Provable In-context Learning for Mixture of Linear Regressions using Transformers

Anonymous authors
Paper under double-blind review

Abstract

We theoretically investigate the in-context learning capabilities of transformers in the context of learning mixtures of linear regression models. For the case of two mixtures, we demonstrate the existence of transformers that can achieve an accuracy, relative to the oracle predictor, of order $\tilde{\mathcal{O}}((d/n)^{1/4})$ in the low signal-to-noise ratio (SNR) regime and $\tilde{\mathcal{O}}(\sqrt{d/n})$ in the high SNR regime, where n is the length of the prompt, and d is the dimension of the problem. Additionally, we derive in-context excess risk bounds of order $\mathcal{O}(L/\sqrt{B})$, where B denotes the number of (training) prompts, and L represents the number of attention layers. The order of L depends on whether the SNR is low or high. In the high SNR regime, we extend the results to K-component mixture models for finite K. Extensive simulations also highlight the advantages of transformers for this task, outperforming other baselines such as the Expectation-Maximization algorithm.

1 Introduction

We investigate the in-context learning ability of transformers in addressing the mixture of regression (MoR) problem (De Veaux, 1989; Jordan & Jacobs, 1994). The MoR model is widely applied in various domains, including clustered federated learning, collaborative filtering, and healthcare (Deb & Holmes, 2000; Viele & Tong, 2002; Kleinberg & Sandler, 2008; Faria & Soromenho, 2010; Ghosh et al., 2020), to address heterogeneity in data, often arising from multiple data sources. We consider linear MoR models where independent and identically distributed samples $(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R}$, for $i = 1, \ldots, n$, are assumed to follow the model $y_i = \langle \beta_i, x_i \rangle + v_i$, where $v_i \sim \mathcal{N}(0, \vartheta^2)$ represents observation noise, independent of x_i, and $\beta_i \in \mathbb{R}^d$ is an unknown regression vector. Specifically, there are K distinct regression vectors $\{\beta_k^*\}_{k=1}^K$, and each β_i is independently drawn from these vectors according to the distribution $\{\pi_k\}_{k=1}^K$. The goal for a new test sample, x_{n+1}, is to predict its label y_{n+1}. Specifically, we are interested in the meta-learning setup for MoR (Kong et al., 2020).

In a recent intriguing work, through a mix of theory and experiments, Pathak et al. (2024) examined the performance of transformers for learning MoR models. However, their theoretical result suffers from the following major drawback: They only showed that the existence of a transformer architecture that is capable of implementing the *oracle* Bayes optimal predictor for the linear MoR problem. That is, they assume the availability of $\{\beta_k^*\}_{k=1}^K$ which are in practice unknown and are to be estimated. Hence, there remains a gap in the theoretical understanding of how transformers actually perform parameter estimation and prediction in MoR. Furthermore, their theoretical result is rather disconnected from their empirical observations which focused on in-context learning. Indeed, they leave open a theoretical characterization of the problem of in-context learning MoR (Pathak et al., 2024, Section 4). In this work, we show that transformers are actually capable of in-context learning linear MoR via implementing the Expectation-Maximization (EM) algorithm, a double-loop algorithm, wherein each inner loop involves multiple steps of gradient ascent.

The EM algorithm is a classic method for estimation and prediction in the MoR model (Balakrishnan et al., 2017; Kwon et al., 2019; Kwon & Caramanis, 2020; Wang et al., 2024). A major limitation of the EM algorithm is its tendency to converge to local maxima rather than the global maximum of the likelihood function. This issue arises because the algorithm's performance crucially depends on the initialization (Jin et al., 2016). To mitigate this, favorable initialization strategies based on spectral methods (Chaganty & Liang, 2013; Zhang et al., 2016; Chen et al., 2020) are typically employed alongside the EM algorithm. Via our experiments, we empirically demonstrate that trained transformers are capable of efficient prediction and estimation in the MoR model, while also considerably avoiding the initialization issues associated with the EM algorithm. In summary, we make the following **contributions** in this work:

- We demonstrate the existence of a transformer capable of learning mixture of two linear regression models by implementing the dual-loops of the EM algorithm. This construction involves the transformer performing multiple gradient ascent steps during each M-step of the EM algorithm. In Theorem 2.1, we derive precise bounds on the transformer's ability to approximate the *oracle predictor* in both low and high signal-to-noise (SNR) regimes. We extend this result to the case of finite-K mixtures in Theorem 4.1 for the high-SNR setting.
- In Theorem 2.2, we establish an excess risk bound for this constructed transformer, demonstrating its ability to achieve low excess risk under population loss conditions. These results collectively show that transformers can provably learn mixtures of linear regression models in-context.
- In Theorem 2.3, we analyze the sample complexity associated with pretraining these transformers using a limited number of in-context learning (ICL) training instances.
- As a byproduct of our analysis, we also derive convergence results with statistical guarantees for the gradient EM algorithm applied to a two-component mixture of regression models, where the M-step involves T steps of gradient ascent. We extend this approach to the multi-component case, improving upon previous works, such as Balakrishnan et al. (2017), which considered only a single step of gradient ascent.

1.1 Related works

Transformers and optimization algorithms: Garg et al. (2022) successfully demonstrated that transformers can be trained to perform in-context learning (ICL) for linear function classes, achieving results comparable to those of the optimal least squares estimator. Beyond their empirical success, numerous studies have sought to uncover the mechanisms by which transformers facilitate ICL. Recent investigations suggest that transformers may internally execute first-order Gradient Descent (GD) to perform ICL, a concept explored in depth by Akyürek et al. (2023), Bai et al. (2024), Von Oswald et al. (2023a), Von Oswald et al. (2023b), Ahn et al. (2024), and Zhang et al. (2024). Specifically, Akyürek et al. (2023) identified fundamental operations that transformers can execute, such as multiplication and affine transformations, showing that transformers can implement GD for linear regression using these capabilities. Building on this, Bai et al. (2024) provided detailed constructions illustrating how transformers can implement convex risk minimization across a wide range of standard machine learning problems, including least squares, ridge, lasso, and generalized linear models (GLMs). Further, Ahn et al. (2024) demonstrated that a single-layer linear transformer, when optimally parameterized, can effectively perform a single step of preconditioned GD. Zhang et al. (2024) expanded on this by showing that every one-step GD estimator, with a learnable initialization, can be realized by a linear transformer block (LTB) estimator.

Moving beyond first-order optimization methods, Fu et al. (2023) revealed that transformers can achieve convergence rates comparable to those of the iterative Newton's Method, which are exponentially faster than GD, particularly in the context of linear regression. These insights collectively highlight the sophisticated computational abilities of transformers in ICL, aligning closely with classical optimization techniques. In addition to exploring how transformers implement these mechanisms, recent studies have also focused on their training dynamics in the context of linear regression tasks; see, for example, Zhang et al. (2023) and Chen et al. (2024). In comparison to the aforementioned works, in the context of MoR, we demonstrate that transformers are capable of implementing double-loop algorithms like the EM algorithm.

EM Algorithm: The analysis of the standard EM algorithm for mixture of Gaussian and linear MoR models has a long-standing history (Wu (1983),McLachlan & Krishnan (2007), Tseng (2004)). Balakrishnan et al. (2017) first proved that EM algorithm converges at a geometric rate to a local region close to the maximum likelihood estimator with explicit statistical and computational rates of convergence. Subsequent works (Kwon et al., 2019; 2021) established improved convergence results for mixture of regression under different SNR conditions. Kwon & Caramanis (2020) extended these results to mixture of regression with many components. Gradient EM algorithm was first analyzed by Wang et al. (2015) and Balakrishnan et al. (2017). It is an immediate variant of the standard EM algorithm where the M-step is achieved by one-step gradient ascent rather than exact maximization. They proved that the gradient EM also can achieve the local convergence with explicit finite sample statistical rate of convergence. Global convergence for the case of two-components mixture of Gaussian model was show by Xu et al. (2016); Daskalakis et al. (2017); Wu & Zhou (2021). The case of unbalanced mixtures was handled by Weinberger & Bresler (2022). Penalized EM algorithm for handling high-dimensional mixture models was analyzed by Zhu et al. (2017), Yi & Caramanis (2015) and Wang et al. (2024), showing that gradient EM can achieve linear convergence to the unknown parameter under mild conditions.

2 Main Results

Mixture of Regression model: In this section, we explore the MoR problem involving two components. The underlying true model is described by the equation:

$$y_i = x_i^\top \beta_i + v_i \tag{1}$$

where $x_i \sim \mathcal{N}(0, I_d)$, $v_i \sim \mathcal{N}(0, \vartheta^2 I_d)$ denotes the noise term with variance ϑ^2, and β_i's are i.i.d. random vectors that taking the value $-\beta^*$ with probability $\frac{1}{2}$ and β^* with probability $\frac{1}{2}$. The parameter β^* is unknown.

Transformer architecture: We focus on transformers that handle the input sequence $H \in \mathbb{R}^{D\times N}$ by integrating attention layers and multi-layer perceptrons (MLPs). These transformers are structured to process the input by effectively mapping the complex interactions and dependencies between data points in the sequence, utilizing the capabilities of attention mechanisms to dynamically weigh the importance of different features in the context of regression analysis.

Definition 2.1. A attention layer with M heads is denoted as $\mathrm{Attn}_{\boldsymbol{\theta}}(\cdot)$ with parameters $\boldsymbol{\theta} = \{(V_m, Q_m, K_m)\}_{m\in[M]} \subset \mathbb{R}^{D\times D}$. On any input sequence $H \in \mathbb{R}^{D\times N}$, we have

$$\widetilde{H} = \mathrm{Attn}_{\boldsymbol{\theta}}(H) := H + \frac{1}{N}\sum_{m=1}^{M} \big(V_m H\big) \times \sigma\big(\big(Q_m H\big)^\top \big(K_m H\big)\big) \in \mathbb{R}^{D\times N}, \tag{2}$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is the activation function and D is the hidden dimension. In the vector form,

$$\tilde{h}_i = \big[\mathrm{Attn}_\theta(H)\big]_i = h_i + \sum_{m=1}^{M} \frac{1}{N} \sum_{j=1}^{N} \sigma\big(\big\langle Q_m h_i, K_m h_j\big\rangle\big) \cdot V_m h_j.$$

Remark 2.1. The prevalent choices for the activation function include the softmax function and the ReLU function. In our analysis in Section 3, Equation 2 employs a normalized ReLU activation, $t \mapsto \sigma(t)/N$, which is used for technical convenience. This modification does not impact the fundamental nature of the study.

Definition 2.2 (Attention only transformer)**.** An L-layer transformer, denoted as $\mathrm{TF}_\theta(\cdot)$, is a composition of L self-attention layers,

$$\mathrm{TF}_\theta(\cdot) = \mathrm{Attn}_{\theta^L} \circ \mathrm{Attn}_{\theta^{L-1}} \circ \cdots \circ \mathrm{Attn}_{\theta^1}(H)$$

where $H \in \mathbb{R}^{D\times N}$ is the input sequence, and the parameter $\boldsymbol{\theta} = \big(\theta^1, \ldots, \theta^L\big)$ consists of the attention layers $\theta^{(\ell)} = \big\{\big(V_m^{(\ell)}, Q_m^{(\ell)}, K_m^{(\ell)}\big)\big\}_{m\in[M^{(\ell)}]} \subset \mathbb{R}^{D\times D}$.

In the theory part, the input sequence $H \in \mathbb{R}^{D\times(n+1)}$ has columns

$$\begin{aligned} h_i &= [x_i, y_i', \mathbf{0}_{D-d-3}, 1, t_i]^\top, \\ h_{n+1} &= [x_{n+1}, y_{n+1}', \mathbf{0}_{D-d-3}, 1, 1]^\top \end{aligned} \tag{3}$$

where $t_i := 1\{i < n+1\}$ is the indicator for the training examples. Then the transformer TF_θ produce the output $\tilde{H} = \mathrm{TF}_\theta(H)$. The prediction $\hat{y}_{n+1}$ is derived from the $(d+1, n+1)$-th entry of $\tilde{H}$, denoted as $\hat{y}_{n+1} = \mathrm{read}_y(\tilde{H}) := \big(\tilde{h}_{n+1}\big)_{d+1}$. Our objective is to develop a fixed transformer architecture that efficiently conducts in-context learning for the mixture of regression problem, thereby providing a prediction $\hat{y}_{n+1}$ for y_{n+1} under an appropriate loss framework. Besides, the constructed transformer in Section 2 can also extract an estimate of the regression components, which is specified in Section 3.

Notation: For a vector $v \in \mathbb{R}^d$, its ℓ_2 norm is denoted by $\|v\|_2$. For a matrix $A \in \mathbb{R}^{d\times d}$, $\|A\|_{\text{op}}$ denotes the operator (spectral) norm of A. For the linear model Equation 1, we denote $\eta = \|\beta^*\|_2/\vartheta$ as the signal-noise-ratio (SNR). We denote the joint distribution of (x, y) in model Equation 1 by $\mathcal{P}_{x,y}$ and the distribution of x by $\mathcal{P}_x$. Besides, we denote the joint distribution of $(x_1, y_1, \ldots, x_n, y_n, x_{n+1}, y_{n+1})$ by $\mathcal{P}$, where $\{x_i, y_i\}_{i=1}^n$ are the input in the training prompt and x_{n+1} is the query sample. Besides, in Section 3, we use $y_i' \in \mathbb{R}$ defined as $y_i' = y_i t_i$ for $i = 1, \ldots, n, n+1$ to simplify our notation.

Evaluation: Let $f : H \mapsto \hat{y} \in \mathbb{R}$ be any procedure that takes a prompt H as input and outputs an estimate $\hat{y}$ on the query y_{n+1}. We define the mean squared error (MSE) by $\mathrm{MSE}(f) := \mathbb{E}_{\mathcal{P}}\big[\big(f(H) - y_{n+1}\big)^2\big]$. Finally, we define the function $f_{n,d,\delta}(a_1, a_2, a_3, a_4, a_5)$ as

$$f_{n,d,\delta}(a_1, a_2, a_3, a_4, a_5) := \Big(\frac{d}{n}\Big)^{a_1} \log^{a_5}(n^{a_2} d^{a_3}/\delta^{a_4}),$$

which will be used in the presentation of theorems in Section 2.1.

2.1 EXISTENCE OF TRANSFORMER FOR MIXTURE OF REGRESSION

In Theorem 2.1, we demonstrate the existence of a transformer capable of approximately implementing the EM algorithm. The performance of the transformer largely depends on the SNR. The threshold order of SNR is given by $\mathcal{O}\big(f_{n,d,\delta}(\frac{1}{4}, 1, 0, 0, \frac{1}{2})\big) = \mathcal{O}\big(d \log^2(n/\delta)/n\big)^{1/4}\big)$ for some small number δ. High SNR means the order of η is greater than $\mathcal{O}\big(f_{n,d,\delta}(\frac{1}{4}, 1, 0, 0, \frac{1}{2})\big)$, while low SNR means the order of η is smaller than $\mathcal{O}\big(f_{n,d,\delta}(\frac{1}{4}, 1, 0, 0, \frac{1}{2})\big)$. Generally, the transformer performs better in the high SNR settings compared to the low SNR settings. In Theorem 2.1, we show that there exists a transformer that implement EM algorithm internally.

Theorem 2.1. *Given input matrix H whose columns are given by Equation 3, there exists a transformer TF_θ, with the number of heads $M^{(\ell)} \leqslant M = 4$ in each attention layers, that can make prediction on y_{n+1} by implementing gradient EM algorithm of MoR problem where T steps of gradient descent are used in each M-step. When T is sufficiently large and the prompt length n satisfies*

$$n \geqslant \mathcal{O}\big(d \log^2(1/\delta)\big), \tag{4}$$

the transformer can achieve the prediction error $\Delta_y := |\,\mathrm{read}_y\big(\mathrm{TF}(H)\big) - x_{n+1}^\top \beta^{OR}|$ of order

$$\Delta_y = \begin{cases} \sqrt{\log(d/\delta)} f_{n,d,\delta}(\frac{1}{4}, 1, 0, 1, \frac{1}{2}) & \eta \leqslant \mathcal{O}\big(f_{n,d,\delta}(\frac{1}{4}, 1, 0, 0, \frac{1}{2})\big) \\ \sqrt{\log(d/\delta)} f_{n,d,\delta}(\frac{1}{2}, 1, 0, 1, 1) & \eta \geqslant \mathcal{O}\big(f_{n,d,\delta}(\frac{1}{4}, 1, 0, 0, \frac{1}{2})\big), \end{cases} \tag{5}$$

with probability at least $1 - \delta$, where β^{OR} is defined as

$$\beta^{OR} := \arg\min_{\beta \in \mathbb{R}^d} \mathbb{E}_{\mathcal{P}_{x,y}}\big[(x^\top\beta - y)^2\big] = \pi_1\beta^* - \pi_2\beta^* \equiv 0, \tag{6}$$

Furthermore, the second-to-last layer approximates β^*:

$$\|\,\mathrm{read}_\beta\big(\mathrm{TF}(H)\big) - \beta^*\|_2 = \begin{cases} \mathcal{O}\big(f_{n,d,\delta}(\frac{1}{4}, 1, 0, 1, \frac{1}{2})\big) & \eta \leqslant \mathcal{O}\big(f_{n,d,\delta}(\frac{1}{4}, 1, 0, 0, \frac{1}{2})\big) \\ \mathcal{O}\big(f_{n,d,\delta}(\frac{1}{2}, 1, 0, 1, 1)\big) & \eta \geqslant \mathcal{O}\big(f_{n,d,\delta}(\frac{1}{4}, 1, 0, 0, \frac{1}{2})\big) \end{cases}$$

with probability at least $1 - \delta$, where $\mathrm{read}_\beta(\mathrm{TF}(H)) = \big[\mathrm{TF}(H)\big]_{d+2,n+1}$ extracts the estimate of β^* in the output matrix.

Details of the proof of Equation 6 and Theorem 2.1 can be found in Appendix B.5. According to Theorem 2.1, the architecture of the constructed transformer varies primarily in the number of layers it includes. In general, with the prompt length n and dimension d held constant, the constructed transformer needs more training samples in the prompt in the low SNR settings to achieve the desired precision. The prediction error is order of $\tilde{\mathcal{O}}(\sqrt{d/n})$ under the high SNR settings, and is $\tilde{\mathcal{O}}((d/n)^{\frac{1}{4}})$ in the low SNR settings. Besides, under the high SNR settings, the constructed transformer needs $\mathcal{O}\big(\log(n/d)\big)$ attention layers, while it needs $\mathcal{O}\big(\log(\log(n/d))\sqrt{n/d}\big)$ attention layers in the low SNR settings.

2.2 Analysis of parameter estimator and prediction error via transformer

In Theorem 2.2, we provide the excess risk bound for the transformer constructed in Theorem 2.1

Theorem 2.2. *For any T being sufficiently large and the prompt length n satisfies condition Equation 4. Define the excess risk $\mathcal{R} := \mathbb{E}_{\mathcal{P}}\Big[(y_{n+1} - \mathrm{read}_y(\mathrm{TF}(H)))^2\Big] - \inf_\beta \mathbb{E}_{\mathcal{P}}\big[(x_{n+1}^\top \beta - y_{n+1})^2\big]$. Then the ICL prediction $\mathrm{read}_y(\mathrm{TF}(H))$ of the constructed transformer in Theorem 2.1 satisfies*

$$\mathcal{R} = \begin{cases} f_{n,d,\delta}(\frac{1}{2}, 1, 0, 0, 1) & 0 < \eta \leqslant \mathcal{O}\big(f_{n,d,\delta}(\frac{1}{4}, 1, 0, 0, \frac{1}{2})\big) \\ f_{n,d,\delta}(1, 1, 0, 0, 2) & \eta \geqslant \mathcal{O}\big(f_{n,d,\delta}(\frac{1}{4}, 1, 0, 0, \frac{1}{2})\big) \end{cases} . \tag{7}$$

Furthermore, $\inf_\beta \mathbb{E}_{\mathcal{P}}\big[(x_{n+1}^\top \beta - y_{n+1})^2\big] = \vartheta^2 + \|\beta^*\|_2^2$.

The main idea behind the proof for Theorem 2.1 and 2.2 is deferred to Section 3. Theorem 2.1 and Theorem 2.2 provide the first quantitative framework for end-to-end ICL in the mixture of regression problems, achieving desired precision. The order of the excess risk of the constructed transformer is order of $\mathcal{O}\big(d \log^2 n/n\big)$ under the high SNR settings, and is order of $\mathcal{O}\big(\sqrt{d/n} \log n\big)$ under the low SNR settings. These results represent an advancement over the findings in Pathak et al. (2024), which do not offer explicit error bounds such as Equation 5 to Equation 7.

2.3 Analysis of pre-training

We now analyze the sample complexity needed to pretrain the transformer with a limited number of ICL training instances. Following the ideas from Bai et al. (2024), we consider the square loss between the in-context prediction and the ground truth label:

$$\ell_{\mathrm{icl}}(\boldsymbol{\theta}; \mathbf{Z}) := \frac{1}{2}\Big[y_{n+1} - \mathrm{clip}_R\left(\mathrm{read}_y\left(\mathrm{TF}_{\boldsymbol{\theta}}(H)\right)\right)\Big]^2,$$

where $\mathbf{Z} := \big(H, y_{n+1}\big)$ is the training prompt, $\boldsymbol{\theta} = \big\{(K_m^{(\ell)}, Q_m^{(\ell)}, V_m^{(\ell)}) : \ell = 1, \ldots, L, m = 1, \ldots, M\big\}$ is the collection of parameters of the transformer and $\mathrm{clip}_R(t) := \mathrm{Proj}_{[-R,R]}(t)$ is the standard clipping operator with (a suitably large) radius $R \geqslant 0$ that varies in different problem setups to prevent the transformer from blowing up on tail events, in all our results concerning (statistical) in-context prediction powers. Additionally, the clipping operator can be employed to control the Lipschitz constant of the transformer $\mathrm{TF}_{\boldsymbol{\theta}}$ with respect to $\boldsymbol{\theta}$. In practical applications, it is common to select a sufficiently large clipping radius R to ensure that it does not alter the behavior of the transformer on any input sequence of interest. Denote $\|\boldsymbol{\theta}\|$ as the norm of transformer given by

$$\|\boldsymbol{\theta}\| := \max_{\ell \in [L]} \Big\{ \max_{m \in [M]} \Big\{ \|Q_m^{(\ell)}\|_{\mathrm{op}}, \|K_m^{(\ell)}\|_{\mathrm{op}} \Big\} + \sum_{m=1}^{M} \|V_m^{(\ell)}\|_{\mathrm{op}} \Big\}.$$

Our pretraining loss is the average ICL loss on B pretraining instances $\mathbf{Z}^{(1:B)} \overset{\text{iid}}{\sim} \pi$, and we consider the corresponding test ICL loss on a new test instance:

$$\hat{L}_{\mathrm{icl}}(\boldsymbol{\theta}) := \frac{1}{B} \sum_{j=1}^{B} \ell_{\mathrm{icl}}\big(\boldsymbol{\theta}; \mathbf{Z}^{(j)}\big) \qquad \text{and} \qquad L_{\mathrm{icl}}(\boldsymbol{\theta}) := \mathbb{E}_{\mathcal{P}}\big[\ell_{\mathrm{icl}}\big(\boldsymbol{\theta}; \mathbf{Z}\big)\big].$$

Our pretraining algorithm is to solve a standard constrained empirical risk minimization problem over transformers with L layers, M heads, and norm bounded by M':

$$\widehat{\boldsymbol{\theta}} := \arg\min_{\boldsymbol{\theta} \in \Theta_{M'}} \widehat{L}_{\mathrm{icl}}(\boldsymbol{\theta}), \quad \Theta_{M'} = \Big\{\boldsymbol{\theta} = (K_m^{(\ell)}, Q_m^{(\ell)}, V_m^{(\ell)}) : \max_{\ell \in [L]} M^{(\ell)} \leqslant M, \|\boldsymbol{\theta}\| \leqslant M'\Big\}. \tag{8}$$

Theorem 2.3 (Generalization for pretraining)**.** *With probability at least $1 - 3\xi$ (over the pretraining instances $\big\{\mathbf{Z}^{(j)}\big\}_{j \in [B]}$), the solution $\widehat{\boldsymbol{\theta}}$ to Equation 8 satisfies*

$$L_{\mathrm{icl}}(\widehat{\boldsymbol{\theta}}) \leqslant \inf_{\boldsymbol{\theta} \in \Theta_{B'}} L_{\mathrm{icl}}(\boldsymbol{\theta}) + \mathcal{O}\left((1 + \eta^{-2}) \log(2nB/\xi) \sqrt{\frac{(L)^2 (MD^2)\iota + \log(1/\xi)}{B}}\right)$$

where $\iota = \log\big(2 + \max\big\{M', B_x, B_y, (2B_y)^{-1}\big\}\big)$, $B_x = \sqrt{\log(ndB/\xi)}$, $B_y = \sqrt{2(1 + \eta^{-2})\|\beta^*\|_2^2 \log(2nB/\xi)}$, *$D$ is the hidden dimension and M is the number of heads.*

Remark 2.2. Under the low SNR settings, the constructed transformer generally requires more attention layers than those in the high SNR settings to achieve the same level of excess risk. In particular, for the constructed transformer in Theorem 2.1, $L = \mathcal{O}\big(T\log(\log(n/d))\sqrt{n/(d\log^2(n/\delta))}\big)$ under the low SNR settings, and $L = \mathcal{O}\big(T\log(n/d)\big)$ under the high SNR settings. Hence, with the number of samples n per prompt and dimension d fixed, the required number of prompts B to achieve a comparable excess pretraining risk under the high SNR settings is smaller than that under the low SNR settings.

3 Transformer Implements the Gradient-EM Algorithm

In this section, we illustrate that the constructed transformers in Theorem 2.1 can solve the MoR problem by implementing EM algorithm internally while GD is used in each M-step. Prior works (e.g. Balakrishnan et al. (2017), Kwon et al. (2019) and Kwon et al. (2021)) focused on the sample-based EM algorithm, typically employing closed-form solutions or one-step gradient approaches in the M-step. For general analysis, we explore the transformer's performance using T-step gradient descent within the EM algorithm. To simplify the analysis, we restrict our stepsize $\alpha \in (0,1)$ in each gradient descent step in M-step.

Attention layer can implement the one-step gradient descent. We first recall how the attention layer can implement one-step GD for a certain class of loss functions as demonstrated by Bai et al. (2024). Let $\ell : \mathbb{R}^2 \to \mathbb{R}$ be a loss function. Let $\widehat{L}_n(\beta) := \frac{1}{n}\sum_{i=1}^n \ell\big(\beta^\top x_i, y_i\big)$ denote the empirical risk with loss function ℓ on dataset $\{(x_i, y_i)\}_{i\in[n]}$, and we denote

$$\beta_{k+1} := \beta_k - \alpha\nabla\widehat{L}_n(\beta_k) \tag{9}$$

as the GD trajectory on $\widehat{L}_n$ with initialization $\beta_0 \in \mathbb{R}^d$ and learning rate $\alpha > 0$. The foundational concept of the construction presented in Theorem 2.1 is derived from Bai et al. (2024). It hinges on the condition that the partial derivative of the loss function, $\partial_s\ell : (s,t) \mapsto \partial_s\ell(s,t)$ (considered as a bivariate function), can be approximated by a sum of ReLU functions, which are defined as follows:

Definition 3.1 (Approximability by sum of ReLUs)**.** A function $g : \mathbb{R}^k \to \mathbb{R}$ is $(\varepsilon_{\text{approx}}, R, M, C)$-approximable by sum of ReLUs, if there exists a "(M, C)-sum of ReLUs" function

$$f_{M,C}(\mathbf{z}) = \sum_{m=1}^M c_m\sigma\big(\mathbf{a}_m^\top[\mathbf{z};1]\big) \quad \text{with} \sum_{m=1}^M |c_m| \leqslant C, \max_{m\in[M]} \|\mathbf{a}_m\|_1 \leqslant 1, \mathbf{a}_m \in \mathbb{R}^{k+1}, c_m \in \mathbb{R}$$

such that $\sup_{\mathbf{z}\in[-R,R]^k} |g(\mathbf{z}) - f_{M,C}(\mathbf{z})| \leqslant \varepsilon_{\text{approx}}$.

Suppose that the partial derivative of the loss function, $\partial_s\ell(s,t)$, is approximable by a sum of ReLUs. Then, T steps of GD, as described in Equation 9, can be approximately implemented by employing T attention layers within the transformer. This result is formally presented in Proposition E.1.

Transformer can implement the gradient-EM algorithm: Proposition E.1 illustrates how the transformer described in Theorem 2.1 is capable of learning from the MoR problem. Using proposition E.1, we can construct a transformer whose architecture consists of attention layers that implement GD for each M-step, followed by additional attention layers responsible for computing the necessary quantities in the E-step. Here is a summary of how the transformer implements the EM algorithm for the mixture of regression problem. Following the notation from Balakrishnan et al. (2017), we define the weight function:

$$w_\beta(x,y) = \frac{\exp\big\{-\frac{1}{2\vartheta^2}\big(y - x^\top\beta\big)^2\big\}}{\exp\big\{-\frac{1}{2\vartheta^2}\big(y - x^\top\beta\big)^2\big\} + \exp\big\{-\frac{1}{2\vartheta^2}\big(y + x^\top\beta\big)^2\big\}}.$$

Denote $\beta^{(t)}$ as the current parameter estimates of β^* in the EM algorithm for the MoR problem. During each M-step, the objective is to maximize the following loss function:

$$Q_n(\beta' \mid \beta^{(t)}) = \frac{-1}{2n}\sum_{i=1}^n \Big(w_{\beta^{(t)}}(x_i, y_i)\big(y_i - x_i^\top\beta'\big)^2 + \big(1 - w_{\beta^{(t)}}(x_i, y_i)\big)\big(y_i + x_i^\top\beta'\big)^2\Big). \tag{10}$$

The update $\beta^{(t+1)}$ is given by $\beta^{(t+1)} = \arg\max_{\beta' \in \Omega} Q_n(\beta' \mid \beta^{(t)})$. Lemma 3.1 below, proved in Section A, demonstrates that the function $\hat{L}_n^{(t)(\beta)}$ minimized in each M-step is approximable by a sum of ReLUs.

Lemma 3.1. For the function $\hat{L}_n^{(t)}(\beta) = \frac{1}{n}\sum_{i=1}^n l^{(t)}(x_i^\top \beta, y_i)$, where

$$l^{(t)}(x_i^\top \beta, y_i) = w_{\beta^{(t)}}(x_i, y_i)(y_i - x_i^\top \beta)^2 + (1 - w_{\beta^{(t)}}(x_i, y_i))(y_i + x_i^\top \beta)^2,$$

it holds that (1) $l^{(t)}(s,t)$ is convex in first argument; and (2) $\partial_s l^{(t)}(s,t)$ is $(0, +\infty, 4, 16)$-approximable by sum of ReLUs.

By Lemma 3.1, we can design attention layers with T layers that implement the T steps of GD for the empirical loss $\hat{L}_n^{(t)}(\beta')$ as outlined in Proposition E.1. We provide a concise demonstration of the entire process below. Starting with an appropriate initialization $\beta^{(0)}$, the first M-step minimizes the loss function:

$$\hat{L}_n^{(0)}(\beta) = \frac{1}{n}\sum_{i=1}^n \left\{ w_{\beta^{(0)}}(x_i, y_i)(y_i - x_i^\top \beta)^2 + (1 - w_{\beta^{(0)}}(x_i, y_i))(y_i + x_i^\top \beta)^2 \right\}.$$

Following Proposition E.1, given the input sequence formatted as $h_i = [x_i; y_i'; 0_d; 0_{D-2d-3}; 1; t_i]$, there exists a transformer with T attention layers that gives the output $\tilde{h}_i = [x_i; y_i'; \beta_T^{(0)}, 0_{D-2d-3}; 1; t_i]$. Furthermore, the existence of a transformer capable of computing the necessary quantities in the M-step is guaranteed by Proposition 1 from Pathak et al. (2024) and we restate this proposition in Section E in appendix.

It is worth mentioning that computing $w_{\beta^{(t)}}(x_i, y_i)$ in each M-step can be easily implemented by affine and softmax operation in Proposition E.2. Similar arguments can be made for the upcoming iterations of the EM algorithm and we summarize these results in Lemma 3.2 and 3.3.

Lemma 3.2. In each E-step, given the input $H^{(T+1)} = \left[h_1^{(T+1)}, \ldots, h_{n+1}^{(T+1)}\right]$ where

$$h_i^{(T+1)} = \left[x_i; y_i'; \beta_T^{(t)}; \mathbf{0}_{D-2d-3}; 1; t_i; w_{\beta_T^{(t-1)}}(x_i, y_i)\right]^\top, \quad i = 1, \ldots, n,$$
$$h_{n+1}^{(T+1)} = \left[x_i; x_{n+1}^\top \beta_T^{(t)}; \beta_T^{(t)}; \mathbf{0}_{D-2d-3}; 1; 1; 0\right]^\top,$$

there exists a transformer $\mathrm{TF}_E^{(t)}$ that can compute $w_{\beta_T^{(t)}}(x_i, y_i)$. Furthermore, the output sequence takes the form of

$$\tilde{h}_i^{(T+1)} = \left[x_i; y_i'; \beta_T^{(t)}; \mathbf{0}_{D-2d-4}; 1; t_i; w_{\beta_T^{(t)}}(x_i, y_i)\right]^\top, \quad i = 1, \ldots, n, \tag{11}$$
$$\tilde{h}_{n+1}^{(T+1)} = \left[x_i; x_{n+1}^\top \beta_T^{(t)}; \beta_T^{(t)}; \mathbf{0}_{D-2d-4}; 1; 1; 0\right]^\top. \tag{12}$$

Lemma 3.3. In each M-step, given the input matrix as Equation 11 and Equation 12, there exists a transformer $\mathrm{TF}_M^{(t)}$ with $T+1$ attention layers that can implement T steps of GD on the loss function $\hat{T}_n^{(t)}(\beta) = \frac{1}{n}\sum_{i=1}^n l^{(t)}(x_i^\top \beta, y_i)$, where $l^{(t)}(x_i^\top \beta, y_i) = w_{\beta_T^{(t)}}(x_i, y_i)(y_i - x_i^\top \beta)^2 + (1 - w_{\beta_T^{(t)}}(x_i, y_i))(y_i + x_i^\top \beta)^2$. Furthermore, the output sequence takes the form of

$$h_i^{(T+1)} = \left[x_i; y_i'; \beta_T^{(t+1)}; \mathbf{0}_{D-2d-3}; 1; t_i; w_{\beta_T^{(t)}}(x_i, y_i)\right]^\top, \quad i = 1, \ldots, n,$$
$$h_{n+1}^{(T+1)} = \left[x_i; x_{n+1}^\top \beta_T^{(t+1)}; \beta_T^{(t+1)}; \mathbf{0}_{D-2d-3}; 1; 1; 0\right]^\top.$$

The above results are proved in Section A. Combining all the architectures into one transformer, we have that there exists a transformer that can implement gradient descent EM algorithm for T_0 iterations (outer loops) and in each M-step (inner loops), it implements T steps of GD for function defined by Equation 10. Finally, following similar procedure in Theorem 1 of Pathak et al. (2024), the output of the transformer will give $\hat{\beta}^{\mathsf{OR}} := \pi_1 \beta_T^{(T_0+1)} - \pi_2 \beta_T^{(T_0+1)}$, which is an estimate of $\beta^{\mathsf{OR}} = \pi_1 \beta^* - \pi_2 \beta^*$ that minimizes the prediction MSE. The output is given by $\tilde{H} \in \mathbb{R}^{D \times (n+1)}$, whose columns are

$$\tilde{h}_i = [x_i, y_i', \beta_T^{(T_0+1)}, \mathbf{0}_{D-2d-4}, 1, t_i]^\top, \qquad i = 1, \ldots, n,$$
$$\tilde{h}_{n+1} = [x_{n+1}, x_{n+1}^\top \hat{\beta}^{\mathsf{OR}}, \beta_T^{(T_0+1)}, \mathbf{0}_{D-2d-4}, 1, 1]^\top.$$

4 MIXTURE OF REGRESSION WITH MORE THAN TWO COMPONENTS

In this section, we illustrate the existence of a transformer that can solve MoR problem with $K \geqslant 3$ components in general. Given the input matrix H as Equation 3 and initialization of $\pi_j^{(0)} = \frac{1}{K}$, there exists a transformer that implements E-steps and computes

$$\gamma_{ij}^{(t+1)} = \frac{\pi_j^{(t)} \prod_{\ell=1}^n \exp\Big(-\frac{1}{2\vartheta^2}\big(y_\ell - x_\ell^\top \beta_j^{(t)}\big)^2\Big)}{\sum_{j'=1}^k \pi_{j'}^{(t)} \prod_{\ell=1}^n \exp\Big(-\frac{1}{2\vartheta^2}\big(y_\ell - x_\ell^\top \beta_{j'}^{(t)}\big)^2\Big)}, \quad \pi_j^{(t+1)} = \frac{1}{n}\sum_{i=1}^n \gamma_{ij}^{(t+1)}, \tag{13}$$

since the computation in Equation 13 only contains scalar product, linear transformation and softmax operation. Next, following same procedure as before, one can construct T attention layers that implement gradient descent of the optimization problem

$$\min_{\beta\in\mathbb{R}^d} \left\{ \sum_{i=1}^K \sum_{\ell=1}^n \gamma_{ij}^{(t+1)} \big(y_\ell - \beta^\top x_\ell\big)^2 \right\}, \quad \text{for all } j \in [K],$$

as the gradient of loss $l(x_\ell^\top \beta, y_\ell) := \sum_{i=1}^k \gamma_{ij}^{(t+1)}\big(y_\ell - \beta^\top x_\ell\big)^2$ is convex in first argument and $\partial_s l(s,t)$ is $(0, +\infty, 4, 16)$ approximable by sum of ReLUs. Hence, the construction in Lemma 3.2 and Lemma 3.3 also holds. For theoretical analysis, we define pairwise distance R_{ij}^*, and $R_{\min}, R_{\max}$ as the smallest and largest distance between regression vectors of any pair of linear models: $R_{ij}^* = \big\|\beta_i^* - \beta_j^*\big\|_2$, $R_{\min} = \min_{i\neq j} R_{ij}^*$, $R_{\max} = \max_{i\neq j} R_{ij}^*$. The SNR of this problem is defined as the ratio of minimum pairwise distance versus standard deviation of noise $\eta := R_{\min}/\vartheta$. Also, we define $\rho_{j\ell} := \pi_\ell^*/\pi_j^*$ for $j \neq \ell$ and $\rho_\pi = \max_j \pi_j^*/\min_j \pi_j^*$ as the ratio of maximum mixing weight and minimum mixing weight, and $\pi_{\min} = \min_j \pi_j^*$. When the number of the components $K = 2$ and $\beta_1^* = -\beta_2^* = \beta^*$, the SNR reduces to $\eta = 2\|\beta^*\|_2/\vartheta$. Finally, the vector that minimizes the mean squared error of the prediction is given by

$$\beta^{\mathsf{OR}} := \arg\min_{\beta\in\mathbb{R}^d} \mathbb{E}_{\mathcal{P}_{x,y}}\big[(x^\top\beta - y)^2\big] = \sum_{\ell=1}^K \pi_\ell^* \beta_\ell^*.$$

The performance of the constructed transformers is guaranteed by Theorem 4.1 below.

Theorem 4.1. *Given the input matrix H in the form of Equation 3, there exists a transformer* TF *with the number of heads $M^{(\ell)} \leqslant M = 4$ in each attention layers. This transformer* TF *can make prediction on y_{n+1} by implementing gradient EM algorithm of MoR problem where T steps of gradient descent is used in each M-step. When L is sufficiently large and the prompt length n satisfies following condition*

$$n \geqslant \mathcal{O}\Big(\max\big\{ d\log^2(dK^2/\delta), \big(K^2/\delta\big)^{1/3}, d\log(K^2/\delta)/\pi_{\min} \big\}\Big),$$

under the SNR condition

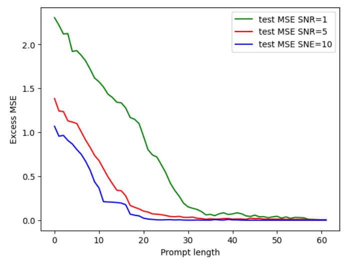
$$\eta \geqslant CK\rho_\pi \log(K\rho_\pi), \quad \textit{for a sufficiently large } C > 0,$$

equipped with $\mathcal{O}\big(T\log\big(n/d\big)\big)$ attention layers, the transformer has the prediction error $\Delta_y := |\operatorname{read}_y\big(\mathrm{TF}(H)\big) - x_{n+1}^\top \beta^{\mathsf{OR}}|$ bounded by

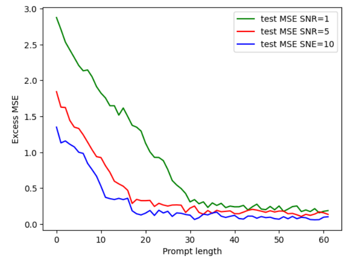
$$\Delta_y \leqslant \mathcal{O}\left(\sqrt{\log(d/\delta)} \left(\sqrt{\frac{d}{n} K\rho_\pi^2 \log^2\big(nK^2/\delta\big)} + \sqrt{\frac{dK\log(K^2/\delta)}{n\pi_{\min}}} \right)\right),$$

with probability at least $1 - 9\delta$.

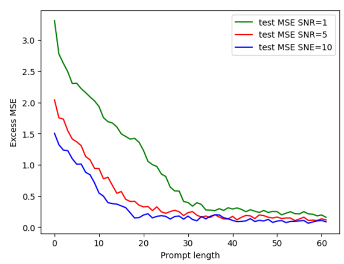
When $K = 2$, the order of prediction error bound reduces to that in Theorem 2.1 under the high SNR settings. The SNR condition required in Theorem 4.1 is stricter than that in Theorem 2.1 due to technical reasons in the proof. However, in the simulation, we see that the actual performance of the transformer is still good in the low SNR scenario when the number of components $K \geqslant 3$.



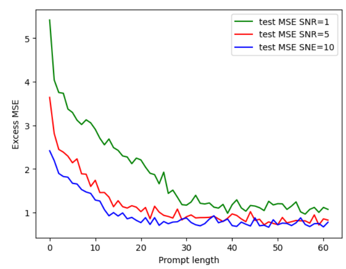
(a) Component $K = 2$



(b) Component $K = 3$



(c) Component $K = 5$



(d) Component $K = 20$

Figure 1: Excess testing risk of the transformer v.s. the prompt length with different SNRs.

5 Simulation Study

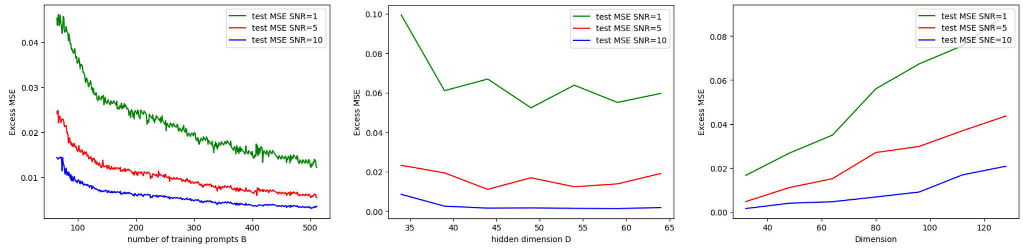
In this section, we present some results of training transformers on the prompts described in Section 2. We trained our transformers using Adam, with a constant step size of 0.001. For the general settings in the experiments, the dimension of samples $d = 32$. The number of training prompts are $B = 64$ by default (B is other value if otherwise stated). The hidden dimension are $D = 64$ by default (D is other value if otherwise stated). The training data x_i's are i.i.d. sampled from standard multivariate Gaussian distribution and the noise v_i's are i.i.d. sampled from normal distribution $\mathcal{N}(0, \vartheta^2)$. The regression coefficients are generated from standard multivariate normal and then normalized by its l_2 norm. Once the coefficient is generated, it is fixed. The excess MSE is reported. Each experiment is repeated by 20 times and the results is averaged over these 20 times.

The initializations of the transformer parameters for all our experiments are random standard Gaussian. As we will see from our results, transformers provide efficient prediction and estimation errors despite this global initialization. A possible explanation for this fact might be the overparametrization naturally available in the transformer architecture and the related need for overparametrization for estimation in mixture models (Dwivedi et al., 2020; Xu et al., 2024); we leave a theoretical investigation of this fact as intriguing future work.

Performance of transformers with different prompt length: In this experiment, we vary the number of components $K = 2, 3, 5$. For each case, we run the experiment with different SNR ($\eta = 1, 5, 10$). The x-axis is the prompt length, and the y-axis is the test MSE. The number of attention layers is given by $L = 4$. The performance results of the transformer are presented in Figure 1.

From Figure 1, we observe the following trends: (1) With the number of prompt lengths and other parameters held constant, the trained transformer exhibits a higher excess mean squared error (MSE) in the low SNR settings. (2) When the prompt length is very small, indicating an insufficient number of samples in the prompt, the resulting excess test MSE is high. However, with a sufficiently large prompt length, the performance of the transformers stabilizes and is effective across all SNR settings, leading to a relatively small excess test MSE. (3) Additionally, when the prompt length and SNR are fixed, an increase in the number of components tends to result in a larger excess test MSE.

Performance of transformers with different number of training prompts: In this experiment, we vary the number of training prompts B from 64 to 512. For each case, we run the experiment



(a) Plot of excess testing risk of the transformer v.s. the number of prompts with different SNRs.

(b) Plot of excess testing risk of the transformer v.s. the hidden dimension D with different SNRs.

(c) Plot of excess testing risk of the transformer v.s. the dimension d with different SNRs.

Figure 2: Effect of number of prompt B, hidden dimension D and input dimension d on the performance of the transformer

with two components ($K = 2$), different SNR ($\eta = 1, 5, 10$). The x-axis is the number of training prompts, and the y-axis is the test MSE. The length of training prompts is $n = 64$.

Figure 2a gives the performance of trained transformer with different number of training prompts under three different SNR settings. Based on Figure 2a, we observe that when the number of training prompts is already sufficiently large, the excess MSE is relatively small. Furthermore, as the number of training prompts increases, there is a general trend of decreasing in the excess MSE.

Performance of transformers with different number of hidden dimension: In this experiment, we vary the hidden dimension $D = 34, 64, 128$. For each case, we run the experiment with two components ($K = 2$), different SNR ($\eta = 1, 5, 10$). The x-axis is the hidden dimension D, and the y-axis is the excess test MSE. The performance of the trained transformer is presented in Figure 2b. In the low SNR settings, increasing the hidden dimension helps in improving the transformer's ability to learn the mixture of regression problem. However, excessively large hidden dimensions can lead to sparsity in the parameter matrix, which may not significantly enhance performance further.

Performance of transformers with different dimension d of samples: In this experiment, we fix the hidden dimension $D = 256$, the number of components $K = 2$, the number of prompts $B = 128$ and the prompt length is given by $n = 64$. The x-axis is the dimension d of the input sample x_i and y-axis is the excess test MSE. In this experiment, we evaluate the performance of the trained transformer for various dimensions $d = 32, 48, 64, 80, 96, 112, 128$. The performance of the transformer are presented in Figure 2c. Observations from this figure indicate that increasing the dimension d significantly raises the excess test MSE. Notably, this increase becomes more pronounced at the lower SNR levels.

6 Discussion

We explored the behavior of transformers in handling linear MoR problems, demonstrating their strong in-context learning capabilities through both theoretical analysis and empirical experiments. Specifically, we showed that transformers can internally implement the EM algorithm for linear MoR tasks. Our findings also reveal that transformer performance improves in high signal-to-noise ratio (SNR) settings and are less suseptible to initializations. Additionally, we examined the sample complexity involved in pretraining transformers with a finite number of ICL training instances, offering valuable insights into their practical performance.

Our empirical and theoretical findings point to several promising directions for future research. First, while our results demonstrate that transformers can internally implement the EM algorithm, investigating the use of looped transformers, as discussed in Giannou et al. (2023), could reduce architectural complexity in in-context linear MoR problems. Next, understanding the training dynamics of transformers for linear MoR problems remains a highly interesting and challenging task. Finally, extending these results to general non-linear MoR models would be a significant and impactful direction for future work.

References

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? Investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=0g0X4H8yN4I`.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, 36, 2024.

Sivaraman Balakrishnan, Martin J Wainwright, and Bin Yu. Statistical guarantees for the EM algorithm: From population to sample-based analysis. *The Annals of Statistics*, 45(1):77–120, 2017.

Arun Tejasvi Chaganty and Percy Liang. Spectral experts for estimating mixtures of linear regressions. In *International Conference on Machine Learning*, pp. 1040–1048. PMLR, 2013.

Sitan Chen, Jerry Li, and Zhao Song. Learning mixtures of linear regressions in subexponential time via fourier moments. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 587–600, 2020.

Siyu Chen, Heejune Sheen, Tianhao Wang, and Zhuoran Yang. Training dynamics of multi-head softmax attention for in-context learning: Emergence, convergence, and optimality. *arXiv preprint arXiv:2402.19442*, 2024.

Constantinos Daskalakis, Christos Tzamos, and Manolis Zampetakis. Ten steps of EM suffice for mixtures of two Gaussians. In *Conference on Learning Theory*, pp. 704–710. PMLR, 2017.

Richard D De Veaux. Mixtures of linear regressions. *Computational Statistics & Data Analysis*, 8 (3):227–245, 1989.

Partha Deb and Ann M Holmes. Estimates of use and costs of behavioural health care: A comparison of standard and finite mixture models. *Health economics*, 9(6):475–489, 2000.

Raaz Dwivedi, Nhat Ho, Koulik Khamaru, Martin J Wainwright, Michael I Jordan, and Bin Yu. Singularity, misspecification and the convergence rate of EM. *The Annals of Statistics*, 48(6): 3161–3182, 2020.

Susana Faria and Gilda Soromenho. Fitting mixtures of linear regressions. *Journal of Statistical Computation and Simulation*, 80(2):201–225, 2010.

Deqing Fu, Tian-Qi Chen, Robin Jia, and Vatsal Sharan. Transformers learn higher-order optimization methods for in-context learning: A study with linear models. *arXiv preprint arXiv:2310.17086*, 2023.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? A case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33:19586–19597, 2020.

Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In *International Conference on Machine Learning*, pp. 11398–11442. PMLR, 2023.

Chi Jin, Yuchen Zhang, Sivaraman Balakrishnan, Martin J Wainwright, and Michael I Jordan. Local maxima in the likelihood of Gaussian mixture models: Structural results and algorithmic consequences. *Advances in neural information processing systems*, 29, 2016.

Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural computation*, 6(2):181–214, 1994.

Jon Kleinberg and Mark Sandler. Using mixture models for collaborative filtering. *Journal of Computer and System Sciences*, 74(1):49–69, 2008.

Jason M Klusowski, Dana Yang, and WD Brinda. Estimating the coefficients of a mixture of two linear regressions by expectation maximization. *IEEE Transactions on Information Theory*, 65 (6):3515–3524, 2019.

Weihao Kong, Raghav Somani, Zhao Song, Sham Kakade, and Sewoong Oh. Meta-learning for mixed linear regression. In *International Conference on Machine Learning*, pp. 5394–5404. PMLR, 2020.

Jeongyeol Kwon and Constantine Caramanis. EM converges for a mixture of many linear regressions. In *International Conference on Artificial Intelligence and Statistics*, pp. 1727–1736. PMLR, 2020.

Jeongyeol Kwon, Wei Qian, Constantine Caramanis, Yudong Chen, and Damek Davis. Global convergence of the EM algorithm for mixtures of two component linear regression. In *Conference on Learning Theory*, pp. 2055–2110. PMLR, 2019.

Jeongyeol Kwon, Nhat Ho, and Constantine Caramanis. On the minimax optimality of the EM algorithm for learning two-component mixed linear regression. In *International Conference on Artificial Intelligence and Statistics*, pp. 1405–1413. PMLR, 2021.

Geoffrey J McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*. John Wiley & Sons, 2007.

Reese Pathak, Rajat Sen, Weihao Kong, and Abhimanyu Das. Transformers can optimally learn regression mixture models. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=sLkj91HIZU`.

Paul Tseng. An analysis of the EM algorithm and entropy-like proximal point methods. *Mathematics of Operations Research*, 29(1):27–44, 2004.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Kert Viele and Barbara Tong. Modeling with mixtures of linear regressions. *Statistics and Computing*, 12:315–330, 2002.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174. PMLR, 2023a.

Johannes Von Oswald, Eyvind Niklasson, Maximilian Schlegel, Seijin Kobayashi, Nicolas Zucchet, Nino Scherrer, Nolan Miller, Mark Sandler, Max Vladymyrov, Razvan Pascanu, et al. Uncovering mesa-optimization algorithms in transformers. *arXiv preprint arXiv:2309.05858*, 2023b.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Ning Wang, Xin Zhang, and Qing Mai. Statistical analysis for a penalized EM algorithm in high-dimensional mixture linear regression model. *Journal of Machine Learning Research*, 25(222): 1–85, 2024.

Zhaoran Wang, Quanquan Gu, Yang Ning, and Han Liu. High dimensional EM algorithm: Statistical optimization and asymptotic normality. *Advances in neural information processing systems*, 28, 2015.

Nir Weinberger and Guy Bresler. The EM algorithm is adaptively-optimal for unbalanced symmetric Gaussian mixtures. *Journal of Machine Learning Research*, 23(103):1–79, 2022.

CF Jeff Wu. On the convergence properties of the EM algorithm. *The Annals of statistics*, pp. 95–103, 1983.

Yihong Wu and Harrison H Zhou. Randomly initialized EM algorithm for two-component Gaussian mixture achieves near optimality in $\mathcal{O}(\sqrt{n})$ iterations. *Mathematical Statistics and Learning*, 4 (3), 2021.

Ji Xu, Daniel J Hsu, and Arian Maleki. Global analysis of expectation maximization for mixtures of two Gaussians. *Advances in Neural Information Processing Systems*, 29, 2016.

Weihang Xu, Maryam Fazel, and Simon S Du. Toward Global Convergence of Gradient EM for Over-Parameterized Gaussian Mixture Models. *arXiv preprint arXiv:2407.00490*, 2024.

Xinyang Yi and Constantine Caramanis. Regularized EM algorithms: A unified framework and statistical guarantees. *Advances in Neural Information Processing Systems*, 28, 2015.

Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *arXiv preprint arXiv:2306.09927*, 2023.

Ruiqi Zhang, Jingfeng Wu, and Peter L Bartlett. In-Context Learning of a Linear Transformer Block: Benefits of the MLP Component and One-Step GD Initialization. *arXiv preprint arXiv:2402.14951*, 2024.

Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I Jordan. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. *Journal of Machine Learning Research*, 17(102): 1–44, 2016.

Rongda Zhu, Lingxiao Wang, Chengxiang Zhai, and Quanquan Gu. High-dimensional variance-reduced stochastic gradient expectation-maximization algorithm. In *International Conference on Machine Learning*, pp. 4180–4188. PMLR, 2017.

A PROOF OF LEMMAS IN SECTION 3

In this section, we provide detailed proofs of the lemmas presented in Section 3.

Proof of Lemma 3.1. Note that

$$l^{(t)}(s,t) = w_{\beta^{(t)}}\big(x_i,y_i\big)(t-s)^2 + \big(1-w_{\beta^{(t)}}(x_i,y_i)\big)(t+s)^2.$$

Taking derivative w.r.t. the first argument yields

$$\partial_s l^{(t)}(s,t) = w_{\beta^{(t)}}(x_i,y_i)(-2)(t-s) + \big(1-w_{\beta^{(t)}}(x_i,y_i)\big)2(t+s),$$
$$\partial_s^2 l^{(t)}(s,t) = 2w_{\beta^{(t)}}\big(x_i,y_i\big) + 2\big(1-w_{\beta^{(t)}}(x_i,y_i)\big) = 2.$$

Hence, $l(s,t)$ is convex in the first argument and

$$\begin{aligned}\partial_s l^{(t)}(s,t) &= 2w_{\beta^{(t)}}(x_i,y_i)(s-t) + 2\big(1-w_{\beta^{(t)}}(x_i,y_i)\big)(s+t)\\ &= 2w_{\beta^{(t)}}(x_i,y_i)[2\sigma((s-t)/2) - 2\sigma(-(s-t)/2)]\\ &\quad + 2\big(1-w_{\beta^{(t)}}(x_i,y_i)\big)[2\sigma((s+t)/2) - 2\sigma(-(s+t)/2)].\end{aligned}$$

Here $c_1 = 4w_{\beta^{(t)}}(x_i,y_i)$, $c_2 = -4w_{\beta^{(t)}}(x_i,y_i)$, $c_3 = 4(1-w_{\beta^{(t)}}(x_i,y_i))$ and $c_4 = -4(1-w_{\beta^{(t)}}(x_i,y_i))$. Now, we have $|c_1|+|c_2|+|c_3|+|c_4| \leqslant 16$ and $\partial_s l(s,t)$ is $(0,+\infty,4,16)$-approximable by sum of ReLUs. $\square$

Proof of Lemma 3.2. Note that the output of M-step after t-th iteration is given by $H^{(T+1)} = \big[h_1^{(T+1)},\ldots,h_{n+1}^{(T+1)}\big]$ where

$$h_i^{(T+1)} = \big[x_i; y_i'; \beta_T^{(t)}; \mathbf{0}_{D-2d-3}; 1; t_i; w_{\beta_T^{(t-1)}}(x_i,y_i)\big]^\top, \quad i=1,\ldots,n$$
$$h_{n+1}^{(T+1)} = \big[x_i; x_{n+1}^\top\beta_T^{(t)}; \beta_T^{(t)}; \mathbf{0}_{D-2d-3}; 1; 1; 0\big]^\top,$$

i.e.

$$H^{(T+1)} = \begin{bmatrix} x_1 & x_2 & \ldots & x_n & x_{n+1}\\ y_1' & y_2' & \ldots & y_n' & x_{n+1}^\top\beta_T^{(t)}\\ \beta_T^{(t)} & \beta_T^{(t)} & \ldots & \beta_T^{(t)} & \beta_T^{(t)}\\ \mathbf{0}_{D-2d-3} & \mathbf{0}_{D-2d-3} & \ldots & \mathbf{0}_{D-2d-3} & \mathbf{0}_{D-2d-3}\\ 1 & 1 & \ldots & 1 & 1\\ t_1 & t_2 & \ldots & t_n & 1\\ w_{\beta_T^{(t-1)}}(x_1,y_1) & w_{\beta_T^{(t-1)}}(x_2,y_2) & \ldots & w_{\beta_T^{(t-1)}}(x_n,y_n) & 0 \end{bmatrix}.$$

After copy down and scale operation, the output is given by

$$H^{(T+1)}(1) = \begin{bmatrix} x_1 & x_2 & \ldots & x_n & x_{n+1}\\ y_1' & y_2' & \ldots & y_n' & x_{n+1}^\top\beta_T^{(t)}\\ \beta_T^{(t)} & \beta_T^{(t)} & \ldots & \beta_T^{(t)} & \beta_T^{(t)}\\ -\beta_T^{(t)} & -\beta_T^{(t)} & \ldots & -\beta_T^{(t)} & -\beta_T^{(t)}\\ \mathbf{0}_{D-3d-3} & \mathbf{0}_{D-3d-3} & \ldots & \mathbf{0}_{D-3d-3} & \mathbf{0}_{D-3d-3}\\ 1 & 1 & \ldots & 1 & 1\\ t_1 & t_2 & \ldots & t_n & 1\\ w_{\beta_T^{(t-1)}}(x_1,y_1) & w_{\beta_T^{(t-1)}}(x_2,y_2) & \ldots & w_{\beta_T^{(t-1)}}(x_n,y_n) & 0 \end{bmatrix}.$$

After affine operation, the output is given by

$$H^{(L+1)}(2) = \begin{bmatrix} x_1 & x_2 & \ldots & x_n & x_{n+1}\\ y_1' & y_2' & \ldots & y_n' & x_{n+1}^\top\beta_L^{(t)}\\ \beta_T^{(t)} & \beta_T^{(t)} & \ldots & \beta_T^{(t)} & \beta_T^{(t)}\\ -\beta_T^{(t)} & -\beta_T^{(t)} & \ldots & -\beta_T^{(t)} & -\beta_T^{(t)}\\ r_1 & r_2 & \ldots & r_n & 0\\ \mathbf{0}_{D-3d-4} & \mathbf{0}_{D-3d-4} & \ldots & \mathbf{0}_{D-3d-4} & \mathbf{0}_{D-3d-4}\\ 1 & 1 & \ldots & 1 & 1\\ t_1 & t_2 & \ldots & t_n & 1\\ w_{\beta_T^{(t-1)}}(x_1,y_1) & w_{\beta_T^{(t-1)}}(x_2,y_2) & \ldots & w_{\beta_T^{(t-1)}}(x_n,y_n) & 0 \end{bmatrix}.$$

After another affine operation, the output is given by

$$H^{(T+1)}(3) = \begin{bmatrix} x_1 & x_2 & \dots & x_n & x_{n+1} \\ y_1' & y_2' & \dots & y_n' & x_{n+1}^\top \beta_T^{(t)} \\ \beta_T^{(t)} & \beta_T^{(t)} & \dots & \beta_T^{(t)} & \beta_T^{(t)} \\ -\beta_T^{(t)} & -\beta_T^{(t)} & \dots & -\beta_T^{(t)} & -\beta_T^{(t)} \\ r_1 & r_2 & \dots & r_n & 0 \\ \tilde{r}_1 & \tilde{r}_2 & \dots & \tilde{r}_n & 0 \\ \mathbf{0}_{D-3d-5} & \mathbf{0}_{D-3d-5} & \dots & \mathbf{0}_{D-3d-5} & \mathbf{0}_{D-3d-5} \\ 1 & 1 & \dots & 1 & 1 \\ t_1 & t_2 & \dots & t_n & 1 \\ w_{\beta_T^{(t-1)}}(x_1, y_1) & w_{\beta_T^{(t-1)}}(x_2, y_2) & \dots & w_{\beta_T^{(t-1)}}(x_n, y_n) & 0 \end{bmatrix}.$$

After softmax operation, the output is given by

$$H^{(T+1)}(4) = \begin{bmatrix} x_1 & x_2 & \dots & x_n & x_{n+1} \\ y_1' & y_2' & \dots & y_n' & x_{n+1}^\top \beta_T^{(t)} \\ \beta_T^{(t)} & \beta_T^{(t)} & \dots & \beta_T^{(t)} & \beta_T^{(t)} \\ -\beta_T^{(t)} & -\beta_T^{(t)} & \dots & -\beta_T^{(t)} & -\beta_T^{(t)} \\ r_1 & r_2 & \dots & r_n & 0 \\ \tilde{r}_1 & \tilde{r}_2 & \dots & \tilde{r}_n & 0 \\ \mathbf{0}_{D-3d-5} & \mathbf{0}_{D-3d-5} & \dots & \mathbf{0}_{D-3d-5} & \mathbf{0}_{D-3d-5} \\ 1 & 1 & \dots & 1 & 1 \\ t_1 & t_2 & \dots & t_n & 1 \\ w_{\beta_T^{(t)}}(x_1, y_1) & w_{\beta_T^{(t)}}(x_2, y_2) & \dots & w_{\beta_T^{(t)}}(x_n, y_n) & 0 \end{bmatrix}.$$

After copy over operation, the output is given by

$$H^{(T+1)}(5) = \begin{bmatrix} x_1 & x_2 & \dots & x_n & x_{n+1} \\ y_1' & y_2' & \dots & y_n' & x_{n+1}^\top \beta_T^{(t)} \\ \beta_T^{(t)} & \beta_T^{(t)} & \dots & \beta_T^{(t)} & \beta_T^{(t)} \\ \mathbf{0}_{D-2d-3} & \mathbf{0}_{D-2d-3} & \dots & \mathbf{0}_{D-2d-3} & \mathbf{0}_{D-2d-3} \\ 1 & 1 & \dots & 1 & 1 \\ t_1 & t_2 & \dots & t_n & 1 \\ w_{\beta_T^{(t)}}(x_1, y_1) & w_{\beta_T^{(t)}}(x_2, y_2) & \dots & w_{\beta_T^{(t)}}(x_n, y_n) & 0 \end{bmatrix}. \tag{14}$$

Finally, this transformer gives the output matrix $H_M^{(T+1)}$ as Equation 14. $\square$

Proof of Lemma 3.3. The conceptual basis of the proof draws from the theorem discussed in Bai et al. (2024). By Proposition C.2 in Bai et al. (2024), there exists a function $f : \mathbb{R}^2 \to \mathbb{R}$ of form

$$f(s,t) = \sum_{m=1}^{4} c_m \sigma\big(a_m s + b_m t + d_m\big) \quad \text{with} \quad \sum_{m=1}^{4} |c_m| \leqslant 16, |a_m| + |b_m| + |d_m| \leqslant 1, \forall m \in [4],$$

such that $\sup_{(s,t)\in\mathbb{R}^2} \big|f(s,t) - \partial_s \ell(s,t)\big| \leqslant \varepsilon$. Next, in each attention layer, for every $m \in [4]$, we define matrices $Q_m, K_m, V_m \in \mathbb{R}^{D\times D}$ such that

$$Q_m h_i = \begin{bmatrix} a_m \beta \\ b_m \\ d_m \\ -2 \\ 0 \\ 0 \end{bmatrix}, \quad K_m h_j = \begin{bmatrix} x_j \\ y_j' \\ 1 \\ R\big(1 - t_j\big) \\ 0 \\ 0 \end{bmatrix}, \quad V_m h_j = -\frac{(N+1)\eta c_m}{N} \cdot \begin{bmatrix} \mathbf{0}_d \\ 0 \\ x_j \\ \mathbf{0}_{D-2p-1} \\ 0 \end{bmatrix}$$

where D is the hidden dimension which is a constant multiple of d. In the last attention layers, the heads $\big\{\big(Q_m^{(T+1)}, K_m^{(T+1)}, V_m^{(T+1)}\big)\big\}_{m=1,2}$ satisfies

$$Q_1^{(T+1)} h_i^{(T)} = \big[x_i; \mathbf{0}_{D-d+1}\big], \quad K_1^{(T+1)} h_j^{(T)} = \big[\beta_T^{(t+1)}; \mathbf{0}_{D-d+1}\big], \quad V_1^{(T+1)} h_j^{(T)} = \big[\mathbf{0}_d; 1; \mathbf{0}_{D-d}\big],$$
$$Q_2^{(T+1)} h_i^{(T)} = \big[x_i; \mathbf{0}_{D-d+1}\big], \quad K_2^{(T+1)} h_j^{(T)} = \big[-\beta_T^{(t+1)}; \mathbf{0}_{D-d}\big], \quad V_2^{(T+1)} h_j^{(T)} = \big[\mathbf{0}_d; -1; \mathbf{0}_{D-d}\big].$$

The output of this transformer gives the matrix

$$H^{(T+1)} = \begin{bmatrix} x_1 & x_2 & \dots & x_n & x_{n+1} \\ y_1' & y_2' & \dots & y_n' & x_{n+1}^\top \beta_T^{(t+1)} \\ \beta_T^{(t+1)} & \beta_T^{(t+1)} & \dots & \beta_T^{(t+1)} & \beta_T^{(t+1)} \\ \mathbf{0}_{D-2d-3} & \mathbf{0}_{D-2d-3} & \dots & \mathbf{0}_{D-2d-3} & \mathbf{0}_{D-2d-3} \\ 1 & 1 & \dots & 1 & 1 \\ t_1 & t_2 & \dots & t_n & 1 \\ w_{\beta_T^{(t)}}(x_1, y_1) & w_{\beta_T^{(t)}}(x_2, y_2) & \dots & w_{\beta_T^{(t)}}(x_n, y_n) & 0 \end{bmatrix}.$$

□

B Proof of Theorem 2.1

In this section, we give the proof of the estimation and prediction bound presented in Theorem 2.1. In Section 3, the transformer described in Lemma 3.3, which is equipped with L layers, implements the M-step of the EM algorithm by performing T steps of gradient descent on the empirical loss $\hat{L}_n^{(t)}(\beta)$. Therefore, it is sufficient to analyze the behavior of the sample-based EM algorithm in which T steps of gradient descent are implemented during each M-step.

To begin, we define some notations that are utilized in the proof. We denote $\tilde{\beta}^{(0)}$ as any fixed initialization for the EM algorithm. The transformer described in Theorem 2.1 addresses the following optimization problem:

$$\text{argmin}\left\{\hat{L}_n^{(0)}(\beta) = \frac{1}{n}\sum_{i=1}^n \left\{w_{\beta^{(0)}}(x_i, y_i)(y_i - x_i^\top \beta)^2 + (1 - w_{\beta^{(0)}}(x_i, y_i))(y_i + x_i^\top \beta)^2\right\}\right\}$$

for some weight function $w_{\beta^{(0)}} \in (0, 1)$. The transformer generates a sequence $\beta_1^{(0)}, \dots, \beta_L^{(0)}$, with $\beta_\ell^{(0)} \to \tilde{\beta}^{(1)}$ as $L \to \infty$. More generally, we denote $\tilde{\beta}^{(t)}$ as the minimizer of the loss function $\hat{L}_n^{(t-1)}(\beta)$ at each M-step. Additionally, $\beta_1^{(0)}, \cdots, \beta_L^{(0)}$ represents the sequence generated by applying L attention layers of the constructed transformer in Lemma 3.3.

The approach to analyzing the convergence behavior of the transformer's output, $\mathrm{TF}(H)$, involves examining the performance of the sample-based gradient EM algorithm. This analysis is conducted by coupling the finite sample EM with the population EM, drawing on methodologies from Balakrishnan et al. (2017) and Kwon et al. (2019).

B.1 Results in population gradient EM algorithm for MoR problem

In this section, we present some results regarding the population EM algorithm. Given the current estimator of the parameter β^* to be $\beta^{(t)}$. The population gradient EM algorithm maximizes (see Balakrishnan et al. (2017) and Kwon et al. (2019))

$$Q\big(\beta \mid \beta^{(t)}\big) = -\frac{1}{2}\mathbb{E}\Big[w_{\beta^{(t)}}(X, Y)\big(Y - \langle X, \beta\rangle\big)^2 + \big(1 - w_{\beta^{(t)}}(X, Y)\big)\big(Y + \langle X, \beta\rangle\big)^2\Big],$$

whose gradient is given by $\mathbb{E}\big[\tanh\big(\frac{1}{\vartheta^2} Y X^\top \beta^{(t)}\big) Y X - \beta\big]$. Rather than using the standard population EM update

$$\tilde{\beta}^{(t+1)} = \arg\max_\beta Q(\beta \mid \beta^{(t)}) = \mathbb{E}\Big[\tanh\Big(\frac{1}{\vartheta^2} Y X^\top \beta^{(t)}\Big) Y X\Big] \tag{15}$$

the output after applying T steps of gradient descent is employed as the subsequent estimator for the parameter β^*, i.e.

$$\beta^{(t+1)} = (1-\alpha)^T \beta^{(t)} + \big(1 - (1-\alpha)^T\big)\mathbb{E}\Big[\tanh\Big(\frac{1}{\vartheta^2} Y X^\top \beta^{(t)}\Big) Y X\Big], \tag{16}$$

where $\alpha \in (0, 1)$ is the step size of the gradient descent.

In each iteration of the population gradient EM algorithm, the current iterate is denoted by β, the next iterate by β' and the standard EM update based on Equation 15 by $\tilde{\beta}'$. We concentrate on a single iteration of the population EM, which yields the next iterate β'. Consequently, Equation 16 becomes:

$$\beta' = (1-\alpha)^T \beta + \big(1-(1-\alpha)^T\big)\tilde{\beta}'. \tag{17}$$

We employ techniques similar to those used in Kwon et al. (2019) for basis transformation. By selecting $v_1 = \beta/\|\beta\|_2$ in the direction of the current iterate and v_2 as the orthogonal complement of v_1 within the span of $\{\beta, \beta^*\}$, we extend these vectors to form an orthonormal basis $\{v_1, \dots, v_d\}$ in $\mathbb{R}^d$. To simplify notation, we define:

$$b_1 := \langle \beta, v_1\rangle = \|\beta\|_2, \qquad b_1^* := \langle \beta^*, v_1\rangle \qquad b_2^* := \langle \beta^*, v_2\rangle, \tag{18}$$

which represent the coordinates of the current estimate β and β^*. The next iterate β' can then be expressed as:

$$\beta' = (1-\alpha)^T b_1 v_1 + \big(1-(1-\alpha)^T\big)\mathbb{E}\Bigg[\tanh\Big(\frac{\alpha_1 b_1}{\vartheta^2}Y\Big)Y\sum_{i=1}^d \alpha_i v_i\Bigg] \tag{19}$$

based on spherical symmetry of Gaussian distribution. The expectation is taken over $\alpha_i \sim \mathcal{N}(0,1)$ and $Y \mid \alpha_i \sim \mathcal{N}(\alpha_1 b_1^* + \alpha_2 b_2^*, \vartheta^2)$. Without loss of generality, we assume that $b_1, b_1^*, b_2^* \geqslant 0$.

Lemma B.1 is analogous to Lemma 1 from Kwon et al. (2019). It provides an explicit expression for β' within the established basis system, demonstrating among other insights that β' resides within the $\operatorname{span}\{\beta, \beta^*\}$. Consequently, all estimators of β^* generated by the population gradient EM algorithm remain confined within the $\operatorname{span}\{\beta^{(0)}, \beta^*\}$

Lemma B.1. Suppose that $\alpha \in (0,1)$. Define $\vartheta_2^2 := \vartheta^2 + b_2^{*2}$. We can write $\beta' = b_1' v_1 + b_2' v_2$, where b_1' and b_2' satisfy

$$b_1' = (1-\alpha)^T b_1 + \big(1-(1-\alpha)^T\big)\big(b_1^* S + R\big), \tag{20}$$

$$b_2' = \big(1-(1-\alpha)^T\big) b_2^* S. \tag{21}$$

Here, $S \geqslant 0$ and $R > 0$ are given explicitly by

$$S := \mathbb{E}_{\alpha_1\sim\mathcal{N}(0,1), y\sim\mathcal{N}(0,\vartheta_2^2)}\big[\tanh\big(\tfrac{\alpha_1 b_1}{\vartheta^2}(y+\alpha_1 b_1^*)\big) + \tfrac{\alpha_1 b_1}{\vartheta^2}(y+\alpha_1 b_1^*)\tanh'\big(\tfrac{\alpha_1 b_1}{\vartheta^2}(y+\alpha_1 b_1^*)\big)\big] \tag{22}$$

and

$$R := \big(\vartheta^2 + \|\beta^*\|_2^2\big)\mathbb{E}_{\alpha_1\sim\mathcal{N}(0,1), y\sim\mathcal{N}(0,\vartheta_2^2)}\Big[\frac{\alpha_1^2 b_1}{\vartheta^2}\tanh'\Big(\frac{\alpha_1 b_1}{\vartheta^2}\big(y+\alpha_1 b_1^*\big)\Big)\Big]. \tag{23}$$

Moreover, $S = 0$ iff $b_1 = 0$ or $b_1^* = 0$.

Proof. The proof of Lemma B.1 is directly adapted from the argument used in Lemma 1 from Kwon et al. (2019), applying Equation 19 for our specific context. In Equation 19, the inner expectation over y is independent of α_i for $i \geqslant 3$. Consequently, taking the expectation over α_i for $i \geqslant 3$ results in zero, confirming that β' remains within the plane spanned by v_1, v_2. This allows us to express β' as $\beta' = b_1' v_1 + b_2' v_2$ with

$$b_1' = (1-\alpha)^T b_1 + \big(1-(1-\alpha)^T\big)\mathbb{E}_{\alpha_1,\alpha_2}\Big[\mathbb{E}_{Y|\alpha_1,\alpha_2}\Big[\tanh\Big(\frac{b_1\alpha_1}{\vartheta^2}Y\Big)Y\Big]\alpha_1\Big], \tag{24}$$

$$b_2' = \big(1-(1-\alpha)^T\big)\mathbb{E}_{\alpha_1,\alpha_2}\Big[\mathbb{E}_{Y|\alpha_1,\alpha_2}\Big[\tanh\Big(\frac{b_1\alpha_1}{\vartheta^2}Y\Big)Y\Big]\alpha_2\Big], \tag{25}$$

where the expectation is taken over $\alpha_i \sim \mathcal{N}(0,1)$, and $y \mid \alpha_i \sim \mathcal{N}(\alpha_1 b_1^* + \alpha_2 b_2^*, \vartheta^2)$. The computation from Equation 24 and Equation 25 to Equation 22 and Equation 23 is identical to that in Lemma 1 of Kwon et al. (2019). $\square$

The findings in Lemma B.1 align with Lemma 1 from Klusowski et al. (2019). As the number of iterations T approaches infinity, the estimator β' converges to the standard population EM update

$$\beta^{(t)} \to \mathbb{E}_{X\sim\mathcal{N}(0,I)}\Big[\Big(\mathbb{E}_{Y|X\sim N(\langle X,\beta^*\rangle,\vartheta^2)}\Big[\tanh\Big(\frac{\langle X,\beta^{(t-1)}\rangle}{\vartheta^2}Y\Big)Y\Big]\Big)X\Big].$$

For any number of steps T, the angle between β' and β^* is consistently smaller than that between β and β^*. This can be observed by noting that:

$$0 \leqslant \tan\angle\big(\beta', \beta\big) = \frac{b_2'}{b_1'} = \frac{\big(1-(1-\alpha)^T\big)b_2^* S}{(1-\alpha)^T b_1 + \big(1-(1-\alpha)^T\big)\big(b_1^* S + R\big)} \leqslant \frac{b_2^*}{b_1^*} = \tan\angle\big(\beta^*, \beta\big). \tag{26}$$

These relationships demonstrate the geometric convergence properties of the estimation process. Motivated by Equation 26, we examine the behavior of the angle between the iterates $\beta^{(t)}$ and β^*. For clarity, we use θ_0, θ, and θ' to denote the angles formed by β^* with $\beta^{(0)}$ (the initial iterate), β (the current iterate), and β' (the next iterate), respectively. Using the coordinate representation of β' Equation 20 and Equation 21, the cosine and sine of θ' can be expressed by

$$\cos\theta' = \frac{(1-\alpha)^T b_1 b_1^* + (1-(1-\alpha)^T)(S\|\beta^*\|_2^2 + R b_1^*)}{\|\beta^*\|_2 \sqrt{(1-\alpha)^{2T} b_1^2 + (1-(1-\alpha)^T)^2 \left(R^2 + S^2\|\beta^*\|_2^2 + 2RSb_1^*\right) + 2(1-\alpha)^T b_1 (1-(1-\alpha)^T)\left(b_1^* S + R\right)}},$$

$$\sin\theta' = \frac{(1-\alpha)^T b_1 b_2^* + (1-(1-\alpha)^T) R b_2^*}{\|\beta^*\|_2 \sqrt{(1-\alpha)^{2T} b_1^2 + (1-(1-\alpha)^T)^2 \left(R^2 + S^2\|\beta^*\|_2^2 + 2RSb_1^*\right) + 2(1-\alpha)^T b_1 (1-(1-\alpha)^T)\left(b_1^* S + R\right)}}.$$

Lemma B.2. There exists a non-decreasing function $\varphi(\lambda)$ on $\lambda \in [0, 1]$ such that

$$\begin{aligned}
\varphi(0) &= \frac{1}{\sqrt{1 + (S/R)^2\|\beta^*\|_2^2 + 2(S/R)b_1^*}}, \\
\varphi(1) &= 1.
\end{aligned}$$

As long as $\theta \in [\frac{\pi}{3}, \frac{\pi}{2})$ and $\alpha \in (0, 1)$, it holds that

$$\sin\theta' \leqslant \varphi((1-\alpha)^T)\sin\theta$$

and

$$\varphi(0) = \frac{1}{\sqrt{1 + (S/R)^2\|\beta^*\|_2^2 + 2(S/R)b_1^*}} \leqslant \left(\sqrt{1 + \frac{2\eta^2}{1+\eta^2}\cos^2\theta}\right)^{-1} < 1.$$

Similarly,

$$\cos\theta' \geqslant \phi((1-\alpha)^T)\cos\theta$$

where

$$\begin{aligned}
\phi(0) &= \sqrt{1 + \frac{b_2^{*2}(3\|\beta^*\|_2^2 + 2\vartheta^2)}{(\|\beta^*\|_2^2 + \vartheta^2)^2 + b_1^{*2}(3\|\beta^*\|_2^2 + 2\vartheta^2)}} > 1, \\
\phi(1) &= 1.
\end{aligned}$$

Proof. We provide the proof for the sine case, and the proof for the cosine case follows a similar approach. Define $\lambda = (1-\alpha)^T \in (0, 1]$, we have

$$\begin{aligned}
\sin\theta' &= \frac{b_2^*}{\|\beta^*\|_2} \frac{\lambda b_1 + (1-\lambda)R}{\sqrt{\big(\lambda b_1 + (1-\lambda)(b_1^* S + R)\big)^2 + \big(\lambda \cdot 0 + (1-\lambda)(b_2^* S)\big)^2}} \\
&= \sin\theta \frac{\lambda b_1 + (1-\lambda)R}{\sqrt{\big(\lambda b_1 + (1-\lambda)(b_1^* S + R)\big)^2 + \big(\lambda \cdot 0 + (1-\lambda)(b_2^* S)\big)^2}}.
\end{aligned}$$

Then we define the function $\varphi(\lambda)$ to be

$$\varphi(\lambda) := \frac{\lambda b_1 + (1-\lambda)R}{\sqrt{\big(\lambda b_1 + (1-\lambda)(b_1^* S + R)\big)^2 + \big(\lambda \cdot 0 + (1-\lambda)(b_2^* S)\big)^2}}.$$

By symmetry, one can assume that the angles $\angle\langle\beta, \beta^*\rangle, \angle\langle\tilde{\beta}', \beta^*\rangle < \frac{\pi}{2}$. The non-decreasing property of $\varphi(\lambda)$ can be easily verified by the fact that β' is located on the line segment between the current iterate β and standard population EM updates $\tilde{\beta}$ based on Equation 17. $\square$

In the remainder of this section, we discuss the convergence of the gradient population EM algorithm in terms of distance, as presented in Theorem B.1.

Theorem B.1. *Assume that $\theta < \pi/8$, and define $\vartheta_2^2 = \vartheta^2 + b_2^{*2}$. If $b_2^* < \vartheta$ or $\frac{\vartheta_2^2}{\vartheta^2} b_1 < b_1^*$, then we have*

$$\|\beta' - \beta^*\|_2 \leqslant \big((1-\alpha)^T + \big(1-(1-\alpha)^T\big)\kappa\big)\|\beta - \beta^*\|_2 + \big(1-(1-\alpha)^T\big)\kappa\big(16\sin^3\theta\big)\|\beta^*\|_2 \frac{\eta^2}{1+\eta^2},$$

where $\kappa = \left(\sqrt{1+\min\left(\frac{\vartheta_2^2}{\vartheta^2}b_1, b_1^*\right)^2/\vartheta_2^2}\right)^{-1}$. Otherwise, we have

$$\|\beta' - \beta^*\|_2 \leqslant \big((1-\alpha)^T + 0.6\big(1-(1-\alpha)^T\big)\big)\|\beta - \beta^*\|_2.$$

Proof. The proof of this theorem is a direct corollary of Theorem 4 from Kwon et al. (2019) by noticing that

$$\begin{aligned}\|\beta' - \beta^*\|_2 &= \big\|(1-\alpha)^T\beta + \big(1-(1-\alpha)^T\big)\tilde{\beta}' - \beta^*\big\|_2 \\ &\leqslant (1-\alpha)^T\|\beta - \beta^*\|_2 + \big(1-(1-\alpha)^T\big)\|\tilde{\beta}' - \beta^*\|_2.\end{aligned}$$

□

B.2 Results in sample-based EM algorithm for MoR problem

In this section, we present results concerning the convergence of the sample-based gradient EM algorithm. We begin by deriving the update rule for the sample-based gradient EM algorithm, which incorporates T steps of gradient descent. Starting from the previous estimate, $\beta^{(t-1)}$, we define $\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^n x_i x_i^\top$. The new estimate, $\beta^{(t)}$, is obtained by applying T steps of gradient descent to the loss function $\hat{L}_n^{(t-1)}(\beta)$, specifically:

$$\begin{aligned}&\beta^{(t)} = \beta_T^{(t-1)} \\ =&\left(I - \frac{\alpha}{n}\sum_{i=1}^n x_i x_i^\top\right)\beta_{T-1}^{(t-1)} + \frac{\alpha}{n}\sum_{i=1}^n \tanh\left(\frac{1}{\vartheta^2}y_i x_i^\top \beta^{(t-1)}\right)y_i x_i \\ =&(I-\alpha\hat{\Sigma})\left[(I-\alpha\hat{\Sigma})\beta_{T-2}^{(t-1)} + \frac{\alpha}{n}\sum_{i=1}^n \tanh\left(\frac{1}{\vartheta^2}y_i x_i^\top \beta^{(t-1)}\right)y_i x_i\right] \\ &\quad + \frac{\alpha}{n}\sum_{i=1}^n \tanh\left(\frac{1}{\vartheta^2}y_i x_i^\top \beta^{(t-1)}\right)y_i x_i \\ =&(I-\alpha\hat{\Sigma})^T\beta^{(t-1)} + \alpha\cdot(\alpha\hat{\Sigma})^{-1}\big(I-(I-\alpha\hat{\Sigma})^T\big)\frac{1}{n}\sum_{i=1}^n \tanh\left(\frac{1}{\vartheta^2}y_i x_i^\top \beta^{(t-1)}\right)y_i x_i.\end{aligned}$$

For the analysis in the remainder of this section, we denote the current iteration as β, the subsequent iteration resulting from T steps of sample-based gradient descent as $\tilde{\beta}'$, and the iteration following T steps of population-based gradient descent as β'. By define $\hat{\mu} := \frac{1}{n}\sum_{i=1}^n \tanh\left(\frac{1}{\vartheta^2}y_i x_i^\top \beta\right)y_i x_i$ and $\mu := \mathbb{E}\tanh\left(\frac{1}{\vartheta^2}YX^\top\beta\right)YX$, we have

$$\begin{aligned}\tilde{\beta}' &= (I-\alpha\hat{\Sigma})^T\beta + \hat{\Sigma}^{-1}\big(I-(I-\alpha\hat{\Sigma})^T\big)\hat{\mu}, \\ \beta' &= (I-\alpha I)^T\beta + \big(I-(I-\alpha I)^T\big)\mu.\end{aligned}$$

In the previous analysis,

$$\tilde{\beta}' - \beta^* = (I - \alpha\hat{\Sigma})^T(\beta - \beta^*) + \big(I - (I - \alpha\hat{\Sigma})^T\big)\big(\hat{\Sigma}^{-1}\hat{\mu} - \beta^*\big),$$

$$\begin{aligned}
\hat{\Sigma}^{-1}\hat{\mu} - \beta^* &= \hat{\Sigma}^{-1}\left(\frac{1}{n}\sum_{i=1}^{n} y_i x_i \tanh\left(\frac{y_i\langle x_i, \beta\rangle}{\vartheta^2}\right) - \mathbb{E}_y \frac{1}{n}\sum_{i=1}^{n} y_i x_i \tanh\left(\frac{y_i\langle x_i, \beta^*\rangle}{\vartheta^2}\right)\right)\\
&= \underbrace{\hat{\Sigma}^{-1}}_{:=I} \underbrace{\left(\frac{1}{n}\sum_{i=1}^{n} y_i x_i \tanh\left(\frac{y_i\langle x_i, \beta\rangle}{\vartheta^2}\right) - \mathbb{E}_y \frac{1}{n}\sum_{i=1}^{n} y_i x_i \tanh\left(\frac{y_i\langle x_i, \beta\rangle}{\vartheta^2}\right)\right)}_{:=II}\\
&\quad + \hat{\Sigma}^{-1} \underbrace{\left(\mathbb{E}_y \frac{1}{n}\sum_{i=1}^{n} y_i x_i \tanh\left(\frac{y_i\langle x_i, \beta\rangle}{\vartheta^2}\right) - \mathbb{E}_y \frac{1}{n}\sum_{i=1}^{n} y_i x_i \tanh\left(\frac{y_i\langle x_i, \beta^*\rangle}{\vartheta^2}\right)\right)}_{:=III}.
\end{aligned}$$

Then $\|I\|_{\mathrm{op}} = 1 + \mathcal{O}\Big(\sqrt{\frac{d}{n}}\Big)$ by standard concentration result and it requires $n \geqslant \mathcal{O}\big(d\log^2(1/\delta)\big)$ in the end. Conditioning on the sample covariance matrix has bounded spectral norm, $\|II\|_2 = O\Big(\sqrt{\frac{d}{n}}\Big)$. Finally, for each fixed β satisfying $\|\beta\|_2 \geqslant \frac{\|\beta^*\|_2}{10}$, and its angle with β^*, θ is less than $\frac{\pi}{70}$, with $n = \mathcal{O}\big(\frac{d}{\epsilon^2}\big)$, $\|III\|_2 \leqslant \big(0.95 + \epsilon/\sqrt{d}\big)\|\beta - \beta^*\|_2$.

This can be improved by

$$\begin{aligned}
\tilde{\beta}' - \beta^* &= (I - \alpha\hat{\Sigma})^T\beta + \hat{\Sigma}^{-1}\big(I - (I - \alpha\hat{\Sigma})^T\big)\frac{1}{n}\sum_{i=1}^{n} \tanh\Big(\frac{1}{\vartheta^2} y_i x_i^\top \beta\Big) y_i x_i - \beta^*\\
&= (I - \alpha\hat{\Sigma})^T(\beta - \beta^*) + \big(I - (I - \alpha\hat{\Sigma})^T\big) \underbrace{\left[\frac{1}{n}\sum_{i=1}^{n} \hat{\Sigma}^{-1} \tanh\Big(\frac{1}{\vartheta^2} y_i x_i^\top \beta\Big) y_i x_i - \beta^*\right]}_{:=A},
\end{aligned}$$

$$\begin{aligned}
A = \hat{\Sigma}^{-1}\Bigg[&\underbrace{\mathbb{E}_{X,Y}\big[XY\Delta_{(X,Y)}(\beta)\big]}_{:=A_1} + \underbrace{\frac{1}{n}\sum_i X_i Y_i \Delta_{(X_i,Y_i)}(\beta) - \mathbb{E}_{X,Y}\big[XY\Delta_{(X,Y)}(\beta)\big]}_{:=A_2}\\
&\underbrace{+\frac{1}{n}\sum_i x_i y_i \tanh\big(y_i x_i^\top \beta^*/\vartheta^2\big) - \mathbb{E}_{y_i|x_i}\Big[\frac{1}{n}\sum_i x_i y_i \tanh\big(y_i x_i^\top \beta^*/\vartheta^2\big)\Big]}_{:=A_3}\Bigg],
\end{aligned}$$

where $\Delta_{(X,Y)}(\beta) := \tanh\big(yx^\top\beta/\vartheta^2\big) - \tanh\big(yx^\top\beta^*/\vartheta^2\big)$. Then

$$\begin{aligned}
&A_1 < 0.9\|\beta - \beta^*\|_2,\\
&A_2 \leqslant (\|\beta - \beta^*\|_2 + 1)\sqrt{d\log^2\big(n\|\beta^*\|_2/\delta\big)/n},\\
&A_3 \leqslant C\sqrt{d\log(1/\delta)/n},
\end{aligned}$$

with probability at least $1 - \delta$.

B.3 Convergence results under the high SNR setting

We first present the results for parameter estimation under the high SNR regime.

Lemma B.3. For any given $r > 0$, there exists a universal constant $c > 0$ such that with probability at least $1 - \delta$.

$$\sup_{\|\beta\|_2 \leqslant r} \big\|\hat{\Sigma}^{-1}\hat{\mu} - \mu\big\|_2 \leqslant cr\sqrt{d\log^2(n/\delta)/n}$$

where

$$\mu = \mathbb{E}\big[XY \tanh\big(YX^\top\beta\big)\big],$$
$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} \tanh\Big(\frac{y_i x_i^\top \beta}{\vartheta^2}\Big) y_i x_i,$$
$$\hat{\Sigma} = \frac{1}{n}\sum_i x_i x_i^\top.$$

Lemma B.4. For each fixed β, with probability at least $1 - \exp(-cn) - 6^d \exp\big(-\frac{nt^2}{72}\big)$

$$\Big\|\frac{1}{n}\sum_{i=1}^{n} y_i \boldsymbol{x}_i \tanh\big(y_i\langle x_i, \beta\rangle\big) - \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{y_i}\big[y_i x_i \tanh\big(y_i\langle x_i, \beta\rangle\big)\big]\Big\|_2 \leqslant t$$

for some absolute constant $c > 0$.

Theorem B.2. *Suppose that $\eta \geqslant \mathcal{O}\big(d\log^2(n/\delta)/n\big)^{1/4}$ for some absolute constant C and $\|\beta^{(0)}\| \geqslant 0.9\|\beta^*\|$ and $\cos\angle\big(\beta^*, \beta^{(0)}\big) \geqslant 0.95.$, let $\{\beta^{(t)}\}$ be the iterates of sample-based gradient EM algorithm, then there exists a constant $\gamma_2 \in (0,1)$ such that*

$$\|\beta^{(t)} - \beta^*\|_2 \leqslant \gamma_2^t + \frac{1}{1-\gamma_2}\mathcal{O}\Big(\sqrt{d\log^2(n/\delta)/n}\Big)$$

holds with probability at least $1 - 5\delta$.

Proof. Without loss of generality, we can assume that $\vartheta = 1$. Denote β as the current iterate, and $\tilde{\beta}'$ as the next sample-based iterate. We first consider

$$\begin{aligned}
\tilde{\beta}' - \beta^* &= (I - \alpha\hat{\Sigma})^T\beta + \hat{\Sigma}^{-1}\big(I - (I - \alpha\hat{\Sigma})^T\big)\frac{1}{n}\sum_{i=1}^{n}\tanh\Big(\frac{1}{\vartheta^2} y_i x_i^\top \beta\Big) y_i x_i - \beta^* \\
&= (I - \alpha\hat{\Sigma})^T(\beta - \beta^*) + \big(I - (I - \alpha\hat{\Sigma})^T\big)\underbrace{\Bigg[\frac{1}{n}\sum_{i=1}^{n}\hat{\Sigma}^{-1}\tanh\Big(\frac{1}{\vartheta^2} y_i x_i^\top \beta\Big) y_i x_i - \beta^*\Bigg]}_{:=A}.
\end{aligned}$$

We prove the results in two cases, i.e. $\eta \geqslant 1$ and $C_0\big(d\log^2(n/\delta)/n\big)^{1/4} \leqslant \eta \leqslant 1$ for some universal constant C_0. When $\eta \geqslant 1$, based on the analysis in Kwon et al. (2021), with probability at least $1 - 5\delta$,

$$\begin{aligned}
\|A\|_2 &\leqslant \Big(0.9 + \sqrt{d\log^2\big(n\|\beta^*\|_2/\delta\big)/n)}\Big)\|\beta - \beta^*\| + C_1\sqrt{d\log^2\big(n\|\beta^*\|_2/\delta\big)/n} \\
&\leqslant \gamma\|\beta - \beta^*\|_2 + C_1\sqrt{d\log^2\big(n\|\beta^*\|_2/\delta\big)/n} \qquad (27)
\end{aligned}$$

where $\gamma = 0.9 + \sqrt{d\log^2\big(n\|\beta^*\|_2/\delta\big)/n)}$. By standard concentration results on $\hat{\Sigma} - I$, it holds that with $n \geqslant \mathcal{O}(d\log^2(1/\delta))$, $\|(I - \alpha\hat{\Sigma})^T\|_{\text{op}} \leqslant (1 - \alpha/2)^T$ with probability at least $1 - \delta$ for appropriately small α. Along with Equation 27,

$$\begin{aligned}
\|\tilde{\beta}' - \beta^*\|_2 &\leqslant \big(1 - \frac{\alpha}{2}\big)^T\|\beta - \beta^*\|_2 + \big(1 - \big(1 - \frac{\alpha}{2}\big)^T\big)\|A\|_2 \\
&\leqslant \Big[\big(1 - \frac{\alpha}{2}\big)^T + \big(1 - \big(1 - \frac{\alpha}{2}\big)^T\big)\gamma\Big]\|\beta - \beta^*\|_2 \\
&\quad + \big(1 - \big(1 - \frac{\alpha}{2}\big)^T\big)C_1\sqrt{d\log^2\big(n\|\beta^*\|_2/\delta\big)/n}. \qquad (28)
\end{aligned}$$

Define $\epsilon(n,\delta) = \big(1 - \big(1 - \frac{\alpha}{2}\big)^T\big)C_1\sqrt{d\log^2\big(n\|\beta^*\|_2/\delta\big)/n}$ and $\gamma_2 = \big(1 - \frac{\alpha}{2}\big)^T + \big(1 - \big(1 - \frac{\alpha}{2}\big)^T\big)\gamma$. As long as $\gamma < 1$, we can iterate over t based on Equation 28 and obtain

$$\begin{aligned}
\|\beta^{(t)} - \beta^*\| &\leqslant \gamma_2\|\beta^{(t-1)} - \beta^*\|_2 + \epsilon(n,\delta) \leqslant \gamma_2^2\|\beta^{(t-2)} - \beta^*\|_2 + (1 + \gamma_2)\epsilon(n,\delta) \\
&\leqslant \gamma_2^t\|\beta^{(0)} - \beta^*\|_2 + \frac{1}{1-\gamma_2}\epsilon(n,\delta).
\end{aligned}$$

In the remaining part of the proof, we present an analysis of the convergence behavior of the sample-based gradient EM algorithm when $C_0\big(d\log^2(n/\delta)/n\big)^{1/4} \leqslant \eta \leqslant 1$. By Lemma 3 from Kwon et al. (2021), it holds that

$$\big\|\mathbb{E}\big[\tanh\big(YX^\top\beta\big)YX\big] - \beta^*\|_2 \leqslant \big(1 - \frac{1}{8}\|\beta^*\|_2^2\big)\|\beta - \beta^*\|_2.$$

To systematically analyze the convergence, we categorize the iterations into several epochs. We define $\bar{C}_0 = \|\beta^{(0)} - \beta^*\|_2$ and assume that during each l^{th} epoch, the distance $\|\beta - \beta^*\|_2$ lies within the interval $[\bar{C}_0 2^{-l-1}, \bar{C}_0 2^{-l}]$. This stratification is conceptual and does not impact the practical implementation of the EM algorithm. The key idea in this part is the same as Kwon et al. (2021). During the l^{th} epoch, the improvement in the population gradient EM updates must exceed the statistical error for convergence to occur, formalized as:

$$\frac{1}{8}\big(1 - (1 - \frac{\alpha}{2})^T\big)\|\beta^*\|_2^2\|\beta - \beta^*\|_2 \geqslant 2cr\sqrt{d\log^2(n/\delta)/n}$$

where c is the constant in Lemma B.3. By setting $r = \|\beta^*\| + \bar{C}_0 2^{-l}$ and using triangle inequality $\|\beta\|_2 \leqslant \|\beta^*\|_2 + \|\beta - \beta^*\|_2$, in l^{th} epoch when

$$\begin{aligned}\frac{1}{8}\big(1 - (1 - \frac{\alpha}{2})^T\big)\|\beta^*\|^2\bar{C}_0 2^{-l-1} &\geqslant 2cr\sqrt{d\log^2(n/\delta)/n}\\ &\geqslant 4c\big(\|\beta^*\| + \bar{C}_0 2^{-l}\big)\sqrt{d\log^2(n/\delta)/n},\end{aligned}$$

is guaranteed to be true, then it holds that

$$\begin{aligned}\|A\|_2 &\leqslant \Big(1 - \frac{1}{16}\|\beta^*\|_2^2\Big)\|\beta - \beta^*\|_2\\ \|\beta' - \beta^*\|_2 &\leqslant \Big[\big(1 - \frac{\alpha}{2}\big)^T + \big(1 - \big(1 - \frac{\alpha}{2}\big)^T\big)\Big(1 - \frac{1}{16}\|\beta^*\|_2^2\Big)\Big]\|\beta - \beta^*\|_2.\end{aligned}$$

Recall that $\eta \geqslant \mathcal{O}\big((d\log^2(n/\delta)/n)^{\frac{1}{4}}\big)$, then with appropriately set constants

$$\|\beta^*\|^2 \geqslant (c_1 + 1)\sqrt{d\log^2(n/\delta)/n},$$

we can deduce that β moves progressively closer to β^* as long as $\bar{C}_0 2^{-l} \leqslant c_2\|\beta^*\|_2^{-1}\sqrt{d\log^2(n/\delta)/n}$. This process requires $\mathcal{O}\big(\|\beta^*\|_2^{-2}\big)$ iterations per epoch, and after $\mathcal{O}(\log(n/d))$ epochs, the error bound $\|\beta - \beta^*\|_2 \leqslant c_2\|\beta^*\|_2^{-1}\sqrt{\frac{d\log^2(n/\delta)}{n}}$ is expected to hold. Thus, the convergence rate for $\beta^{(t)}$ towards β^* is quantified as:

$$\|\beta^{(t)} - \beta^*\|_2 \leqslant \gamma_2^t\|\beta^{(0)} - \beta^*\|_2 + \frac{1}{1-\gamma_2}\sqrt{d\log^2(n/\delta)/n}.$$

□

B.4 Convergence results under Low SNR settings

We present several auxiliary lemmas that will be utilized in analyzing the convergence results for sample-based gradient EM iterates.

Lemma B.5 (Lemma 6 in Kwon et al. (2021))**.** There exists some universal constants $c_u > 0$ such that,

$$\|\beta\|_2\big(1 - 4\|\beta\|_2^2 - c_u\|\beta^*\|_2^2\big) \leqslant \big\|\mathbb{E}\big[\tanh\big(YX^\top\beta\big)YX\big]\big\|_2 \leqslant \|\beta\|_2\big(1 - \|\beta\|_2^2 + c_u\|\beta^*\|_2^2\big).$$

Theorem B.3. *When $\eta \leqslant C_0(d\log^2(n/\delta)/n)^{1/4}$, there exist universal constants $C_3, C_4 > 0$ such that the sample-based gradient EM updates initialized with $\|\beta^{(0)}\|_2 \leqslant 0.2$ return $\beta^{(t)}$ that satisfies*

$$\|\beta^{(t)} - \beta^*\|_2 \leqslant \mathcal{O}\Big(\big(d\log^2 n/n\big)^{\frac{1}{4}}\Big)$$

with probability at least $1-\delta$ after $t \geqslant C_4\big(1 - \big(1 - \alpha/2\big)^T\big)^{-1}\log(\log(n/d))\sqrt{n/\big(d\log^2(n/\delta)\big)}$ iterations.

Proof. The proof argument follows the similar localization argument used in Theorem B.2. Define $\epsilon(n,\delta) := C\sqrt{d\log^2(n/\delta)/n}$ with some absolute constant $C > 0$. We assume that we start from the initialization region where $\|\beta\|_2 \leqslant \epsilon^{\alpha_0}(n,\delta)$ for some $\alpha_0 \in [0, 1/2)$. Suppose that $\epsilon^{\alpha_{l+1}}(n,\delta) \leqslant \|\beta\|_2 \leqslant \epsilon^{\alpha_l}(n,\delta)$ at the l^{th} epoch for $l \geqslant 0$. We let $C > 0$ sufficiently large such that

$$\epsilon(n,\delta) \geqslant 4c_u\|\beta^*\|_2^2 + 4\sup_{\beta\in\mathbb{B}(\beta^*,r_l)} \left\|\mu - \hat{\Sigma}^{-1}\hat{\mu}\right\|_2/r_l$$

with $r_l = \epsilon_n^{\alpha_l}$. During this period, from Lemma B.5 on contraction of population EM, and Lemma B.3 concentration of finite sample EM, we can check that

$$\begin{aligned}
\|\hat{\Sigma}^{-1}\hat{\mu}\|_2 &\leqslant \|\beta\|_2 - 0.5\|\beta\|_2^3 + c_u\|\beta\|_2\|\beta^*\|_2^2 + \sup_{\beta\in\mathbb{B}(\beta^*,r)} \left\|\mu - \hat{\Sigma}^{-1}\hat{\mu}\right\| \\
&\leqslant \|\beta\|_2 - \frac{1}{2}\epsilon^{3\alpha_{l+1}}(n,\delta) + \frac{1}{4}\epsilon^{\alpha_l+1}(n,\delta), \\
\|\tilde{\beta}'\|_2 &\leqslant \big(1-\frac{\alpha}{2}\big)^T\|\beta\|_2 + \Big(1-\big(1-\frac{\alpha}{2}\big)^T\Big)\|\hat{\Sigma}^{-1}\hat{\mu}\|_2 \\
&\leqslant \|\beta\|_2 + \Big(1-\big(1-\frac{\alpha}{2}\big)^T\Big)\Big[-\frac{1}{2}\epsilon^{3\alpha_{l+1}}(n,\delta) + \frac{1}{4}\epsilon^{\alpha_l+1}(n,\delta)\Big].
\end{aligned}$$

Note that this inequality is valid as long as $\epsilon^{\alpha_{l+1}}(n,\delta) \leqslant \|\beta\|_2 \leqslant \epsilon^{\alpha_l}(n,\delta)$. Now we define a sequence α_l by

$$\alpha_l = (1/3)^l(\alpha_0 - 1/2) + 1/2$$

and $\alpha_l \to 1/2$ as $l \to \infty$. With this choice of α_l, $\epsilon_n^{\alpha_l} \to (d/n)^{1/4}$. Hence during the l^{th} epoch, we have

$$\left\|\tilde{\beta}'\right\|_2 \leqslant \|\beta\|_2 - \frac{1}{4}\Big(1-\big(1-\frac{\alpha}{2}\big)^T\Big)\epsilon^{\alpha_l+1}(n,\delta).$$

Furthermore, the number of iterations required in l^{th} epoch is

$$t_l := \frac{\big(\epsilon^{\alpha_l}(n,\delta) - \epsilon^{\alpha_{l+1}}(n,\delta)\big)}{\Big(1-\big(1-\frac{\alpha}{2}\big)^T\Big)\epsilon^{\alpha_l+1}(n,\delta)} \leqslant \Big(1-\big(1-\frac{\alpha}{2}\big)^T\Big)^{-1}\epsilon^{-1}(n,\delta).$$

When it gets into $(l+1)^{th}$ epoch. the behavior can be analyzed in the same way and after going through l epochs in total, we have $\|\beta\|_2 \leqslant \epsilon^{\alpha_{l+1}}(n,\delta)$. At this point, the total number of iterations (counted in terms of steps of gradient descent) is bounded by

$$l\Big(1-\big(1-\frac{\alpha}{2}\big)^T\Big)^{-1}\epsilon^{-1}(n,\delta).$$

By taking $l = C\big(1-\big(1-\alpha/2\big)^T\big)\log(1/\theta)$ for some universal constant C such that α_l is $1/2-\theta$ for arbitrarily small $\theta > 0$, it holds that

$$\|\beta^{(t)}\|_2 \leqslant \epsilon^{1/2-\theta}(n,\delta) \leqslant c\big(d\log^2(n/\delta)/n\big)^{1/4-\theta/2}$$

with high probability as long as $t \geqslant \epsilon^{-1}(n,\delta)l \gtrsim \sqrt{d/n}\big(1-\big(1-\alpha/2\big)^T\big)\log(1/\theta)$ where c is some universal constant. By taking $\theta = C/\log(d/n)$ and using triangle inequalities, it holds that $\|\beta^{(t)}\|_2 \leqslant c\big(d\log^2(n/\delta)/n\big)^{1/4}$ and $\|\beta^{(t)} - \beta^*\|_2 \leqslant c_1\big(d\log^2(n/\delta)/n\big)^{1/4}$ where c_1 is some universal constant under low SNR settings.

To finish the proof, we replace δ by $\delta/\log(1/\theta)$ and take the union bound of the concentration of sample gradient EM operators for all $l = 1,\ldots,C\big(1-\big(1-\alpha/2\big)^T\big)\log(1/\theta)$, such that the argument holds for all epochs. $\square$

B.5 Proof of Theorem 2.1

Proof of Equation 6. For the data generated based on model Equation 1 with two components, $\beta_{n+1} = -\beta^*$ with probability $\frac{1}{2}$ and $\beta_{n+1} = \beta^*$ with probability $\frac{1}{2}$. For any choice of $\beta \in \mathbb{R}^d$,

$$\begin{aligned}
\mathbb{E}_{\mathcal{P}_{x,y}}[(y_{n+1} - x_{n+1}^\top \beta)^2] &= \mathbb{E}_{\mathcal{P}_{x,y}}[(x_{n+1}^\top \beta_{n+1} - x_{n+1}^\top \beta + v_{n+1})^2] \\
&= \vartheta^2 + \mathbb{E}_{\mathcal{P}_{x,y}}\Big[\big(x_{n+1}^\top \beta_{n+1} - x_{n+1}^\top \beta\big)^2\Big] \\
&= \vartheta^2 + \mathbb{E}_{\mathcal{P}_{x,y}} \operatorname{tr} x_{n+1} x_{n+1}^\top (\beta_{n+1} - \beta)(\beta_{n+1} - \beta)^\top \\
&= \vartheta^2 + \mathbb{E}_{\mathcal{P}_{x,y}} \operatorname{tr}(\beta_{n+1} - \beta)(\beta_{n+1} - \beta)^\top \\
&= \vartheta^2 + \mathbb{E}_{\mathcal{P}_{x,y}} \|\beta_{n+1} - \beta\|_2^2 \\
&= \vartheta^2 + \frac{1}{2}\|\beta^* - \beta\|_2^2 + \frac{1}{2}\|\beta^* + \beta\|_2^2.
\end{aligned}$$

Therefore, $\mathbb{E}_{\mathcal{P}_{x,y}}[(y_{n+1} - x_{n+1}^\top)^2]$ is minimized at $\hat{\beta} = \frac{1}{2}\beta^* - \frac{1}{2}\beta^* = 0$ and the optimal risk is given by $\vartheta^2 + \|\beta^*\|_2^2$. And same results holds if the estimator β depends on previous training instance $(x_1, y_1, \ldots, x_n, y_n)$ and the expectation is taken w.r.t. $\mathcal{P}$. ☐

Proof of Theorem 2.1. The existence of the transformer follows from Lemma 3.2 and Lemma 3.3. The output of the transformer is given by

$$\hat{y}_{n+1} = \operatorname{read}_y\big(\mathrm{TF}(H)\big) = x_{n+1}^\top \hat{\beta}^{\mathsf{OR}}$$

where $\hat{\beta}^{\mathsf{OR}}$ is given by

$$\hat{\beta}^{\mathsf{OR}} := \pi_1 \hat{\beta} - (1 - \pi_1)\hat{\beta}$$

with $\hat{\beta} = \operatorname{read}_\beta(\mathrm{TF}(H))$ for $L = \mathcal{O}\Big(T\big(1 - \big(1 - \alpha/2\big)^T\big)^{-1} \log(\log(n/d))\sqrt{n/\big(d\log^2(n/\delta)\big)}\Big)$ in the low SNR settings and $\mathcal{O}\Big(T \log\big(\frac{n \log n}{d}\big)\Big)$ in the high SNR settings. Note that $\|\hat{\beta}^{\mathsf{OR}} - \beta^{\mathsf{OR}}\|_2 \leqslant \pi_1\|\beta^* - \hat{\beta}\|_2 + (1 - \pi_1)\|\beta^* - \hat{\beta}\|_2 \leqslant \|\beta^* - \hat{\beta}\|_2$.

- Under thelow SNR regime, after $T_0 \geqslant \mathcal{O}\Big(\log(\log(n/d))\sqrt{n/\big(d\log^2(n/\delta)\big)}\Big)$ outer loops,
$$\|\beta^{\mathsf{OR}} - \hat{\beta}^{\mathsf{OR}}\|_2 \leqslant \mathcal{O}\Big(\Big(\frac{d\log(n/\delta)}{n}\Big)^{\frac{1}{4}}\Big)$$
with probability at least $1 - 5\delta$.
- Under the high SNR regime, after $T_0 \geqslant \mathcal{O}\Big(\log\big(\frac{n \log n}{d}\big)\Big)$ outer loops,
$$\|\beta^{\mathsf{OR}} - \hat{\beta}^{\mathsf{OR}}\|_2 \leqslant \mathcal{O}\left(\sqrt{\frac{d\log^2(n/\delta)}{n}}\right)$$
with probability at least $1 - 5\delta$.

Then we can bound the error $\big|x_{n+1}^\top \beta^{\mathsf{OR}} - x_{n+1}^\top \hat{\beta}^{\mathsf{OR}}\big|$ as

$$\big|x_{n+1}^\top \beta^{\mathsf{OR}} - x_{n+1}^\top \hat{\beta}^{\mathsf{OR}}\big| \leqslant \big\|x_{n+1}\big\|_2 \big\|\beta^{\mathsf{OR}} - \hat{\beta}^{\mathsf{OR}}\big\|_2.$$

By standard concentration results on Euclidean norm of standard Gaussian random vectors, $\|x_{n+1}\|_2 \leqslant 2\sqrt{\log \frac{d}{\delta}}$ with probability at least $1 - \delta$. Combining everything above with Theorem B.3 and Theorem B.2 yields the results. ☐

Proof of Theorem 2.2. The oracle estimator that minimizes the MSE, i.e. $\mathrm{MSE}(f) = \mathbb{E}_{\mathcal{P}}\big[\big(f(H) - y_{n+1}\big)^2\big]$ is given by Equation 6. We would like to bound

$$\mathbb{E}_{\mathcal{P}}\Big[\big(y_{n+1} - \operatorname{read}_y(\mathrm{TF}(H))\big)^2\Big] - \inf_\beta \mathbb{E}_{\mathcal{P}}\Big[\big(x_{n+1}^\top \beta - y_{n+1}\big)^2\Big].$$

Note that the $\mathbb{E}_{\mathcal{P}}\Big[\big(y_{n+1}-\mathrm{read}_y(\mathrm{TF}(H))\big)^2\Big]$ is given by

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{P}}\Big[\big(x_{n+1}^\top\hat{\beta}^{\mathsf{OR}}-y_{n+1}\big)^2\Big]=\mathbb{E}_{\mathcal{P}}\Big[\big(x_{n+1}^\top\big(\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}+\beta^{\mathsf{OR}}\big)-y_{n+1}\big)^2\Big]\\
=&\mathbb{E}_{\mathcal{P}}\Big[\big(x_{n+1}^\top\big(\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\big)\big)^2\Big]+2\mathbb{E}_{\mathcal{P}}\Big[\big(\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\big)^\top x_{n+1}\big(x_{n+1}^\top\beta^{\mathsf{OR}}-y_{n+1}\big)\Big]+\mathbb{E}_{\mathcal{P}}\Big[\big(x_{n+1}^\top\beta^{\mathsf{OR}}-y_{n+1}\big)^2\Big].
\end{aligned}
$$

Hence, when $\pi_1=\pi_2=\frac{1}{2}$, $\beta^{\mathsf{OR}}=\pi_1\beta^*-\pi_2\beta^*=0$,

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{P}}\Big[\big(y_{n+1}-\mathrm{read}_y(\mathrm{TF}(H))\big)^2\Big]-\inf_{\beta}\mathbb{E}_{\mathcal{P}}\Big[\big(x_{n+1}^\top\beta-y_{n+1}\big)^2\Big]\\
=&\mathbb{E}_{\mathcal{P}}\Big[\big(x_{n+1}^\top\big(\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\big)\big)^2\Big]+2\mathbb{E}_{\mathcal{P}}\Big[\big(\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\big)^\top x_{n+1}x_{n+1}^\top\beta^{\mathsf{OR}}\Big]\\
=&\mathbb{E}_{\mathcal{P}}\Big[\big(\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\big)^\top x_{n+1}x_{n+1}^\top\big(\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\big)\Big]\\
=&\mathbb{E}_{\mathcal{P}}\Big[\mathrm{tr}\Big(x_{n+1}x_{n+1}^\top\big(\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\big)\big(\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\big)^\top\Big)\Big]\\
=&\mathbb{E}_{\mathcal{P}}\big\|\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\big\|_2^2.
\end{aligned}
$$

- Under the high SNR settings, it holds that

  $$
  \mathbb{P}\Big(\|\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\|_2\leqslant\mathcal{O}\Big(\sqrt{d\log^2(n/\delta)/n}\Big)\Big)\geqslant 1-\delta.
  $$

  Hence, by integrating the tail probabilities we have

  $$
  \begin{aligned}
  \mathbb{E}\|\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\|_2^2&=\int_0^{+\infty}\mathbb{P}\big(\big\|\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\big\|_2\geqslant\sqrt{t}\big)dt\\
  &=\Big[\int_0^{c_1}+\int_{c_1}^{+\infty}\Big]\mathbb{P}\big(\big\|\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\big\|_2\geqslant\sqrt{t}\big)dt\\
  &\leqslant\int_0^{c_1}1dt+\int_{c_1}^{+\infty}\mathbb{P}\big(\big\|\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\big\|_2\geqslant\sqrt{t}\big)dt\\
  &\leqslant c_1+\int_{c_1}^{+\infty}\mathbb{P}\big(\big\|\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\big\|_2\geqslant\sqrt{t}\big)dt.
  \end{aligned}
  $$

  Setting $\sqrt{t}=\mathcal{O}\Big(\sqrt{d\log^2(n/\delta)/n}\Big)$ and solving for δ give us $\delta\leqslant n\exp\big\{-\sqrt{nt/d}\big\}$. By taking $c_1=\frac{Cd\log^2 n}{n}$ for some absolute constant C, it holds that

  $$
  \begin{aligned}
  \mathbb{E}\|\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\|_2^2&\leqslant\mathcal{O}\bigg(\frac{d\log^2 n}{n}\bigg)+\int_{c_1}^{+\infty}n\exp\big\{-\sqrt{nt/d}\big\}dt\\
  &=\mathcal{O}\bigg(\frac{d\log^2 n}{n}\bigg)+\mathcal{O}\bigg(\frac{(2d+1)\log n}{n}\bigg)\\
  &=\mathcal{O}\bigg(\frac{d\log^2 n}{n}\bigg).
  \end{aligned}
  $$

- Under the low SNR settings, it holds that

  $$
  \mathbb{P}\Big(\|\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\|_2\leqslant\mathcal{O}\big(d^{\frac{1}{4}}\log^{\frac{1}{2}}(n/\delta)/n^{\frac{1}{4}}\big)\Big)\geqslant 1-\delta.
  $$

Hence,

$$\begin{aligned}\mathbb{E}\|\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\|_2^2 &= \int_0^{+\infty}\mathbb{P}\big(\big\|\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\big\|_2 \geqslant \sqrt{t}\big)dt\\ &= \Big[\int_0^{c_1}+\int_{c_1}^{+\infty}\Big]\mathbb{P}\big(\big\|\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\big\|_2 \geqslant \sqrt{t}\big)dt\\ &\leqslant \int_0^{c_1}1dt+\int_{c_1}^{+\infty}\mathbb{P}\big(\big\|\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\big\|_2 \geqslant \sqrt{t}\big)dt\\ &\leqslant c_1+\int_{c_1}^{+\infty}\mathbb{P}\big(\big\|\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\big\|_2 \geqslant \sqrt{t}\big)dt.\end{aligned}$$

Similarly, setting $\sqrt{t}=\mathcal{O}\Big(d^{\frac14}\log^{\frac12}(n/\delta)/n^{\frac14}\Big)$ and solving for δ give us $\delta\leqslant n\exp\Big\{-\sqrt{n/dt}\Big\}$. By taking $c_1=C\sqrt{d\log^2 n/n}$ for some absolute constant C, it holds that

$$\begin{aligned}\mathbb{E}\|\hat{\beta}^{\mathsf{OR}}-\beta^{\mathsf{OR}}\|_2^2 &\leqslant \mathcal{O}\left(\sqrt{\frac{d\log^2 n}{n}}\right)+\int_{c_1}^{+\infty}n\exp\Big\{-d^{-\frac14}\sqrt{t}\Big\}dt\\ &= \mathcal{O}\Big(\sqrt{d/n}\log n\Big).\end{aligned}$$

Combining everything together, it holds that

$$\begin{aligned}&\mathbb{E}_{\mathcal{P}}\Big[(y_{n+1}-\mathrm{read}_y(\mathrm{TF}(H)))^2\Big]-\inf_{\beta}\mathbb{E}_{\mathcal{P}}\big[(x_{n+1}^{\top}\beta-y_{n+1})^2\big]\\ &=\begin{cases}\mathcal{O}\Big(\frac{d\log^2 n}{n}\Big) & \eta\geqslant\mathcal{O}\big((d\log^2(n/\delta)/n)^{\frac14}\big)\\ \mathcal{O}\Big(\sqrt{d/n}\log n\Big) & \eta\leqslant\mathcal{O}\big((d\log^2(n/\delta)/n)^{\frac14}\big)\end{cases}.\end{aligned}$$

□

C Proof of Theorem 2.3 in Section 2.3

Proposition C.1 (Proposition A.4 Bai et al. (2024)). *Suppose that $\{X_\theta\}_{\theta\in\Theta}$ is a zero-mean random process given by*

$$X_\theta := \frac{1}{N}\sum_{i=1}^{N}f(z_i;\theta)-\mathbb{E}_z[f(z;\theta)],$$

where $z_1,\cdots,z_N$ are i.i.d samples from a distribution $\mathbb{P}_z$ such that the following assumption holds:

(a) *The index set Θ is equipped with a distance ρ and diameter D. Further, assume that for some constant A, for any ball Θ' of radius r in Θ, the covering number admits upper bound $\log N\big(\delta;\Theta',\rho\big)\leqslant d\log(2Ar/\delta)$ for all $0<\delta\leqslant 2r$.*

(b) *For any fixed $\theta\in\Theta$ and z sampled from $\mathbb{P}_z$, the random variable $f(z;\theta)$ is a $\mathrm{SG}\big(B^0\big)$-sub-Gaussian random variable.*

(c) *For any $\theta,\theta'\in\Theta$ and z sampled from $\mathbb{P}_z$, the random variable $f(z;\theta)-f(z;\theta')$ is a $\mathrm{SG}\big(B^1\rho\big(\theta,\theta'\big)\big)$-subGaussian random variable.*

Then with probability at least $1-\delta$, it holds that

$$\sup_{\theta\in\Theta}\big|X_\theta\big|\leqslant CB^0\sqrt{\frac{d\log(2A\kappa)+\log(1/\delta)}{N}},$$

where C is a universal constant, and we denote $\kappa=1+B^1D/B^0$.

Furthermore, if we replace the SG *in assumption (b) and (c) by* SE*, then with probability at least* $1-\delta$*, it holds that*

$$\sup_{\theta\in\Theta}\left|X_{\theta}\right| \leqslant CB^{0}\left[\sqrt{\frac{d\log(2A\kappa)+\log(1/\delta)}{N}}+\frac{d\log(2A\kappa)+\log(1/\delta)}{N}\right].$$

For any $p\in[1,\infty]$, let $\|H\|_{2,p} := \left(\sum_{i=1}^{n}\|h_i\|_2^p\right)^{1/p}$ denote the column-wise $(2,p)$-norm of H. For any radius $R>0$, we denote $\mathcal{H}_R := \left\{H : \|H\|_{2,\infty}\leqslant R\right\}$ be the ball of radius R under norm $\|\cdot\|_{2,\infty}$.

Lemma C.1 (Corollary J.1 Bai et al. (2024))**.** For a single attention layer $\boldsymbol{\theta}_{\text{attn}} = \left\{(V_m,Q_m,K_m)\right\}_{m\in[M]}\subset\mathbb{R}^{D\times D}$ and any fixed dimension D, we consider

$$\Theta_{\text{attn},B'} := \left\{\boldsymbol{\theta}_{\text{attn}} : \left\|\boldsymbol{\theta}_{\text{attn}}\right\|\leqslant B'\right\}.$$

Then for $H\in\mathcal{H}_R, \boldsymbol{\theta}_{\text{attn}}\in\Theta_{\text{attn},B}$, the function $\left(\boldsymbol{\theta}_{\text{attn}},H\right)\mapsto \text{Attn}_{\boldsymbol{\theta}_{\text{attn}}}(H)$ is (B^2R^3)-Lipschitz w.r.t. $\boldsymbol{\theta}_{\text{attn}}$ and $(1+B^3R^2)$-Lipschitz w.r.t. H. Furthermore, for the function TF^{R} given by

$$\text{TF}^{R} : (\boldsymbol{\theta},H)\mapsto \text{clip}_R\left(\text{Attn}_{\theta_{\text{attn}}}(H)\right).$$

TF^{R} is B_Θ-Lipschitz w.r.t $\boldsymbol{\theta}$ and L_H-Lipschitz w.r.t. H, where $B_\Theta := B^2R^3$ and $B_H := 1+B^3R^2$.

Proposition C.2 (Proposition J.1 Bai et al. (2024))**.** *For a fixed number of heads* M *and hidden dimension* D*, we consider*

$$\Theta_{\text{TF},L,B'} = \left\{\boldsymbol{\theta}=\boldsymbol{\theta}_{\text{attn}}^{(1:L)} : M^{(\ell)}=M, D^{(\ell)}=D, \|\boldsymbol{\theta}\|\leqslant B'\right\}.$$

Then the function TF^R *is* $\left(LB_H^{L-1}B_\Theta\right)$*-Lipschitz w.r.t* $\boldsymbol{\theta}\in\Theta_{\text{TF},L,B}$ *for any fixed* $\mathbf{H}$.

Proof. Define events

$$\mathcal{E}_y := \left\{\max_{i\in[n+1],j\in[B]}\left\{\left|y_i^{(j)}\right|\right\}\leqslant B_y\right\},$$

$$\mathcal{E}_x := \left\{\max_{i\in[n+1],j\in[B]}\left\{\left\|x_i^{(j)}\right\|_2\right\}\leqslant B_x\right\},$$

and the random process

$$X_{\boldsymbol{\theta}} := \frac{1}{B}\sum_{j=1}^{B}\ell_{\text{icl}}\left(\boldsymbol{\theta};\mathbf{Z}^{(j)}\right)-\mathbb{E}_{\mathbf{Z}}\left[\ell_{\text{icl}}(\boldsymbol{\theta};\mathbf{Z})\right]$$

where $\mathbf{Z}^{(1:B)}$ are i.i.d. copies of $\mathbf{Z}\sim \mathrm{P}$, drawn from the distribution P. The next step involves applying Proposition C.1 to the process $\{X_{\boldsymbol{\theta}}\}$ conditioning on events $\mathcal{E}_x\cap\mathcal{E}_y$. To proceed, we must verify the following preconditions:

(a) By [Wainwright (2019), Example 5.8], it holds that $\log N\left(\delta;\mathrm{B}_{\|\cdot\|}(r),\|\cdot\|\right)\leqslant L(3MD^2)\log(1+2r/\delta)$, where $\mathrm{B}_{\|\cdot\|}(r)$ is any ball of radius r under norm $\|\cdot\|$.

(b) $\left|\ell_{\text{icl}}\left(\boldsymbol{\theta};\mathbf{Z}\right)\right|\leqslant 4B_y^2$ and hence $4B_y^2$-sub-Gaussian.

(c) $\left|\ell_{\text{icl}}(\boldsymbol{\theta};\mathbf{Z})-\ell_{\text{icl}}(\widetilde{\boldsymbol{\theta}};\mathbf{Z})\right|\leqslant 2B_y\left(LB_H^{L-1}B_\Theta\right)\|\boldsymbol{\theta}-\widetilde{\boldsymbol{\theta}}\|$ by Proposition C.2, where $B_\Theta := B'^2R^3$ and $B_H := 1+B'^3R^2$.

Therefore, by Proposition C.1, conditioning on $\mathcal{E}_x\cap\mathcal{E}_y$ with probability at least $1-\xi$,

$$\sup_{\boldsymbol{\theta}}\left|X_{\boldsymbol{\theta}}\right|\leqslant\mathcal{O}\left(B_y^2\sqrt{\frac{L(MD^2)\iota+\log(1/\xi)}{B}}\right)$$

where $\iota = 20L\log\left(2+\max\left\{B', R, (2B_y)^{-1}\right\}\right)$. Note that y_i is sub-Gaussian with parameter at most $\sqrt{\vartheta^2+\|\beta^*\|_2^2} = \sqrt{(1+\eta^{-2})\|\beta^*\|_2^2}$. Then by taking

$$
\begin{aligned}
B_x &= \sqrt{d\log(nB/\xi)},\\
B_y &= \sqrt{2(1+\eta^{-2})\|\beta^*\|_2^2\log(2nB/\xi)},\\
R &= 2\max\{B_x, B_y\},
\end{aligned}
$$

we have $\mathbb{P}(\mathcal{E}_y) \geqslant 1-\xi$ and $\mathbb{P}(\mathcal{E}_x) \geqslant 1-\xi$ by union bound. Hence, with probability at least $1-3\xi$,

$$
\sup_{\boldsymbol{\theta}} |X_{\boldsymbol{\theta}}| \leqslant \mathcal{O}\left((1+\eta^{-2})\log(2nL/\xi)\sqrt{\frac{L(MD^2)\iota+\log(1/\xi)}{B}}\right)
$$

where

$$
\iota = 20L\log\left(2+\max\left\{B', R, (2B_y)^{-1}\right\}\right)
$$

is a log factor. □

D Proof of Theorem 4.1

Given the estimate $\beta_j^{(t-1)}$ and $\pi_j^{(t-1)}$, at step $t-1$, the population EM algorithm is defined by the updates

$$
\begin{aligned}
w_j^{(t)}(X,Y) &= \frac{\pi_j^{(t-1)}\exp\left\{-\frac{1}{2}\left(Y-X^\top\beta_j^{(t-1)}\right)^2\right\}}{\sum_{\ell\in[K]}\pi_\ell^{(t-1)}\exp\left\{-\frac{1}{2}\left(Y-X^\top\beta_\ell\right)^2\right\}},\\
\tilde{\beta}_j^{(t)} &= \left(\mathbb{E}\left[w_j^{(t)}(X,Y)XX^\top\right]\right)^{-1}\left(\mathbb{E}\left[w_j^{(t)}(X,Y)XY\right]\right),\\
\tilde{\pi}_j^{(t)} &= \mathbb{E}\left[w_j^{(t)}(X,Y)\right].
\end{aligned}
$$

In the sample version of the gradient EM algorithm, we define $\hat{\Sigma}_w^{(t)} = \frac{1}{n}\sum_{i=1}^n w_{ij}^{(t)}(x_i,y_i)x_ix_i^\top$. The new estimate, $\beta^{(t)}$, is obtained by applying L steps of gradient descent to the loss function

$$
\hat{L}_n^{(t)}(\beta) = \frac{1}{2n}\sum_{i=1}^n w_{ij}^{(t)}(x_i,y_i)\left(y_i - x_i^\top\beta\right)^2
$$

starting from $\beta_j^{(t-1)}$. Specifically,

$$
\beta_j^{(t)} = \left(I-\alpha\hat{\Sigma}_w^{(t)}\right)^T\beta_j^{(t-1)} + \left(I-\left(I-\alpha\hat{\Sigma}_w^{(t)}\right)^T\right)\frac{1}{n}\sum_{i=1}^n\left[\hat{\Sigma}_w^{(t)}\right]^{-1}w_{ij}^{(t)}(x_i,y_i)y_ix_i.
$$

In the finite sample gradient version of EM, the estimation error at the next iteration in this problem is

$$
\beta_j^{(t)} - \beta_j^* = \left(I-\alpha\hat{\Sigma}_w^{(t)}\right)^T\left(\beta_j^{(t-1)}-\beta_j^*\right) + \left(I-\left(I-\alpha\hat{\Sigma}_w^{(t)}\right)^T\right)\left[\frac{1}{n}\sum_{i=1}^n\left[\hat{\Sigma}_w^{(t)}\right]^{-1}w_{ij}^{(t)}(x_i,y_i)y_ix_i - \beta_j^*\right].
$$

Define

$$
w_j^*(X,Y) = \frac{\pi_j^*\exp\left(-\frac{1}{2}\left(Y-X^\top\beta_j^*\right)^2\right)}{\sum_{l=1}^K\pi_j^*\exp\left(-\frac{1}{2}\left(Y-X^\top\beta_j^*\right)^2\right)},
$$

then we have

$$
\mathbb{E}\left[w_j^*(X,Y)X\left(Y-X^\top\beta_j^*\right)\right] = \pi_1^*\mathbb{E}\left[X\left(Y-X^\top\beta_j^*\right)\right] = 0,
$$

since true parameters are a fixed point of the EM iteration. Hence,

$$
\begin{aligned}
\beta_j^{(t)} - \beta_j^* &= \left(I-\alpha\hat{\Sigma}_w^{(t-1)}\right)^T\left(\beta_j^{(t-1)}-\beta_j^*\right) + \left(I-\left(I-\alpha\hat{\Sigma}_w^{(t)}\right)^T\right)\left(\hat{\Sigma}_w^{(t)}\right)^{-1}\left[e_B+B\right],\\
e_B &= \frac{1}{n}\sum_{i=1}^n w_{ij}^{(t)}(x_i,y_i)\left(y_i-x_i^\top\beta_j^*\right)x_i - \mathbb{E}\left[w_j^{(t)}(X,Y)X\left(Y-X^\top\beta_j^*\right)\right],\\
B &= \mathbb{E}\left[w_j^{(t)}(X,Y)X\left(Y-X^\top\beta_j^*\right)\right] - \mathbb{E}\left[w_j^*(X,Y)X\left(Y-X^\top\beta_j^*\right)\right].
\end{aligned}
$$

In Kwon & Caramanis (2020), the following results are proved.

Lemma D.1 ((Kwon & Caramanis, 2020)). Under SNR condition

$$\eta \geqslant CK\rho_\pi \log(K\rho_\pi)$$

with sufficiently large $C > 0$ and initialization condition

$$\max_\ell |\pi_\ell^{(t-1)} - \pi_\ell^*| \leqslant \frac{\pi_{\min}}{2},$$
$$\max_\ell \|\beta_\ell^{(t-1)} - \beta_\ell^*\|_2 \leqslant \frac{c\eta}{K\rho_\pi \log(K\rho_\pi)},$$

for sufficiently small $c > 0$. Given $n \geqslant \mathcal{O}\big(\max\{d\log^2(dK^2/\delta), (K^2/\delta)^{1/3}\}\big)$ samples, we get

$$\|e_B\|_2 \leqslant \sqrt{\frac{K\pi_j^{*2}}{\pi_{\min}}}\sqrt{\frac{d}{n}\log^2\left(nK^2/\delta\right)}\max_\ell \|\beta_\ell^{(t-1)} - \beta_l^*\|_2 + \sqrt{\frac{K\pi_j^{*2}}{\pi_{\min}}}\sqrt{\frac{d}{n}\log^2\left(nK^2/\delta\right)}$$

with probability at least $1-\delta$.

Lemma D.2 ((Kwon & Caramanis, 2020)). Under SNR condition

$$\eta \geqslant CK\rho_\pi \log(K\rho_\pi)$$

with sufficiently large $C > 0$ and initialization condition

$$\max_\ell |\pi_\ell^{(t-1)} - \pi_\ell^*| \leqslant \frac{\pi_{\min}}{2},$$
$$\max_\ell \|\beta_\ell^{(t-1)} - \beta_\ell^*\|_2 \leqslant \frac{c\eta}{K\rho_\pi \log(K\rho_\pi)},$$

for sufficiently small $c > 0$. There exits some universal constant $c'_B \in (0, 1/2)$

$$B \leqslant c'_B \pi_j^* \max_\ell \|\beta_\ell^{(t-1)} - \beta_\ell^*\|_2.$$

Now, it remains to bound the maximum eigenvalue and minimum eigenvalue of the weighted sample covariance matrix $\hat{\Sigma}_w^{(t)}$. Define the event

$$\mathcal{E}_j = \{\text{the sample comes from } j\text{-th component}\}.$$

Note that

$$\frac{1}{n}\sum_{i=1}^n w_{ij}^{(t)}(x_i, y_i)x_i x_i^\top 1_{\mathcal{E}_j} \preceq \hat{\Sigma}_w^{(t)} = \frac{1}{n}\sum_{i=1}^n w_{ij}^{(t)}(x_i, y_i)x_i x_i^\top \preceq \frac{1}{n}\sum_{i=1}^n x_i x_i^\top.$$

By standard concentration results on $\hat{\Sigma} - I$, it holds that with $n \geqslant \mathcal{O}(d\log(1/\delta))$,

$$\lambda_{\max}(\hat{\Sigma}_w^{(t)}) \leqslant \lambda_{\max}(\hat{\Sigma}) \leqslant \frac{3}{2}$$

with probability at least $1-\delta$. The concentration of $\frac{1}{n}\sum_{i=1}^n w_{ij}^{(t)}(x_i, y_i)x_i x_i^\top 1_{\mathcal{E}_j}$ comes from standard concentration argument for random matrix with sub-exponential norm Vershynin (2018). Since $w_{ij}^{(t)} \in (0, 1)$ and x_i is standard multivariate Gaussian, then by Appendix B.2 in Kwon & Caramanis (2020), it holds that

$$\left\|\frac{1}{n}\sum_i w_{ij}^{(t)} x_i x_i^\top 1_{\mathcal{E}_j} - \mathbb{E}\big[w_j^{(t)}(X, Y)XX^\top 1_{\mathcal{E}_j}\big]\right\|_2 \leqslant O\left(\sqrt{\pi_j^*}\sqrt{\frac{d\log(K^2/\delta)}{n}}\right)$$

with probability at least $1-\delta$. By Lemma A.3 in Kwon & Caramanis (2020), it holds that under the same SNR condition

$$\lambda_{\min}\big(\mathbb{E}\big[w_j^{(t)}(X, Y)XX^\top\big]\big) \geqslant \frac{\pi_j^*}{2}.$$

Therefore, as long as $n \geqslant \mathcal{O}\big(d\log(K^2/\delta)/\pi_{\min}\big)$, it holds that

$$\lambda_{\min}\big(\hat{\Sigma}_w^{(t)}\big) \geqslant \lambda_{\min}\Big(\frac{1}{n}\sum_i w_{ij}^{(t)} x_i x_i^\top 1_{\mathcal{E}_1}\Big) \geqslant \frac{\pi_j^*}{4}.$$

Therefore, we have under the same SNR and initialization condition, as long as

$$n \geqslant \mathcal{O}\Big(\max \big\{ d \log^2(dK^2/\delta), \big(K^2/\delta\big)^{1/3}, d\log(K^2/\delta)/\pi_{\min} \big\} \Big),$$

it holds that for appropriately small α,

$$\|\big(I - \alpha\hat{\Sigma}_w^{(t)}\big)^T\|_2 \leqslant \max\{|1-3\alpha/2|, (1-\pi_{\min}\alpha/4)\}^T := \gamma_T, \tag{29}$$

$$\|e_B\|_2 \leqslant \sqrt{\frac{K\pi_j^{*2}}{\pi_{\min}}}\sqrt{\frac{d}{n}\log^2\big(nK^2/\delta\big)} \max_\ell \|\beta_\ell^{(t-1)} - \beta_\ell^*\|_2 + \sqrt{\frac{K\pi_j^{*2}}{\pi_{\min}}}\sqrt{\frac{d}{n}\log^2\big(nK^2/\delta\big)}, \tag{30}$$

$$B \leqslant \frac{\pi_j^*}{2}\max_\ell \|\beta_\ell^{(t-1)} - \beta_\ell^*\|_2, \tag{31}$$

$$\big\|\big[\hat{\Sigma}_w^{(t)}\big]^{-1}\big\|_2 \leqslant \frac{4}{\pi_{\min}}. \tag{32}$$

For appropriately small α, we have $\gamma_T \in (0,1)$ Therefore, combining Equation 29, Equation 30, Equation 31 and Equation 32 together, we have

$$\begin{aligned}\beta_j^{(t)} - \beta_j^{(*)} \leqslant &\Bigg[\gamma_T + (1-\gamma_T)\Bigg(\sqrt{\frac{K\pi_j^{*2}}{\pi_{\min}}}\sqrt{\frac{d}{n}\log^2\big(nK^2/\delta\big)} + \frac{\pi_j^*}{2}\Bigg)\Bigg]\max_\ell \|\beta_\ell^{(t-1)} - \beta_\ell^*\|_2 \\ &+ \sqrt{\frac{K\pi_j^{*2}}{\pi_{\min}}}\sqrt{\frac{d}{n}\log^2\big(nK^2/\delta\big)}\end{aligned}$$

with probability at least $1-5\delta$.

To derive the concentration results for $\big|\frac{1}{n}\sum_i w_{ij}^{(t)}(x_i,y_i) - \mathbb{E}\big[w_j^{(t)}(X,Y)\big]\big|$, we define following events

$$\begin{aligned}\mathcal{E}_{\ell,1} &= \big\{|v| \leqslant \tau_\ell\big\}, \\ \mathcal{E}_{\ell,2} &= \Big\{4(|\langle X,\Delta_j\rangle| \vee |\langle X,\Delta_\ell\rangle|) \leqslant |\langle X, \beta_\ell^* - \beta_j^*\rangle|\Big\}, \\ \mathcal{E}_{\ell,3} &= \Big\{|\langle X, \beta_\ell^* - \beta_j^*\rangle| \geqslant 4\sqrt{2}\tau_\ell\Big\}, \\ \mathcal{E}_{\ell,\text{ good}} &= \mathcal{E}_{\ell,1}\cap\mathcal{E}_{\ell,2}\cap\mathcal{E}_{\ell,3},\end{aligned}$$

where $\Delta_\ell = \beta_\ell^{(t-1)} - \beta_\ell^*$, then we have the decomposition

$$w_{ij}^{(t)}(x_i,y_i) = \left(\sum_{\ell\neq j}^K w_{ij}^{(t)}(x_i,y_i)1_{\mathcal{E}_\ell\cap\mathcal{E}_{\ell,good}} + w_{ij}^{(t)}(x_i,y_i)1_{\mathcal{E}_\ell\cap\mathcal{E}_{\ell,good}^c}\right) + w_{ij}^{(t)}(x_i,y_i)1_{\mathcal{E}_j}.$$

Therefore, we could bound

$$\begin{aligned}&\left|\frac{1}{n}\sum_i w_{ij}^{(t)}(x_i,y_i)1_{\mathcal{E}_\ell\cap\mathcal{E}_{\ell,good}} - \mathbb{E}\Big[w_{ij}^{(t)}(x_i,y_i)1_{\mathcal{E}_\ell\cap\mathcal{E}_{\ell,\text{ good}}}\Big]\right|, \\ &\left|\frac{1}{n}\sum_i w_{ij}^{(t)}(x_i,y_i)1_{\mathcal{E}_\ell\cap\mathcal{E}_{\ell,good}^c} - \mathbb{E}\Big[w_{ij}^{(t)}(x_i,y_i)1_{\mathcal{E}_\ell\cap\mathcal{E}_{\ell,good}^c}\Big]\right|, \\ &\left|\frac{1}{n}\sum_i w_{ij}^{(t)}(x_i,y_i)1_{\mathcal{E}_j} - \mathbb{E}\Big[w_{ij}^{(t)}(x_i,y_i)1_{\mathcal{E}_j}\Big]\right|,\end{aligned}$$

respectively. For the first part, note that

$$\begin{aligned}\big\|w_{ij}^{(t)}(x_i,y_i)1_{\mathcal{E}_\ell\cap\mathcal{E}_{\ell,good}^c}\big\|_{\psi_2} &= \sup_{p\geqslant 1} p^{-1/2}\mathbb{E}\Big[\big|w_{ij}^{(t)}(x_i,y_i)\big|^p \mid \mathcal{E}_\ell\cap\mathcal{E}_{\ell,good}\Big]^{1/p} \\ &\leqslant C\rho_{\ell j}\exp\big(-\tau_\ell^2\big).\end{aligned}$$

Therefore, with probability at least $1-\delta/K^2$,

$$\Big|\tfrac{1}{n}\sum_i w_{ij}^{(t)}(x_i,y_i)1_{\mathcal{E}_\ell\cap\mathcal{E}_{\ell,good}} - \mathbb{E}\Big[w_{ij}^{(t)}(x_i,y_i)1_{\mathcal{E}_\ell\cap\mathcal{E}_{\ell,\text{good}}}\Big]\Big| \leqslant \mathcal{O}\Big(\rho_{\ell j}\exp\big(-\tau_\ell^2\big)\sqrt{\pi_\ell^*}\sqrt{\tfrac{1}{n}\log(K^2/\delta)}\Big).$$

For the second part, note that

$$\|w_{ij}^{(t)}(x_i,y_i)1_{\mathcal{E}_\ell\cap\mathcal{E}_{\ell,good}^c}\|_{\psi_2} = \sup_{p\geqslant 1} p^{-1/2}\mathbb{E}_{\mathcal{D}}\Big[\big|w_{ij}^{(t)}(x_i,y_i)\big|^p \mid \mathcal{E}_\ell\cap\mathcal{E}_{\ell,\text{ good}}^c\Big]^{1/p} \leqslant 1,$$
$$\mathbb{P}\big(\mathcal{E}_\ell\cap\mathcal{E}_{\ell,\text{ good}}^c\big) \leqslant \mathcal{O}\big(\pi_\ell^*/(K\rho_\pi)\big).$$

Therefore,

$$\Big|\tfrac{1}{n}\sum_i w_{ij}^{(t)}(x_i,y_i)1_{\mathcal{E}_\ell\cap\mathcal{E}_{\ell,good}^c} - \mathbb{E}\big[w_{ij}^{(t)}(x_i,y_i)1_{\mathcal{E}_\ell\cap\mathcal{E}_{\ell,good}^c}\big]\Big| \leqslant \mathcal{O}\Big(\sqrt{\tfrac{\pi_\ell^*}{K\rho_\pi}\vee\tfrac{\log(K^2/\delta)}{n}}\sqrt{\tfrac{\log(K^2/\delta)}{n}}\Big).$$

Similar to the second part, we have the following concentration result for the last part:

$$\left|\frac{1}{n}\sum_i w_{ij}^{(t)}(x_i,y_i)1_{\mathcal{E}_j} - \mathbb{E}\big[w_{ij}^{(t)}(x_i,y_i)1_{\mathcal{E}_j}\big]\right| \leqslant \mathcal{O}\left(\sqrt{\pi_j^*\vee\frac{\log(K^2/\delta)}{n}}\sqrt{\frac{\log(K^2/\delta)}{n}}\right).$$

Combining three parts together, we have

$$\begin{aligned}\Big|\frac{1}{n}\sum_i w_{ij}^{(t)}(x_i,y_i) - \mathbb{E}\big[w_j^{(t)}(X,Y)\big]\Big| &\leqslant \mathcal{O}\bigg(\sqrt{\tfrac{1}{n}\log(K^2/\delta)}\Big(\textstyle\sum_{\ell\neq j}^K\rho_{\ell j}\exp\big(-\tau_\ell^2\big)\sqrt{\pi_\ell^*}+\sqrt{\tfrac{\pi_j^*}{K}}\Big)+\sqrt{\tfrac{\pi_j^*\log(K^2/\delta)}{n}}\bigg)\\ &\leqslant \mathcal{O}\bigg(\sqrt{\tfrac{K\log(K^2/\delta)}{n\pi_{\min}}}\sqrt{\tfrac{\pi_j^*}{K}}\Big(\textstyle\sum_{\ell\neq j}^K\frac{\sqrt{\rho_{\ell,j}}\sqrt{\pi_j^*}}{K\rho_\pi}+\sqrt{\tfrac{\pi_j^*}{k}}\Big)+\sqrt{\tfrac{K\log(K^2/\delta)}{n\pi_{\min}}}\pi_j^*\bigg)\\ &\leqslant \mathcal{O}\bigg(\sqrt{\tfrac{K\log(K^2/\delta)}{n\pi_{\min}}}\pi_j^*\bigg).\end{aligned}$$

with probability at least $1-3\delta$. Therefore,

$$\begin{aligned}|x_{n+1}^\top\hat\beta^{\mathsf{OR}} - x_{n+1}^\top\beta^{\mathsf{OR}}| &\leqslant \|x_{n+1}\|_2\|\hat\beta^{\mathsf{OR}}-\beta^{\mathsf{OR}}\|_2\\ &\leqslant \|x_{n+1}\|_2\Big(\max_j|\hat\pi_j-\pi_j^*|\max_j\|\beta_j^*\|_2+\max_j\{\hat\pi_j\}\max_j\|\hat\beta_j-\beta_j^*\|_2\Big)\\ &\leqslant \sqrt{\log(d/\delta)}\left(\sqrt{\frac{K\log(K^2/\delta)}{n\pi_{\min}}}\pi_j^*+\sqrt{\frac{K\pi_j^{*2}}{\pi_{\min}}}\sqrt{\frac{d}{n}\log^2\big(nK^2/\delta\big)}\right)\end{aligned}$$

with probability at least $1-9\delta$.

E Auxiliary Results

Proposition E.1 (Proposition C.2 in Bai et al. (2024)). *Let $\ell(\cdot,\cdot):\mathbb{R}^2\to\mathbb{R}$ be a loss function such that $\partial_1\ell$ is (ε,R,M,C)-approximable by sum of relus with $R=\max\{B_xB_w,B_y,1\}$. Let $\widehat{L}_n(\beta):=\frac{1}{n}\sum_{i=1}^n\ell\big(\beta^\top x_i,y_i\big)$ denote the empirical risk with loss function ℓ on dataset $\{(x_i,y_i)\}_{i\in[n]}$. Then, for any $\varepsilon>0$, there exists an attention layer $\{(Q_m,K_m,V_m)\}_{m\in[M]}$ with M heads such that, for any input sequence that takes form $h_i=\big[x_i;y_i';\beta;0_{D-2d-3};1;t_i\big]$ with $\|\beta\|_2\leqslant B_w$, it gives output*

$$\widetilde{h}_i = \big[\mathrm{Attn}_{\boldsymbol{\theta}}(H)\big]_i = \big[x_i;y_i';\widetilde{\beta};0_{D-2d-3};1;t_i\big]$$

for all $i\in[N+1]$, where

$$\big\|\widetilde{\beta}-\big(\beta-\eta\nabla\widehat{L}_n(\beta)\big)\big\|_2 \leqslant \varepsilon\cdot\big(\eta B_x\big).$$

Proposition E.2 (Proposition 1 in Pathak et al. (2024)). *Given any input matrix $H\in\mathbb{R}^{p\times q}$ that output a matrix $H'\in\mathbb{R}^{p\times q}$, following operators can be implemented by a single layer of an autoregressive transformer:*

- *copy_down*$\big(H; k, k', \ell, \mathcal{I}\big)$: *For columns with index $i \in \mathcal{I}$, outputs H' where*

$$H'_{k':\ell',i} = H_{k:\ell,i}$$

and the remaining entries are unchanged. Here, $\ell' = k' + (\ell - k)$ and $k' \geqslant k$, so that entries are copied "down" within columns $i \in \mathcal{I}$. Note, we assume $\ell \geqslant k$ and that $k' \leqslant q$ so that the operator is well-defined.

- *copy_over*$\big(H; k, k', \ell, \mathcal{I}\big)$: *For columns with index $i \in \mathcal{I}$, outputs H' with*

$$H'_{k':\ell',i} = H_{k:\ell,i-1}.$$

The remaining entries stay the same. Here entries from column $i-1$ are copied "over" to column i.

- $\mathrm{mul}\left(H; k, k', k'', \ell, \mathcal{I}\right)$: *For columns with index $i \in \mathcal{I}$, outputs H' where*

$$H'_{k''+t,i} = H_{k+t,i}H_{k'+t,i}, \quad \textit{for } t \in \{0, \dots, \ell - k\}.$$

Note that $\ell'' = k'' + \delta''$ where $W \in \mathbb{R}^{\delta'' \times \delta}, W' \in \mathbb{R}^{\delta'' \times \delta'}$ and $\ell = k + \delta, \ell' = k' + \delta'$. We assume $\delta, \delta', \delta'' \geqslant 0$. The remaining entries of H are copied over to H', unchanged.

- *scaled_agg*$\big(H; \alpha, k, \ell, k', i, \mathcal{I}\big)$: *Outputs a matrix H' with entries*

$$H_{k'+t,i} = \alpha \sum_{j \in \mathcal{I}} H_{k+t,j} \quad \textit{for } t \in \{0, 1, \dots, \ell - k\}.$$

The set $\mathcal{I}$ is causal, so that $\mathcal{I} \subset [i-1]$. The remaining entries of H are copied over to H', unchanged.

- $\mathrm{soft}\left(H; k, \ell, k'\right)$: *For the final column q, outputs a matrix H' with entries*

$$H'_{k'+t,q} = \frac{\mathrm{e}^{H_{k+t,q}}}{\sum_{t'=0}^{\ell-k} \mathrm{e}^{H_{k+t',q}}}, \quad \textit{for } t \in \{0, 1, \dots, \ell - k\}.$$

The remaining entries of H are copied over to H', unchanged.