathbf{p}_{N+1} \end{bmatrix} \in \mathbb{R}^{D\times(N+1)}, \quad \mathbf{p}_i := \begin{bmatrix} \mathbf{0}_{D-(d+3)} \\ 1 \\ 1\{i < N+1\} \end{bmatrix} \in$$

$\{\mathbf{p}_i\}$ are fixed vectors consisting of ones, zeros, and indicator for being the trained token (similar to a positional encoding vector). We assume

$$\|\mathbf{x}_i\|_2 \leq B_x, |y_i| \leq B_y, a.s.$$

Output:

$$\widetilde{\mathbf{H}} = \mathsf{TF}_\theta(\boldsymbol{H})$$

$$\hat{y}_{N+1} = \widetilde{\mathrm{read}}_y(\widetilde{\mathbf{H}}) := \mathrm{clip}_R\left(\left(\widetilde{\mathbf{h}}_{N+1}\right)_{d+1}\right)$$

Masked Attention layer

Decoder TFs are the same as encoder TFs, except that the attention layers are replaced by masked attention layers with a specific decoder-based (causal) attention mask.

Definition 4

Masked Attention Layer *A masked attention layer with* M *heads is denoted as* $\mathrm{MAttn}_{\boldsymbol{\theta}}(\cdot)$ *with parameters* $\boldsymbol{\theta} = \left\{(\boldsymbol{V}_m, \boldsymbol{Q}_m, \boldsymbol{K}_m) \in (\mathbb{R}^{D\times D})^3\right\}_{m=1}^{M}$. *On any input sequence* $\boldsymbol{H} \in \mathbb{R}^{D\times N'}$ with $N' \le N$,

$$\widetilde{\boldsymbol{H}} = \mathrm{MAttn}_{\boldsymbol{\theta}}(\boldsymbol{H}) :=$$
$$\boldsymbol{H} + \sum_{m=1}^{M} (\boldsymbol{V_m H}) \times \Big((MSK_{1:N',1:N'}) \circ \sigma(\boldsymbol{Q_m H})^\top (\boldsymbol{K_m H}) \Big) \in \mathbb{R}^{D\times N'},$$

Masked Attention Layer

In the definition, $\circ$ denotes the entry-wise (Hadamard) product of two matrices, and $\mathrm{MSK} \in \mathbb{R}^{N \times N}$ is the mask matrix given by

$$\mathrm{MSK} = \begin{bmatrix} 1 & 1/2 & 1/3 & \cdots & 1/N \\ 0 & 1/2 & 1/3 & \cdots & 1/N \\ 0 & 0 & 1/3 & \cdots & 1/N \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 1/N \end{bmatrix}.$$

In vector form, we have

$$\widetilde{\boldsymbol{h}}_i = [\mathrm{Attn}_{\boldsymbol{\theta}}(\boldsymbol{H})]_i = \boldsymbol{h}_i + \sum_{m=1}^{M} \frac{1}{i} \sum_{j=1}^{i} \sigma(\langle \boldsymbol{Q}_m \boldsymbol{h}_i, \boldsymbol{K}_m \boldsymbol{h}_j \rangle) \cdot \boldsymbol{V}_m \boldsymbol{h}_j.$$

Decoder-based Transformer

Definition 5

(Decoder-based Transformer). *An L-layer decoder-based Transformer, denoted as $\mathrm{DTF}_{\boldsymbol{\theta}}(\cdot)$, is a composition of L self-attention layers each followed by an MLP layer:*

$$\mathbf{H}^{(L)} = \mathrm{DTF}_{\boldsymbol{\theta}}(\mathbf{H}^{(0)}), \quad \text{where } \mathbf{H} = \mathbf{H}^{(0)} \in \mathbb{R}^{D\times N}$$

is the input sequence, and

$$\mathbf{H}^{(\ell)} = \mathrm{MLP}_{\boldsymbol{\theta}^{(\ell)}_{\mathrm{mlp}}}\left(\mathrm{MAttn}_{\boldsymbol{\theta}^{(\ell)}_{\mathrm{mattn}}}(\mathbf{H}^{(\ell-1)})\right), \quad \ell \in \{1,\dots,L\}.$$

Above, the parameter $\boldsymbol{\theta} = (\boldsymbol{\theta}^{(1:L)}_{\mathrm{mattn}}, \boldsymbol{\theta}^{(1:L)}_{\mathrm{mlp}})$ consists of the attention layers $\boldsymbol{\theta}^{(\ell)}_{\mathrm{mattn}} = \{(\mathbf{V}^{(\ell)}_m, \mathbf{Q}^{(\ell)}_m, \mathbf{K}^{(\ell)}_m)\}_{m\in[M]} \subset \mathbb{R}^{D\times D}$ and the MLP layers $\boldsymbol{\theta}^{(\ell)}_{\mathrm{mlp}} = (\mathbf{W}^{(\ell)}_1, \mathbf{W}^{(\ell)}_2) \in \mathbb{R}^{D\times D'} \times \mathbb{R}^{D'\times D}$.

Input Matrix

Input:

$$\mathbf{H} = \begin{bmatrix} \boldsymbol{x}_1 & \mathbf{0} & \dots & \boldsymbol{x}_N & \mathbf{0} & \boldsymbol{x}_{N+1} \\ 0 & y_1 & \dots & 0 & y_N & 0 \\ \boldsymbol{p}_1 & \boldsymbol{p}_2 & \dots & \boldsymbol{p}_{2N-1} & \boldsymbol{p}_{2N} & \boldsymbol{p}_{2N+1} \end{bmatrix} \in \mathbb{R}^{D\times(2N+1)},$$

$$\boldsymbol{p}_i := \begin{bmatrix} \mathbf{0}_{D-(d+4)} \\ \lceil i/2 \rceil \\ 1 \\ \mathrm{mod}(i+1, 2) \end{bmatrix} \in \mathbb{R}^{D-(d+1)}$$

Input Matrix

Input:

$$\mathbf{H} = \begin{bmatrix} \boldsymbol{x}_1 & \mathbf{0} & \dots & \boldsymbol{x}_N & \mathbf{0} & \boldsymbol{x}_{N+1} \\ 0 & y_1 & \dots & 0 & y_N & 0 \\ \boldsymbol{p}_1 & \boldsymbol{p}_2 & \dots & \boldsymbol{p}_{2N-1} & \boldsymbol{p}_{2N} & \boldsymbol{p}_{2N+1} \end{bmatrix} \in \mathbb{R}^{D\times(2N+1)},$$

$$\boldsymbol{p}_i := \begin{bmatrix} \mathbf{0}_{D-(d+4)} \\ \lceil i/2 \rceil \\ 1 \\ \mathrm{mod}(i+1, 2) \end{bmatrix} \in \mathbb{R}^{D-(d+1)}$$

Output:

$$\widetilde{\mathbf{H}} = \mathsf{DTF}_\theta(\boldsymbol{H})$$

$$\hat{y}_{N+1} = \widetilde{\mathrm{read}}_y(\widetilde{\mathbf{H}}) := \mathrm{clip}_R\left(\left(\widetilde{\mathbf{h}}_{2N+1}\right)_{d+1}\right)$$

Comparsion of Input Matrix

- The input format for decoder-based Transformer is different from the input format for encoder-based Transformers.
- The main difference is that $(\boldsymbol{x}_i, y_i)$ are in different tokens in the former, whereas $(_i, y_i)$ are in the same token in the latter.
- The reason for the former (i.e., different tokens in decoder) is that we want to avoid every $[\boldsymbol{x}_i; 0]$ token seeing the information of y_i, since we will evaluate the loss at every token.
- The reason for the latter (i.e., the same token in encoder) is for presentation convenience: since we only evaluate the loss at the last token, it is not necessary to alternate between $[\boldsymbol{x}_i; 0]$ and $[\mathbf{0}; y_i]$ to avoid information leakage.

Input Format Conversion

(Proof to be added)

Proposition 6

Input format conversion There exists a 2-layer decoder TF with 3 heads per layer, hidden dimension 2 and $\|\boldsymbol{\theta}|_2 \leq 12$ such that upon taking input $\boldsymbol{H}$ of format for decoder-based Transformers, it outputs $\widetilde{\boldsymbol{H}} = \mathrm{DTF}(\boldsymbol{H})$ with

$$\widetilde{\mathbf{H}} = \begin{bmatrix} \boldsymbol{x}_1 & \mathbf{0} & \dots & \boldsymbol{x}_N & \mathbf{0} & \boldsymbol{x}_{N+1} \\ 0 & y_1 & \dots & 0 & y_N & 0 \\ \boldsymbol{p}_1 & \boldsymbol{p}_2 & \dots & \boldsymbol{p}_{2N-1} & \boldsymbol{p}_{2N} & \boldsymbol{p}_{2N+1} \end{bmatrix} \in \mathbb{R}^{D \times (2N+1)},$$

In particular, this format contains the format for encoders as a submatrix, by restricting to the $\{1, 2, \dots, D-1, D-2, D\}$ rows and $\{2, 4, \dots, 2N-2, 2N, 2N+1\}$ columns.

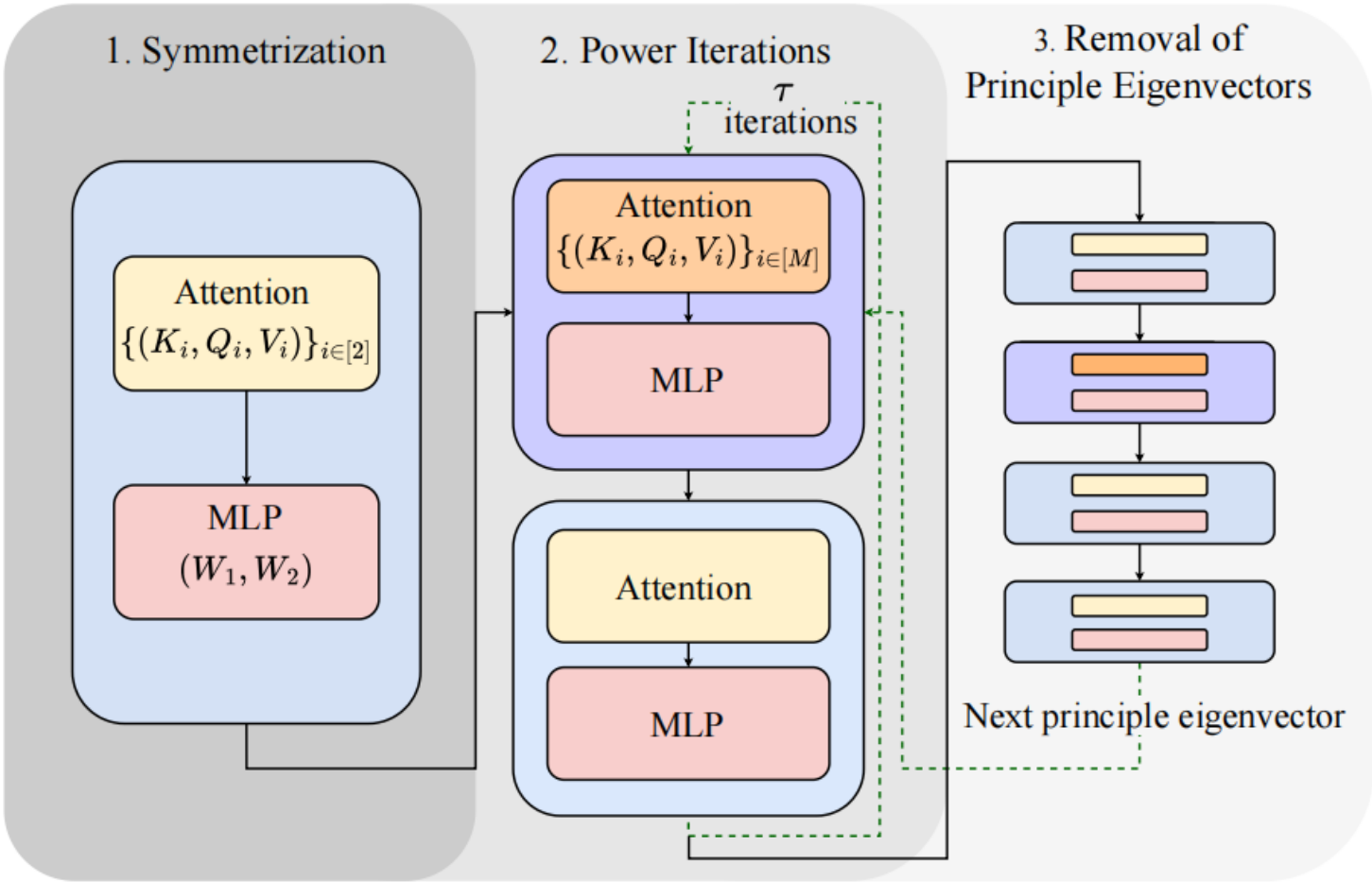
Overview

Power Method

Algorithm 1: Power Method for the Left Singular Vectors

Data: Matrix $\boldsymbol{X} \in \mathbb{R}^{d \times N}$, Number of Iterations τ
Symmertize $A = XX^\top \in \mathbb{R}^{d \times d}$;
Let the set of eigenvectors be $\mathcal{V} = \{\}$. Initialize $A_1 \leftarrow A$;
for $\ell \leftarrow 1$ *to* k **do**
    Sample a random vector $\boldsymbol{v}_{0,\ell} \in \mathbb{S}^{N-1}$. Initialize $\boldsymbol{v}_\ell^{(0)} \leftarrow \boldsymbol{v}_{0,\ell}$;
    for $t \leftarrow 1$ *to* τ **do**
        Apply the procedure to obtain the principle eigenvector $\boldsymbol{v}_\ell^{(t)} = \frac{A_\ell \boldsymbol{v}_\ell^{(t-1)}}{\| A_\ell \boldsymbol{v}_\ell^{(t-1)} \|_2}$;
    Let $\mathcal{V} \leftarrow \mathcal{V} \cup \{\boldsymbol{v}_\ell^{(\tau)}\}$;
    Compute the eigenvalue estimate $\widehat{\lambda}_\ell \leftarrow \| A_\ell \boldsymbol{v}_\ell^{(\tau)} \|_2$;
    Update the matrix by $A_{\ell+1} = A_\ell - \widehat{\lambda}_\ell \boldsymbol{v}_\ell^{(\tau)} \boldsymbol{v}_\ell^{(\tau),\top}$;
return $\mathcal{V}$;

Power Method



Pretraining via Supervised Learning

We construct the input of the Transformer as a *context-augmented matrix* given by the following:

$$\mathbf{H} = \begin{bmatrix} \mathbf{X} \\ \mathbf{P} \end{bmatrix} \in \mathbb{R}^{D \times N}, \quad \mathbf{P} = \begin{bmatrix} \widetilde{p}_{1,1}, \ldots, \widetilde{p}_{1,N} \\ \widetilde{p}_{2,1}, \ldots, \widetilde{p}_{2,N} \\ \vdots \\ \widetilde{p}_{D-d,1}, \ldots, \widetilde{p}_{D-d,N} \end{bmatrix} \in \mathbb{R}^{(D-d) \times N},$$

The auxillary matrix $\mathbf{P}$ contains contextual information; the design also maked sure P is unrelated to X. The experiments show that P is not necesssary for the pre-trained Transformer to performer PCA with high accuracy.

Pretraining via Supervised Learning

We construct the input of the Transformer as a *context-augmented matrix* given by the following:

$$\mathbf{H} = \begin{bmatrix} \mathbf{X} \\ \mathbf{P} \end{bmatrix} \in \mathbb{R}^{D \times N}, \quad \mathbf{P} = \begin{bmatrix} \widetilde{p}_{1,1}, \ldots, \widetilde{p}_{1,N} \\ \widetilde{p}_{2,1}, \ldots, \widetilde{p}_{2,N} \\ \vdots \\ \widetilde{p}_{D-d,1}, \ldots, \widetilde{p}_{D-d,N} \end{bmatrix} \in \mathbb{R}^{(D-d) \times N},$$

The auxillary matrix $\mathbf{P}$ contains contextual information; the design also maked sure P is unrelated to X. The experiments show that P is not necesssary for the pre-trained Transformer to performer PCA with high accuracy. **Output:**

$$TF_\theta(\mathbf{H}) = \begin{bmatrix} \hat{v}_1^\top & \ldots & \hat{v}_k^\top \end{bmatrix}^\top \in \mathbb{R}^{d \times k}$$

which corresponds to the estimated principal eigenvectors of the matrix $\mathbf{X}$.

The Learning Problem

Consider a set of samples $\{\mathbf{X}^{(i)}\}_{i\in[n]}$ i.i.d. sampled from some distribution $p_{\mathbf{X}}$, we construct their oracle top-k principal components as $\mathbf{V}^{(i)} = \begin{bmatrix} \mathbf{v}_1^{i,\top} & \dots & \mathbf{v}_k^{i,\top} \end{bmatrix}^\top$ and the context-augmented input matrix as $\mathbf{H}_i$ for each $\mathbf{X}_i$. Then, the pretraining procedure is given by minimizing the following objective for some convex loss function $L(\cdot,\cdot) : \mathbb{R}^{d\times k} \times \mathbb{R}^{d\times k} \to \mathbb{R}$,

$$\hat{\theta} = \underset{\theta\in\Theta(B_\theta,B_M)}{\arg\min} \sum_{i=1}^{n} L(TF_\theta(\mathbf{H}^{(i)}), \mathbf{V}^{(i)}). \tag{1}$$

Here we consider $\Theta(B_\theta) := \{\theta : \|\theta\| \le B_\theta, \max_\ell M^\ell \le B_M\}$ to be the space of parameters. We also consider guarantees when $L(\mathbf{x}_1, \mathbf{x}_2) := \|\mathbf{x}_1 - \mathbf{x}_2\|_2$ in the theory.

The Design of Auxillary Matrix

Our design of the matrix $\mathbf{P}$ consists of three parts:

1. *Place Holder.* For $\ell \in \{1\} \cup [4 : k+3]$ and $i \in [N]$, we let $\widetilde{\mathbf{p}}_{\ell,i} = \mathbf{0} \in \mathbb{R}^{d\times 1}$. The placeholders in $\mathbf{P}$ record the intermediate results in the forward propagation. Recall that k is the number of eigenvectors we hope to recover.
2. *Identity Matrix.* We let $\begin{bmatrix}\widetilde{\mathbf{p}}_{2,1} & \dots & \widetilde{\mathbf{p}}_{2,N}\end{bmatrix} = \begin{bmatrix}\mathbf{I}_d & \mathbf{0}_{d\times(N-d)}\end{bmatrix}$. The identity matrix in $\mathbf{P}$ helps us screen out all the covariates $\mathbf{X}$ in the forward propagation.
3. *Random Samples on the Hypersphere.* We let $\widetilde{\mathbf{p}}_{3,1}, \dots \widetilde{\mathbf{p}}_{3,k}$ be the i.i.d. samples uniformly distributed on $\mathbb{S}^{d-1}$. The random samples on the sphere correspond to the initial vectors $\mathbf{v}_{0,\ell}$ for $\ell \in [k]$ in algorithm 1.

The auxiliary matrix is designed for purely technical reasons. Moreover, our experiments suggest that such an auxiliary matrix is not necessary for the task.

Transformer Approximation of Power Method

Theorem 7

Assume that the eigenvalues of $\mathbf{X}\mathbf{X}^\top$ *to be* $\lambda_1 > \lambda_2 > \ldots > \lambda_k > \ldots$. *Let* $\Delta := \min_{1\le i<j\le k} |\lambda_i - \lambda_j|$. *Assume that the initialized vectors* $\widetilde{\mathbf{p}}_{3,1}, \ldots \widetilde{\mathbf{p}}_{3,N}$ *satisfy* $\widetilde{\mathbf{p}}_{3,i}^\top \mathbf{v}_i \ge \delta$ *for all* $i \in [k]$ *and make the rest of the vectors* $\mathbf{0}$. *Then, there exists a Transformer model with the number of layers* $L = 2\tau + 4k + 1$ *and the number of heads* $B_M \le \lambda_1^d \frac{C}{\epsilon^2}$ *with* $\tau \le \frac{\log(1/\epsilon_0\delta)}{\epsilon_0}$ *such that for all* $\epsilon_0, \epsilon > 0$, *the final output* $\left[v_1, \ldots, v_k\right]$ *given by the Transformer model achieves*

$$\|v_{\eta+1} - \mathbf{v}_{\eta+1}\|_2 \le C\tau\epsilon\lambda_1^2 + \frac{C\lambda_1\sqrt{\epsilon_0}}{\Delta}\prod_{i=1}^{k}\frac{5\lambda_{i+1}}{\Delta}.$$

Theorem 7 Continued

Moreover, consider the accuracy of multiple $\mathbf{v}$s as a whole. There exists $\theta \in \Theta(B_{\lambda_1}, B_M)$ that satisfies

$$L(TF_\theta(\mathbf{H}), \mathbf{V}) \leq C\tau\epsilon k\lambda_1^2 + C\frac{\epsilon_0\lambda_1^2}{\Delta^2}\sum_{\eta=1}^{k-1}\prod_{i=1}^{\eta}\frac{25\lambda_{i+1}^2}{\Delta^2}^{1/2}.$$

- The first error term comes from the approximation of the Power Method iterations by transformers.
- The second error term comes from finite iteration.

A Special Case

The eigenvalues $\lambda_1 \asymp \lambda_2 \asymp \ldots \asymp \lambda_k \asymp \Delta$. Then our results boil down to

$$|TF_\theta(\mathbf{H}) - \left[\mathbf{v}_1^\top, \mathbf{v}_2^\top, \ldots, \mathbf{v}_k^\top\right]^\top|_2 \lesssim \tau\epsilon k\lambda_1^2 + \frac{\lambda_1}{\Delta}\sqrt{k\epsilon_0}.$$

These results hide dimension d in the universal constant. We note that the dimension significantly affects the approximation bound of Transformers. This is mainly due to the limitations given by approximating high dimensional functions by ReLU neural networks.
In the above theorem, our results rely on the random initialization of $\mathbf{P}$. We show that the conditions on $\widetilde{\mathbf{p}}_{3,1}, \ldots, \widetilde{\mathbf{p}}_{3,N}$ can be achieved through sampling from isotropic Gaussians, given by the following lemma.

Lemma

Lemma 8

Consider $\mathbf{x}$ *to be a random vector sampled uniformly at random on* $\mathbb{S}^{d-1}$. *Let* $\mathbf{v}$ *be any unit length vector, then we have for all* $\delta < \frac{1}{2}d^{-1}$, $\mathbb{P}(|\mathbf{v}^\top \mathbf{x}| \leq \delta) \leq \frac{1}{\sqrt{\pi}}\sqrt{\delta} + \exp(-C\delta^{-\frac{1}{2}})$. *Therefore, for all* $\delta < \frac{1}{2}d^{-1}$, *the event in theorem 7 is achieved with*

$$\mathbb{P}\exists i \in [k] \text{ such that } \mathbf{x}_i^\top \mathbf{v}_i \leq \frac{\delta}{\sqrt{d}} \leq \frac{k\sqrt{\delta}}{\sqrt{\pi}} + k\exp(-C\delta^{-1}).$$

Given the approximation error provided by theorem 7, we further provide the generalization error bound for the ERM defined by$\hat{\theta} = \arg\min_{\theta \in \Theta(B_\theta, B_M)} \sum_{i=1}^{n} L(TF_\theta(\mathbf{H}^{(i)}), \mathbf{V}^{(i)})$. This requires us to consider the following regularity conditions on the underlying distribution of $\mathbf{X}\mathbf{X}^\top$ (which also translates to the distribution for the pre-training instances $\mathbf{X}$).

Assumption 1 (Problems for Pretraining)

The distribution of $\mathbf{X}\mathbf{X}^\top$ supports on the following space

$$\mathbb{X} := \{\mathbf{A} : \mathbf{A} \in \mathbf{S}_{++}^d, B_X \geq \lambda_1(\mathbf{A}) > \lambda_2(\mathbf{A}) > \ldots > \lambda_k(\mathbf{A}), \inf_{1 \leq i < j \leq k} \lambda_i(\mathbf{A}) - \lambda_j(\mathbf{A}) \geq \Delta\}.$$

The above assumption can be generalized to a distribution that supports $\mathbb{X}$ with high probability. Examples of such distribution include the Wishart distribution under the Gaussian design.

Given the above assumption, we are ready to state the generalization bound.

Proposition

Proposition 9

Under assumption 1 and using the notations given by theorem 7, with probability at least $1-\xi$*, the ERM solution* θ *satisfies*

$$\mathbb{E}[L(TF_\theta(\mathbf{H}),\mathbf{V})\Big|\theta] \leq \inf_{\theta\in\Theta(B_\theta,B_M)} \mathbb{E}[L(TF_\theta(\mathbf{H}),\mathbf{V})] + C\sqrt{\frac{k^3 L B_M d^2 \log(B_X+k)+\log(1/\xi)}{n}},$$

where the expectation is taken over the new sample $\mathbf{X}$*.*

Corollary

Corollary 10

Under assumption 1, with probability at least $1 - \xi - \frac{k\sqrt{\delta}}{\sqrt{\pi}} - k\exp(-C\delta^{-1/2})$ *for all* $\delta < d^{-1}$ *we have for all* $\epsilon, \epsilon_0 > 0$,

$$L_{PCA}(\theta, \mathbf{P}) := \mathbb{E}[L(TF_\theta(\mathbf{H}), \mathbf{V}) \Big| \theta, \mathbf{P}] \lesssim \tau\epsilon k\lambda_1^2 + \frac{\epsilon_0\lambda_1^2}{\Delta^2} \sum_{\eta=1}^{k-1} \prod_{i=1}^{\eta} \frac{25\lambda_{i+1}^2}{\Delta^2}^{1/2}$$

$$+ \sqrt{\frac{k^3 \log(\delta/\epsilon_0) B_X^d d^2 \log(B_X + k) + \log(1/\xi)}{n\epsilon_0\epsilon^2}}.$$

If we consider optimizing the bound w.r.t. ϵ_0 and ϵ, we obtain that $L_{PCA}(\theta, \mathbf{P}) \lesssim n^{-1/5}$ with high probability, given that the parameters and the dimension d are of constant scales.

Proof of Theorem 7

Our proof can be disected into the following setps: We construct a Transformer with fixed parameters that performs:

1. The computation of the symmetrized covariate matrix;
2. The approximation of the power method;
3. The removal of the principal eigenvectors;
4. Adjust the dimension of the output through multiplying the two matrices $\widetilde{\mathbf{W}}_0$ and $\widetilde{\mathbf{W}}_1$ on the left and right.

1. The Covariate Matrix

Construct $\mathbf{H} = \begin{bmatrix} \mathbf{X}_1, \ldots, \mathbf{X}_N \\ \widetilde{\mathbf{p}}_{1,1}, \ldots, \widetilde{\mathbf{p}}_{1,N} \\ \widetilde{\mathbf{p}}_{2,1}, \ldots, \widetilde{\mathbf{p}}_{2,N} \\ \vdots \\ \widetilde{\mathbf{p}}_{\ell,1}, \ldots, \widetilde{\mathbf{p}}_{\ell,N} \end{bmatrix} = \begin{bmatrix} \mathbf{X} \\ \mathbf{P} \end{bmatrix}$, we let the number of heads $m = 2$, and construct the first covariate layer as follows,

$$\mathbf{V}_1^{cov} = I_D = -\mathbf{V}_2^{cov}, \quad \mathbf{Q}_1^{cov,\top}\mathbf{K}_1^{cov} = -\mathbf{Q}_2^{\top}\mathbf{K}_2 = \begin{bmatrix} \mathbf{0}_{N+1\times d} & I_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{D\times} \tag{2}$$

$$\widetilde{\mathbf{p}}_{1,\ell,j} = \mathbf{0}, \qquad \widetilde{\mathbf{p}}_{2,\ell,j} = \begin{cases} 1_{\ell=j} & \text{when } \ell \le d \\ 0 & \text{when } \ell > d \end{cases}. \tag{3}$$

1. The Covariate Matrix

Under the above constructions, we obtain that

$$\mathbf{Q}_1^\top \mathbf{K}_1 \mathbf{H} = \begin{bmatrix} I_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{D\times N}, \quad \mathbf{Q}_2^\top \mathbf{K}_2 \mathbf{H} = \begin{bmatrix} -I_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{D\times N},$$
$$\sigma(\mathbf{H}^\top \mathbf{Q}_1^\top \mathbf{K}_1 \mathbf{H}) + \sigma(\mathbf{H}^\top \mathbf{Q}_2^\top \mathbf{K}_2 \mathbf{H}) = \left[\mathbf{X}^\top, \mathbf{0}\right] \in \mathbb{R}^{N\times N}.$$

We further obtain that

$$\frac{1}{N}\sum_{m=1}^{M} (\mathbf{V}_m \mathbf{H}) \times \sigma((\mathbf{Q}_m \mathbf{H})^\top (\mathbf{K}_m \mathbf{H})) = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{X}\mathbf{X}^\top \in \mathbb{R}^{d\times d} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{D\times D}.$$

Therefore, the output is given by $\widetilde{\mathbf{H}}^{cov} = \begin{bmatrix} \mathbf{X} \\ \mathbf{X}\mathbf{X}^\top, \mathbf{0} \\ \widetilde{\mathbf{p}}_{2,1}, \ldots, \widetilde{\mathbf{p}}_{2,N} \\ \widetilde{\mathbf{p}}_{\ell,1}, \ldots, \widetilde{\mathbf{p}}_{\ell,N} \end{bmatrix}$.

2. The Power Iteration

Step 1: Obtaining the vector given by $\mathbf{X}\mathbf{X}^\top \mathbf{v}$.
Step 2: Approximation of the value of the inverse norm given by $1/\|\mathbf{X}\mathbf{X}^\top \mathbf{v}\|_2$. We show that one can use the multihead ReLU Transformer to achieve both goals simulatenously, whose parameters are given by

$$\mathbf{V}_1^{pow,1} = -\mathbf{V}_2^{pow,1} = \begin{bmatrix} \mathbf{0}_{(3d+1)\times(2d+1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}_{(d)\times(2d+1)} & I_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{Q}_1^{pow,1} = -\mathbf{Q}_1^{pow,1} = \begin{bmatrix} \mathbf{0}_{(d+1)\times(d+1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}_{d\times(d+1)} & I_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{K}_1^{pow,1} = \mathbf{K}_2^{pow,1} = \begin{bmatrix} \mathbf{0}_{(3d+1)\times(3d+1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}_{d\times(3d+1)} & I_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \qquad \widetilde{\mathbf{p}}_{4,j} = \mathbf{0} \text{ for all } j \in [N].$$

2. The Power Iteration

We can calculate that the output of the first power iteration layer is given by

$$\widetilde{\mathbf{H}}^{pow,1} = \begin{bmatrix} \mathbf{X} \\ \widetilde{\mathbf{y}}^\top \\ \mathbf{X}\mathbf{X}^\top, \mathbf{0} \\ \widetilde{\mathbf{p}}_{2,1}, \ldots, \widetilde{\mathbf{p}}_{2,N} \\ \widetilde{\mathbf{p}}_{3,1}, \ldots, \widetilde{\mathbf{p}}_{3,N} \\ \mathbf{X}\mathbf{X}^\top \widetilde{\mathbf{p}}_{3,1}, \mathbf{0} \\ \widetilde{\mathbf{p}}_{5,1}, \ldots, \widetilde{\mathbf{p}}_{5,N} \\ \vdots \\ \widetilde{\mathbf{p}}_{\ell,1}, \ldots, \widetilde{\mathbf{p}}_{\ell,N} \end{bmatrix}.$$

Then, using lemma, we design an extra attention layer that performs the normalizing procedure, with the following parameters for all $m \in [M]$,

$$\mathbf{V}_m^{pow,2} = \begin{bmatrix} \mathbf{0}_{d\times(4d+1)} & c_m\mathbf{I}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \qquad \mathbf{Q}_m^{pow,2} = \begin{bmatrix} \mathbf{0}_{d\times(2d+1)} & \mathbf{I}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{K}_m^{pow,2} = \begin{bmatrix} \mathbf{0}_{1\times(3d+1)} & \mathbf{a}_m^\top & \mathbf{0} \\ \vdots & & \\ \mathbf{0}_{1\times(3d+1)} & \mathbf{a}_m^\top & \mathbf{0} \\ \mathbf{0}_{(D-d)\times(3d+1)} & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

2. The Power Iteration

Then, given $\mathbf{V}_m^{pow,2}$ we can show that under the condition given by lemma, we have

$$\left\| \sum_{m=1}^{M} \mathbf{V}_m^{pow,2} \widetilde{\mathbf{H}}^{pow,1} \sigma((\mathbf{Q}_m^{pow,2} \widetilde{\mathbf{H}}^{pow,1})^\top (\mathbf{K}_m^{pow,2} \widetilde{\mathbf{H}}^{pow,1})) \right. - \left[\begin{array}{c} \mathbf{0}_{4d+} \\ \frac{\mathbf{X}\mathbf{X}^\top \widetilde{\mathbf{p}}_{3,1}}{\|\mathbf{X}\mathbf{X}^\top \widetilde{\mathbf{p}}_{3,1}\|_2} - \mathbf{X} \\ \mathbf{0} \end{array} \right.$$

$$< \epsilon,$$

Moreover, we can further achieve that

$$\left\| \sum_{m=1}^{M} \mathbf{V}_m^{pow,2j} \widetilde{\mathbf{H}}^{pow,1} \sigma((\mathbf{Q}_m^{pow,2} \widetilde{\mathbf{H}}^{pow,1})^\top (\mathbf{K}_m^{pow,2} \widetilde{\mathbf{H}}^{pow,1})) \right. - \left[\begin{array}{c} \mathbf{0}_{4d-} \\ \frac{\mathbf{X}\mathbf{X}^\top \widetilde{\mathbf{p}}_{3,1}}{\|\mathbf{X}\mathbf{X}^\top \widetilde{\mathbf{p}}_{3,1}\|_2} - \\ \mathbf{0} \end{array} \right.$$

$$< \epsilon \|\mathbf{X}\mathbf{X}^\top \widetilde{\mathbf{p}}_{3,1}\|_2.$$

2. The Power Iteration

Hence, using the fact that $\widetilde{\mathbf{H}}^{pow,2} = \widetilde{\mathbf{H}}^{pow,1} + \sum_{i=1}^{m} \mathbf{V}_m^{pow,2} \widetilde{\mathbf{H}}^{pow,1} \sigma((\mathbf{Q}_m^{pow,2} \widetilde{\mathbf{H}}^{pow,1})^\top (\mathbf{K}_m^{pow,2} \widetilde{\mathbf{H}}^{pow,1}))$, we obtain that

$$\left\| \widetilde{\mathbf{H}}^{pow,2} - \begin{bmatrix} \mathbf{X} \\ \widetilde{\mathbf{y}} \\ \mathbf{X}\mathbf{X}^\top, \mathbf{0} \\ \widetilde{\mathbf{p}}_{2,1}, \ldots, \widetilde{\mathbf{p}}_{2,N} \\ \widetilde{\mathbf{p}}_{3,1}, \ldots, \widetilde{\mathbf{p}}_{3,N} \\ \frac{\mathbf{X}\mathbf{X}^\top \widetilde{\mathbf{p}}_{3,1}}{\|\mathbf{X}\mathbf{X}^\top \widetilde{\mathbf{p}}_{3,1}\|_2}, \ldots \mathbf{0} \\ \vdots \end{bmatrix} \right\|_2 < \epsilon \|\mathbf{X}\mathbf{X}^\top \widetilde{\mathbf{p}}_{3,1}\|_2.$$

Then we construct another attention layer, which performs similar calculations as that of $pow, 1$ but switch the rows of $\widetilde{\mathbf{p}}_{3,1}$ with that of $\frac{\mathbf{X}\mathbf{X}^\top \widetilde{\mathbf{p}}_{3,1}}{\|\mathbf{X}\mathbf{X}^\top \widetilde{\mathbf{p}}_{3,1}\|_2}$. Our construction for the third layer is given by

2. The Power Iteration

$$\mathbf{V}_1^{pow,3} = -\mathbf{V}_2^{pow,3} = \begin{bmatrix} \mathbf{0}_{(3d+1)\times(2d+1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}_{d\times(2d+1)} & I_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{Q}_1^{pow,3} = -\mathbf{Q}_2^{pow,3} = \begin{bmatrix} \mathbf{0}_{(3d+1)\times(d+1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}_{d\times(d+1)} & I_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{K}_1^{pow,3} = \mathbf{K}_2^{pow,3} = \begin{bmatrix} \mathbf{0}_{(4d+1)\times(4d+1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}_{d\times(4d+1)} & I_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \qquad \widetilde{\mathbf{p}}_{4,j} = \mathbf{0} \text{ for all } j \in [N].$$

Consider we are doing in total of τ power iterations, we can set for all $\tau \in \mathbb{N}^*$,

$$\mathbf{V}_m^{pow,2\tau+1} = \mathbf{V}_m^{pow,3}, \quad \mathbf{Q}_m^{pow,2\tau+1} = \mathbf{Q}_m^{pow,3}, \quad \mathbf{K}_m^{pow,2\tau+1} = \mathbf{K}_m^{pow,3},$$
$$\mathbf{V}_m^{pow,2\tau+2} = \mathbf{V}_m^{pow,4}, \quad \mathbf{Q}_m^{pow,2\tau+2} = \mathbf{Q}_m^{pow,4}, \quad \mathbf{K}_m^{pow,2\tau+2} = \mathbf{K}_m^{pow,4}.$$

Therefore, taking another layer of normalization, we can show that

$$\left\| \widetilde{\mathbf{H}}^{pow,3} - \begin{bmatrix} \mathbf{X} \\ \widetilde{\mathbf{y}} \\ \mathbf{X}\mathbf{X}^\top, \mathbf{0} \\ \widetilde{\mathbf{p}}_{2,1}, \ldots, \widetilde{\mathbf{p}}_{2,N} \\ \widetilde{\mathbf{p}}_{3,1}, \ldots, \widetilde{\mathbf{p}}_{3,N} \\ \frac{(\mathbf{X}\mathbf{X}^\top)^2 \widetilde{\mathbf{p}}_{3,1}}{\|\mathbf{X}\mathbf{X}^\top \widetilde{\mathbf{p}}_{3,1}\|_2^2}, \mathbf{0} \\ \widetilde{\mathbf{p}}_{5,1}, \ldots, \widetilde{\mathbf{p}}_{5,N} \\ \vdots \\ \widetilde{\mathbf{p}}_{\ell,1}, \ldots, \widetilde{\mathbf{p}}_{\ell,N} \end{bmatrix} \right\|_2 \leq 2\epsilon \|\mathbf{X}\mathbf{X}^\top\|_2.$$

Then, using the sublinearity of errors, we can show that for $\tau \in \mathbb{N}$,

$$\left\| \widetilde{\mathbf{H}}^{pow,2\tau+2} - \begin{bmatrix} \mathbf{X} \\ \widetilde{\mathbf{y}} \\ \mathbf{X}\mathbf{X}^\top, \mathbf{0} \\ \widetilde{\mathbf{p}}_{2,1}, \ldots, \widetilde{\mathbf{p}}_{2,N} \\ \widetilde{\mathbf{p}}_{3,1}, \ldots, \widetilde{\mathbf{p}}_{3,N} \\ \widetilde{\mathbf{p}}_{3,1}^{(\tau)}, \mathbf{0} \\ \widetilde{\mathbf{p}}_{5,1}, \ldots, \widetilde{\mathbf{p}}_{5,N} \\ \vdots \\ \widetilde{\mathbf{p}}_{\ell,1}, \ldots, \widetilde{\mathbf{p}}_{\ell,N} \end{bmatrix} \right\|_\infty \leq \tau\epsilon \|\mathbf{X}\mathbf{X}^\top\|_2, \quad \widetilde{\mathbf{p}}_{3,1}^{(\tau)} = \frac{\mathbf{X}\mathbf{X}^\top \widetilde{\mathbf{p}}_{3,1}^{(\tau-1)}}{\|\mathbf{X}\mathbf{X}^\top \widetilde{\mathbf{p}}_{3,1}^{(\tau-1)}\|_2}$$

$$\widetilde{\mathbf{p}}_{3,1}^{(0)} = \widetilde{\mathbf{p}}_{3,1}$$

2. The Power Iteration

If we denote $\mathbf{v}_i$ as the eigenvector corresponds to the i th largest eigenvalue of $\mathbf{X}\mathbf{X}^\top$. Let the eigenvalues of $\mathbf{X}\mathbf{X}^\top$ be denoted by $\lambda_1 > \lambda_2 > \cdots > \lambda_n$. Given $|\widetilde{\mathbf{p}}_{3,1}^\top \mathbf{v}_1| > \delta$ and $|\sqrt{\lambda_1} - \sqrt{\lambda_2}| = \Omega(1)$. Theorem 3.11 in blum2020foundations page 53 shows that given $k = \frac{\log(1/\epsilon_0\delta)}{2\epsilon_0}$ and $\|\widetilde{\mathbf{p}}_{3,1}^{(\tau)}\|_2 = \|\mathbf{v}_1\|_2 = 1$, one immediately obtains that

$$\widetilde{\mathbf{p}}_{3,1}^{(\tau),\top}\mathbf{v}_1 \geq 1 - \epsilon_0, \qquad \|\widetilde{\mathbf{p}}_{3,1}^{(\tau)} - \mathbf{v}_1\|_2 = \sqrt{2 - 2\mathbf{v}_1^\top \widetilde{\mathbf{p}}_{3,1}^{(\tau)}} = \sqrt{2\epsilon_0}.$$

And we also consider the approximation of the maximum eigenvalue. Note that using $\|\mathbf{v}_1\|_2 = 1$, we have

$$\begin{aligned}
\|\mathbf{X}\mathbf{X}^\top\|_2 = \|\mathbf{X}\mathbf{X}^\top\mathbf{v}_1\|_2 &= \|\mathbf{X}\mathbf{X}^\top\widetilde{\mathbf{p}}_{3,1}^{(\tau)} + \mathbf{X}\mathbf{X}^\top(\mathbf{v}_1 - \widetilde{\mathbf{p}}_{3,1}^{(\tau)})\|_2 \\
&\leq \|\mathbf{X}\mathbf{X}^\top\widetilde{\mathbf{p}}_{3,1}^{(\tau)}\|_2 + \|\mathbf{X}\mathbf{X}^\top(\mathbf{v}_1 - \widetilde{\mathbf{p}}_{3,1}^{(\tau)})\|_2 \\
&\leq \|\mathbf{X}\mathbf{X}^\top\widetilde{\mathbf{p}}_{3,1}^{(\tau)}\|_2 + \|\mathbf{X}\mathbf{X}^\top\|_2\|\mathbf{v}_1 - \widetilde{\mathbf{p}}_{3,1}^{(\tau)}\|_2.
\end{aligned}$$

Similarly we can also derive that $\|\mathbf{X}\mathbf{X}^\top\|_2 \geq \|\mathbf{X}\mathbf{X}^\top\widetilde{\mathbf{p}}_{3,1}^{(\tau)}\|_2 - \|\mathbf{X}\mathbf{X}^\top\|_2\|\mathbf{v}_1 - \widetilde{\mathbf{p}}_{3,1}\|_2$. Then we show that

3. The Removal of principal Eigenvectors.

Step 1: The computation of the estimated eigenvalue $\|\mathbf{X}\mathbf{X}^\top \widetilde{\mathbf{p}}_{3,1}\|_2$.
Step 2: The construction of the low rank update $\widetilde{\mathbf{p}}_{3,1}\widetilde{\mathbf{p}}_{3,1}^\top$. For step (1), we consider the following construction:

$$\mathbf{V}_1^{rpe,1} = -\mathbf{V}_2^{rpe,1} = \begin{bmatrix} \mathbf{0}_{(3d+1)\times(2d+1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}_{d\times(2d+1)} & I_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{Q}_1^{rpe,1} = -\mathbf{Q}_2^{rpe,1} = \begin{bmatrix} \mathbf{0}_{(d+1)\times(d+1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}_{d\times(d+1)} & I_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{K}_1^{rpe,1} = \mathbf{K}_2^{rpe,1} = \begin{bmatrix} \mathbf{0}_{(4d+1)\times(4d+1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0}_{d\times(4d+1)} & I_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

3. The Removal of principal Eigenvectors.

Note that the above construction is similar to the first layer of the power method. Under this construction, we can show that

$$\widetilde{\mathbf{H}}^{rpe,1} = \widetilde{\mathbf{H}}^{pow,2\tau+2} + \sum_{m\in\{1,2\}} \mathbf{V}_m^{rpe,1}\sigma((\mathbf{Q}_m^{rpe,1}\widetilde{\mathbf{H}}^{pow,2\tau+2})^\top(\mathbf{K}_m^{rpe,1}\widetilde{\mathbf{H}}^{pow,2\tau+2})),$$

$$\left\|\widetilde{\mathbf{H}}^{rpe,1} - \underbrace{\begin{bmatrix} \mathbf{X} \\ \widetilde{\mathbf{y}} \\ \mathbf{X}\mathbf{X}^\top, \mathbf{0} \\ \widetilde{\mathbf{p}}_{2,1}, \ldots, \widetilde{\mathbf{p}}_{2,N} \\ \widetilde{\mathbf{p}}_{3,1}, \ldots, \widetilde{\mathbf{p}}_{3,N} \\ \widetilde{\mathbf{p}}_{3,1}^{(\tau)}, \mathbf{0} \\ \mathbf{X}\mathbf{X}^\top\widetilde{\mathbf{p}}_{3,N}^{(\tau)}, \mathbf{0} \\ \widetilde{\mathbf{p}}_{6,1}, \ldots, \widetilde{\mathbf{p}}_{6,N} \\ \vdots \\ \widetilde{\mathbf{p}}_{\ell,1}, \ldots, \widetilde{\mathbf{p}}_{\ell,N} \end{bmatrix}}_{=:\mathbf{H}^{rpe,1}}\right\|_2 \leq C\tau\epsilon\|\mathbf{X}\mathbf{X}^\top\|_2^2, \qquad \widetilde{\mathbf{p}}_{5,i} = \mathbf{0}, \quad \forall i \in$$

3. The Removal of principal Eigenvectors.

Then, we construct the next layer, using the notations in lemma, for $M \geq \|\mathbf{X}\mathbf{X}^\top\|_2^d \frac{C(d)}{\epsilon^2}$ for all $m \in [M]$ we have

$$\mathbf{V}_m^{rpe,2} = \begin{bmatrix} \mathbf{0}_{d\times(4d+1)} & d_m\mathbf{I}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \qquad \mathbf{Q}_m^{rpe,2} = \begin{bmatrix} \mathbf{0}_{d\times(2d+1)} & \mathbf{I}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{K}_m^{rpe,2} = \begin{bmatrix} \mathbf{0}_{1\times(5d+1)} & \mathbf{b}_m^\top & \mathbf{0} \\ \vdots & & \\ \mathbf{0}_{1\times(5d+1)} & \mathbf{b}_m^\top & \mathbf{0} \\ \mathbf{0}_{(D-d)\times(5d+1)} & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Given the above construction, we subsequently show that

$$(\mathbf{Q}_m^{rpe,2}\widetilde{\mathbf{H}}^{rpe,1})^\top = \begin{bmatrix} I_{d\times d} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \qquad \mathbf{K}_m^{rpe,2}\widetilde{\mathbf{H}}^{rpe,1} = \begin{bmatrix} \mathbf{b}_m^\top\mathbf{X}\mathbf{X}^\top\widetilde{\mathbf{p}}_{3,1}^{(\tau)} & \mathbf{0} \\ \vdots & \vdots \\ \mathbf{b}_m^\top\mathbf{X}\mathbf{X}^\top\widetilde{\mathbf{p}}_{3,1}^{(\tau)} & \mathbf{0} \\ \mathbf{0}_{(D-d)\times 1} & \mathbf{0} \end{bmatrix}.$$

3. The Removal of principal Eigenvectors.

Hence, given the construction of $\mathbf{V}_m^{rpe,2}$, we can show that $\widetilde{\mathbf{H}}^{rpe,2}$ satisfies

$$\begin{aligned}
\widetilde{\mathbf{H}}^{rpe,2} &= \widetilde{\mathbf{H}}^{rpe,1} + \sum_{m\in[M]} \mathbf{V}_m^{rpe,2}\widetilde{\mathbf{H}}^{rpe,1} \times \sigma((\mathbf{K}_m^{rpe,2}\widetilde{\mathbf{H}}^{rpe,1})^\top(\mathbf{Q}_m^{rpe}\widetilde{\mathbf{H}}^{rpe,1})) \\
&= \mathbf{H}^{rpe,1} + \sum_{m\in[M]} \mathbf{V}_m^{rpe,2}\mathbf{H}^{rpe,1} \times \sigma((\mathbf{K}_m^{rpe,2}\mathbf{H}^{rpe,1})^\top(\mathbf{Q}_m^{rpe,2}\mathbf{H}^{rpe,1})) \\
&\quad + (\widetilde{\mathbf{H}}^{rpe,1} - \mathbf{H}^{rpe,1}) + \sum_{m\in[M]} \mathbf{V}_m^{rpe,2}\widetilde{\mathbf{H}}^{rpe,1} \times \sigma((\mathbf{K}_m^{rpe,2}\widetilde{\mathbf{H}}^{rpe,1})^\top \mathbf{Q}_m^{rpe,2} \\
&\quad - \sum_{m\in[M]} \mathbf{V}_m^{rpe,2}\widetilde{\mathbf{H}}^{rpe,1} \times \sigma((\mathbf{K}_m^{rpe,2}\widetilde{\mathbf{H}}^{rpe,1})^\top \mathbf{Q}_m^{rpe,2}\widetilde{\mathbf{H}}^{rpe,1}).
\end{aligned}$$

Denote the sum of the first two terms as $\hat{\mathbf{H}}^{rpe,1}$.

3. The Removal of principal Eigenvectors.

We note that by lemma we can show that

$$\left\| H^{rpe,1} - \begin{bmatrix} \mathbf{X} \\ \widetilde{\mathbf{y}} \\ \mathbf{X}\mathbf{X}^\top, \mathbf{0} \\ \widetilde{\mathbf{p}}_{2,1} \cdots, \widetilde{\mathbf{p}}_{2,N} \\ \widetilde{\mathbf{p}}_{3,1} \cdots, \widetilde{\mathbf{p}}_{3,N} \\ \widetilde{\mathbf{p}}_{3,1}^{(\tau)}, \mathbf{0} \\ \|\mathbf{X}\mathbf{X}^\top \widetilde{\mathbf{p}}_{3,N}^{(\tau)}\|_2^{\frac{1}{2}} \widetilde{\mathbf{p}}_{3,1}^{(\tau)}, \mathbf{0} \\ \vdots \\ \widetilde{\mathbf{p}}_{\ell,1}, \cdots \widetilde{\mathbf{p}}_{\ell,N} \end{bmatrix} \right\|_2 \leq C\tau\epsilon \|\mathbf{X}\mathbf{X}^\top\|_2^2.$$

Then the rest of the proof focuses on showing that the rest of the terms are small. Already we have

$$\|\widetilde{\mathbf{H}}^{rpe,1} - \mathbf{H}^{rpe,1}\|_2 \leq \tau\epsilon \|\mathbf{X}\mathbf{X}^\top\|_2^2.$$

And for the last term, we can show that

3. The Removal of principal Eigenvectors.

$$\Big\| \sum_{m\in[M]} \mathbf{V}_m^{rpe,2}\widetilde{\mathbf{H}}^{rpe,1} \times \sigma((\mathbf{K}_m^{rpe,2}\widetilde{\mathbf{H}}^{rpe,1})^\top(\mathbf{Q}_m^{rpe,2}\widetilde{\mathbf{H}}^{rpe,1}))- \sum_{m\in[M]} \mathbf{V}_m^{rpe,2}\mathbf{H}^{rpe,1} \times \sigma((\mathbf{K}_m^{rpe,2}\mathbf{H}^{rpe,1})^\top(\mathbf{Q}_m^{rpe,2}\mathbf{H}^{rpe,1}))\Big\|_2 \leq C\tau\epsilon\|\mathbf{X}\mathbf{X}^\top\|_2^2.$$

Collecting the above pieces, we finally show that

$$\left\|\widetilde{\mathbf{H}}^{rpe,2} - \begin{bmatrix} \mathbf{X} \\ \mathbf{X}\mathbf{X}^\top, \mathbf{0} \\ \widetilde{\mathbf{p}}_{2,1}, \ldots, \widetilde{\mathbf{p}}_{2,N} \\ \widetilde{\mathbf{p}}_{3,1}, \ldots, \widetilde{\mathbf{p}}_{3,N} \\ \widetilde{\mathbf{p}}_{3,1}^{(\tau)}, \mathbf{0} \\ \|\mathbf{X}\mathbf{X}^\top\widetilde{\mathbf{p}}_{3,1}^{(\tau)}\|_2^{\frac{1}{2}}\widetilde{\mathbf{p}}_{3,1}^{(\tau)}, \mathbf{0} \\ \widetilde{\mathbf{p}}_{6,1}, \ldots, \widetilde{\mathbf{p}}_{6,N} \\ \vdots \\ \widetilde{\mathbf{p}}_{\ell,1}, \ldots, \widetilde{\mathbf{p}}_{\ell,N} \end{bmatrix}\right\|_2 \leq C\tau\epsilon\|\mathbf{X}\mathbf{X}^\top\|_2^2.$$

3. The Removal of principal Eigenvectors.

Then we construct another layer to remove the principal components from the matrix $\mathbf{X}\mathbf{X}^\top$, given by

$$-\mathbf{V}_1^{rpe,3} = \mathbf{V}_2^{rpe,3} = \begin{bmatrix} \mathbf{0}_{(d+1)\times(4d+1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{Q}_1^{rpe,3} = -\mathbf{Q}_2^{rpe,3} = \begin{bmatrix} \mathbf{0}_{d\times(4d+1)} & I_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$$\mathbf{K}_1^{rpe,3} = \mathbf{K}_2^{rpe,3} = \begin{bmatrix} \mathbf{0}_{d\times(4d+1)} & I_d & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

3. The Removal of principal Eigenvectors.

Therefore, we can further show that

$$\widetilde{\mathbf{H}}^{rpe,3} = \widetilde{\mathbf{H}}^{rpe,2} + \sum_{m=1}^{2} \mathbf{V}_m^{rpe,3} \widetilde{\mathbf{H}}^{rpe,2} \times \sigma((\mathbf{Q}_m^{rpe,3} \widetilde{\mathbf{H}}^{rpe,2})^\top \mathbf{K}_m^{rpe,3} \widetilde{\mathbf{H}}^{rpe,2})$$

satisfies

$$\left\| \widetilde{\mathbf{H}}^{rpe,3} - \begin{bmatrix} \mathbf{X} \\ \mathbf{X}\mathbf{X}^\top - \|\mathbf{X}\mathbf{X}^\top \widetilde{\mathbf{p}}_{3,1}^{(\tau)}\|_2 \widetilde{\mathbf{p}}_{3,1}^{(\tau)} \widetilde{\mathbf{p}}_{3,1}^{(\tau),\top}, \mathbf{0} \\ \widetilde{\mathbf{p}}_{2,1}, \ldots, \widetilde{\mathbf{p}}_{2,N} \\ \widetilde{\mathbf{p}}_{3,1}, \ldots, \widetilde{\mathbf{p}}_{3,N} \\ \widetilde{\mathbf{p}}_{3,1}^{(\tau)}, \mathbf{0} \\ \widetilde{\mathbf{p}}_{5,1}, \ldots, \widetilde{\mathbf{p}}_{5,N} \\ \vdots \\ \widetilde{\mathbf{p}}_{\ell,1}, \ldots, \widetilde{\mathbf{p}}_{\ell,N} \end{bmatrix} \right\|_2 \leq C\tau\epsilon \|\mathbf{X}\mathbf{X}^\top\|_2^2.$$

3. The Removal of principal Eigenvectors.

And then we proceed to recover the rest of the k principal eigenvectors using similar model architecture given by the ones used by the Power Iterations. For the computation over the τ-th eigenvector, we denote $\widetilde{\mathbf{H}}^{pow,\eta,1}$ till $\widetilde{\mathbf{H}}^{pow,\eta,\tau}$ to be the intermediate states corresponding to the η-th power iteration. We denote $\widetilde{\mathbf{H}}^{rpe,\eta,\tau_0}$ to be the output of η-th removal of principal eigenvector layers for the τ-th eigenvector. Furthermore, we iteratively define

$$\mathbf{A}_1 = \mathbf{X}\mathbf{X}^\top - \|\mathbf{X}\mathbf{X}^\top \widetilde{\mathbf{p}}_{3,1}^{(\tau)}\|_2 \widetilde{\mathbf{p}}_{3,1}^{(\tau)} \widetilde{\mathbf{p}}_{3,1}^{(\tau),\top}, \qquad \mathbf{A}_{i+1} = \mathbf{A}_i - \|\mathbf{A}_i \widetilde{\mathbf{p}}_{3,i}^{(\tau)}\|_2 \widetilde{\mathbf{p}}_{3,i}^{(\tau)} \widetilde{\mathbf{p}}_{3,i}^{(\tau)},$$

Then, applying the subadditivity of the 2-norm, we can show that

3. The Removal of principal Eigenvectors.

$$\left\| \widetilde{\mathbf{H}}^{rpe,4,k} - \begin{bmatrix} \mathbf{X} \\ \mathbf{A}_{k+1}, \mathbf{0} \\ \widetilde{\mathbf{p}}_{2,1}, \ldots, \widetilde{\mathbf{p}}_{2,N} \\ \widetilde{\mathbf{p}}_{3,1}, \ldots, \widetilde{\mathbf{p}}_{3,N} \\ \widetilde{\mathbf{p}}_{3,1}^{(\tau)}, \mathbf{0} \\ \widetilde{\mathbf{p}}_{3,2}^{(\tau)}, \mathbf{0} \\ \vdots \\ \widetilde{\mathbf{p}}_{3,k}^{(\tau)}, \mathbf{0} \end{bmatrix} \right\|_2 \leq C\tau k\epsilon \|\mathbf{X}\mathbf{X}^\top\|_2^2.$$

For simplicity, we denote $\widetilde{\mathbf{A}} = \begin{bmatrix} \mathbf{X} \\ \mathbf{A}_{k+1}, \mathbf{0} \\ \widetilde{\mathbf{p}}_{2,1}, \ldots, \widetilde{\mathbf{p}}_{2,N} \\ \widetilde{\mathbf{p}}_{3,1}, \ldots, \widetilde{\mathbf{p}}_{3,N} \end{bmatrix}$ and $\widetilde{\mathbf{P}} = \begin{bmatrix} \widetilde{\mathbf{p}}_{3,1}^{(\tau)} \\ \widetilde{\mathbf{p}}_{3,2}^{(\tau)} \\ \vdots \\ \widetilde{\mathbf{p}}_{3,k}^{(\tau)} \end{bmatrix}$ from here.

4. Finishing Up

Our construction gives the following:

$$\widetilde{\mathbf{W}}_0 = \left[\mathbf{0}, I_{kd}\right], \qquad \widetilde{\mathbf{W}}_1 = \begin{bmatrix} 1 \\ \mathbf{0}_{N-1} \end{bmatrix}.$$

And we can show that

$$\left\| \widetilde{\mathbf{W}}_0 \widetilde{\mathbf{H}}^{rpe,4,k} \widetilde{\mathbf{W}}_1 - \begin{bmatrix} \widetilde{\mathbf{p}}_{3,1}^{(\tau)} \\ \widetilde{\mathbf{p}}_{3,2}^{(\tau)} \\ \vdots \\ \widetilde{\mathbf{p}}_{3,k}^{(\tau)} \end{bmatrix} \right\|_2 \leq C\tau k\epsilon \|\mathbf{X}\mathbf{X}^\top\|_2^2.$$

We further use the result given by lemma, denote $a_\eta := \|\mathbf{v}_\eta - \widetilde{\mathbf{p}}_{3,\eta}^{(\tau)}\|_2$, $\widehat{\lambda}_\eta = \|\mathbf{A}_\eta \widetilde{\mathbf{p}}_{3,\eta}^{(\tau)}\|_2$, and $b_\eta := |\lambda_\eta - \widehat{\lambda}_\eta|$ for $\eta \in [k]$, we obtain that for all $\eta \geq 1$, given the number of iterations $\tau \geq C\frac{\log(1/\epsilon_0\delta)}{2\epsilon_0}$ where the constant value C depends on d,

4. Finishing Up

$$a_{\eta+1} \leq \frac{\max_{i\in[\eta]} b_i + \sum_{i=1}^{\eta} 2\lambda_i a_i}{\Delta}, \qquad b_{\eta+1} \leq \frac{2\lambda_{\eta+1}}{\Delta} \max_{i\in[\eta]} b_\eta + \sum_{i=1}^{\eta} 2\lambda_i a_i + \lambda$$

Further note that the starting point is given by $a_1 \leq \sqrt{2\epsilon_0}$, $b_1 \leq \lambda_1\sqrt{2\epsilon_0}$. Introducing $A_\eta = \sum_{i=1}^{\eta} 2\lambda_i a_i$, we obtain that $A_{\eta+1} = \sum_{i=1}^{\eta+1} 2\lambda_i a_i = A_\eta + 2\lambda_{\eta+1} a_{\eta+1}$ which alternatively implies that

$$\frac{1}{2\lambda_{\eta+1}}(A_{\eta+1} - A_\eta) \leq \frac{\max_{i\in[\eta]} b_i + A_\eta}{\Delta},$$

$$b_{\eta+1} \leq \frac{2\lambda_{\eta+1}}{\Delta} \max_{i\in[\eta]} b_\eta + A_\eta + \lambda_{\eta+1}\sqrt{2\epsilon_0}.$$

4. Finishing Up

We use the fact $\frac{\lambda_\eta}{\Delta} > 1$ for all $\eta \in [k]$ to show the following

$$A_{\eta+1} + \max_{i\in[\eta+1]} b_i \leq \frac{5\lambda_{\eta+1}}{\Delta}(A_\eta + \max_{i\in[\eta]} b_i) + \lambda_1\sqrt{2\epsilon_0}, \qquad A_1 + b_1 = 2\lambda_1\sqrt{2\epsilon_0}$$

which implies that

$$A_{\eta+1} + \max_{i\in[\eta+1]} b_i + \frac{\lambda_1\sqrt{2\epsilon_0}}{\frac{5\lambda_1}{\Delta} - 1} \leq \frac{5\lambda_1}{\Delta} A_\eta + \max_{i\in[\eta]} b_i + \frac{\lambda_1\sqrt{2\epsilon_0}}{\frac{5\lambda_1}{\Delta} - 1},$$

$$A_{\eta+1} + \max_{i\in[\eta+1]} b_i + \frac{\lambda_1\sqrt{2\epsilon_0}}{\frac{5\lambda_1}{\Delta} - 1} \leq A_1 + b_1 + \frac{\lambda_1\sqrt{2\epsilon_0}}{\frac{5\lambda_1}{\Delta} - 1} \prod_{i=1}^{\eta} \frac{5\lambda_{i+1}}{\Delta}$$

$$= \lambda_1\sqrt{2\epsilon_0}2 + \frac{1}{\frac{5\lambda_1}{\Delta} - 1} \prod_{i=1}^{\eta} \frac{5\lambda_{i+1}}{\Delta}.$$

4. Finishing Up

Therefore, for $\eta \leq k$, we have for all $\eta \in [k-1]$,

$$a_{\eta+1} \leq \frac{1}{\Delta}\lambda_1\sqrt{2\epsilon_0}2 + \frac{1}{\frac{5\lambda_1}{\Delta}-1}\prod_{i=1}^{\eta}\frac{5\lambda_{i+1}}{\Delta} - \frac{\lambda_1\sqrt{2\epsilon_0}}{\frac{5\lambda_1}{\Delta}-1},$$

$$b_{\eta+1} \leq \frac{2\lambda_\eta\lambda_1\sqrt{2\epsilon_0}}{\Delta}2 + \frac{1}{\frac{5\lambda_1}{\Delta}-1}\prod_{i=1}^{\eta}\frac{5\lambda_{i+1}}{\Delta} + \lambda_{\eta+1}\sqrt{2\epsilon_0}.$$

Therefore collecting pieces, we conclude that there exists a transformer with number of layers $2\tau + 4k + 1$ and number of heads $M \leq \lambda_1^d \frac{C(d)}{\epsilon^2}$ such that the final output $v_1, \ldots, v_k$ given by the Transformer model satisfy $\forall \eta \in [k-1]$,

$$\|v_{\eta+1} - \mathbf{v}_{\eta+1}\|_2 \leq C\tau\epsilon\lambda_1^2 + \frac{1}{\Delta}\lambda_1\sqrt{2\epsilon_0}2 + \frac{1}{\frac{5\lambda_1}{\Delta}-1}\prod_{i=1}^{\eta}\frac{5\lambda_{i+1}}{\Delta} - \frac{\lambda_1\sqrt{2\epsilon_0}}{\frac{5\lambda_1}{\Delta}-1}.$$

And the rest of the result directly follows.

Overview

Sparse Recovery: Formulated as LASSO Problems

$$\beta^{(k+1)} = \mathcal{S}_{\alpha/L}(\beta^{(k)} - \frac{1}{L}\mathbf{X}^\top(\mathbf{X}\beta^{(k)} - \mathbf{y})).$$

Here α is a coefficient controlling the sparsity penalty. We denote the transpose of the i-th row in $\mathbf{X}$ by $\mathbf{x}_i$, i.e., $\mathbf{x}_i = [\mathbf{X}^\top]_{:,i}$.
Algorithms:

- Gradient Descent: inefficent
- Learning Iterative Shrinkage Thresholding Algorithm (LISTA): feed-forward neural network
- Classical variants of LISTA, e.g. LISTA-CP The update rule in the k-th iteration:

$$\beta^{(k+1)} = \mathcal{S}_{\theta^{(k)}}\left(\beta^{(k)} - (\mathbf{D}^{(k)})^\top(\mathbf{X}\beta^{(k)} - \mathbf{y})\right), \tag{4}$$

  where $\{\theta^{(k)}, \mathbf{D}^{(k)}\}$ are learnable parameters.
- Transformer: LISTA-VM(LISTA with varying measurements)

Transformer Architecture

In this work, we set the activation function as the element-wise ReLU function. Next, we define the one-layer decoder-based Transformer structure.

Definition 11 (Transformer layer)

A one-layer decoder-based Transformer is parameterized by $\mathbf{\Theta} := \{\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}, (\mathbf{V}_m, \mathbf{Q}_m, \mathbf{K}_m)_{m\in[M]}\}$, denoted as $_\theta$. Therefore, give input sequence $\mathbf{H} \in \mathbb{R}^{d\times N}$, the output sequence is:

$$TF_\theta(\mathbf{H}) = MLP_{\{\mathbf{W}_1,\mathbf{W}_2,\mathbf{b}\}}\Big(Attn_{\{(\mathbf{V}_m,\mathbf{Q}_m,\mathbf{K}_m)\}}(\mathbf{H})\Big).$$

Embedding

Given an in-context sparse recovery instance $\mathcal{I} = (\mathbf{X}, \mathbf{y}, \mathbf{x}_{N+1})$ we embed the instance into an input sequence $\mathbf{H}^{(1)} \in \mathbb{R}^{(2d+2)\times(2N+1)}$ as follows:

$$\mathbf{H}^{(1)}(\mathcal{I}) = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_1 & \cdots & \mathbf{x}_N & \mathbf{x}_N & \mathbf{x}_{N+1} \\ 0 & y_1 & \cdots & 0 & y_N & 0 \\ \beta_1^{(1)} & \beta_2^{(1)} & \cdots & \beta_{2N-1}^{(1)} & \beta_{2N}^{(1)} & \beta_{2N+1}^{(1)} \\ 1 & 0 & \cdots & 1 & 0 & 1 \end{bmatrix}, \tag{5}$$

where $\{\beta_i^{(1)}\}_{i\in[2N+1]} \in \mathbb{R}^d$ are implicit parameter vectors initialized as $\mathbf{0}_d$, and $\mathbf{x}_i$ is the i-th column of the transposed measurement matrix, i.e, $[\mathbf{X}^\top]_{:,i}$. We note that a similar embedding structure is adopted in bai2023transformers.

Assumption 2

For $\mathbf{x} \sim P_{\mathbf{x}}$ and $\beta^* \sim P_\beta$, we assume $\|\mathbf{x}\| \leq b_{\mathbf{x}}$ and $\|\beta^*\|_1 \leq b_\beta$ almost surely. Besides, we consider the noiseless scenario where $\epsilon = \mathbf{0}$.

Equivalence between ICL and LISTA-VM

Theorem 12 (Equivalence between ICL and LISTA-VM)

With the Transformer structure described above, under Assumption 2, there exists a set of parameters in the Transformer so that for any $k \in [1:K]$, $n \in [N]$, *we have*

$$\beta_{2n+1}^{(k+1)} = \mathcal{S}_{\theta^{(k)}}\left(\beta_{2n+1}^{(k)} - \frac{1}{2n+1}\mathbf{M}^{(k)}[\mathbf{X}]_{1:n,:}^{\top}([\mathbf{X}]_{1:n,:}\beta_{2n+1}^{(k)} - \mathbf{y}_{1:n})\right), \quad (6)$$

where $\mathbf{M}^{(k)} \in \mathbb{R}^{d\times d}$ *is embedded in the* k*-th Transformer layer.*

Convergence of ICL

Theorem 13 (Convergence of ICL)

Let $\delta \in (0,1)$, $N_0 = 8(4S-2)^2 \frac{\log d + \log S - \log \delta}{c}$, $\alpha_n = -\log\left(1 - \frac{2}{3}\gamma + \gamma(2S-1)\sqrt{\frac{\log d - \log \delta}{nc}} + \sqrt{\frac{\log S - \log \delta}{nc}}\right)$, *where* c *is a positive constant and* γ *is a positive constant satisfies* $\gamma \leq \frac{3}{2}$. *For a* K*-layer Transformer model with the structure described previously, under Assumption 2, there exists a set of parameters such that for any randomly generated sparse recovery instance and any* $n \in [N_0 : N]$, *with probability at least* $1 - \delta$, *we have*

$$\left\| \beta_{2n+1}^{(K+1)} - \beta^* \right\| \leq b_\beta e^{-\alpha_n K}.$$

Approximation of Read-out Functions

Theorem 14

Under the same setting, for any $n \in [N_0 + 1 : N + 1]$, *with probability at least* $1 - n\delta - \delta'$, *we have*

$$\|y_n - \hat{y}_n\| \leq b_{\mathbf{x}}(1 - \frac{2}{3}\gamma)^K + \frac{c_4 K}{\sqrt{n}}(1 - \frac{2}{3}\gamma)^{K-1}$$

for a linear read-out function, and with probability at least $1 - \delta$, *we have* $\|y_n - \hat{y}_n\| \leq c_5 e^{-\alpha_n K}$ *for a query read-out function, where* c_4, c_5 *are constants.*

References

1. Bai(2023) "Transformers as statisticians: Provable in-context learning with in-context algorithm selection."
2. Vaswani(2017) "Attention is all you need."
3. He(2024) "Can transforeners perform PCA?"
4. Liu(2025) "On the learn-to-optimize capabilities of Transformer in in-context sparse recovery"