# Structured Low-Rank Tensors for Generalized Linear Models

**Anonymous authors**
**Paper under double-blind review**

## Abstract

Recent works have shown that imposing tensor structures on the coefficient tensor in regression problems can lead to more reliable parameter estimation and lower sample complexity compared to vector-based methods. This work investigates a new low-rank tensor model, called Low Separation Rank (LSR) in Generalized Linear Model (GLM) problems. The LSR model – which generalises the well-known Tucker and CANDECOMP/PARAFAC (CP) models – is imposed onto the coefficient tensor in the GLM model. This work proposes a block coordinate descent algorithm for parameter estimation in LSR-structured tensor GLMs. Additionally, it derives a minimax lower bound on the error threshold on estimating the coefficient tensor in LSR tensor GLM problems. The minimax bound is proportional to the intrinsic degrees of freedom in the LSR tensor GLM problem, suggesting that its sample complexity may be significantly lower than that of vectorised GLMs. This result can also be specialised to lower bound the estimation error in CP and Tucker-structured GLMs. The derived bounds are comparable to tight bounds in the literature for Tucker linear regression, and the tightness of the minimax lower bound is further assessed numerically. Finally, numerical experiments on synthetic datasets demonstrate the efficacy of the proposed LSR tensor model for three regression types (linear, logistic ad Poisson), and experiments on a collection of medical imaging datasets demonstrate the usefulness of the LSR model over other tensor models (Tucker and CP) on real, imbalanced data with limited available samples.

## 1 Introduction

In machine learning, regression models are used to understand the relationship between a set of independent variables (also known as covariates) and a dependent outcome. More formally, given a vector of covariates, $\mathbf{x}$, and outcome, $y$, jointly distributed according to $\mathbb{P}_{\mathbf{x},y}$, the goal is to predict $y$ when given $\mathbf{x}$. Under the Mean Squared Error (MSE) criterion, the solution would be finding the conditional mean $\mathbb{E}_{y|\mathbf{x}}[y|\mathbf{x}]$, which is obtained by modeling the conditional probability $\mathbb{P}_{y|\mathbf{x}}$ and estimating the model class parameters. Typically, the model class parameters are denoted as $\mathbf{b}$ and estimated through a set of $n$ training data samples $\{\mathbf{x}_i, y_i\}_{i=1}^n$ , where $\mathbf{x}_i$ is the $i^{th}$ sample vector of covariates. There are many regression models that may suit different types of prediction problems. Some common examples are linear, logistic and Poisson regression, all of which fall under a broader class of models called Generalized Linear Models (GLMs). GLMs were introduced to encompass classes of models that cannot be appropriately modeled as a simple 'linear-response model' (McCullagh & Nelder, 2019). In particular, GLMs refer to a parametric statistical framework that seeks to model the conditional probability of a scalar response variable (denoted as $y_i$) that follows an exponential family distribution. Many distributions are easily expressible as an exponential family distribution, and thus GLMs allow one to study a broader class of regression problems.

GLM regression can handle a wide array of datasets for a multitude of applications, yet modern day technologies are creating data-intensive environments and collecting increasingly high-dimensional data. That is, data in the form of variegated and structured multi-dimensional arrays (tensors), where the number of available data samples is far smaller than the number of variables (the dimensionality of the data). Prominent examples of two-dimensional arrays include biological imaging data such as electroencephalography (EEG) and fiber-bundle imaging (Dumas et al., 2019). Examples of three-dimensional arrays include functional Magnetic Resonance Images (fMRI) (Bellec et al., 2017) and Magnetic Resonance Angiography (MRA) (Yang et al., 2020). Though such data has been used in many instances throughout the literature (Li et al., 2018; Hung & Wang, 2013; Zhou et al., 2013; Dumas et al., 2019), classical parameter estimation methods require vector-structured covariates and estimate a corresponding coefficient vector. There are two major concerns associated with multidimensional (a.k.a tensor) data and its vectorisation. First, vectorising data that was originally in tensor form destroys its underlying structure (which often contains rich information valuable for regression analysis). Secondly, the resulting vector model exhibits a very large number of parameters, in the sense that in the high-dimensional setting the GLM regression model becomes ill-posed. We term this problem as 'the curse of dimensionality' (Hung & Wang, 2013; Zhang & Jiang, 2018).

A common solution to the curse of dimensionality in tensor GLM problems is the imposition of structure on the model parameters and thus is the focus of this work. If the covariates are tensor structured, we expect to estimate coefficient tensors whilst exploiting the multidimensional structure of the data and rich information lying in the correlation between tensor modes. Common structures include: The addition of a sparsity regulariser or a low-rank inducing regulariser to the regression problem (Abramovich & Grinshtein, 2018; Seber & Lee, 2003; Zhang & Jiang, 2018; An & Zhang, 2020; Raskutti et al., 2019); The imposition of some tensor factorisation on the model parameters (Ahmed et al., 2020; Li et al., 2018; Zhou et al., 2013; Zhang et al., 2020; Tan et al., 2012; Zhang & Jiang, 2016; Wu et al., 2022; Taki et al., 2021); Both of the above. The imposition of such structures should ultimately lead to estimation of fewer parameters, improving the computational complexity and performance of our parameter estimation problem.

The most commonly used tensor factorisations are the CANDECOMP/PARAFAC (CP) and Tucker decompositions (Kolda & Bader, 2009). These decompositions impose a compact structure on the coefficient tensors, thereby restricting the class of possible solutions in the parameter estimation problem. Compared to simple vector regression, these decompositions can decrease the number of training samples needed for reliable coefficient estimation (also referred to as 'sample complexity'). A smaller sample complexity can lower the variance of the model, yet the restrictive nature of these decompositons can also reduce the representation power of the coefficient tensors for many classes of tensors, causing a non-favourable bias-variance tradeoff.

In this work we impose a new decomposition on the coefficient tensors that we will promptly refer to as the Low Separation Rank (LSR) model. In fact we will show that the LSR decomposition is a generalisation of the Tucker decomposition. The LSR model maintains a lower sample complexity than vectorisation-based tensor GLM regression but is less restrictive than Tucker or CP. Though estimating an LSR-structured tensor introduces a greater sample complexity than the aforementioned decompositions, this increase is compensated by a stronger representation power, leading to better parameter estimation performance. In other words, we show that the increase in variance of the LSR model is conquered by its stronger representation power for a larger class of tensors (ergo, its decrease in bias), leading to a more favourable bias-variance tradeoff (model compactness vs representation power).

### 1.1 Contributions

In this paper we make the following contributions. We introduce the LSR-structured tensor problem under the GLM framework that we appropriately denote as LSR-TGLM. GLMs encompass various regression types including linear, logistic and Poisson. We focus on the high-dimensional setting and discuss the various parameters of an LSR-structured tensor (such as 'rank' and 'seperataion rank', terms we will introduce in Section 2) that reduce the sample complexity of parameter estimation in GLMs. We also compare sample complexities between different tensor decomposition models and the LSR model.

Additionally, we explore two problems at the core of this work. First, we propose a parameter estimation algorithm (which we name LSRTR) for the LSR-TGLM problem. The main idea is that parameter estimation in GLMs can be achieved through Maximum Likelihood Estimation (MLE) (McCullagh & Nelder, 2019). However, for structured tensor settings, such as estimating CP or Tucker structured tensors, the objective function of the MLE problem is highly non-convex (Li et al., 2018; Zhang & Jiang, 2016; Zhou et al., 2013; Tan et al., 2012). This is also true when the tensor is LSR-structured. To overcome this, we observe that the problem can be partitioned into several convex sub-problems that can then be solved alternately. On the basis thereof we propose a block coordinate descent algorithm to find the MLE of the LSR-structured coefficient tensor. Secondly, we investigate the fundamental error thresholds of the LSR-TGLM problem under study. We derive a minimax lower bound on the estimation error for the LSR-TGLM problem. The minimax bound is useful for assessing the performance of the proposed algorithm and ascertaining the parameters that may affect the sample complexity of the parameter estimation problem. The bound is general and can be specialised to various regression types under the GLM framework. We also specialise our results for the Tucker and CP regression frameworks. Obtaining the bound requires a special construction of a packing set of LSR-structured tensors. The methods we develop are systematic and can be appropriate in other works that consider similar topological properties of structured tensors (i.e., LSR-structured tensors). We also assess the tightness of our bounds in two ways: 1) Through a numerical study where we show that the ratio of the empirical error through LSRTR and the minimax bound is approximately constant with increasing sample size, and 2) We specialise our minimax bound to the Tucker linear regression case and show that our bound matches the optimal error rates for Tucker linear regression found in recent works (Zhang et al., 2020).

Finally, we evaluate the performance of our algorithm through extensive numerical experiments on synthetic data. We also test the performance of imposing the LSR structure on several classification problems with multidimensional medical imaging datasets. We show that while the LSR model outperforms the vector model, its rich representation power also allows for enhanced performance over the Tucker (and CP) case.

### 1.2 Relation To Prior Work

Regression problems have been a major focus of high dimensional statistics for many years (Giraud, 2021). Some works on linear and logistic regression impose sparsity on the model parameter in order to reduce the sample complexity of the vector-based regression problems (Abramovich & Grinshtein, 2018; Sun & Zhang, 2012). However, in very high dimensional regimes such as when the data is tensor-structured, i.e., we have $\{\underline{\mathbf{X}}_i\}_{i=1}^n$, sparsity assumptions do not provide enough reduction in the sample complexity (Raskutti et al., 2019; Lee & Courtade, 2020). Several works overcome the limitations of sparse vector regression by extending regular regression to the high-dimensional and low-rank matrix settings. Low-rank assumptions on data have been used ubiquitously throughout the literature in order to reduce the sample complexity of estimation problems (Barnes & Ozgur, 2019; Shi et al., 2014). Such works propose regularized matrix linear and logistic regression models to obtain low-rank and/or sparse estimates of the coefficient matrix in regression problems, such as those on inference on images or graph data (Hung & Wang, 2013; Zhang & Jiang, 2018; Shi et al., 2014; An & Zhang, 2020; Berthet & Baldin, 2020). Some works directly impose low-rank structures on coefficient matrices through the rank-$r$ singular value decomposition (SVD) (Taki et al., 2021).

Additionally, though tensors and their decompositions have long since been introduced in the literature (Kolda & Bader, 2009), their applications in regression analysis have recently become established. Analogous to the low-rank matrix regression works, a variety of works have introduced low-rank structures on coefficients tensors for tensor regression problems. For logistic regression, Tan et al. (Tan et al., 2012) first introduced using a low-rank and/or sparse CANDECOMP/PARAFAC (CP) decomposition. A more flexible generalisation of this work is imposing the Tucker decomposition in the coefficient tensor in tensor logistic regression (Zhang & Jiang, 2016; Wu et al., 2022). The Tucker structure has also been introduced for tensor linear regression (Zhang et al., 2020; Ahmed et al., 2020; Wu et al., 2022).

More works to our interest generalise tensor linear and logistic regression works by imposing the CP and Tucker decompositions in tensor GLMs (Li et al., 2018; Zhou et al., 2013). Both structures have been shown to significantly reduce the number of learnable parameters, leading to efficient estimation and prediction in a variety of regression problems, particularly with medical imaging data. The aforementioned works develop efficient estimation algorithms and provide empirical results on their performance. The proposed approaches outperform vector-based methods (in terms of estimation and prediction accuracy) in the high-dimensional setting when the number of available samples is limited. However, these matrix and tensor structures are aimed at being compact (in the sense that they decrease the number of learnable parameters in a given problem), and are therefore also quite restrictive in their representation power of the true coefficient tensor. A more general and flexible tensor model is required to achieve accurate and efficient estimation while maintaining a useful level of compactness.

In terms of theoretical guarantees, various regression works provide local identifiability guarantees of the proposed CP and Tucker tensor models for GLMs, and asymptotic consistency and normality results for the MLE estimator of the model parameter (Li et al., 2018; Zhang et al., 2020; Zhou et al., 2013). Some works on high-dimensional regression also provide sample complexity bounds of the proposed model in the form of risk upper bounds (Zhang et al., 2020; Ahmed et al., 2020) or minimax lower bounds (Barnes & Ozgur, 2019; Zhang et al., 2020; Raskutti et al., 2019; Foster et al., 2018; Abramovich & Grinshtein, 2018; Lee & Courtade, 2020; Abramovich & Grinshtein, 2016; Raskutti et al., 2011); however, these works are specific to vector-based logistic regression or Tucker linear regression. Current works in tensor logistic regression or tensor-based GLMs do not provide any theoretical guarantees for sample complexity (upper or lower bounds).

In terms of the LSR model, a special case of the LSR model was introduced by Tsiligkaridis and Hero for covariance estimation problems (Tsiligkaridis & Hero, 2013). An extension of this model has only recently been used on tensor data for dictionary learning (Ghassemi et al., 2019). In terms of the GLM framework, to the best of our knowledge, this work is the first to consider the LSR model in regression problems.

### 1.3 Organization

The organization of this paper is as follows. In Section 2 we establish a background on various tensor models, as well as the LSR tensor model. In Section 3 we formulate the LSR-TGLM model and introduce two objectives: parameter estimation and minimax risk. In Section 4 we discuss the estimation problem of LSR-structured coefficient tensors in GLMs and propose an efficient algorithm for parameter estimation. In Section 5 we provide a numerical study of the LSRTR algorithm with synthetic data. In Section 6 we introduce a sample complexity bound in the form of a minimax lower bound on the estimation error of the low-rank LSR-GLM model and provide a formal proof in Section 6.3. We provide experimental results on real data in Section 7 and conclude our work in Section 8. Proofs of lemmas for the main theorem are provided in the appendix.

## 2 Preliminaries

We use the following notation convention throughout the paper: $x$, $\mathbf{x}$, $\mathbf{X}$ and $\underline{\mathbf{X}}$ denote scalars, vectors, matrices and tensors, respectively. Given a fixed tensor $\underline{\mathbf{X}}$, $\mathbf{x} \triangleq \text{vec}(\underline{\mathbf{X}})$ is the column-wise vectorisation of $\underline{\mathbf{X}}$. The tensor $\underline{\mathbf{I}}_m$ is the $m \times \cdots \times m$ identity tensor, such that for any tensor $\underline{\mathbf{S}}$, $\underline{\mathbf{I}} \cdot \underline{\mathbf{S}} = \underline{\mathbf{S}} \cdot \underline{\mathbf{I}} = \underline{\mathbf{S}}$. Given $n$ $K$-mode tensors $\{\underline{\mathbf{X}}_i\}_{i=1}^n$ of dimension $m_1 \times m_2 \times \cdots \times m_K$, $\mathcal{X}$ is the combined (aggregated) tensor of $n$ samples of dimension $m_1 \times m_2 \times \cdots \times m_K \times n$. For a positive integer $K$, the set $[K] = \{1, 2, \ldots, K\}$ so that $(\mathbf{X}_k)_{k\in[K]}$ is the ordered set $(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_K)$. Additionally we have $-[K]$ so that $(\mathbf{X}_k)_{k\in-[K]} = (\mathbf{X}_K, \mathbf{X}_{K-1}, \ldots, \mathbf{X}_1)$. For a matrix $\mathbf{X}$, the vector $\mathbf{x}^{(j)}$ is the $j^{th}$ column of $\mathbf{X}$ and the vector $\mathbf{x}^{T(j)}$ is its $j^{th}$ row. For a vector $\mathbf{x}$, $\mathbf{x}(j)$ is the $j^{th}$ element of $\mathbf{x}$. If $x \in \mathbb{R}$ then $\lfloor x \rfloor$ is the greatest integer less than or equal to $x$. We use the standard notation $\|\mathbf{x}\|_p$ for the $p$-norm ($p \geq 1$)of a vector $\mathbf{x}$. For a matrix $\mathbf{X}$ or tensor $\underline{\mathbf{X}}$, the Frobenius norms are $\|\mathbf{X}\|_F$ and $\|\underline{\mathbf{X}}\|_F$, respectively. For two vectors $\mathbf{x}_1$ and $\mathbf{x}_2$, $\mathbf{x}_1 \circ \mathbf{x}_2$ denotes their outer product. Similarly, for two matrices $\mathbf{X}_1$ and $\mathbf{X}_2$, $\mathbf{X}_1 \otimes \mathbf{X}_2$ denotes their Kronecker product. The inner product between two vectors, matrices or tensors is denoted as $\langle \cdot, \cdot \rangle$. For a set of $K$ vectors $\{\mathbf{x}_i\}_{i=1}^K$, $\mathbf{x}_1 \circ \mathbf{x}_2 \circ \cdots \circ \mathbf{x}_K$ produces a $k$-dimensional rank-1 tensor. For $K$ matrices $\{\mathbf{X}_i\}_{i=1}^K$, $\bigotimes_{k\in[K]} \mathbf{X}_k \triangleq \mathbf{X}_1 \otimes \mathbf{X}_2 \otimes \cdots \otimes \mathbf{X}_K$ produces a '$K$-order Kronecker-structured matrix'. We call a matrix a $K$-order kronecker-structured matrix if it is a product of $K \geq 2$ matrices. The mode-$k$ matricisation of tensor $\underline{\mathbf{X}}$ is $\underline{\mathbf{X}}_{(k)}$ (Kolda & Bader, 2009), and given a matrix $\mathbf{B}$, $\underline{\mathbf{X}} \times_k \mathbf{B}$ denotes the multiplication of $\underline{\mathbf{X}}$ by $\mathbf{B}$ along mode $k$. Finally, for all $k \in [K]$ we have $\underline{\mathbf{X}} \times_{[K]} \mathbf{B}_k \triangleq \underline{\mathbf{X}} \times_1 \mathbf{B}_1 \times_2 \cdots \times_K \mathbf{B}_K$.

We now formally define GLMs for vector-structured covariates, as seen in the literature.

**Definition 1** (Vector-structured Generalized Linear Models)**.** *Consider an observation $y$, a vector of covariates $\mathbf{x} \in \mathbb{R}^m$, and a bias $z$ and regression coefficient vector $\mathbf{b}$, both to be estimated. Let $y$ be a response variable generated from a distribution in the exponential family with probability mass/density function as follows:*

$$\mathbb{P}(y, \eta) = b(y) \exp(\eta T(y) - a(\eta)). \tag{1}$$

*Here, $\eta$ is called the natural parameter, $T(y)$ is the sufficient statistic and $a(\eta)$ is the log-partition (or cumulant) function. Consider a given regression problem of estimating $y$ given $\mathbf{x}$. This requires minimizing the MSE as follows:*

$$\widehat{y}_{MMSE} = \arg\min_{\widehat{y}} \mathbb{E}_{\mathbf{x},y}[(\widehat{y} - y)^T(\widehat{y} - y)] \tag{2}$$

*The Minimum MSE (MMSE) solution is the expected value of $y$ conditioned on $\mathbf{x}$, or $\mathbb{E}[y|\mathbf{x}] = \mu$. Now, let $H$ be the support of $y$ and consider a strictly increasing and invertible link function $g(\cdot) : H \to \mathbb{R}$. The Generalized Linear Model is then defined as*

$$g(\mu) = \eta \triangleq \langle \mathbf{b}, \mathbf{x} \rangle + z, \tag{3}$$

*where the natural parameter $\eta$ is the linear predictor and $\mu = g^{-1}(\eta)$.*

The core idea behind GLMs is that though many regression problems are not linear, the distribution of the observation $y$ is only affected by the linear combination $\langle \mathbf{b}, \mathbf{x} \rangle + z$. If the distribution of $y$ falls under the exponential family (which it often does), then $y$ is related to the linear predictor via $g(\cdot)$. In the case of linear regression where $y \in \mathbb{R}$ is assumed to be a continuous response variable with Gaussian distribution, we have $\mu = \langle \mathbf{b}, \mathbf{x} \rangle$ and $g(\cdot)$ is just the identity function. In logistic regression $y$ is Bernoulli distributed, taking values $y \in \{0, 1\}$, and we have $\mu = \frac{1}{1+\exp(-\langle \mathbf{b}, \mathbf{x} \rangle)}$ and $g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$. In Poisson regression $y \in \mathbb{N}$ is a Poisson distributed random variable that expresses the count of an event occurring in a fixed time interval, while $\mu = \exp(\langle \mathbf{b}, \mathbf{x} \rangle)$and $g(\mu) = \log(\mu)$.

This work is based on structured tensor decompositions. For a more comprehensive tutorial on tensor decompositions, see the survey of Kolda and Bader (Kolda & Bader, 2009). We will now list some necessary preliminaries regarding tensors and tensor structures.

**Definition 2** (CP Decomposition (Kolda & Bader, 2009; Zhou et al., 2013))**.** *Consider a $K$-mode tensor $\underline{\mathbf{B}} \in \mathbb{R}^{m_1 \times \cdots \times m_K}$. The rank-$r$ CP decomposition decomposes $\underline{\mathbf{B}}$ into a sum of $r$ rank-1 tensors as follows:*

$$\underline{\mathbf{B}} = \sum_{i \in [r]} \mathbf{b}_{1,i} \circ \cdots \circ \mathbf{b}_{K,i}, \tag{4}$$

*where $\mathbf{b}_{k,i} \in \mathbb{R}^{m_k},\ k \in [K],\ i \in [r]$ is a column vector. Equivalently, (4) can be expressed in vector form as follows:*

$$\mathrm{vec}(\underline{\mathbf{B}}) \triangleq \mathbf{b} = \sum_{i \in [r]} \mathbf{b}_{K,i} \otimes \cdots \otimes \mathbf{b}_{1,i}. \tag{5}$$

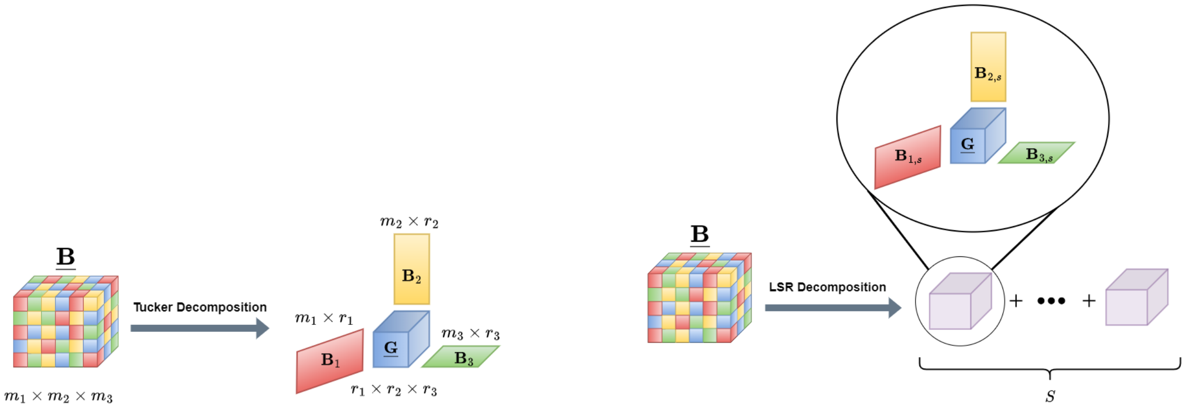
**Definition 3** (Tucker decomposition (Kolda & Bader, 2009; Li et al., 2018))**.** *Consider a $K$-mode tensor $\underline{\mathbf{B}} \in \mathbb{R}^{m_1 \times \cdots \times m_K}$. The rank-$(r_1, \ldots, r_k)$ Tucker decomposition decomposes $\underline{\mathbf{B}}$ as follows:*

$$\underline{\mathbf{B}} = \underline{\mathbf{G}} \times_1 \mathbf{B}_1 \times_2 \cdots \times_K \mathbf{B}_K, \tag{6}$$

*where $\underline{\mathbf{G}} \in \mathbb{R}^{r_1 \times \cdots \times r_K}$ denotes the core tensor and $\{\mathbf{B}_k \in \mathbb{R}^{m_k \times r_k}\}_{k \in [K]}$ denote the factor matrices. Equivalently, by defining $\mathbf{g} \triangleq \mathrm{vec}(\underline{\mathbf{G}})$, (6) can be expressed in vector form as follows:*

$$\mathrm{vec}(\underline{\mathbf{B}}) \triangleq \mathbf{b} = \big(\mathbf{B}_K \otimes \cdots \otimes \mathbf{B}_1\big)\mathbf{g}. \tag{7}$$

A visual depiction of the Tucker model is in Figure 1a. We pause to make some important observations. First, the Tucker model decomposes a tensor into a core tensor $\underline{\mathbf{G}}$ of dimension $r_1 \times \cdots \times r_K$ which has been multiplied by a factor matrix $\mathbf{B}_k$ along each mode $k \in [K]$. Assuming $\underline{\mathbf{G}}$ is small (i.e., $r_k \ll m_k\ \forall k \in [K]$), the factor matrices $\mathbf{B}_k\ \forall k \in [K]$ are low-rank matrices. Additionally, the CP model appears as a specialised case of the Tucker model as follows: Fix the number of basis vectors along all modes to some $r \in \mathbb{R}$ (i.e., fix the rank of all factor matrices to $r$) and impose $\underline{\mathbf{G}} \in \mathbb{R}^{\overset{k\ times}{r \times \cdots \times r}}$ as a diagonal tensor. The CP model lends a



(a)

(b)

Figure 1: $(a)$: A third-order tensor under the Tucker model. Tensor $\underline{\mathbf{B}}$ is decomposed into a core tensor $\underline{\mathbf{G}}$ and factor matrices $\mathbf{B}_k$ multiplied along the $k^{th}$ mode of $\underline{\mathbf{G}}$ for $k \in [3]$. The CP model appears as a special case of the Tucker model where $\underline{\mathbf{G}}$ is a diagonal tensor of equal dimension along each mode. $(b)$: A third-order tensor under the LSR decomposition. Tensor $\underline{\mathbf{B}}$ is comprised of a sum of Tucker-structured tensors, with core tensor $\underline{\mathbf{G}}$ fixed across all summands.

desirable compactness as the number of learnable parameters in a CP-structured tensor can be far fewer than that of an unstructured tensor. Despite this, however, the restrictive nature of the CP model (specifically its rank restriction, where each factor matrix must have equal rank) renders it unfavourable against the more 'rank-flexible' Tucker model.

Our second observation refers to (7). The vectorisation of a Tucker-structured tensor shows the Kronecker-structured matrix composed of the $K$ factor matrices. However this 'Kronecker' structure – which implies that the coefficient vector $\mathbf{b}$ is composed of separable sub-matrices weighted by some vector $\mathbf{g} = \text{vec}(\underline{\mathbf{G}})$ – is also quite restrictive. To overcome this, we introduce the concept of matrix separation rank.

**Definition 4** (Matrix Separation Rank (Tsiligkaridis & Hero, 2013))**.** *Fix* $\mathbf{m} = (m_1, m_2, \ldots, m_K) \in \mathbb{N}^K$ *and* $\mathbf{r} = (r_1, r_2, \ldots, r_K) \in \mathbb{N}^K$*, and set* $\widetilde{m} = \prod_{k \in [K]} m_k$ *and* $\widetilde{r} = \prod_{k\in[K]} r_k$*. Then for a matrix* $\mathbf{B} \in \mathbb{R}^{\widetilde{m} \times \widetilde{r}}$*, its separation rank* $\mathfrak{S}^K_{m,p}(\cdot)$ *is the minimum number* $S$ *of* $K$*-order Kronecker-structured matrices such that*

$$\mathbf{B} = \sum_{s\in[S]} \mathbf{B}_{(1,s)} \otimes \cdots \otimes \mathbf{B}_{(K,s)}, \tag{8}$$

*where* $\mathbf{B}_{(k,s)} \in \mathbb{R}^{m_k \times r_k}$.

We will show that the expression in (8) induces a generalisation of (7). It poses that a matrix can be expressed as a sum of $S$ Kronecker-structured matrices, with $S = 1$ being a specialised case and what we observe in the Tucker model when the tensor is vectorised, as shown in (7). Having introduced the concept of separation rank, we are now ready to define Low Separation Rank (LSR) tensors.

**Definition 5** (Low Separation Rank (LSR) Tensor Decomposition)**.** *Consider a* $K$*-mode tensor* $\underline{\mathbf{B}} \in \mathbb{R}^{m_1 \times \cdots \times m_K}$*. The rank-*$(r_1, \ldots, r_k)$ *LSR decomposition with separation rank* $S$ *decomposes* $\underline{\mathbf{B}}$ *as follows:*

$$\underline{\mathbf{B}} = \sum_{s\in[S]} \underline{\mathbf{G}} \times_1 \mathbf{B}_{(1,s)} \times_2 \cdots \times_K \mathbf{B}_{(K,s)}, \tag{9}$$

*where* $\underline{\mathbf{G}} \in \mathbb{R}^{r_1 \times \cdots \times r_K}$ *denotes the core tensor and* $\mathbf{B}_{(k,s)} \in \mathbb{R}^{m_k \times r_k}$*,* $k \in [K]$*,* $s \in [S]$ *denote the Kronecker-structured factor matrices. Equivalently, by defining* $\mathbf{g} \triangleq \text{vec}(\underline{\mathbf{G}})$*, (9) can be expressed in vector form as follows:*

$$\text{vec}(\underline{\mathbf{B}}) \triangleq \mathbf{b} = \sum_{s\in[S]} \left(\mathbf{B}_{(K,s)} \otimes \cdots \otimes \mathbf{B}_{(1,s)}\right) \mathbf{g}. \tag{10}$$

A visual depiction of the LSR model is in Figure 1b. For the purposes of this work, we pose some constraints on the LSR decomposition. First, we assume that the $K \times S$ factor matrices $\mathbf{B}_{(k,s)}$ are low rank (i.e., $r_k \ll m_k \ \forall k \in [K]$) and have orthonormal columns. Secondly, the LSR-model corresponds to tensor $\underline{\mathbf{B}}$ having a separation rank that is relatively small so that $1 \leq S < \min\left(\prod_{k\in[K]} m_k, \prod_{k\in[K]} r_k\right)$. Thus, defining $\mathbb{O}^{m\times r}$ as the manifold of matrices with orthogonal columns in $\mathbb{R}^{m\times r}$, and for a fixed tensor rank $(r_1, r_2, \ldots, r_K)$ and separation rank $S$, the LSR structured tensor $\underline{\mathbf{B}}$ belongs to the following parameter space $\mathcal{P}_{\{r_k\},S}$:

$$\mathcal{P}_{\{r_k\},S} \triangleq \Bigg\{ \underline{\mathbf{B}}' = \sum_{s\in[S]} \underline{\mathbf{G}}' \times_1 \mathbf{B}'_{(1,s)} \times_2 \cdots \times_K \mathbf{B}'_{(K,s)} \in \mathbb{R}^{m_1\times\cdots\times m_K} : \underline{\mathbf{G}}' \in \mathbb{R}^{r_1\times\cdots\times r_K}, \\ \operatorname{rank}(\underline{\mathbf{B}}') = (r_1, \ldots, r_K),\ \mathbf{B}_{(k,s)} \in \mathbb{O}^{m_k\times r_k},\ k \in [K],\ s \in [S] \Bigg\} \tag{11}$$

The orthonormality assumption on the columns of $\mathbf{B}_{(k,s)}$, is a commmon assumption made in the GLM literature (Zhang & Jiang, 2016). Similar assumptions such as unit norm columns and columns with fixed entries are also common (Zhou et al., 2013; Li et al., 2018). At this point, we do not investigate the uniqueness of the LSR decomposition, and leave the discussion of this important topic to future work. We have described how the LSR structure in (9) is a generalisation of the Tucker (and therefore of CP) decomposition. Table 1 reviews the number of learnable parameters for all models. The CP model contains the least number of parameters, especially if $r$ is small. The LSR model is the most complex of the three models in that it has more parameters than the Tucker model. The working hypothesis throughout this work is that the price we are likely to pay for an increase in sample complexity (by using the LSR model for tensor GLM problems) is worth the gain we are likely to achieve in representation power and estimation accuracy.

| | **CP** | **Tucker** | **LSR** |
|---|---|---|---|
| **Parameters** | $\sum_{k=1}^{K}(m_k r) + r$ | $\sum_{k=1}^{K}(m_k r_k) + \prod_{k=1}^{K} r_k$ | $S\sum_{k=1}^{K}(m_k r_k) + \prod_{k=1}^{K} r_k$ |

Table 1: Number of learnable parameters in the three tensor models (CP, Tucker and LSR).

## 3 Problem Statement

We are now ready to propose the Low Separation Rank Tensor Generalized Linear Model (LSR-TGLM). Consider a response variable $y$ with probability distribution belonging to the exponential family with parameter $\eta$, as in (1). Consider also tensor structured covariates $\underline{\mathbf{X}} \in \mathbb{R}^{m_1\times_1\cdots\times_k m_K}$ and a coefficient tensor $\underline{\mathbf{B}} \in \mathbb{R}^{m_1\times_1\cdots\times_k m_K}$ that assumes a low-rank LSR structure as shown in (9). Given a link function $g(\cdot)$, the LSR-TGLM model assumes $\eta$ is given by

$$g(\mu) = \eta \triangleq \left\langle \sum_{s\in[S]} \underline{\mathbf{G}} \times_1 \mathbf{B}_{(1,s)} \times_2 \cdots \times_K \mathbf{B}_{(K,s)}, \underline{\mathbf{X}} \right\rangle + z. \tag{12}$$

It is important to note here that we assume that the tensor rank $(r_1, r_2, \ldots, r_K)$ and LSR rank $S$ are known. We will discuss this point in more detail in Section 3.1. Additionally for algebraic simplicity, from this point forward we will consider the standard case where $z = 0$ in (3) and without loss of generality we will express the LSR-TGLM model as

$$g(\mu) = \left\langle \sum_{s\in[S]} \underline{\mathbf{G}} \times_1 \mathbf{B}_{(1,s)} \times_2 \cdots \times_K \mathbf{B}_{(K,s)}, \underline{\mathbf{X}} \right\rangle. \tag{13}$$

The LSR-TGLM model in (13) can be written as a standard GLM by vectorizing the parameters as follows

$$g(\mu) = \langle \operatorname{vec}(\underline{\mathbf{B}}), \operatorname{vec}(\underline{\mathbf{X}}) \rangle = \left\langle \sum_{s\in[S]} \left(\mathbf{B}_{(K,s)} \otimes \cdots \otimes \mathbf{B}_{(1,s)}\right) \mathbf{g}, \mathbf{x} \right\rangle.$$

The sum of Kronecker-structured matrices in (14) clearly appears in the LSR-TGLM model in (13). We can now formally define the two goals that are at the core of this work: 1) Parameter estimation for LSR-TGLM and 2) Minimax lower bound on the estimation error.

### 3.1 Parameter Estimation for LSR-TGLM

The objective in regression theory is to predict an outcome $y$ based on $\underline{\mathbf{X}}$, which is achieved by first estimating the model parameter. For LSR-TGLMs, we wish to find an estimate of $\underline{\mathbf{B}}$ that best fits the model in (13). The underlying (true)$\underline{\mathbf{B}}$ is a low-rank LSR-structured tensor belonging to a constraint set $\mathcal{C}$:

$$\mathcal{C} \triangleq \left\{ \underline{\mathbf{B}} \in \mathbb{R}^{m_1 \times \cdots \times m_K} : \underline{\mathbf{B}} \in \mathcal{P}_{\{r_k\},S}, r_k \leq R_K, S \leq \min\left\{ \prod_k m_k, \prod_k r_k \right\}, k \in [K] \right\}. \tag{14}$$

This is the set of all LSR-structured tensors with constrained tensor rank tuple $(r_1, r_2, \ldots, r_K)$ and constrained separation rank $S$. Now consider the space of tensor-structured covariates $\mathbb{X} \subset \mathbb{R}^{m_1 \times m_2 \times \cdots \times m_K}$, and space of scalar observations $\mathbb{Y} \subset \mathbb{R}$. There exists a probability measure, denoted as $\mathcal{P}_{xy}$, that allows a learning procedure to randomly draw points $\{\underline{\mathbf{X}} \in \mathbb{X}, y \in \mathbb{Y}\}$ from the product space $\mathbb{X} \times \mathbb{Y}$. We do not have access to $\mathcal{P}_{xy}$, but we can draw $n$ independent observations $\{\underline{\mathbf{X}}_i, y_i\}_{i=1}^n$. Therefore, we find an estimate of $\underline{\mathbf{B}}$ based on $n$ independent observations through Maximum Likelihood Estimation (MLE). This makes the objective function the negative log-likelihood:

$$\mathcal{L}_n(\underline{\mathbf{B}}) = \sum_{i=1}^{n} \log(b(y_i)) + \sum_{i=1}^{n} \left( \langle \underline{\mathbf{B}}, \underline{\mathbf{X}}_i \rangle^T T(y_i) - a(\langle \underline{\mathbf{B}}, \underline{\mathbf{X}}_i \rangle) \right). \tag{15}$$

To find the MLE, we minimize (15) over the constraint set $\mathcal{C}$:

$$\underset{\underline{\mathbf{B}} \in \mathcal{C}}{\arg\min} \sum_{i=1}^{n} \log(b(y_i)) + \sum_{i=1}^{n} \left( \langle \underline{\mathbf{B}}, \underline{\mathbf{X}}_i \rangle T(y_i) - a(\langle \underline{\mathbf{B}}, \underline{\mathbf{X}}_i \rangle) \right). \tag{16}$$

It follows from Lemma 1 from Ghassemi et. al (Ghassemi et al., 2019) that since finding the rank $r$ of a tensor is NP-hard (Håstad, 1989), then finding the separation rank of a Kronecker structured matrix is also NP-hard. The problem in (16) is therefore intractable. To mitigate this issue we first assume that the tensor rank $(r_1, r_2, \ldots, r_K)$ and separation rank $S$ are known and we solve the following factorized problem for parameter estimation for LSR-TGLMs:

$$\begin{aligned} \underset{\{\mathbf{B}_{(k,s)}\}, \underline{\mathbf{G}}}{\arg\min} &= \sum_{i=1}^{n} \log(b(y_i)) + \sum_{i=1}^{n} \left\langle \sum_{s=1}^{S} \underline{\mathbf{G}} \times_{[K]} \mathbf{B}_k, \underline{\mathbf{X}}_i \right\rangle T(y_i) - a\left( \left\langle \sum_{s=1}^{S} \underline{\mathbf{G}} \times_{[K]} \mathbf{B}_k, \underline{\mathbf{X}}_i \right\rangle \right). \\ &\text{subject to } \mathbf{B}_{(k,s)} \in \mathbb{O}^{m_k \times r_k}. \end{aligned} \tag{17}$$

The expression in (17) provides a tractable relaxation for (16), where the coefficient tensor is explicitly written in terms of the core tensor $\underline{\mathbf{G}}$ and factor matrices $\mathbf{B}_{(k,s)}$ of the low-rank LSR structure. Thus, in this work, we will study the problem in (17).

### 3.2 Minimax Lower Bound for LSR-TGLM

Our second goal is to derive a lower bound on the minimax risk of estimating LSR-structured coefficient tensors for the LSR-TGLM problem in (13). Minimax bounds can be useful tools in developing an insight into the parameters on which an achievable error of a given problem might depend and shed light on the benefits of imposing tensor structures in regression problems. They also provide a means of quantifying the performance of existing algorithms.

We adopt a local analysis and assume that the LSR-TGLM's underlying (true) $\underline{\mathbf{B}}$ resides within a neighbourhood with known radius around a fixed point. However, for a sufficiently large neighborhood, the minimax lower bounds derived in this work effectively become independent of the radius. Therefore we assume that for a fixed tensor rank $(r_1, r_2, \ldots, r_K)$ and separation rank $S$, the underlying $\underline{\mathbf{B}}$ belongs to the set

$$\mathcal{B}_d(\underline{\mathbf{0}}) \triangleq \{\underline{\mathbf{B}}' \in \mathcal{P}_{\{r_k\},S} : \rho(\underline{\mathbf{B}}', \underline{\mathbf{0}}) < d\}, \tag{18}$$

the open ball of radius $d$ with distance metric $\rho = \|\cdot\|_F$, which resides in the parameter space $\mathcal{P}_{\{r_k\},S}$ defined in (11). Thus $\mathcal{B}_d(\mathbf{0}) \subset \mathcal{P}_{\{r_k\},S}$ and $\underline{\mathbf{B}}$ has energy bounded since $\|\underline{\mathbf{B}}\|_F^2 < d^2$. Note that we fix the reference point as the tensor of all zero-elements $\underline{\mathbf{0}}$ without loss of generality. Indeed, any neighbourbood $\mathcal{B}_d(\underline{\mathbf{A}})$ around a point $\underline{\mathbf{A}} \in \mathcal{P}_{\{r_k\},S}$ is just a translation from $\mathcal{B}_d(\underline{\mathbf{0}})$ with known distance $\|\underline{\mathbf{A}}\|_F$. Additionally, we note that the point $\underline{\mathbf{0}}$ also belongs to the parameter space $\mathcal{P}_{\{r_k\},S}$. The minimax risk is defined as the minimum worst-case behaviour for any estimator. Mathematically, it is expressed as follows.

$$\varepsilon^* = \inf_{\widehat{\underline{\mathbf{B}}}} \sup_{\underline{\mathbf{B}} \in \mathcal{B}_d(\underline{\mathbf{0}})} \mathbb{E}_{\mathbf{y},\underline{\mathbf{X}}^c} \left\{ \phi(\widehat{\underline{\mathbf{B}}}, \underline{\mathbf{B}}) \right\}. \tag{19}$$

Here, $\widehat{\underline{\mathbf{B}}}$ denotes an estimator of $\underline{\mathbf{B}}$, the metric $\phi$ is a non-decreasing function with $\phi(0) = 0$. If we define $\phi = \|\cdot\|_F^2 \triangleq \mathbb{R}_+ \to \mathbb{R}_+$ with $\phi(0) = 0$ then the minimax risk is simply the worst-case Mean Squared Error (MSE) for the best estimator, i.e.,

$$\varepsilon^* = \inf_{\widehat{\underline{\mathbf{B}}}} \sup_{\underline{\mathbf{B}} \in \mathcal{B}_d(\underline{\mathbf{0}})} \mathbb{E}_{\mathbf{y},\underline{\mathbf{X}}^c} \left\{ \left\| \widehat{\underline{\mathbf{B}}} - \underline{\mathbf{B}} \right\|_F^2 \right\}. \tag{20}$$

We point out that the minimax risk in (20) is an inherent property of the LSR-TGLM parameter estimation problem. It holds for all possible estimators and can be used to assess the performance of all possible algorithms. We pause to note that existing minimax bounds on the parameter estimation problem in GLMs or regression models provided in the literature, (Abramovich & Grinshtein, 2016; Lee & Courtade, 2020; Raskutti et al., 2011), cannot be utilised here for two reasons. Primarily, these bounds do not account for the impact of the structural assumptions we make on our model. In fact, we require bounds that accurately reflect the sample complexity of estimation algorithms for LSR structured coefficient tensors. Secondly, the link function in GLMs makes part of our analysis non-trivial and fundamentally different to such works. The derived minimax bound in this section conveniently generalizes the CP and Tucker tensor structures and can be specialized to existing bounds in the literature such as that for Tucker structured linear regression by Zhang et al. (Zhang et al., 2020).

We now introduce some standard lemmas and assumptions used in this work.

**Assumption 1** (Covariate Distribution)**.** *For* $\underline{\mathbf{X}} \in \mathbb{R}^{m_1 \times \cdots \times m_K}$*, define* $\mathrm{vec}(\underline{\mathbf{X}}) \triangleq \mathbf{x}$ *and* $\widetilde{m} = \prod_{k \in [K]} m_k$*. Then,* $\mathbb{E}[\mathbf{x}] = \mathbf{0}$ *and* $\mathbb{E}[\mathbf{x}\mathbf{x}^T] = \mathbf{\Sigma}_x$*.*

**Lemma 1** ((McCullagh & Nelder, 2019))**.** *Any observation $y$ generated according to a distribution from the exponential family has mean $a'(\eta)$, i.e., the first derivative of $a(\eta)$, and variance $a''(\eta)$, i.e., the second derivative of $a(\eta)$.*

**Assumption 2.** *The first derivative of the cumulant function, $a(\eta)$, with respect to $\eta$ is bounded uniformly by a constant $M \geq 0$: $a'(\eta) \leq M$.*

**Lemma 2.** *Consider the standard GLM problem in Definition 1 with negative log-likelihood function*

$$\sum_{i=1}^{n} \log(b(y_i)) + \sum_{i=1}^{n} \left( \langle \mathbf{b}, \mathbf{x}_i \rangle^T T(y_i) - a(\langle \mathbf{b}, \mathbf{x}_i \rangle) \right). \tag{21}$$

*The gradient of* (21) *with respect to* $\mathbf{b}$ *is* $\sum_{i=1}^{n} \left( T(y_i) - g^{-1}(\langle \mathbf{b}, \mathbf{x}_i \rangle) \right) \mathbf{x}_i$*.*

The proof of Lemma 2 follows the same steps explained in (McCullagh & Nelder, 2019). Assumption 1 states that $\mathrm{vec}(\underline{\mathbf{X}}) \triangleq \mathbf{x}$ is a zero-mean Gaussian random variable with covariance matrix $\sigma_x^2$. We do not place any further assumptions on the distribution of $\underline{\mathbf{X}}$. Assumption 2 implies that the mean of any GLM observation $y$ is bounded. This is a common assumption made in the literature (Lee & Courtade, 2020). For binary logistic regression, e.g., this assumption is satisfied with $M = 1$. For linear or Poisson regression, this assumption implies that the energy of $y$ is bounded. Though the range of $y$ is $\mathbb{R}$ and $\mathbb{N}$ for linear and Poisson regression, respectively, any outcome can be bounded by a positive constant $M$ with high probability in most instances.

## 4 Estimation Problem and Algorithm

To solve (17) we propose an approach similar to those found in (Li et al., 2018; Zhou et al., 2013; Zhang & Jiang, 2016; Tan et al., 2012). The main idea behind this approach is to recognize that although the objective function in (17) is non-convex in all elements $\underline{\mathbf{G}}$ and $\{\mathbf{B}_{(k,s)}\}_{k\in[K],s\in[S]}$ jointly, it is convex with respect to each element separately. In this work, we propose a Block Coordinate Descent (BCD) algorithm that estimates each element of the LSR structured tensor $\underline{\mathbf{B}}$ alone while holding all other elements constant. We note for the reader that BCD is simply Alternating Minimization (AM) when there are only two factors to be estimated. Additionally, one may use any solver to solve each convex sub-problem to estimate each element, however, we choose to solve it via gradient steps in this work. First, for every $k' \in [K]$ and $s' \in [S]$, we estimate the factor matrix $\mathbf{B}_{k',s'}$ while keeping all other factor matrices $\{\mathbf{B}_{(k,s}\}_{k\in[K]\backslash k', s\in[S]\backslash s'}$ and core tensor $\underline{\mathbf{G}}$ fixed. Secondly, we estimate the core tensor $\underline{\mathbf{G}}$ while keeping all factor matrices $\{\mathbf{B}_{(k,s}\}_{k\in[K],s\in[S]}$ fixed. We derive the optimization problem for each step of the algorithm below.

### 4.1 Algorithm: Estimating Factor Matrices

To estimate the factor matrices we perform gradient descent over our objective function and project each iterate onto the set of orthogonal matrices, with the objective function corresponding to the block $\mathbf{B}_{(k',s')}$ given by:

$$
\begin{aligned}
&\mathcal{L}_n\left(\mathbf{B}_{(k',s')}\right) \\
&\quad= \sum_{i=1}^{n}\left(\left\langle \mathbf{B}_{(k',s')}, \mathbf{X}_{i_{(k')}}\left(\bigotimes_{-[K]\backslash k'} \mathbf{B}_{(k,s')}\right)\mathbf{G}^T_{(k')}\right\rangle + \sum_{s\in[S]\backslash s'}\left\langle \underline{\mathbf{G}} \times_{[K]} \mathbf{B}_{(k,s)}, \underline{\mathbf{X}}_i\right\rangle\right) T(y_i) \\
&\quad\quad - a\left(\left\langle \mathbf{B}_{(k',s')}, \mathbf{X}_{i_{(k')}}\left(\bigotimes_{-[K]\backslash k'} \mathbf{B}_{(k,s')}\right)\mathbf{G}^T_{(k')}\right\rangle + \sum_{s\in[S]\backslash s'}\left\langle \underline{\mathbf{G}} \times_{[K]} \mathbf{B}_{(k,s)}, \underline{\mathbf{X}}_i\right\rangle\right).
\end{aligned} \tag{22}
$$

In order to derive the expression in (22) for every $\mathbf{B}_{k',s'}$ for $k' \in [K]$ and $s' \in [S]$, notice that:

$$
\begin{aligned}
\langle \underline{\mathbf{B}}, \underline{\mathbf{X}} \rangle &= \left\langle \underline{\mathbf{G}} \times_1 \mathbf{B}_{(1,s')} \times_2 \cdots \times_K \mathbf{B}_{(K,s')}, \underline{\mathbf{X}} \right\rangle + \sum_{s\in[S]\backslash s'} \left\langle \underline{\mathbf{G}} \times_1 \mathbf{B}_{(1,s)} \times_2 \cdots \times_K \mathbf{B}_{(K,s)}, \underline{\mathbf{X}} \right\rangle \\
&= \left\langle \mathbf{B}_{(k',s')}, \mathbf{X}_{(k')}\left(\mathbf{B}_{(K,s')} \otimes \cdots \otimes \mathbf{B}_{(k'+1,s')} \otimes \mathbf{B}_{(k'-1,s')} \otimes \cdots \otimes \mathbf{B}_{(1,s')}\right)\mathbf{G}^T_{(k')} \right\rangle \\
&\quad + \sum_{s\in[S]\backslash s'} \left\langle \underline{\mathbf{G}} \times_1 \mathbf{B}_{(1,s)} \times_2 \cdots \times_K \mathbf{B}_{(K,s)}, \underline{\mathbf{X}} \right\rangle.
\end{aligned} \tag{23}
$$

Now we define the following notations in order to keep the expressions in (24) concise. From the first summand in (24) we define

$$
\boldsymbol{\omega}_{(k',s')} = \mathrm{vec}\left(\mathbf{X}_{(k')}\left(\mathbf{B}_{(K,s')} \otimes \cdots \otimes \mathbf{B}_{(k'+1,s')} \otimes \mathbf{B}_{(k'-1,s')} \otimes \cdots \otimes \mathbf{B}_{(1,s')}\right)\mathbf{G}^T_{(k')}\right), \tag{24}
$$

and from the second summand in (24) we define

$$
\gamma_{(k',s')} = \sum_{s\in[S]\backslash s'} \left\langle \underline{\mathbf{G}} \times_1 \mathbf{B}_{(1,s)} \times_2 \cdots \times_K \mathbf{B}_{(K,s)}, \underline{\mathbf{X}} \right\rangle. \tag{25}
$$

We can then express (24) as

$$
\langle \underline{\mathbf{B}}, \underline{\mathbf{X}} \rangle = \left\langle \mathrm{vec}(\mathbf{B}_{(k',s')}, \boldsymbol{\omega}_{(k',s')}) \right\rangle + \gamma_{(k',s')} = \left\langle \left[\mathrm{vec}(\mathbf{B}_{(k',s')}), 1\right], \left[\boldsymbol{\omega}_{(k',s')}, \gamma_{(k',s')}\right] \right\rangle. \tag{26}
$$

By defining $\widetilde{\mathbf{x}} \triangleq \left[\boldsymbol{\omega}_{(k',s')}, \gamma_{(k',s')}\right]$, we rewrite (22) as

$$
\underset{\mathbf{B}_{(k',s')} \in \mathbb{O}^{m_{k'} \times r_{k'}}}{\arg\min} \sum_{i=1}^{n} \left(\left\langle \left[\mathrm{vec}(\mathbf{B}_{(k',s')}), 1\right], \widetilde{\mathbf{x}}_i \right\rangle\right) T(y_i) - a\left(\left\langle \left[\mathrm{vec}(\mathbf{B}_{(k',s')}), 1\right], \widetilde{\mathbf{x}}_i \right\rangle\right). \tag{27}
$$

The most important insight here is that the resulting problem can be viewed as a parameter estimation problem for GLMs with $\mathbf{B}_{(k',s')}$ as the 'parameter' and $\widetilde{\mathbf{x}}_i$ as the 'predictor' or structured covariates. Estimating $\mathbf{B}_{(k',s')}$ alone in particular results in a low-dimensional problem with $m_{k'}r_{k'}$ parameters. We solve (22) via projected gradient descent. Using the above defined notations, the gradient of $\mathcal{L}_n(\mathbf{B}_{(k',s')})$ with respect to $\mathbf{B}_{(k',s')}$ is

$$\frac{\partial \mathcal{L}_n}{\partial \mathrm{vec}(\mathbf{B}_{k',s'})} = \sum_{i=1}^{n} \left(T(y_i) - g^{-1}\left(\left\langle \left[\mathrm{vec}(\mathbf{B}_{(k',s')}), 1\right], \widetilde{\mathbf{x}}_i \right\rangle\right)\right) \widetilde{\mathbf{x}}_i, \tag{28}$$

and the projection operator $\mathcal{H} : \mathbb{R}^{m_{k'} \times r_{k'}} \longrightarrow \mathbb{O}^{m_{k'} \times r_{k'}}$ that projects the obtained iterate onto the manifold of orthogonal matrices is defined as

$$\mathcal{H}(\mathbf{B}_{(k',s')}) \triangleq \underset{\widehat{\mathbf{B}} \in \mathbb{O}^{m_{k'} \times r_{k'}}}{\arg\min} \left\| \widehat{\mathbf{B}} - \mathbf{B}_{(k',s')} \right\|_F^2. \tag{29}$$

The solution to the problem in (29) is simply obtained using the QR decomposition of $\mathbf{B}_{(k',s')}$.

### 4.2 Algorithm: Estimating the Core Tensor

For core tensor $\underline{\mathbf{G}}$, we are only required to perform gradient descent. The optimization problem is

$$\underset{\underline{\mathbf{G}}}{\arg\min}\ \mathcal{L}_n\left(\underline{\mathbf{G}}\right), \text{ where} \tag{30}$$

$$\mathcal{L}_n\left(\underline{\mathbf{G}}\right) = \sum_{i=1}^{n} \left( \left\langle \mathbf{g}, \sum_{s \in [S]} \left( \bigotimes_{-[K]} \mathbf{B}_{(k,s)} \right)^T \mathbf{x}_i \right\rangle \right) T(y_i) - a \left( \left\langle \mathbf{g}, \sum_{s \in [S]} \left( \bigotimes_{-[K]} \mathbf{B}_{(k,s)} \right)^T \mathbf{x}_i \right\rangle \right).$$

In order to see the derivative of the expression in (30), notice that,

$$\langle \underline{\mathbf{B}}, \underline{\mathbf{X}} \rangle = \langle \mathrm{vec}(\underline{\mathbf{B}}), \mathrm{vec}(\underline{\mathbf{X}}) \rangle = \sum_{s \in [S]} \left\langle \left( \bigotimes_{-[K]} \mathbf{B}_{(k,s)} \right) \mathbf{g}, \mathbf{x} \right\rangle = \left\langle \mathbf{g}, \sum_{s \in [S]} \left( \bigotimes_{-[K]} \mathbf{B}_{(k,s)} \right)^T \mathbf{x} \right\rangle. \tag{31}$$

With a slight overload of notation we also define the following:

$$\widetilde{\mathbf{x}} = \sum_{s \in [S]} \left( \mathbf{B}_{(K,s)} \otimes \cdots \otimes \mathbf{B}_{(1,s)} \right)^T \mathbf{x}, \tag{32}$$

and further express (31) and (30) as

$$\langle \underline{\mathbf{B}}, \underline{\mathbf{X}} \rangle = \langle \mathbf{g}, \widetilde{\mathbf{x}} \rangle \tag{33}$$

and

$$\underset{\underline{\mathbf{G}}}{\arg\min} \sum_{i=1}^{n} \left( \langle \mathbf{g}, \widetilde{\mathbf{x}}_i \rangle \right) T(y_i) - a\left( \langle \mathbf{g}, \widetilde{\mathbf{x}}_i \rangle \right), \tag{34}$$

respectively. Once again, the resulting problem can be viewed as a parameter estimation problem for GLMs with $\underline{\mathbf{G}}$ as the 'parameter' and $\widetilde{\mathbf{x}}_i$ as the 'predictor' or structured covariates. Estimating $\underline{\mathbf{G}}$ alone results in a low-dimensional problem with $\prod_{k \in [K]} r_k$ parameters. We solve (30) via gradient descent, where the gradient of $\mathcal{L}_n(\underline{\mathbf{G}})$ with respect to $\underline{\mathbf{G}}$ is

$$\frac{\partial \mathcal{L}_n}{\partial \mathrm{vec}(\underline{\mathbf{G}})} = \sum_{i=1}^{n} \left( T(y_i) - g^{-1}\left( \langle \mathbf{g}, \widetilde{\mathbf{x}}_i \rangle \right) \right) \widetilde{\mathbf{x}}_i. \tag{35}$$

### 4.3 Final Algrithm: LSRTR

We summarize the procedure discussed above in Algorithm 1 and we name our algorithm Low Separation Rank Tensor Regression (LSRTR). We also show the prediction procedure performed using the estimated coefficient tensor in Algorithm 2, where we calculate the mean $\mathbb{E}[y|\underline{\mathbf{X}}]$ of the posterior distribution based on the estimated coefficient tensor $\underline{\mathbf{B}}$ from Algorithm 1. The posterior mean allows us to make predictions for an observation $y$ and report confidence probabilities.

**Algorithm 1** LSRTR: A block coordinate descent algorithm for LSR-TGLMs

```
Input: $n$ training samples $\{\underline{\mathbf{X}}_i, y_i\}_{i=1}^n$, step size $\alpha$, separation rank $S$, tensor rank $\{r_1, r_2, \ldots, r_K\}$.
Initialise: Factor matrices $\mathbf{B}^0_{(k,s)}$ $\forall\, k \in [K], s \in [S]$, core tensor $\underline{\mathbf{G}}^0$ and $t \leftarrow 0$.
repeat:
    for $s' \in [S]$ do
        for $k' \in [K]$ do
            $\widetilde{\mathbf{B}}^{(t)}_{(k',s')} \leftarrow \mathrm{vec}\left(\mathbf{B}^{(t)}_{(k',s')}\right) - \alpha \sum_{i=1}^n \left(T(y_i) - g^{-1}\left(\left\langle \left[\mathrm{vec}(\mathbf{B}_{(k',s')}), 1\right], \widetilde{\mathbf{x}}_i\right\rangle\right)\right) \widetilde{\mathbf{x}}_i$
            $\mathbf{B}^{(t+1)}_{(k',s')} \leftarrow \mathcal{H}\left(\widetilde{\mathbf{B}}^{(t)}_{(k',s')}\right)$
        end for
    end for
    $\widetilde{\underline{\mathbf{G}}}^{(t+1)} \leftarrow \mathrm{vec}(\underline{\mathbf{G}}^{(t)}) - \alpha \sum_{i=1}^n \left(T(y_i) - g^{-1}\left(\langle \mathbf{g}, \widetilde{\mathbf{x}}_i \rangle\right)\right) \widetilde{\mathbf{x}}_i$
    $t \leftarrow t + 1$
until convergence
return $\widehat{\underline{\mathbf{B}}} \leftarrow \sum_{s \in [S]} \underline{\mathbf{G}}^{(t)} \times_1 \mathbf{B}^{(t)}_{(1,s)} \times_2 \cdots \times_K \mathbf{B}^{(t)}_{(K,s)}$
```

**Algorithm 2** Posterior prediction for LSR-TGLMs

```
Input Estimate $\widehat{\underline{\mathbf{B}}} \in \mathbb{R}^{m_1 \times \cdots \times m_K}$ and $n_{te}$ test data points $\{\underline{\mathbf{X}}_i\}_{i=1}^{n_{te}}$
Output Expectation $\widehat{\boldsymbol{\mu}} = [\mathbb{E}[y_1|\underline{\mathbf{X}}_1], \mathbb{E}[y_2|\underline{\mathbf{X}}_2] \ldots \mathbb{E}[y_{n_{te}}|\underline{\mathbf{X}}_{n_{te}}]$
Define: $\underline{\mathcal{X}} \triangleq \left[\underline{\mathbf{X}}_1, \underline{\mathbf{X}}_2, \ldots, \underline{\mathbf{X}}_{n_{te}}\right]]$
Compute $\widehat{\boldsymbol{\mu}}$ for input $\underline{\mathcal{X}}$ as: $\widehat{\boldsymbol{\mu}} = g^{-1}(\langle \widehat{\underline{\mathbf{B}}}, \underline{\mathcal{X}} \rangle$
return $\widehat{\boldsymbol{\mu}}$
```

In Section 5 we assess the performance of our proposed algorithm on various GLM problems for 2-dimensional and 3-dimensional synthetic data. Moreover, in Section 7, we use the LSR-TGLM model and the proposed LSRTR algorithm to solve classification problems on several real world medical imaging datasets. Though we do not provide any convergence guarantees for Algorithm 1, we make some promising remarks on its practical performance and leave the theoretical analysis for future works.

## 5 Numerical Study

In this section we provide a comprehensive numerical study in order to assess the efficacy of the LSR-TGLM regression model and the corresponding proposed LSRTR algorithm. The objectives of this study are three-fold. First, we investigate the performance of our algorithm, particularly: 1) The performance of our proposed approach (parameter estimation and prediction) against a substantial growth in number of parameters, and 2) The performance of our algorithm, particularly when faced with a large increase in sample size. Secondly, we compare the estimation and prediction accuracy gains of the LSR model to current regression models and algorithms in the literature (Zhou et al., 2013; Li et al., 2018; Zhang et al., 2020; McCullagh & Nelder, 2019; Seber & Lee, 2003).

### 5.1 Experiments on Synthetic Data

We begin with experiments on synthetic data for three GLM problems: linear regression, logistic regression and Poisson regression. The purpose of the experiments is to answer the following questions:

1. When the underlying coefficient tensor has an LSR structure, does the proposed LSRTR algorithm lead to reliable parameter estimation? Specifically, does our algorithm achieve a reduction in sample complexity and estimation error compared to vector-based methods (for a fixed sample size)?
2. Does our proposed algorithm: 1) provide reliable parameter estimation and prediction against a substantial growth in number of parameters and 2) have per-iteration computation time comparable to other tensor-based methods, when faced with a large increase in sample size?

In our experiments we generate an $(r_1, r_2, \ldots, r_K)$-rank LSR structured coefficient tensor $\underline{\mathbf{B}} \in \mathbb{R}^{m_1 \times \cdots \times m_K}$ with separation rank $S$ in (9) as follows. The entries of the core tensor $\underline{\mathbf{G}}$ are sampled from a Gaussian$(0, \frac{1}{r_1 r_2 \ldots r_K})$ distribution. Each factor matrix $\mathbf{B}_{(k,s)}$ is constructed by first constructing matrix $\mathbf{A}_{(k,s)} \in \mathbb{R}^{m_k \times r_k}$ with independent standard normal entries. The QR decomposition is then applied to $\mathbf{A}_{(k,s)}$ in order to obtain the orthonormal matrix $\widetilde{\mathbf{A}}_{(k,s)} \in \mathbb{O}^{m_k \times m_k}$. The first $r_k$ columns are then extracted from $\widetilde{\mathbf{A}}_{(k,s)}$ in order to obtain $\mathbf{B}_{(k,s)}$. With the above construction, we have $\|\underline{\mathbf{B}}\|_F^2 \leq S^2 \widetilde{r} \, \|\underline{\mathbf{G}}\|_F^2$ (we derive a more precise bound on $\|\underline{\mathbf{B}}\|_F^2$ in Section 6.3). We also generate $n$ data samples $\{\underline{\mathbf{X}}, y\}$ with $n_{tr}$ samples used for training and estimation, and $n_{te}$ samples used for testing and prediction. We generate independent tensor-structured covariates $\{\underline{\mathbf{X}}_i\}_{i=1}^n$ with zero-mean Gaussian entries and covariance $\mathbf{\Sigma}_x$, and observations $\{y_i\}_{i=1}^n$ which are randomly generated according to the probabilistic model in (13) and with appropriate link function $g(\mu)$. For linear regression, the observation $y_i$ conditioned on data $\underline{\mathbf{X}}_i$ follows a Gaussian distribution, i.e., $y_i \sim \mathcal{N}\left(\langle \underline{\mathbf{B}}, \underline{\mathbf{X}}_i, \rangle, \sigma_x^2 \underline{\mathbf{I}}_m\right)$. In logistic regression, the observation $y_i$ conditioned on data $\underline{\mathbf{X}}_i$ follows a Bernoulli distribution, i.e., $y_i \sim$ Bernoulli $\left(\frac{1}{1+\exp(-\langle \underline{\mathbf{B}}, \underline{\mathbf{X}}_i \rangle)}\right)$. Finally, in Poisson regression, the observation $y_i$ conditioned on data $\underline{\mathbf{X}}_i$ follows a Poisson distribution, i.e., $y_i \sim$ Poisson $(\exp(\langle \underline{\mathbf{B}}, \underline{\mathbf{X}}_i \rangle))$.

The experiment for each GLM problem is performed for various model sizes $\{m_k\}_{k \in [K]}$, tensor ranks $\{r_k\}_{k \in [K]}$ and separation rank $S$. The coefficient tensor $\underline{\mathbf{B}}$ and data samples $\{\underline{\mathbf{X}}_i, y_i\}_{i=1}^n$ are generated as described above and each experiment is repeated for increasing value of $n_{tr}$. We compute and report the following:

1. A coefficient tensor $\widehat{\underline{\mathbf{B}}}$ estimated through the following learning methods: $(i)$ The LSRTR method (Algorithm 1), $(ii)$ Low-rank Tucker model methods, specifically, a 'block relaxation' algorithm for parameter estimation of Tucker-structured GLMs proposed by Li et. al (we will call this procedure TTR)(Li et al., 2018), and a similar iterative procedure for logistic Tucker regression (LTuR) with Frobenius norm regularization proposed by Zhang et. al (Zhang & Jiang, 2016), $(iii)$ unstructured (vector-based) methods for regression. In the case of $(i.i.i)$ we use Least Squares (LS) for linear regression, a first-order method we name LR (Seber & Lee, 2003) for logistic regression, and a GLM fitting algorithm we name PR, with log link function from (McCullagh & Nelder, 2019) for Poisson regression. The performance of each method is evaluated via the normalised estimation error defined as $\frac{\left\|\underline{\mathbf{B}} - \widehat{\underline{\mathbf{B}}}\right\|_F^2}{\|\underline{\mathbf{B}}\|_F^2}$.
2. A predicted response vector $\widehat{\mathbf{y}} = [\widehat{y}_1, \widehat{y}_2, \ldots, \widehat{y}_{n_{te}}] = [\widehat{\mu}_1, \widehat{\mu}_2, \ldots, \widehat{\mu}_{n_{te}}]$ where $\widehat{\mu}_i = g^{(-1)}\left(\left\langle \widehat{\underline{\mathbf{B}}}, \underline{\mathbf{X}}_i \right\rangle\right)$. The prediction accuracy for linear, logistic and Poisson regression is evaluated via the normalised squared error defined as $\frac{\|\widehat{\mathbf{y}} - \mathbf{y}\|_2^2}{\|\mathbf{y}\|_2^2}$, the Mean Absolute Error (MAE) defined as $\frac{\|\widehat{\mathbf{y}} - \mathbf{y}\|_1}{n_{te}}$, and the normalised squared logarithmic error defined as $\frac{\left\|\log(\widehat{\mathbf{y}}+1) - \log(\mathbf{y}+1)\right\|_2^2}{\|\log(\mathbf{y}+1)\|_2^2}$, respectively.

Finally, for each GLM problem, unless otherwise stated, we conduct 50 repetitions of each experiment (for a fixed coefficient tensor $\underline{\mathbf{B}}$) and average the error over the repetitions.

#### 5.1.1 2D Synthetic Data

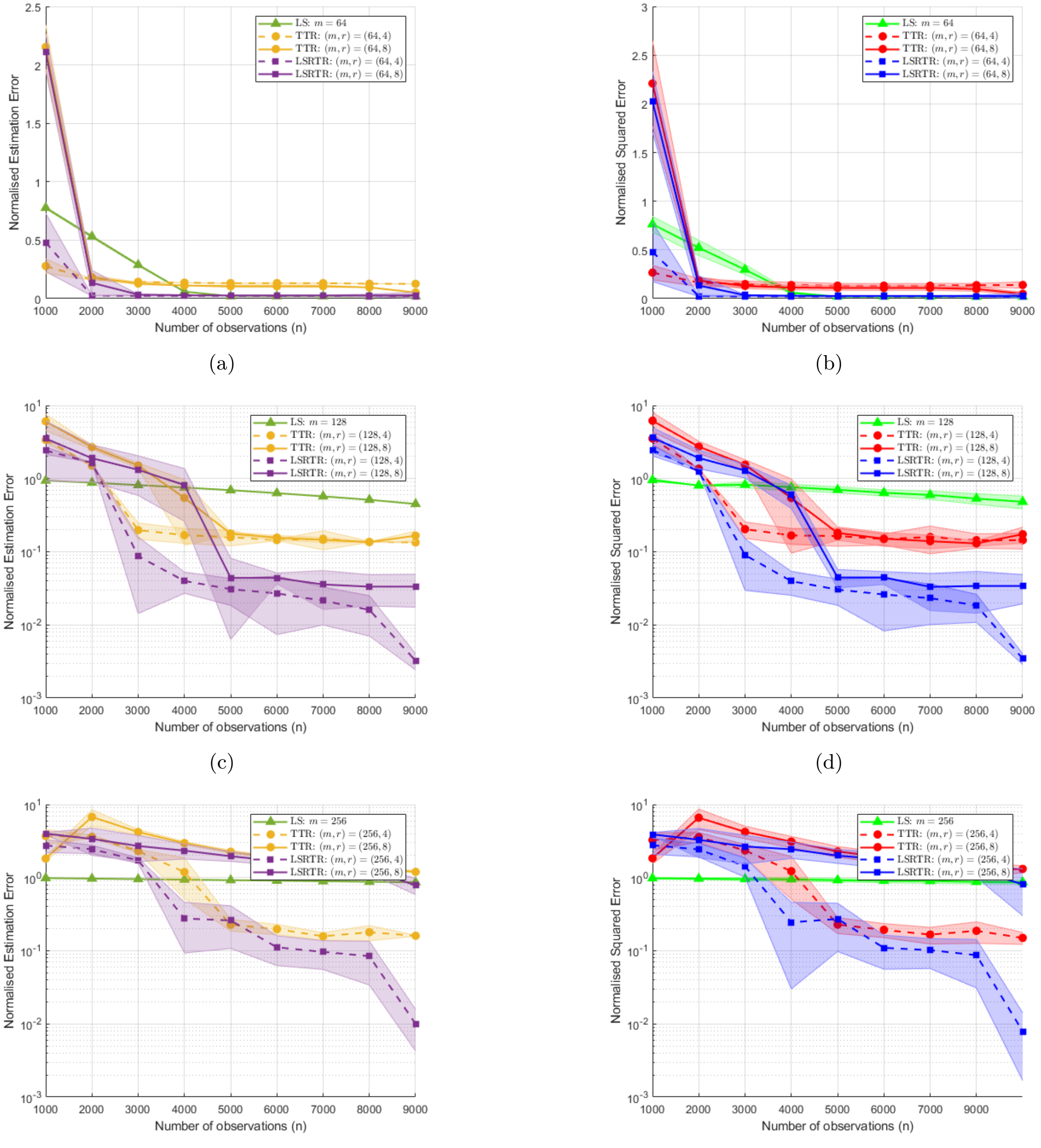
We begin with the two-dimensional (matrix) case with $K = 2$ as many medical imaging data fall under this case (x-rays, EEG, fiber-bundle images.). In this set of experiments the underlying coefficient matrix $\mathbf{B}$ has dimensions $m \times m$ and rank $r$. The dimensions and rank of $\mathbf{B}$ are thus represented as the tuple $(m, r)$. For all GLM problems we fix the separation rank $S = 2$; thus the number of learnable parameters is $S(mr + mr) + r^2 = 4mr + r^2$. We now proceed with the specific experimental setups of the three GLM problems under study.

For linear regression we consider various model sizes and ranks, i.e., $m \in \{64, 128, 256\}$, $r \in \{4, 8\}$. We range sample size $n_{tr}$ from (roughly) the degrees of freedom (number of learnable parameters) of the smallest model size to no smaller than the degrees of freedom of the largest model size. The smallest model size is for the tuple $(m = 64,\ r = 4)$ and the largest model size is for the tuple $(m = 256,\ r = 8)$, which, under the LSR structure, have $4(64 \times 4) + 4^2 = 1040$ and $4(256 \times 8) + 8^2 = 8256$ learnable parameters, respectively. We compare the performance of LSRTR to TTR and LS. Figure 2 reports the mean normalised estimation and prediction accuracy, respectively, across 50 repetitions. The shaded regions are one standard deviation of mean estimation and prediction accuracy, based on 50 replications.

For logistic and Poisson regression we consider $m \in \{32, 64, 128\}$ and $r \in \{4, 8\}$. Since the models are non-linear, we follow the heuristic rule of ranging sample size $n$ from (roughly) 5 times the degrees of freedom of the smallest model size to no smaller than that of the largest model size (Li et al., 2018). The smallest model size is for the tuple $(m = 32,\ r = 4)$ and the largest model size is for the tuple $(m = 128,\ r = 8)$, which, under the LSR structure, have $4(32 \times 4) + 4^2 = 528$ and $4(128 \times 8) + 8^2 = 4160$ learnable parameters, respectively. We compare the performance of the LSRTR method to *i)* LTuR and LR and *ii)* TTR and PR for logistic and Poisson regression, respectively. Figure 3 and Figure 4 report the estimation and prediction accuracy for logistic and Poisson regression, respectively, with shaded regions depicting one standard deviation of mean accuracy, based on 50 replications.

We pause to make a few remarks. Firstly, in the case of synthetic data, the rank $r$ and separation rank $S$ are assumed to be known and other algorithmic parameters (the step size $\alpha$) are set using separate validation experiments. Secondly, though the generated coefficient $\mathbf{B}$ is a matrix, the numerical study by Li et al., (Li et al., 2018), shows that the low-rank Tucker model effectively estimates two-dimensional parameters and the TTR algorithm performs well in the matrix setting even with huge increases in model and sample sizes. The study by Li et al. also shows how TTR generalises CP tensor regression algorithms and improves upon the performance of several other methods (such as PCA and Bayesian regression methods) in terms of estimation and prediction accuracy in experiments with synthetic data and for several regression types (linear and logistic) (Li et al., 2018). For these reasons we limit our scope of comparative algorithms for matrix-structured regression to TTR.

First we note that since we report normalised errors, we do not investigate any errors greater than 1 in Figures 2-4. The plots in Figures 2-4 show that, for all algorithms in all GLM problems, the estimation and prediction accuracy improves and the shaded regions (which characterise the standard deviation of the error points over 50 repetitions) decay with an increase in observations. In particular, on the whole, the tensor-based methods LSRTR, TTR and LTuT outperform the vector-based methods LR, LS and PR, particularly with larger model size $m$. This is because the number of observations provided in the experiments is typically much lower than the sample complexity of the vector methods. For example, consider linear regression with $m = 256$. LS requires at least $256 \times 256 = 65,536$ observations for reliable parameter estimation, while if we set $r = 4$ and $S = 2$, TTR and LSRTR require only 2064 and 4112 observations for reliable parameter estimation, respectively. Additionally, for TTR, LTuR and LSRTR, as the core tensor dimensions grow (i.e., as we increase the Tucker rank of our tensor models), we require a relatively larger sample size to achieve better accuracy. This is not surprising as a larger core tensor increases the number of parameters to be learned. Figures 2-4 also show that for all GLM problems under study (namely linear, logistic and Poisson regression), the LSRTR algorithm shows ‘consistent’ performance for estimation and prediction. What we mean by this, is that, even with a substantial growth in model size, the the estimation error induced by LSRTR decreases with increasing sample size, and the LSRTR algorithm can still achieve acceptable estimation and prediction errors as long as the number of observations provided is large enough.



(a)

(b)

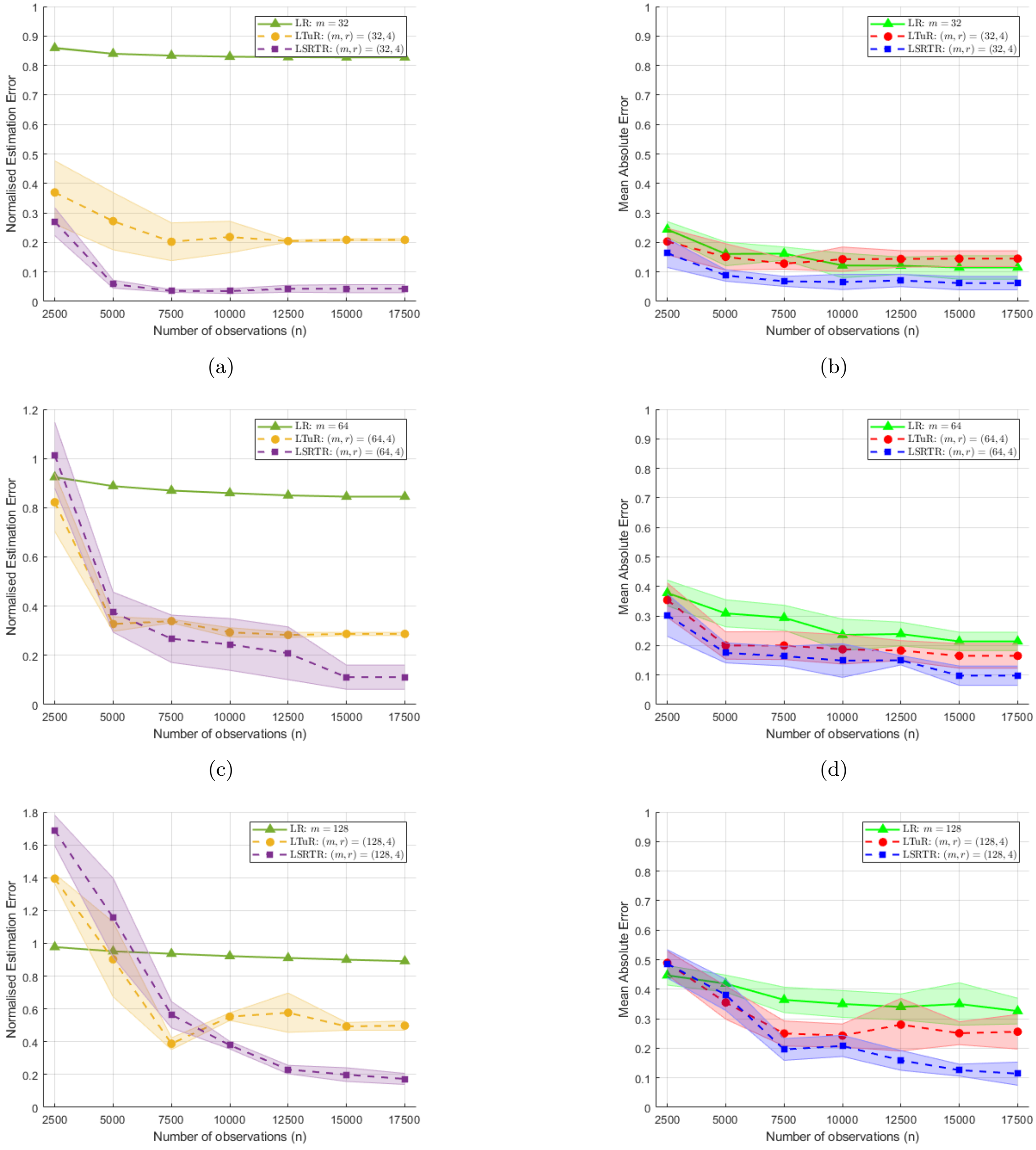
(c)

(d)

(e)

(f)

Figure 2: Comparison of LSRTR with LS and TTR for two-dimensional synthetic data when $m = 64,\ 128,\ 256$, $r = 4, 8$ and $S = 2$. Normalised estimation error for $m = 64,\ 128,\ 256$ is shown in $(a),\ (c),\ (e)$, respectively. Normalised prediction error for $m = 64,\ 128,\ 256$ is shown in $(b),\ (d),\ (f)$, respectively. Each marker represents the mean normalised estimation/prediction errors, over 50 repetitions. The shaded regions correspond to one standard deviation of the mean normalised errors.

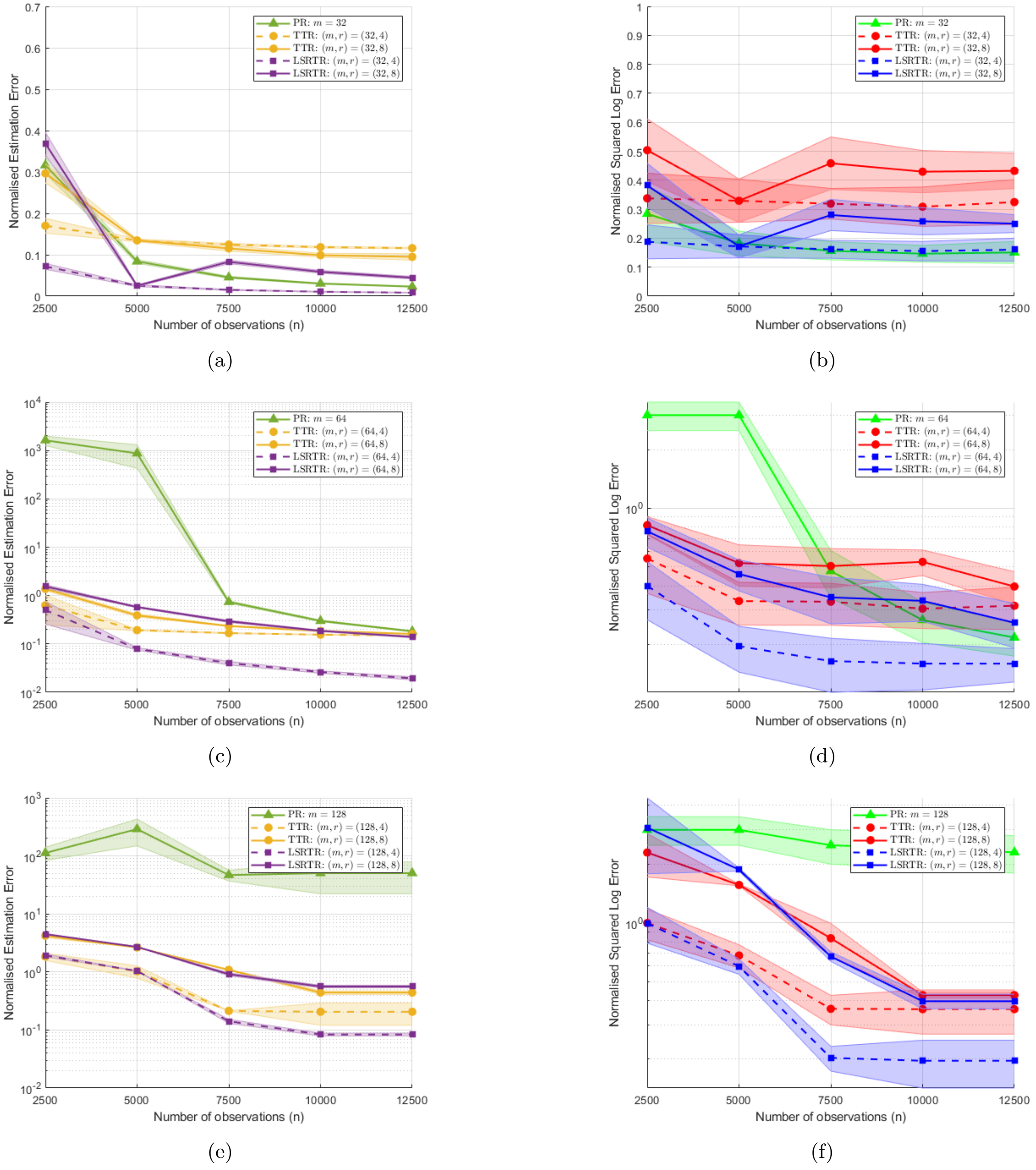


(a) (b)

(c) (d)

(e) (f)

Figure 3: Comparison of LSRTR with LR and LTuR for two-dimensional synthetic data when $m = 32,\ 64,\ 128$, $r = 4$ and $S = 2$. Normalised estimation error for $m = 32,\ 64,\ 128$ is shown in $(a)$, $(c)$, $(e)$, respectively. Normalised prediction error for $m = 32,\ 64,\ 128$ is shown in $(b)$, $(d)$, $(f)$, respectively. Each marker represents the mean normalised estimation/prediction errors, over 50 repetitions. The shaded regions correspond to one standard deviation of the mean normalised errors.



(a) (b)

(c) (d)

(e) (f)

Figure 4: Comparison of LSRTR with PR and TTR for two-dimensional synthetic data when $m = 32,\ 64,\ 128$, $r = 4,\ 8$ and $S = 2$. Normalised estimation error for $m = 32,\ 64,\ 128$ is shown in $(a)$, $(c)$, $(e)$, respectively. Normalised prediction error for $m = 32,\ 64,\ 128$ is shown in $(b)$, $(d)$, $(f)$, respectively. Each marker represents the mean normalised estimation/prediction errors, over 50 repetitions. The shaded regions correspond to one standard deviation of the mean normalised errors.

Additionally, we can see that LSRTR algorithm can predict using new data and can avoid over-fitting even with a very large number of observations.

We also gain some additional insights from the results. 1) If the underlying parameter model $\mathbf{B}$ is LSR structured, then LSRTR can recover $\mathbf{B}$ with a relatively low number of observations, in the sense that the number of observations required is proportional to the intrinsic degrees of freedom in the LSR model (rather than the extrinsic dimensionality of $\mathbf{B}$). This implies that though we do not provide any theoretical guarantees for Algorithm 1, it can be employed successfully in practice. 2) More interestingly, if the underlying parameter model $\mathbf{B}$ is LSR structured, then imposing the Tucker model on $\mathbf{B}$ in the parameter estimation procedure – as in TTR and LTuR algorithms – results in less accurate estimation compared to LSRTR. Particularly, we see that for relatively large model sizes, such as in Figures 2e and 2f, or for non-linear models, such as in Figures 3e, and 4e, TTR and LTuR algorithms' performance plateaus, even when the number of observations has far exceeded the degrees of freedom of a Tucker structured coefficient matrix $\mathbf{B}$. In fact, even LS and LR outperform TTR and LTuR when the model size is small enough and enough observations have been provided. Though we expect LSRTR to perform better than TTR or LTuR when $\mathbf{B}$ is originally LSR structured, the results suggest that if regression problems with real data have model parameters that are approximately LSR structured, then TTR or LTuR may not be suitable algorithms and we can motivate the use of algorithms such at LSRTR that allow for the consideration of a richer class of tensors that may lead to more reliable estimation.

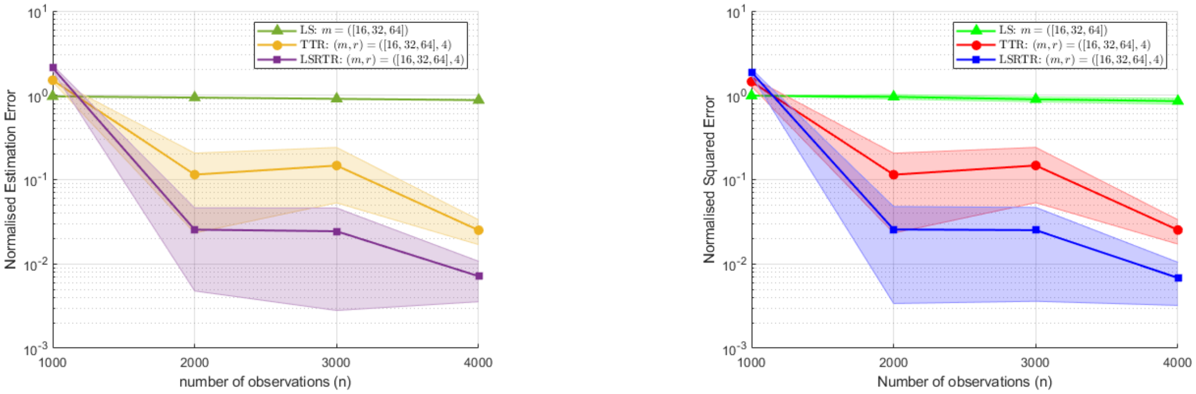
We make specific remarks on Figures 3c and 3e. In the case of $m = 64$, we see that LSRTR shows lower estimation error at $n = 17500$ (0.11 vs 0.29), yet has a larger standard deviation (shaded region) over 50 experiment replications (0.04 vs 0.006). This is an acceptable standard deviation, especially considering the reduction in estimation error compared to the Tucker-based LTuR algorithm. In the case of $m = 128$, LTuR shows lower estimation error than LSRTR until about $n = 7500$ (errors of 0.6 vs 0.4). This could be due to the fact that LSRTR is operating in the undersampling regime for $n = 7500$ (as LSR has more learnable parameters than Tucker). Additionally, we see in Figure 3e that the estimation error for LTuR increases after $n = 7500$, and in Figure 3f that LSRTR outperforms LTuR, which suggests that LTuR may be overfitting the GLM model.

We also make a specific remark on Figure 4a, where we witness a slight increase in estimation error for $(m, r) = (32, 8)$ between $n = 5000$ and $n = 7500$ (0.025 vs 0.08). However, we see that the estimation error continues to decrease after $n = 7500$, and we do not witness this phenomena in any other experiment in Figure 4.

### 5.2 3D Synthetic Data

We have evaluated the performance of the LSRTR algorithm against an increase in model and sample size for $K = 2$. Now, we wish to further explore the performance of our method compared to the other state-of-the-art methods with 3-dimensional data ($K = 3$). In these experiments, the underlying coefficient tensor $\underline{\mathbf{B}}$ has dimensions $m_1 \times m_2 \times m_3$ and Tucker rank $\{r, r, r\}$. The dimensions and rank of $\underline{\mathbf{B}}$ are thus represented as the tuple $(\mathbf{m}, r)$ where $\mathbf{m} = [m_1, m_2, m_3]$. We study the linear regression problem and we fix the separation rank $S = 1$, i.e., $\underline{\mathbf{B}}$ is Tucker structured. Thus $\underline{\mathbf{B}}$ is a 3-dimensional Tucker tensor of dimension $m_1 \times m_2 \times m_3$ and rank $r$ along each mode, and the number of learnable parameters is $S(m_1 r + m_2 r + m_3 r) + r^3 = r(m_1 + m_2 + m_3 + r^2)$. We consider the model size $\mathbf{m} = [16, 32, 64],\ r = 4$. The model size under the Tucker structure is $(16+32+64+4^2)4 = 512$. We range sample size $n_{tr}$ from (roughly) two times the degrees of freedom to roughly eight times the degrees of freedom. We compare the performance of LSRTR to TTR and LS. Figure 5 reports the estimation accuracy and prediction accuracy, respectively, with shaded regions depicting one standard deviation of mean accuracy, based on 50 replications. For LSRTR the rank $r$ is assumed to be known and other parameters (the separation rank $S$ and the step size $\alpha$) are set using separate validation experiments. The separation rank was thus chosen as $S = 2$.

We observe from figure 5 the same observations as in the previous set of experiments. We also conclude that even with the underlying coefficient tensor being Tucker structured our proposed LSRTR method in this experiment exhibits better statistical performance with fewer observations than 'Tucker specific' algorithm TTR as well as LS. This suggests that the LSR model induces a richer class of solutions with greater



(a)

(b)

Figure 5: Comparison of LSRTR with LS and TTR for three-dimensional synthetic data when $\mathbf{m} = [16, 32, 64]$, $r = 4$, $S = 1$. For LSRTR, $S = 2$ was chosen. Normalised estimation error is shown in $(a)$. Normalised prediction error is shown in $(b)$. Each marker represents the mean normalised estimation/prediction errors, over 50 repetitions. The shaded regions correspond to one standard deviation of the mean normalised errors.

| | $\mathbf{n = 100}$ | $\mathbf{n = 300}$ | $\mathbf{n = 500}$ |
|---|---|---|---|
| **LSRTR** | $0.1$ $(7e-5)$ | $0.37$ $(2e-4)$ | $0.65$ $(5e-4)$ |
| **TTR** | $0.048$ $(3e-5)$ | $0.18$ $(1e-4)$ | $0.31$ $(2e-4)$ |

Table 2: Per-iteration computation time (in seconds) of LSRTR and TTR over varying sample size ($n = 100$, $n = 300$, $n = 500$). We report the mean (and variance) over 50 repetitions.

representation power that is able to capture meaningful information in the data that the more compact state-of-the-art tensor models do not – even with increased sample size. The results also suggest that the projection step in LSRTR (imposing orthonormal factor matrices), may contribute to the performance of LSRTR in terms of estimation/prediction error (as TTR does not impose such a constraint).

Finally, we numerically investigate the per-iteration computational complexity of LSRTR vs TTR for three-dimensional synthetic data, for varing sample size, $n$. We repeat the previous experiment, i.e., $\mathbf{m} = [16, 32, 64]$, $r = 4$, $S = 2$, over varying sample size ($n = 100$, $n = 300$, $n = 500$), and report the mean per-iteration computation time (in seconds) over 50 repetitions of the experiment, for LSRTR and TTR. Table 2 shows the results. We can see that LSRTR and TTR have comparable per-iteration computation times. In fact LSRTR has computation time of roughly $S = 2$ times that of TTR.

In Section 6, we derive a lower bound on the minimax risk of the estimation error in the LSR-TGLM parameter estimation problem in (13). We numerically assess the tightness of the derived bound. Furthermore, in Section 7, we evaluate the LSR-TGLM model and the LSRTR algorithm on several medical imaging datasets.

## 6 Minimax Lower Bounds for Tensor-Structured GLMs

In this section we derive lower bounds on the minimax risk of the LSR-TGLM problem. In the results we will derive, the minimax risk depends explicitly on the parameters of the LSR-structured tensor model, namely $\{r_k\}_{k\in[K]}$, $\{m_k\}_{k\in[K]}$, $S$, the distribution of $\underline{\mathbf{X}}$ and the number of samples $n$. Specifically, we derive lower bounds on $\varepsilon^*$ using an argument for estimation problems based on Fano's inequality (Yu, 1997; Taki et al., 2021). This approach relates the minimax risk of coefficient tensor $\underline{\mathbf{B}}$ to a multiple hypothesis testing problem. It states that if there exists an estimator (let us call it $\widehat{\underline{\mathbf{B}}}(\mathbf{y}, \mathcal{X})$, and which can be estimated by any parameter estimation algorithm), with error matching the minimax risk $\varepsilon^*$ (i.e $\left\|\widehat{\underline{\mathbf{B}}}(\mathbf{y}, \mathcal{X}) - \underline{\mathbf{B}}\right\|_F^2 = \varepsilon^*$), then this estimator $\widehat{\underline{\mathbf{B}}}(\mathbf{y}, \mathcal{X})$ can be used to solve a multiple hypothesis testing problem (MHTP) (Khas'minskii, 1979). In this MHTP, the hypothesis set consists of a collection, $\mathcal{B}_L$, of $L$ LSR-structured coefficient tensors, where the goal of the estimator, $\widehat{\underline{\mathbf{B}}}(\mathbf{y}, \mathcal{X})$, is to detect the correct 'generating' LSR-structured tensor. Fano's inequality provides a fundamental limit/bound on the error probability for the multiple hypothesis testing problem. This limit in turn provides a lower bound on the minimax risk $\varepsilon^*$. As mentioned in Section 3, our analysis is local and our approach to deriving a lower bound on the minimax risk is information-theoretic and thus involves the analysis of the mutual information (defined as $\mathbb{I}(\mathbf{y}; l)$) between observations $\mathbf{y} = [y_1 \dots y_n]$ and hypothesis $l \in [L]$;

$$\mathbb{I}(\mathbf{y}; l) \triangleq \mathbb{E}_{\mathbf{y},l}\left[\log \frac{f(\mathbf{y}, l)}{f(\mathbf{y}) f(l)}\right], \tag{36}$$

where $f(\cdot)$, is a probability distribution. We are now ready to state our main result.

**Theorem 6.** *Consider the rank-$(r_1, \cdots, r_K)$ and separation rank $S$ LSR-TGLM problem in* (13) *with $n$ i.i.d observations, $\left\{\underline{\mathbf{X}}_i, y_i\right\}_{i=1}^n$, where $\mathrm{vec}(\underline{\mathbf{X}}_i) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_x)$ and the true coefficient tensor $\|\underline{\mathbf{B}}\|_F^2 < d^2$. The minimax risk $\varepsilon^*$ is then lower bounded by*

$$\varepsilon^* \geq \frac{\left(\frac{1}{16}\right) S \sum_{k\in[K]}(m_k - 1)r_k + \prod_{k\in[K]}(r_k - 1) - 1}{128M \left\|\Sigma_x\right\|_2 n}. \tag{37}$$

Theorem 6 can be specialised to the Tucker and CP decomposition regression problems in the existing literature. That is, when $S = 1$ we have the Tucker model, and when $r_1 = r_2 = \cdots = r_K = r$ and $\underline{\mathbf{G}}$ is constructed as a diagonal tensor, we have the CP model. With these specializations we obtain the following corollaries:

**Corollary 1.** *Consider the rank-$(r_1, \cdots, r_K)$ Tucker-TGLM problem (Li et al., 2018; Zhang et al., 2020; Zhang & Jiang, 2016) with $n$ i.i.d observations, $\left\{\underline{\mathbf{X}}_i, y_i\right\}_{i=1}^n$, where $\mathrm{vec}(\underline{\mathbf{X}}_i) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_x)$ and the true coefficient tensor $\|\underline{\mathbf{B}}\|_F^2 < d^2$. The minimax risk is lower bounded by*

$$\varepsilon^* \geq \frac{\left(\frac{1}{16}\right) \sum_{k\in[K]}(m_k - 1)r_k + \prod_{k\in[K]}(r_k - 1) - 1}{128M \left\|\Sigma_x\right\|_2 n}. \tag{38}$$

**Corollary 2.** *Consider the rank-$r$ CP-TGLM problem (Zhou et al., 2013; Tan et al., 2012) with $n$ i.i.d observations, $\left\{\underline{\mathbf{X}}_i, y_i\right\}_{i=1}^n$, where $\mathrm{vec}(\underline{\mathbf{X}}_i) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_x)$ and the true coefficient tensor $\|\underline{\mathbf{B}}\|_F^2 < d^2$. The minimax risk is lower bounded by*

$$\varepsilon^* \geq \frac{\left(\frac{1}{16}\right) \sum_{k\in[K]}(m_k - 1)r + (r - 1) - 1}{128M \left\|\Sigma_x\right\|_2 n}, \tag{39}$$

Table 3 provides a summary of the minimax lower bounds from this work and the relevant current literature. Specifically Table 3 shows the order-wise lower bounds on the minimax risk for logistic regression, linear regression and GLMs for several tensor structures on the model parameter. We make three remarks prior to our discussion of Table 3. First, we define the terms $\widetilde{m} = \prod_{k\in[K]} m)k$ and $\widetilde{r} = \prod_{k\in[K]} r_k$. Additionally, $\sigma_y^2$ is the variance of observation $y$, $L$ is a fixed constant and an upper bound on the second derivative of the cumulant function $(a^{''}(\eta))$, and we place a dash $(-)$ in the cells where there are no specific results in the existing literature. However, note that our derived minimax bounds for tensor-structured GLMs cover these gaps in the literature (namely the case of CP-structred linear and logistic regression, and Tucker-structured Logistic regression). Secondly, the order-wise lower bounds reported for the unstructured case are in fact bounds on the minimax risk for prediction error, rather than estimation error. Nonetheless we include them in our comparison as the sample complexities of estimation and prediction in a given problem tend to be proportional. Thirdly, here we have we reported lower bound on the minimax risk for non-sparse regression, as we are not studying sparsity in the scope of this work. In the first two rows of the table, we report the

| | **Structure of $\underline{\mathbf{B}}$** | | | |
|---|---|---|---|---|
| **Regression** | **Unstructured** | **CP** | **Tucker** | **LSR** |
| **Linear** | $\frac{\sigma_y^2 \widetilde{m}}{n}$ (Raskutti et al., 2011) | – | $\frac{\sigma_y^2 \left( \sum_{k \in [K]} m_k r_k - r_k^2 + \widetilde{r} \right)}{n}$ (Zhang et al., 2020) | – |
| **Logistic** | $\frac{\widetilde{m}}{n}$ (Abramovich & Grinshtein, 2016) | – | – | – |
| **GLM** | $\frac{\sigma_y^2 \widetilde{m}}{Ln}$ (Lee & Courtade, 2020) | $\frac{\sum_{k \in [K]} m_k r + r}{M \|\Sigma_x\|_2 n}$ | $\frac{\sum_{k \in [K]} m_k r_k + \widetilde{r}}{M \|\Sigma_x\|_2 n}$ | $\frac{\left( S \sum_{k \in [K]} m_k r_k \right) + \widetilde{r}}{M \|\Sigma_x\|_2 n}$ |

Table 3: Summary of order-wise lower bounds on the minimax risk for linear regression, logistic regression and GLM settings and for various tensor structures (unstructured, CP, Tucker and LSR).

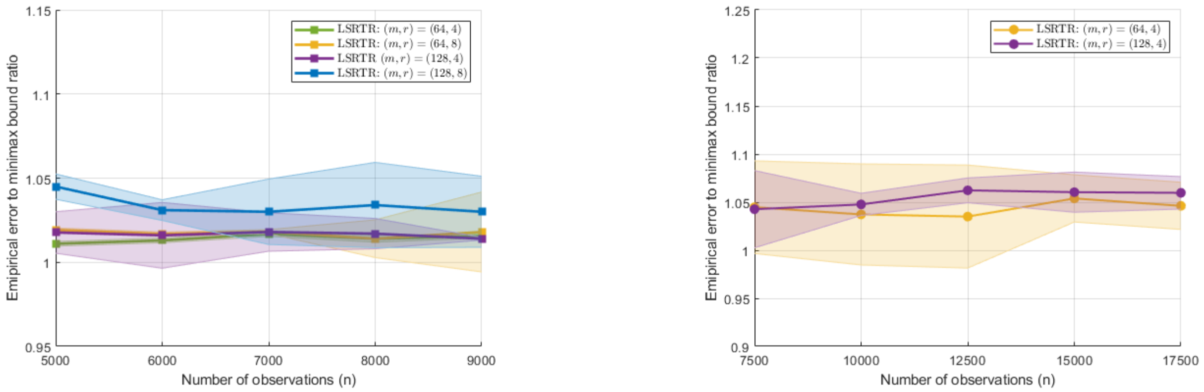
order-wise minimax lower bounds for linear and logistic regression. In all cells of the first two rows, we see that the minimax risk decreases proportionally with increasing sample size. In the unstructured case, the minimax risk is proportional to the number of learnable parameters in the model ($\prod_k m_k$). For linear regression, however, this only applies if the number of parameters is less than $n$. This condition puts this result at a disadvantage for the high-dimensional setting under study. We can see that the minimax risk of linear regression in the Tucker-structured tensor case is proportional to the number of parameters in the Tucker model ($\sum_k m_k r_k + \prod_k r_k$). The numerators ($\prod_k m_k$) and ($\sum_k m_k r_k + \prod_k r_k$) give insights into the parameters on which an achievable minimax risk might depend, and thus give insights into the sample complexity of the parameter estimation problem. We can see that imposing a low-rank Tucker structure may significantly decrease the number of learnable parameters.

The third row is the row of interest as it summarizes results for GLMs. Similar to the previous rows, the minimax risk decreases with an increase in sample size. In the unstructured case the sample complexity is ($\prod_k m_k$). We also have a dependence on $\sigma_y^2$ and $\frac{1}{L}$. Intuitively speaking, this means that the minimax risk increases with the variance of $y$. Theorem 6 and Corrolaries 1 and 2 make up the last three columns. The minimax risk is proportional to the number of parameters for the CP, Tucker and LSR case. This is intuitively pleasing as it suggests that imposing either the CP, Tucker or LSR structure on $\underline{\mathbf{B}}$, one can significantly reduce the sample complexity from the unstructured case to $\sum_k m_k r + r$, $\sum_k m_k r_k + \prod_k r_k$, and $S \sum_k m_k r_k \prod_k r_k$, respectively, and that we can develop algorithms (such as LSRTR in Algorithm 1) that meet these bounds. As previously mentioned, the CP model induces the lowest number of parameters and the LSR model has $(S-1) \sum_k m_k r_k$ more parameters than the Tucker model. Additionally, our results also imply an inverse relationship between the minimax risk and $\|\mathbf{\Sigma}_x\|_2$ and $M$, which is due to an artifact of our analysis.

Lastly, though we do not provide any theoretical guarantees for the tightness of our bounds in Theorem 6 and Corollaries 1 and 2, we can see that our minimax bound for the Tucker-GLM meets that of the Tucker linear regression works by Zhang et al., (Zhang et al., 2020), (we do not consider the $r_k^2$ term in (Zhang et al., 2020) as that only accounts for the non-singular transformation indeterminacy, which we do not discuss in the scope of this work). Zhang et al. have shown that their bounds are indeed optimal, which suggests that the minimax lower bounds derived in Theorem 6 and Corrolaries 1 and 2 are also tight. Moreover, we now provide a numerical analysis to assess the tightness of our bounds.

## 6.1 Tightness of Theorem 6

We utilise the results of our experiments on synthetic data in Section 5 to investigate the tightness of the minimax lower bound in (6). Though we do not provide any theoretical guarantees for the tightness of our result, we have shown that our result matches an existing bound in the literature for the specific case of Tucker linear regression. Now, we perform an empirical assessment of our result. Figure 6 shows the ratio of the mean empirical error (over 50 repetitions) of LSRTR from our experiments in Section 5.1.1 to our obtained lower bound in (6). We do this for linear and logistic regression and we can see the ratio is approximately constant as a function of sample size for both regression types. Such a result suggests that our bound may be achievable in the sense that we can develop algorithms that meet the minimax lower bound and take advantage of the LSR tensor structure to lower the sample complexity of GLM problems.



(a)

(b)

Figure 6: Comparison of the empirical error in LSRTR to the minimax lower bound for linear and logistic regression. $(a)$ shows the ratio of the empirical error in linear regression to the minimax lower bound for $m = 64, 128$ and $r = 4, 8$. $(b)$ shows the ratio of the empirical error in logistic regression to the minimax bound for $m = 64, 128$ and $r = 4$.

## 6.2 Roadmap for Proof of Theorem 6

We begin by introducing a few important concepts. First, in order to set up the multiple hypothesis testing problem consider the constructed packing set

$$\mathcal{B}_L = \{\underline{\mathbf{B}}_l \in \mathcal{P}_{\{r_k\},S} : l \in [L]\} \subset \mathcal{B}_d(\underline{\mathbf{0}}),\ \ L \in \mathbb{N} \tag{40}$$

from which the true index $l$ corresponding to the true coefficient tensor is generated uniformly at random. In order to ensure a tight bound on our minimax risk, $\mathcal{B}_L$ must possess following properties: $(i)$ The minimum packing distance between any two distinct hypotheses $\underline{\mathbf{B}}_l, \underline{\mathbf{B}}_{l'} \in \mathcal{B}_L$, $l, l' \in [L]$, $l \neq l'$ is large. Specifically, for a strictly positive parameter $\delta$ we require a construction such that $\|\underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}\|_F \geq \sqrt{8\delta}$. $(ii)$ The hypothesis testing problem is hard in the sense that it is difficult to detect the true index $l$ (and thus the true coefficient tensor) based on an observation $y$. $(iii)$ The construction of $\mathcal{B}_L$ must induce lower bounds on the minimax risk that reflects the reduction in the sample complexity of the LSR tensor structure, i.e., we require a bound that exhibits a relation between the intrinsic degrees of freedom of the LSR tensor $(\{m_k\}_{k\in[K]}, \{r_k\}_{k\in[K]}, S)$ and the number of samples $(n)$. Secondly, since the objective of the multiple hypothesis testing problem is to detect the index $l$, it is solved through the minimum distance decoder

$$\widehat{l}(\mathbf{y}) \triangleq \underset{l'\in[L]]}{\arg\min} \left\| \widehat{\underline{\mathbf{B}}} - \underline{\mathbf{B}}_{l'} \right\|_F , \tag{41}$$

where $\widehat{\underline{\mathbf{B}}}$ is estimated through any learning algorithm, such as Algorithm 1 proposed in this work. In the LSR-TGLM problem we have the following hypothesis detection criteria:

- If $\left\|\widehat{\underline{\mathbf{B}}} - \underline{\mathbf{B}}_l\right\|_F \leq \sqrt{2\delta}$: $\mathbb{P}(\widehat{l}(\mathbf{y}) \neq l) = 0$.
- If $\left\|\widehat{\underline{\mathbf{B}}} - \underline{\mathbf{B}}_l\right\|_F > \sqrt{2\delta}$: A detection error **might** occur, $\mathbb{P}(\widehat{l}(\mathbf{y}) \neq l) \geq 0$.

The minimum distance decoder detects the true hypothesis if the estimate $\widehat{\underline{\mathbf{B}}}$ lies within the open ball $\mathcal{B}(\underline{\mathbf{B}}_l, \sqrt{2\delta})$, and a detection error can only occur if $\left\|\widehat{\underline{\mathbf{B}}} - \underline{\mathbf{B}}_l\right\|_F > \sqrt{2\delta}$. Thus the probability of error is bounded by

$$\mathbb{P}(\widehat{l}(\mathbf{y}) \neq l) \leq \mathbb{P}\Big(\left\|\widehat{\underline{\mathbf{B}}} - \underline{\mathbf{B}}_l\right\|_F \geq \sqrt{2\delta}\Big). \tag{42}$$

In this approach we interpret the coefficient estimation problem as a communication problem. From the set in (40), the source selects the true hypothesis $\underline{\mathbf{B}}_l$ (the true coefficient tensor) uniformly at random. The hypothesis $\underline{\mathbf{B}}_l$ then generates the channel output according to a 'channel model' as in (13) with a chosen link function $g(\cdot)$. Specifically, the channel outputs $y_i$ such that $\mathbb{E}[y_i|\underline{\mathbf{X}}_i] = \mu_i = g^{-1}(\langle \underline{\mathbf{B}}_l, \underline{\mathbf{X}}_i \rangle)$. The minimum distance decoder then successfully recovers the true hypothesis if the estimator $\widehat{\underline{\mathbf{B}}}$ is within a ball of radius $\sqrt{2\delta}$ around $\underline{\mathbf{B}}_l$. Thirdly, we recognise that the problem under study is supervised and the response variables $y_i \ \forall i \in [n]$ are conditioned on input data $\underline{\mathbf{X}}_i \ \forall i \in [n]$. On the basis thereof we opt to evaluate the conditional mutual information $\mathbb{I}(\mathbf{y}; l|\mathbf{X}) \geq \mathbb{I}(\mathbf{y}; l)$. We use the aforementioned concepts to achieve our result in (37)

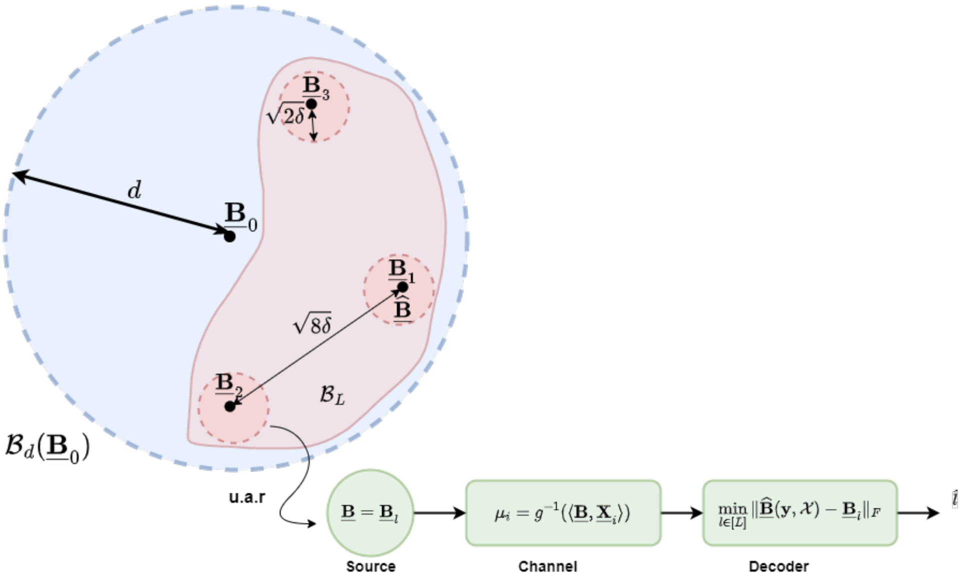

$\underline{\mathbf{B}}_3$
$\sqrt{2\delta}$
$d$
$\underline{\mathbf{B}}_0$
$\underline{\mathbf{B}}_1$
$\widehat{\underline{\mathbf{B}}}$
$\sqrt{8\delta}$
$\underline{\mathbf{B}}_2$
$\mathcal{B}_L$
$\mathcal{B}_d(\underline{\mathbf{B}}_0)$
u.a.r
$\underline{\mathbf{B}} = \underline{\mathbf{B}}_l$
Source
$\mu_i = g^{-1}(\langle \underline{\mathbf{B}}, \underline{\mathbf{X}}_i \rangle)$
Channel
$\min_{l \in [L]} \|\widehat{\underline{\mathbf{B}}}(\mathbf{y}, \mathcal{X}) - \underline{\mathbf{B}}_i\|_F$
Decoder
$\widehat{l}$

Figure 7: Information theoretic approach for deriving bounds on the minimax risk. Consider a packing set $\mathcal{B}_L = \{\underline{\mathbf{B}}_l : l \in [L]\} \subset \mathcal{B}_d(\underline{\mathbf{0}})$ with minimum distance between any two elements as $\sqrt{8\delta}$. The true coefficient tensor is selected uniformly at random (u.a.r) and the LSR-TGLM model generates the channel output. We detect the true coefficient tensor using the minimum distance decoder. Here, the minimum distance decoder will detect $\underline{\mathbf{B}}_1$ as the true hypothesis

.

through the accomplishment of the following tasks: 1) We must construct the set $\mathcal{B}_L$ as an exponentially large (with respect to the dimensions of an LSR structured tensor) family of $L$ distinct LSR structured tensors, and satisfying the properties described above. This involves constructing $KS+1$ individual sets (for the $K \times S$ factor matrices $\{\mathbf{B}_{(k,s)}\}_{k\in[k],\ s\in[S]}$ and core tensor $\underline{\mathbf{G}}$) and deriving conditions under which all sets can exists simultaneously (with high probability). 2) We must find tight bounds on the conditional mutual information $\mathbb{I}(\mathbf{y}; l|\mathcal{X})$. As explained in Section 3, 'Off-the-shelf' packing sets and bounds on the minimax risk for the unstructured (vector-based) parameter estimation problem are not useful in our application as our goal is to understand the benefits of imposing the LSR structure on the problem's sample complexity.

The novelty in our work is that we explicitly leverage this structure leading to a hypothesis set $\mathcal{B}_L$ with a structure than cannot be achieved through current methods. Lastly, though our analysis is local and depends on a fixed tensor in a neighbourhood with known radius, our minimax risk becomes independent of this radius if it becomes sufficiently large.

### 6.3 Proof of Theorem, 6

Four lemmas lay the foundation to the formal proof of Theorem 6. Through Lemmas 4, 6, 7 and 8 we achieve the following: We construct the hypothesis set $\mathcal{B}_L$ containing $L$ distinct tensors. Since each $\underline{\mathbf{B}}_l \in \mathcal{B}_L$ is LSR structured, as defined in (9), then we construct $\mathcal{B}_L$ by constructing $KS+1$ individual sets that correspond to $K \times S$ sets for $\mathbf{B}_{(k,s)}$ ($k \in [K], s \in [S]$) and a set for $\underline{\mathbf{G}}$. Our construction of $\mathcal{B}_L$ results in tight upper and lower bounds on the distance $\|\underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}\|_F^2$ for any two distinct $l, l' \in [L]$, sampled uniformly at random (u.a.r), which will be used to derive bounds on the mutual information $\mathbb{I}(\mathbf{y}; l)$ and minimax risk $\varepsilon^*$. Complete proofs of these lemmas are in the appendix. Lemma 4 (and Corollary 5) introduce the sets of 'generating' vectors (and matrices) from which we will later construct $\underline{\mathbf{G}}$ (and $\mathbf{B}_{(k,s)}$ for all $k \in [k]$, $s \in [S]$), that are in turn used to construct each $\underline{\mathbf{B}}_l$. Specifically, for some $\alpha \in \mathbb{N}$, our aim is to construct a set of ($\alpha$)-dimensional vectors with entries sampled uniformly at random from $\left\{\frac{-1}{\sqrt{\alpha}}, \frac{1}{\sqrt{\alpha}}\right\}$, with some designated minimum packing distance in the Hamming metric between any two vectors. Topologically speaking, this can also be viewed as the construction of a subset of an ($\alpha$)-dimensional hypercube with a required minimum Hamming distance between any two distinct elements in the set. Lemma 4 and its subsequent corollary utilise the Gilbert-Vershamov bound on constructing binary codes with minimum distance in order to construct the 'generating' sets we have just described. The following can be found in the book by Tsybakov (Tsybakov, 2009, Lemma 2.9), where we restate the bound here in the interest of keeping this work self-contained.

**Lemma 3** (Gilbert-Vershamov Bound, Lemma 2.9 (Tsybakov, 2009))**.** *Let $d \geq 8$. Then there exists a subset $\mathcal{C} \subset \{0,1\}^d$ of size*

$$|\mathcal{C}| \geq 2^{d/8} \tag{43}$$

*such that for any pair $\mathbf{v}, \mathbf{v}' \in \mathcal{C}$, we have $\|\mathbf{v} - \mathbf{v}'\|_1 \geq \frac{d}{8}$.*

**Corollary 3.** *For any $d \geq 8$ there exists a set of binary vectors $\mathcal{C} \subseteq \{-\alpha, \alpha\}^d$ of size $|\mathcal{C}| \geq 2^{d/8}$ such that for any pair $\mathbf{v}, \mathbf{v}' \in \mathcal{C}$, we have $\|\mathbf{v} - \mathbf{v}'\|_0 \geq \frac{d}{8}$.*

Lemma 4 and Lemma 5 introduce a set of 'generating' binary vectors and matrices, respectively, with minimum distance.

**Lemma 4.** *Let $r_k > 0$ $\forall k \in [K]$ and $\widetilde{r} = \prod_{k \in [K]} r_k$. For any integers $(r_1, r_2, \cdots, r_k)$ such that $\widetilde{r} \geq 8$, there exists a collection of $F \geq 2^{(\widetilde{r}-1)/8}$ vectors $\{\mathbf{s}_f \in \frac{1}{\sqrt{\widetilde{r}-1}}\{-1,1\}^{\widetilde{r}-1} \colon f \in [F]\}$ such that for all $f, f' \in [F]$ we have $\|\mathbf{s}_f - \mathbf{s}_{f'}\|_0 \geq (\widetilde{r}-1)/8$.*

**Lemma 5.** *Let $m_k > 0$ and $r_k > 0$ $\forall k \in [K]$. For any integers $(m_k, r_k)$ such that $(m_k - 1)r_k \geq 8$ there exists a collection of $P = 2^{(m_k-1)r_k/8}$ matrices $\{\mathbf{S}_p \in \frac{1}{\sqrt{(m_k-1)r_k}}\{-1,1\}^{(m_k-1)r_k} \colon$*
*$p \in [P]\}$ such that for all $p, p' \in [P]$ we have $\|\mathbf{S}_p - \mathbf{S}_{p'}\|_0 \geq (m_k - 1)r_k/8$.*

The sets generated in Lemma Lemma 4 and Lemma 5 can be used to construct the set $\mathcal{B}_L$ of $L$ tensors in (40). We now introduce Lemma 6, which derives conditions on $L$ such that the sets in Lemma 4 and Lemma 5 exist simultaneously. In Lemma 6 we construct $\mathcal{B}_L$ with a certain set of properties. Note that every element in $\mathcal{B}_L$ must have an LSR structure. That is, every element is comprised of $K \times S$ low-rank and orthogonal factor matrices (of dimensions $m_k \times r_k$, $k \in [K]$), and a core tensor (of dimensions $r_1 \times \cdots \times r_K$), that we carefully construct from the set of 'generating' matrices and vectors defined in Lemma 5 and Lemma 4, respectively, according to (9). Additionally, any two distinct elements $\underline{\mathbf{B}}_l, \underline{\mathbf{B}}_{l'} \in \mathcal{B}_L$ must be a suitable distance apart. Lemma 6 derives upper and lower bounds on this distance; the lower bound determines the minimum packing distance, while the upper bound is used to derive upper bound on the conditional mutual information in Lemma 7.

**Lemma 6.** *Define $\widetilde{r} = \prod_{k \in [K]} r_k$. Let $S > 0$, $m_k > 0$, $r_k > 0$, and $(m_k - 1)r_k \geq 8$ $\forall k \in [K]$ There exists a collection of $L$ tensors $\mathcal{B}_L \triangleq \{\underline{\mathbf{B}}_l : l \in [L]\} \subset \mathcal{B}_d(\underline{\mathbf{0}})$ for some $d > 0$ of cardinality*

$$L \geq 2^{\frac{1}{8}\left[S \sum_{k \in [K]} (m_k - 1) r_k + (\widetilde{r} - 1)\right]} \tag{44}$$

*such that for any*

$$\frac{1}{S}\sqrt{\frac{32(\widetilde{r} - 1)}{\widetilde{r}}} < \varepsilon \leq \frac{d}{S}\sqrt{\frac{\widetilde{r} - 1}{\widetilde{r}}}, \tag{45}$$

*we have*

$$S^2 \frac{\widetilde{r}}{\widetilde{r} - 1}\varepsilon^2 \leq \|\underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}\|_F^2 \leq 4S^2 \frac{\widetilde{r}}{\widetilde{r} - 1}\varepsilon^2. \tag{46}$$

The bounds in (46) match (up to a constant) and are tight, which help ensure that the upper bound on the conditional mutual information is also tight. Until now, we have completed the tasks of constructing the set $\mathcal{B}_L$ with the packing distance $8\delta = \frac{S^2}{4}\frac{\widetilde{r}}{\widetilde{r}-1}\varepsilon^2$ and with packing number (cardinality) parametrised by the parameters $\{r_k\}_{k \in [K]}, \{m_k\}_{k \in [K]}, S$. We transition to the next task of deriving upper bounds on the conditional mutual information, $\mathbb{I}(\mathbf{y}; \mathcal{X}|)$. Upper bounds on $\mathbb{I}(\mathbf{y}; \mathcal{X}|)$ involve the evaluation of the Kullback-Leibler (KL) divergence. For this we employ well-established results in the literature (Cover & Thomas, 2012; Jung et al., 2016). Specifically, define $f_l(\mathbf{y}|\mathcal{X})$ and $f_{l'}(\mathbf{y}|\mathcal{X})$ as the conditional probability distribution of $\mathbf{y}$ given $\mathcal{X}$ with any two distinct coefficient tensors $\underline{\mathbf{B}}_l$ and $\underline{\mathbf{B}}_{l'}$ respectively. Denote $D_{KL}$ as the relative entropy, then

$$\mathbb{I}(\mathbf{y}; l|\mathcal{X}) \leq \frac{1}{L^2}\sum_{l,l'} \mathbb{E}_{\mathcal{X}} D_{KL}(f_l(\mathbf{y}|\mathcal{X}) || f_{l'}(\mathbf{y}|\mathcal{X})). \tag{47}$$

We note here that we require the hypothesis test to be hard, which corresponds to a relatively small KL divergence. Therefore, our aim is to derive tight upper bounds on the KL divergence in (47), by relying on the tight bounds in (46). Lemma 7 thus derives an upper bound on $\mathbb{I}(\mathbf{y}; l|\underline{\mathbf{X}}^c)$.

**Lemma 7.** *Consider the LSR-TGLM problem given by the model in* (13) *such that $\underline{\mathbf{B}} \in \mathcal{B}_d(\underline{\mathbf{0}})$ for some $d > 0$, and consider the set $\mathcal{B}_L$ constructed in Lemma 6. Consider $n$ i.i.d observations $y_i$ following a probability distribution belonging to the exponential family when conditioned on $vec(\underline{\mathbf{X}}_i) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_x)$. Then we have:*

$$\mathbb{I}(\mathbf{y}; l|\mathcal{X}) \leq 4S^2 Mn \|\mathbf{\Sigma}_x\|_2 \frac{\widetilde{r}}{\widetilde{r} - 1}\varepsilon^2. \tag{48}$$

The final step is to lower bound $\mathbb{I}(\mathbf{y}; l|\mathcal{X})$ using Fano's inequality (Yu, 1997), stated below:

$$\mathbb{I}(\mathbf{y}; l|\mathcal{X}) \geq \mathbb{I}(\mathbf{y}; l) \geq \left(1 - \mathbb{P}(\widehat{l}(\mathbf{y}) \neq l)\right)\log_2(L) - 1. \tag{49}$$

Evaluating (49) is simple. For this we refer the reader back to (42) in Section 6.2, which bounds the error probability $\mathbb{P}(\widehat{l}(\mathbf{y}) \neq l)$ for the recovery of hypothesis $\underline{\mathbf{B}}_l$ using the minimum distance decoder. Lemma 8 bounds this probability, with proof that follows exactly that for Lemma 8 in (Jung et al., 2016), and thus is omitted in this work.

**Lemma 8.** *Consider the minimum distance decoder in* (41). *Consider also the LSR-TGLM regression problem in* (13) *with minimax risk $\varepsilon^*$. Suppose $\varepsilon^* \leq \delta$ for some non-negative scalar $\delta$. If the $L$ tensors destributed in Lemma 6 satisfy $\|\underline{\mathbf{B}} - \underline{\mathbf{B}}_{l'}\| \geq 8\delta$ then the detection error of the minimum distance decoder is upper bounded by $\mathbb{P}(\widehat{l}(\mathbf{y}) \neq l) \leq \frac{1}{2}$.*

*Proof of Theorem 6.* Fix $(r_1, r_2, \ldots, r_K)$. From Lemma 6 for any $\varepsilon > 0$ satisfying (45) there exists a packing set $\mathcal{B}_L \subset \mathcal{B}_d(\underline{\mathbf{0}})$ with cardinality $L$ and packing distance $\|\underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}\|_F^2$ satisfying (44) and (46), respectively.

Also, from Lemma 7, with the set $\mathcal{B}_L$, $\mathbb{I}(\mathbf{y}; l|\mathcal{X})$ satisfies (48). Now suppose there exists an estimator $\underline{\widehat{\mathbf{B}}}$ which guarantees a risk $\varepsilon^* = \frac{S^2 r}{32(r-1)}\varepsilon^2$. If we set $8\delta = \frac{S^2}{4}\frac{\widetilde{r}}{\widetilde{r-1}}\varepsilon^2$ then from Lemma 8 for the set $\mathcal{B}_L$ and we have $\mathbb{P}(\widehat{l}(\mathbf{y}) \neq l) \leq \frac{1}{2}$. Combining Lemmas 7 and 8 we have

$$\frac{1}{2}\log_2 L - 1 \leq \mathbb{I}(\mathbf{y}; l|\mathcal{X}) \leq 128Mn\left\|\mathbf{\Sigma}_x\right\|_2 \varepsilon^*, \tag{50}$$

which can be rearranged to achieve the result in (6). □

## 7 Experiments on Medical Imaging Data

We move on to investigating our approach on medical imaging data. Regression analysis of medical imaging data can be a useful tool in medical decision making. We chose this type of data for several reasons. Medical images are usually multi-dimensional, and since data acquisition in medical sciences is expensive, the regression problems under study with such data is very high dimensional. Another major reason is that medical images model complex biological structures (such as brain maps). We expect the model parameter in the corresponding regression model to model this complexity. Lastly, the set of observations in medical imgaing data is usually severely unbalanced. In other words, positive medical diagnoses are relatively rare, and thus a given set of observations will contain far fewer positive diagnoses than negative diagnoses. The high-dimensionality, complexity and unbalanced observation set of medical imaging data serve as an excellent assessment of the efficiency of our proposed model and algorithm. Here we study three datasets; ABIDE Autism (Craddock et al., 2013; Lodhi & Bajwa, 2020), ADHD200 (Bellec et al., 2017) and Vessel MNIST 3D (Yang et al., 2021).

### 7.1 ABIDE Autism

The Autism Brain Imaging Data Exchange (ABIDE) dataset contains the resting state fMRI data collected from 98 subjects. The data has already been preprocessed for motion realignment and correction, slice timing correction and image normalisation. Each data sample corresponds to 111 cortical and sub-cortical brain regions scanned over 116 time periods (Craddock et al., 2013). Therefore each sample is a $111 \times 116$ data matrix of fMRI data from one subject. Each observation is a binary response variable $y \in \{0, 1\}$ depicting a subjects' diagnosis of either having autism ($y = 1$) or not ($y = 0$). The goal is to classify the subjects as either having or not having autism. We perform a single train-test split procedure and we choose 40 autistic and 40 control samples uniformly at random for training. We also choose 14 test subjects the same way. The case-control ratio is this dataset is 1 : 1. ABIDE Autism is the only balanced dataset in this set of experiments.

### 7.2 ADHD200

ADHD200 is a repository of resting state fMRI images of subjects from 8 research sites: Peking University, Brown University, Kennedy Krieger Institute, Donders Institute, NYU Child Study Center, Oregon Health and Science University, University of Pittsburgh and Washington University in St. Louis. The data includes brain maps of fractional Amplitude of Low-Frequency Fluctuations (fALFF) of 959 child subjects. fALFF brain maps can be useful in predicting Attention Deficit Hyperactivity Disorder (ADHD) in children. The data has been preprocessed, (consortium, 2012), and we also perform standard preprocessing of the data such as removing missing observations or images with poor quality (a list of images with poor quality has been made public, (consortium, 2012)). Each data sample corresponds to a 3-dimensional patient fMRI brain scan (T1-weighted image) of size $121 \times 145 \times 121$. Figure 8 shows the axial, sagittal and coronal views (views across each tensor mode) of a fMRI data sample . Each observation is a binary response variable $y \in \{0, 1\}$ depicting a child subject's diagnosis of either having ADHD ($y = 1$) or being typically developing ($y = 0$). The objective here is classification of children subjects with ADHD. The dataset has already undergone a train-test split procedure. The training dataset consists of 762 samples, 280 of which are labeled as ADHD (hyperactive/impulsive, inattentive and combined) and 482 of which are labeled as typical (control). The testing dataset consists of 197 samples, where 76 and 93 are labeled as ADHD and control, respectively. The data is slightly imbalanced where the case-control ratio in the training and testing sets is 3 : 5 and 4 : 5, respectively.

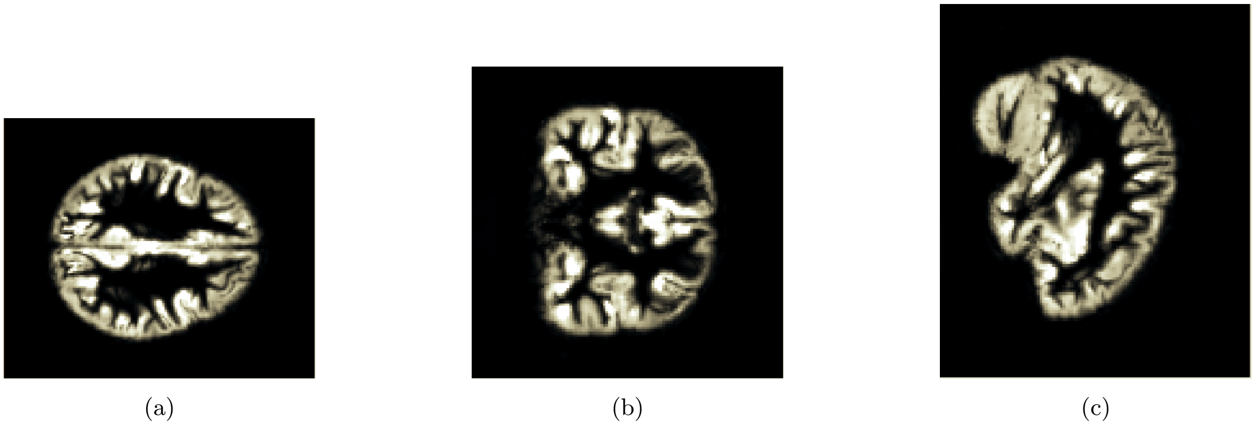

(a) (b) (c)

Figure 8: Axial ($a$), coronal ($b$), and sagittal ($c$) views of an fMRI data sample.

### 7.3 Vessel MNIST 3D

MedMNIST (Yang et al., 2021) is a collection of medical imaging datasets suitable for various tasks such as classification and regression. One of the datasets is the Vessel 3D dataset (Yang et al., 2020) containing $1,909$ reconstructions of vessel segments from brain Magnetic Resonance Angiography (MRA) data of differents subjects. Each data point corresponds to 3-dimensional vessel segment of dimension $28 \times 28 \times 28$. Each vessel segment may or may not exhibit an intracranial aneurysm and the dataset is diverse in the sense that it includes a variety of shapes and scales of intracranial aneurysms. The data has also been preprocessed to restore incomplete scans made by neurosurgeons and to remove duplicates data. Each observation is a binary response variable $y \in \{0, 1\}$ depicting a vessel segment diagnosis of either having an aneurysm ($y = 1$) or being healthy ($y = 0$). The goal is to therefore classify each vessel segment as either exhibiting an aneurysm or being healthy. The dataset has already undergone a train-test split procedure. The training dataset consists of 1335 samples, 150 of which are labeled as 'aneurysm' and 1185 of which are labeled as 'healthy'. The testing dataset consists of 382 samples, where 43 and 339 are labeled as 'aneurysm' and 'healthy', respectively. The data is severely imbalanced where the case-control ratio in both the training and testing sets is 3 : 25.

#### 7.3.1 Results of Experiments on Medical Imaging Data

All three medical imaging datasets support a logistic regression problem. We estimate the unknown coefficient tensor $\underline{\mathbf{B}}$ through our LSRTR algorithm and use our estimate to predict the responses of the test data through a logistic regression model, i.e., we calculate the posterior probability using the sigmoid function $s = (1/(1 + \exp\langle\underline{\mathbf{B}}, \underline{\mathbf{X}}_i\rangle))$ for every test subject $i$ and use a cut-off threshold of 0.5 to classify each response as 1 or 0, where the positive class is chosen if $s > 0.5$. Parameters including the Tucker rank $\{r_k\}_{k\in[K]}$, the separation rank $S$ and the step size $\alpha$ are set – for each dataset – using separate validation experiments. On all three datasets we compare our method with four other methods, two of which have been previously introduced, namely, LR and LTuR. The third method is a low-rank CP model method (which we name LCPR), specifically a block coordinate descent approach for parameter estimation of CP structured logistic regression models proposed by Tan et al. (Tan et al., 2012). The fourth method is a Support Vector Machine (SVM) with Gaussian Kernel (Hearst et al., 1998). In the case of the Vessel MNIST dataset, we also compare our method with an established baseline in the literature for Medical MNIST datasets: ResNet 50, which is residual neural network that is 50 layers deep. This neural network has also been augmented with a convolutional layer designed to handle 3-dimensional data. To evaluate the performance of each method we report the following scores: Specificity, defined as the true negative rate, sensitivity, defined as the true positive rate, the F1 score, the average accuracy score defined as the MAE, and the Area Under the Curve

(AUC). We note that we penalise false positives and false negatives with the same severity and compute the MAE to be consistent with the current literature (Li et al., 2018; Zhou et al., 2013; Zhang & Jiang, 2016). We also note that it is important to use the F1 score and AUC, as they are more appropriate metrics in the case of class imbalance, compared to average accuracy.

Table 4 summarizes the description of each dataset and Tables 5a, 5b, and 5c summarize the results. The chosen rank $(r_k)_{k\in[K]}$ for ABIDE Autism, ADHD200 and Vessel MNIST 3D are $(6,6)$, $(6,6,6)$ and $(3,3,3)$, respectively. The chosen LSR rank for LSRTR for ABIDE Autism, ADHD200 and Vessel MNIST 3D are $S=2$, $S=3$ and $S=2$, respectively. The results show that the LSRTR method performs well in terms of sensitivity and avergage accuracy for all datasets and the vector based-method LR has poorer performance with all datasets, where for some datasets all observations are predicted as positive. Particularly with the ADHD200 dataset, we are faced with the challenge of efficient estimation as the sample size is very small with respect to the model size (762 vs 2097152). The relatively good performance of LSRTR suggests that LSRTR performs well in the high-dimensional setting (i.e., when the sample size is significantly smaller than the number of covariates). We notice that LR performs very poorly with ADHD200, as there are over 2 million parameters to estimate. With the chosen rank, the LSR model decreases the dimensionality to $2322S+216$ which enables significantly more efficient classification. LTuR and LTR decrease the dimensionality to $2322+216$ and 2322, respectively, but exhibit poorer performance than LSRTR (and even SVM), suggesting that CP and Tucker models cause a loss of representation power that is essential to parameter estimation for some datasets. With the Vessel MNIST dataset we have more training samples than ADHD200 and a lower-dimensional problem (only 21952). For this reason all methods perform relatively well, with LSRTR having one of the best performances. When compared to ResNet 50, our method approaches its performance in terms of F1 score and beats its performance in terms of average accuracy. Additionally, we note that an advantage of using GLMs over neural networks is that GLMs naturally provide prediction confidence probabilities (not just accuracy scores) that complement the specificity and sensitivity scores through the posterior probabilities. Such confidence probabilities are important information for medical professionals to consider during in medical studies, as it provides valuable interpretability of the classification results. Additionally, unlike the ResNet 50 (and additional convolutional layer) neural network, LSRTR is easily extendable beyond the 3D case, and is trivially applied to other regression types (such as linear and Poisson), and does not require fine tuning for each case.

We make a similar remark on Table 5a. The sensitivity for LSRTR is 1, which is likely due to the fact that we use only 14 test samples from the ABIDE Autism dataset. In order to provide better interpretability, Figure 9 shows a histogram of the posterior probabiltities for the 14 test data samples. We see that all of the samples labeled as $y=1$ are classified as such since their posterior probabilities are all over 0.5. Most samples of Class 1 are classified with high confidence (in the sense that posterior probabilities are well above the 0.5 threshold). Only one sample of Class 0 was misclassified.

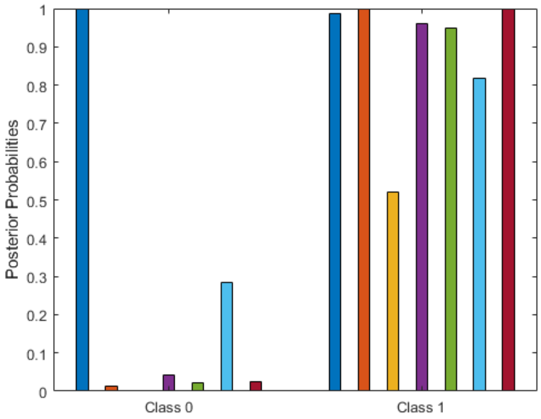


Figure 9: Histogram of posterior probabilities for test data in ABIDE Autism Dataset.

| **Dataset** | **Covariate Dimensions** | **Train Size** | **Test Size** |
|---|---|---|---|
| **ABIDE Autism** | $(111, 116)$ | 80 | 14 |
| **ADHD200** | $(121, 145, 121)$ | 762 | 169 |
| **Vessel MNIST 3D** | $(28, 28, 28)$ | 1335 | 382 |

Table 4: Description of three medical imaging datasets (ABIDE ADHD, ADHD 200 and Vessel MNIST 3D). Described are the model dimensions and train and test sizes for all three datasets.

| | **SVM** | **LR** | **LCPR** | **LTuR** | **LSRTR** |
|---|---|---|---|---|---|
| **Sensitivity** | 0.71 | 0.71 | 0.71 | 0.71 | 1 |
| **Specificity** | 0.14 | 0.71 | 0.85 | 0.85 | 0.85 |
| **F**1 **score** | 0.55 | 0.71 | 0.77 | 0.77 | **0.93** |
| **AUC** | 0.42 | 0.51 | 0.84 | 0.84 | **0.9** |
| **Average Accuracy** | 0.43 | 0.71 | 0.78 | 0.78 | **0.92** |

(a) **ABIDE Autism**

| | **SVM** | **LR** | **LCPR** | **LTuR** | **LSRTR** |
|---|---|---|---|---|---|
| **Sensitivity** | 0.41 | 1 | 0.62 | 0.51 | 0.83 |
| **Specificity** | 0.78 | 0 | 0.34 | 0.52 | 0.4 |
| **F**1 **score** | 0.5 | 0.62 | 0.35 | 0.48 | **0.65** |
| **AUC** | 0.62 | 0.5 | 0.56 | 0.62 | **0.67** |
| **Average Accuracy** | **0.62** | 0.45 | 0.47 | 0.51 | 0.59 |

(b) **ADHD**200

| | **SVM** | **LR** | **LCPR** | **LTuR** | **LSRTR** | **ResNet** 50 |
|---|---|---|---|---|---|---|
| **Sensitivity** | 0.39 | 0.53 | 0.26 | 0.32 | 0.47 | 0.85 |
| **Specificity** | 0.95 | 0.55 | 0.946 | 0.94 | 0.96 | 0.86 |
| **F**1 **score** | 0.44 | 0.21 | 0.3 | 0.37 | 0.55 | **0.57** |
| **AUC** | 0.84 | 0.52 | 0.6 | 0.66 | 0.81 | **0.9** |
| **Average Accuracy** | 0.89 | 0.55 | 0.869 | 0.87 | **0.91** | 0.85 |

(c) **Vessel MNIST 3D**

Table 5: Comparison of LSRTR with LR, LCPR and LTuR for classification diagnosis of test subjects in datasets: $(a)$ABIDE Autism $(b)$ ADHD200 and $(c)$ Vessel MNIST 3D. In the case of Vessel MNIST 3D, LSRTR is also compared to ResNet 50.

## 8 Conclusion

In this work, we investigate the LSR model on tensor-structured GLM problems. Specifically, we imposed a low-rank LSR structure on the coefficient tensor in GLMs. The parameter estimation problem is highly non-convex, and we propose a block coordinate descent algorithm, called LSRTR. Each convex sub-problem in LSRTR estimates a separate element/component of the LSR-structured coefficient tensor. In our theoretical analysis, we provide a minimax lower bound on the estimation error of the parameter estimation problem of LSR-structured tensors and specialise these bounds for Tucker-structured and CP-structured tensors. These bounds show that LSR reduces sample complexity compared to that of the vector case. We evaluate the tightness of these bounds numerically and through the comparison of the specific case of Tucker regression to tight bounds in the literature. The methods we use may be of interest to readers as they can also be utilised for deriving minimax bounds on other LSR-structured estimation problems. Furthermore, we evaluate the LSRTR algorithm and the LSR-model on several synthetic and real datasets. These experiments demonstrate that the LSR model is less restrictive than other tensor models (such as Tucker or CP), and can can be effectively employed for analysis on balanced and imbalanced medical imaging data. Some possible future work include theoretical analysis of the tightness of the minimax bounds, and theoretical guarantees of the LSRTR algorithm.

## References

Felix Abramovich and Vadim Grinshtein. Model selection and minimax estimation in generalized linear models. *IEEE Transactions on Information Theory*, 62(6):3721–3730, 2016.

Felix Abramovich and Vadim Grinshtein. High-dimensional classification by sparse logistic regression. *IEEE Trans. on Information Theory*, 65(5):3068–3079, 2018.

Talal Ahmed, Haroon Raja, and Waheed U Bajwa. Tensor regression using low-rank and sparse Tucker decompositions. *SIAM Journal on Mathematics of Data Science*, 2(4):944–966, 2020.

Baiguo An and Beibei Zhang. Logistic regression with image covariates via the combination of $\ell_1$ and Sobolev regularizations. *PLOS One*, 15(6):e0234975, 2020.

Leighton Pate Barnes and Ayfer Ozgur. Minimax bounds for distributed logistic regression. *arXiv preprint arXiv:1910.01625*, 2019.

Pierre Bellec, Carlton Chu, Francois Chouinard-Decorte, Yassine Benhajali, Daniel S Margulies, and R Cameron Craddock. The Neuro Bureau ADHD-200 preprocessed repository. *Neuroimage*, 144:275–286, 2017.

Quentin Berthet and Nicolai Baldin. Statistical and computational rates in graph logistic regression. In *Proc. Int. Conf. on Artificial Intelligence and Statistics (PMLR)*, pp. 2719–2730, 2020.

ADHD-200 consortium. The adhd-200 consortium: a model to advance the translational potential of neuroimaging in clinical neuroscience. *Frontiers in systems neuroscience*, 6:62, 2012.

Thomas M Cover and Joy A Thomas. *Elements of Information Theory.* John Wiley & Sons, 2012.

Cameron Craddock, Yassine Benhajali, Carlton Chu, Francois Chouinard, Alan Evans, András Jakab, Budhachandra Singh Khundrakpam, John David Lewis, Qingyang Li, Michael Milham, et al. The Neuro Bureau preprocessing initiative: Open sharing of preprocessed neuroimaging data and derivatives. *Frontiers in Neuroinformatics*, 7:27, 2013.

John P Dumas, Muhammad A Lodhi, Batoul A Taki, Waheed U Bajwa, and Mark C Pierce. Computational endoscopy—a framework for improving spatial resolution in fiber bundle imaging. *Optics Letters*, 44(16): 3968–3971, 2019.

Dylan J Foster, Satyen Kale, Haipeng Luo, Mehryar Mohri, and Karthik Sridharan. Logistic regression: The importance of being improper. In *Proc. Conf. On Learning Theory (PMLR)*, pp. 167–208, 2018.

Mohsen Ghassemi, Zahra Shakeri, Anand D Sarwate, and Waheed U Bajwa. Learning mixtures of separable dictionaries for tensor data: Analysis and algorithms. *IEEE transactions on signal processing*, 68:33–48, 2019.

Christophe Giraud. *Introduction to High-dimensional Statistics.* Chapman and Hall/CRC, 2021.

Johan Håstad. Tensor rank is NP-complete. In *Proc. International Colloquium on Automata, Languages, and Programming*, pp. 451–460. Springer, 1989.

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

Hung Hung and Chen-Chien Wang. Matrix variate logistic regression model with application to EEG data. *Biostatistics*, 14(1):189–202, 2013.

Alexander Jung, Yonina C Eldar, and Norbert Görtz. On the minimax risk of dictionary learning. *IEEE Trans. on Inf. Theory*, 62(3):1501–1515, 2016.

Rafail Z Khas'minskii. A lower bound on the risks of non-parametric estimates of densities in the uniform metric. *Theory of Probability & Its Applications*, 23(4):794–798, 1979.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

Kuan-Yun Lee and Thomas Courtade. Minimax bounds for generalized linear models. *Proc. Advances in Neural Information Processing Systems*, 33:9372–9382, 2020.

Xiaoshan Li, Da Xu, Hua Zhou, and Lexin Li. Tucker tensor regression and neuroimaging analysis. *Statistics in Biosciences*, 10(3):520–545, 2018.

Muhammad Asad Lodhi and Waheed U Bajwa. Learning product graphs underlying smooth graph signals. *arXiv preprint arXiv:2002.11277*, 2020.

Peter McCullagh and John A Nelder. *Generalized Linear Models.* Routledge, 2019.

Frank Nielsen. Statistical divergences between densities of truncated exponential families with nested supports: Duo bregman and duo jensen divergences. *Entropy*, 24(3):421, 2022.

Kaare Brandt Petersen, Michael Syskind Pedersen, et al. The matrix cookbook. *Technical University of Denmark*, 7(15):510, 2008.

Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over \ell_q *−balls. IEEE transactions on information theory*, 57(10):6976–6994, 2011.

Garvesh Raskutti, Ming Yuan, and Han Chen. Convex regularization for high-dimensional multiresponse tensor regression. *The Annals of Statstics*, 47(3):1554–1584, 2019.

George AF Seber and Alan J Lee. *Linear Regression Analysis*, volume 330. John Wiley & Sons, 2003.

Jianing V Shi, Yangyang Xu, and Richard G Baraniuk. Sparse Bilinear Logistic Regression. *arXiv preprint arXiv:1404.4104*, 2014.

Tingni Sun and Cun-Hui Zhang. Scaled sparse linear regression. *Biometrika*, 99(4):879–898, 2012.

Batoul Taki, Mohsen Ghassemi, Anand D Sarwate, and Waheed U Bajwa. A minimax lower bound for low-rank matrix-variate logistic regression. In *Proc. 2021 55th Asilomar Conference on Signals, Systems, and Computers*, pp. 477–484. IEEE, 2021.

Xu Tan, Yin Zhang, Siliang Tang, Jian Shao, Fei Wu, and Yueting Zhuang. Logistic tensor regression for classification. In *Proc. 2012 PMLR Int. Conf. on Intelligent Science and Intelligent Data Engineering*, pp. 573–581. Springer, 2012.

Theodoros Tsiligkaridis and Alfred O Hero. Covariance estimation in high dimensions via Kronecker product expansions. *IEEE Transactions on Signal Processing*, 61(21):5347–5360, 2013.

Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation.* Springer New York, NY, 2009.

Martin J Wainwright. Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting. *IEEE Trans. on Inf. Theory*, 55(12):5728–5741, 2009.

Ying Wu, Dan Chen, Chaoqian Li, and Niansheng Tang. Bayesian tensor logistic regression with applications to neuroimaging data analysis of Alzheimer's disease. *Statistical Methods in Medical Research*, pp. 09622802221122409, 2022.

Jiancheng Yang, Rui Shi, and Bingbing Ni. Medmnist classification decathlon: A lightweight automl benchmark for medical image analysis. In *Prpc. 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pp. 191–195. IEEE, 2021.

Xi Yang, Ding Xia, Taichi Kin, and Takeo Igarashi. Intra: 3d intracranial aneurysm dataset for deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2656–2666, 2020.

Bin Yu. Assouad, Fano, and Le Cam. In *Festschrift for Lucien Le Cam*, pp. 423–435. Springer, 1997.

Anru R Zhang, Yuetian Luo, Garvesh Raskutti, and Ming Yuan. ISLET: Fast and optimal low-rank tensor regression via importance sketching. *SIAM J. on Mathematics of Data Science*, 2(2):444–479, 2020.

Jianguang Zhang and Jianmin Jiang. Decomposition-based tensor learning regression for improved classification of multimedia. *Journal of Visual Communication and Image Representation*, 41:260–271, 2016.

Jianguang Zhang and Jianmin Jiang. Rank-optimized logistic matrix regression toward improved matrix data classification. *Neural Computation*, 30(2):505–525, 2018.

Hua Zhou, Lexin Li, and Hongtu Zhu. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013.

## A Supporting results

*Proof of Lemma 3.* By Lemma 3 there exists a packing with minimum distance $\frac{d}{8}$ in the $\ell_1$ norm containing $2^{d/8}$ binary vectors. Mapping $0 \to -\alpha$ and $1 \to \alpha$ for all vectors in this packing shows that there exists a packing of at least $2^{d/8}$ vectors in $\{-\alpha, \alpha\}^d$ with minimum distance $\frac{d\alpha}{4}$ in the $\ell_1$ norm. For any pair $\mathbf{v}, \mathbf{v}'$ in that packing we have $\|\mathbf{v} - \mathbf{v}'\|_0 \geq \frac{d}{8}$, since every entry of $\mathbf{v} - \mathbf{v}'$ is in $\{-2\alpha, 0, 2\alpha\}$. ☐

*Proof of Lemma 4.* This is a direct consequence of Corollary 3. ☐

The proof of Lemma 6 contains the derivation of a tight upper bound on $\|\underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}\|_F^2$. The following lemma is a component needed to achieve such an upper bound; specifically, we will see that it proves useful in deriving an upper bound on $\|\underline{\mathbf{B}}_l\|_F^2$, for any $l \in [L]$.

**Lemma 9.** *Let $\mathbf{x} \in \mathbb{R}^m$ be any $m$-dimensional real vector and $\mathbf{O} \in \mathbb{O}^{m\times m}$ be any orthogonal basis of $\mathbb{R}^m$. Define $u_i = |\cos\theta_i|$, where $\theta_i$ is the angle between any possible $\mathbf{x}$ and basis vector $\mathbf{o}_i$. The function $f = \sum_{i=1}^{m}(u_i+1)^2$ is minimized when $u_i = \frac{-1}{\sqrt{m}}$, or when $\mathbf{x}$ is equiangular to all basis vectors $\mathbf{o}_i, \forall i \in [m]$.*

*Proof of Lemma 9.* Consider the function $f = \sum_{i=1}^{m}(u_i+1)^2$. Additionally, for a basis $\mathbf{O} \in \mathbb{O}^{m\times m}$ and $x \in R^m$, $\sum_{i=1}^m (\cos\theta_i)^2 = 1$, thus we have the equality constraint $g = \sum_{i=1}^{m} u_i^2 - 1 = 0$. Denote $\lambda$ as the Lagrange multiplier, and thus the Lagrange function is defined as

$$L_g = \sum_{i=1}^{m}(u_i+1)^2 - \lambda(\sum_{i=1}^{m} u_i^2 - 1). \tag{51}$$

The partial derivative of $L_g$ with respect to $u_i, \forall i \in [m]$ is

$$\frac{\partial L_g}{\partial u_i} = 2(u_i+1) - 2\lambda u_i,\ \forall i \in [m]. \tag{52}$$

The partial derivative of $L_g$ with respect to $\lambda$ is

$$\frac{\partial L_g}{\partial \lambda} = -\sum_{i=1}^{m} u_i^2 + 1. \tag{53}$$

By setting (52) and (53) to zero we get

$$u_i = \frac{-1}{1-\lambda}\ \forall i \in [m], \tag{54}$$

and

$$-\sum_{i=1}^{m} u_i^2 + 1 = 0, \tag{55}$$

respectively. What we have in (54) and (55) is a system of $m+1$ equations and $m+1$ unknowns. We solve the system to find the critical points of $f(\cdot)$, which are $u_i = \frac{-1}{\sqrt{m}}$ and $u_i = \frac{1}{\sqrt{m}}$. Since $u_i \geq 0$, the solution is $u_i = \frac{1}{\sqrt{m}}, \forall i \in [m]$ minimizes the function $f(\cdot)$. $\square$

*Proof of Lemma 6.* Fix the following arbitrary real orthonormal bases: $\mathbf{Q}$ of $\mathbb{R}^{\widetilde{r}}$, and $K$ sets of $r_k$ bases, $\left\{\mathbf{U}_{k,j}\right\}_{j=1}^{r_k}$ of $\mathbb{R}^{m_k}, \forall k \in [K]$.

Next, consider the following hypercubes or subsets thereof: 1) The set of $F$ vectors $\{\mathbf{s}_f\}$ from Lemma 4:

$$\mathbf{s}_f \in \left\{\frac{-1}{\sqrt{\widetilde{r}-1}}, \frac{+1}{\sqrt{\widetilde{r}-1}}\right\}^{\widetilde{r}-1}, \tag{56}$$

where $f \in [F]$, and 2) $S \times K$ sets of $P_{(k,s)}$ matrices $\forall k \in [K], \forall s \in [S]$, from Lemma 5:

$$\mathbf{S}_{p_{(k,s)}} \in \left\{\frac{-1}{\sqrt{(m_k-1)r_k}}, \frac{+1}{\sqrt{(m_k-1)r_k}}\right\}^{(m_k-1)\times r_k} \quad \forall k \in [K],\ s \in [S], \tag{57}$$

where $p_{(k,s)} \in [P_{(k,s)}]$.

We proceed with the following steps in order to construct the final set $\mathcal{B}_L$ of coefficient tensors from the sets in (56) and (57). Since $\mathcal{B}_L \subset \mathcal{B}_d(\underline{\mathbf{0}})$, we assume that the energy of any $\underline{\mathbf{B}}_l$ is upper bounded by $d^2$. We will construct $\underline{\mathbf{G}}_f$, and matrices with orthonormal columns, namely $\mathbf{B}_{p_{(k,s)}} \forall k \in [K],\ s \in [S]$, all of which will be used to construct every $\underline{\mathbf{B}}_l \in \mathcal{B}_L$. Specifically, due to our LSR model, any tensor $\underline{\mathbf{B}}_l$ will have a low-rank LSR structure.

We use the notation $(f,i)$ to denote the $i^{th}$ step in constructing the $f^{th}$ element of $\underline{\mathbf{G}}_f$. Hence in the first step, we construct vectors $\mathbf{g}_{(f,1)} \in \mathbb{R}^{\widetilde{r}}$ for $f \in [F]$, using $\mathbf{Q}$ and $\mathbf{s}_f$, as follows:

$$\mathbf{g}_{(f,1)} = \mathbf{Q}\begin{bmatrix}\sqrt{\frac{1}{\sqrt{\widetilde{r}-1}}} \\ \mathbf{s}_f\end{bmatrix}, \forall f \in [F]. \tag{58}$$

From (58), since $\|\mathbf{s}_f\|_2^2 = 1$ we have:

$$\left\|\mathbf{g}_{(f,1)}\right\|_2^2 = \left\|\mathbf{Q}\begin{bmatrix}\sqrt{\frac{1}{\sqrt{\widetilde{r}-1}}} \\ \mathbf{s}_f\end{bmatrix}\right\|_2^2 = \frac{\widetilde{r}}{\widetilde{r}-1}.$$

Now, we define each vector $\mathbf{g}_f$ as

$$\mathbf{g}_f = \frac{\varepsilon}{\sqrt{\widetilde{r}}}\mathbf{g}_{(f,1)}, \forall f \in [F], \tag{59}$$

for some positive number $\varepsilon$.

Similaryly, we use the notation $(p_{(k,s)}, i)$ to denote the $i^{th}$ step in constructing the $p_{(k,s)}^{th}$ element of $\mathbf{B}_{p_{(k,s)}}$. Hence, we construct matrices $\mathbf{B}_{(p_{(k,s)},1)} \in \mathbb{R}^{m_k \times r_k}$, for $p_{(k,s)} \in [P_{(k,s)}], \forall k \in [K], s \in [S]$. Define $\mathbf{B}_{(p_{(k,s)},1)}^{(j)}$ as the $j^{th}$ column of $\mathbf{B}_{(p_{(k,s)},1)}, \forall k \in [K], s \in [S]$, and $\mathbf{S}_{p_{(k,s)}}^{(j)}$ as the $j^{th}$ column of $\mathbf{S}_{p_{(k,s)}}$. Let the columns be constructed as follows:

$$\mathbf{B}_{(p_{(k,s)},1)}^{(j)} = \mathbf{U}_{k,j}\begin{bmatrix}1 \\ \mathbf{S}_{p_{(k,s)}}^{(j)}\end{bmatrix}, \forall p_{(k,s)} \in [P_{(k,s)}], k \in [K], s \in [S]. \tag{60}$$

From (60) we have:

$$\left\|\mathbf{B}_{(p_{(k,s)},1)})^{(j)}\right\|_2^2 = = \frac{r_k+1}{r_k} \; \forall k \in [K].$$

We now construct matrices $\mathbf{B}_{p_{(k,s)}} \in \mathbb{R}^{m_k \times r_k}$, for $p_{(k,s)} \in [P_{(k,s)}], k \in [K], s \in [S]$. The construction of each $\mathbf{B}_{p_{(k,s)}}$ follows the same procedure for all $k \in [K]$ and $s \in [S]$. Define $\mathbf{B}^{(j)}_{p_{(k,s)}} \in \mathbb{R}^{m_k}$ as the $j^{th}$ column of $\mathbf{B}_{p_{(k,s)}}$, for $j \in [r_k]$. We set

$$\mathbf{B}^{(1)}_{p_{(k,s)}} = \frac{\mathbf{B}^{(1)}_{(p_{(k,s)},1)}}{\left\|\mathbf{B}^{(1)}_{(p_{(k,s)},1)}\right\|_2}, \tag{61}$$

and define

$$\mathbf{a}^{(j+1)} \triangleq \mathbf{B}^{(j+1)}_{(p_{(k,s)},1)} - \sum_{j'=1}^{j} \langle \mathbf{B}^{(j+1)}_{(p_{(k,s)},1)}, \mathbf{B}^{(j')}_{p_{(k,s)}} \rangle \mathbf{B}^{(j')}_{p_{(k,s)}}, \tag{62}$$

and

$$\mathbf{B}^{(j+1)}_{p_{(k,s)}} \triangleq \frac{\mathbf{a}^{j+1}}{\|\mathbf{a}^{j+1}\|_2}. \tag{63}$$

The steps in (61), (62) and (63) constitute the well-known Gram-Schmidt process. Thus, the set of vectors $\mathbf{B}^{(j)}_{(p_{(k,s)},1)}$, for $j \in [r_k]$, $p_{(k,s)} \in [P_{(k,s)}], k \in [K]$ and $s \in [S]$ are orthonormal, i.e, $\left\|\mathbf{B}^{(j)}_{p_{(k,s)}}\right\|_2^2 = 1$ and $\mathbf{B}^{(j)}_{p_{(k,s)}} \perp \mathbf{B}^{(j')}_{p_{(k,s)}}$, for any two distinct $j, j' \in [r_k]$. Consequently, $\left(\mathbf{B}_{p_{(k,s)}}\right)^T \left(\mathbf{B}_{p_{(k,s)}}\right) = \mathbf{I}_{r_k}$. Now by defining the set

$$\mathcal{L} \triangleq \left\{ (f, \left(p_{(k,s)}\right)_{k \in [K],\; s \in [S]}) : f \in [F], p_{(k,s)} \in [P_{(k,s)}], k \in [K], s \in [S] \right\} \tag{64}$$

as the set of all tuples, $(f, p_{(1,1)}, \dots, p_{(1,S)}, \dots, p_{(K,1)}, \dots, p_{(K,S)})$, we have

$$L = |\mathcal{L}| \overset{(a)}{\geq} 2^{(1/8)\left[(\widetilde{r}-1) + S\sum_{k=1}^{K}(m_k-1)r_k\right]}, \tag{65}$$

where $(a)$ follows from Lemma 4 and Corollary 5. We define the set of coefficient tensors, $\mathcal{B}_L$ as,

$$\mathcal{B}_L \triangleq \left\{ \underline{\underline{\mathbf{B}}}_l = \sum_{s=1}^{S} \underline{\underline{\mathbf{G}}}_f \times_1 \mathbf{B}_{p_{(1,s)}} \times_2 \mathbf{B}_{p_{(2,s)}} \cdots \times_K \mathbf{B}_{p_{(K,s)}} : l \in [L], f \in [F], p_{(k,s)} \in [P_{(k,s)}], k \in [K], s \in [S] \right\}, \tag{66}$$

and we restrict $\varepsilon$ such that

$$\frac{1}{S}\sqrt{\frac{32(\widetilde{r}-1)}{\widetilde{r}}} < \varepsilon < \frac{d}{S}\sqrt{\frac{\widetilde{r}-1}{\widetilde{r}}}. \tag{67}$$

We make the final note that, due to the Kronecker product, we can express $\text{vec}(\underline{\underline{\mathbf{B}}}_l)$ as:

$$\text{vec}(\underline{\underline{\mathbf{B}}}_l) = \sum_{s=1}^{S} (\mathbf{B}_{p_{(K,s)}} \otimes \mathbf{B}_{p_{(K-1,s)}} \otimes \cdots \otimes \mathbf{B}_{p_{(1,s)}})\mathbf{g}_f. \tag{68}$$

We have the following remaining tasks at hand: 1) We must show that the energy of any $\underline{\underline{\mathbf{B}}}_l$ is less than $d^2$. 2) We must derive upper and lower bounds on the distance $\left(\|\underline{\underline{\mathbf{B}}}_l - \underline{\underline{\mathbf{B}}}_{l'}\|_F^2\right)$ between any two distinct tensors

$\underline{\mathbf{B}}_l, \underline{\mathbf{B}}_{l'}, \in B_L$. We begin by showing $\|\underline{\mathbf{B}}_l\|_F^2 < d^2$:

$$\begin{aligned}
\|\underline{\mathbf{B}}_l\|_F^2 &= \left\| \sum_{s=1}^{S} \underline{\mathbf{G}}_f \times_1 \mathbf{B}_{p_{(1,s)}} \times_2 \mathbf{B}_{p_{(2,s)}} \times_3 \cdots \times_K \mathbf{B}_{p_{(K,s)}} \right\|_F^2 \\
&= \left\| \sum_{s=1}^{S} \left(\mathbf{B}_{p_{(K,s)}} \otimes \mathbf{B}_{p_{(K-1,s)}} \otimes \cdots \otimes \mathbf{B}_{p_{(1,s)}}\right) \mathbf{g}_f \right\|_2^2 \\
&\overset{(b)}{\leq} \left\| \sum_{s=1}^{S} \mathbf{B}_{p_{(K,s)}} \otimes \cdots \otimes \mathbf{B}_{p_{(1,s)}} \right\|_F^2 \|\mathbf{g}_f\|_2^2 \qquad (69) \\
&\overset{(c)}{\leq} \left( \sum_{s=1}^{S} \left\|\mathbf{B}_{p_{(K,s)}} \otimes \cdots \otimes \mathbf{B}_{p_{(1,s)}}\right\|_F \right)^2 \|\mathbf{g}_f\|_2^2 \qquad (70) \\
&\overset{(d)}{=} S^2 \prod_k \|\mathbf{B}_{k,p_k}\|_F^2 \, \|\mathbf{g}_f\|_2^2 = \frac{S^2 \widetilde{r} \varepsilon^2}{\widetilde{r}-1} \overset{(e)}{<} d^2, \qquad (71)
\end{aligned}$$

where $(b)$ follows from the fact that for any matrix $\mathbf{A}$ and any vector $\mathbf{a}$, $\|\mathbf{A}\mathbf{a}\|_2 \leq \|\mathbf{A}\|_2 \|\mathbf{a}\|_2$ and the fact that $\|\mathbf{A}\|_2 \leq \|\mathbf{A}\|_F$ (Petersen et al., 2008). Additionally, $(c)$ follows the triangle inequality and $(d)$ from the fact that the matrix norm of the Kronecker product is the product of the matrix norms. Additionally, $(e)$ holds due to (67).

We proceed with deriving lower and upper bounds on $\|\underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}\|_F^2$ for any two distinct $\underline{\mathbf{B}}_l, \underline{\mathbf{B}}_{l'} \in B_L$. We first denote the square matrix $\widetilde{\mathbf{B}}_{p_{(k,s)}} \in \mathbb{R}^{m_k \times m_k}$ as the completed orthonormal matrix of each low-rank matrix $\mathbf{B}_{p_{(k,s)}} \in \mathbb{R}^{m_k \times r_k}$. Also, $\widetilde{\underline{\mathbf{G}}}_f \in \mathbb{R}^{m_1 \times \cdots \times m_K}$ has entries $\widetilde{\underline{\mathbf{G}}}_f(\cdot)$ defined as follows:

$$\begin{cases} \widetilde{\underline{\mathbf{G}}}_f(1\colon r_1, \ldots, 1\colon r_K) = \underline{\mathbf{G}}_f(1\colon r_1, \ldots, 1\colon r_K) \\ \widetilde{\underline{\mathbf{G}}}_f(r_1+1\colon m_1, \ldots, r_K+1\colon m_K) = \underline{\mathbf{G}}_f(1\colon r_1, \ldots, 1\colon r_K). \end{cases} \tag{72}$$

Also define $\widetilde{\underline{\mathbf{B}}}_l = \sum_{s=1}^{S} \widetilde{\underline{\mathbf{G}}}_f \times_1 \widetilde{\mathbf{B}}_{p_{(1,s)}} \times_2 \cdots \times_K \widetilde{\mathbf{B}}_{p_{(K,s)}}$ for any $l \in [L]$. We these definitions, we have the eqaulity $\|\underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}\|_F^2 = \left\|\widetilde{\underline{\mathbf{B}}}_l - \widetilde{\underline{\mathbf{B}}}_{l'}\right\|_F^2$. Defining $\bigotimes_{k=K}^{k=1} \widetilde{\mathbf{B}}_{p_{(k,s)}} \triangleq \widetilde{\mathbf{B}}_{p_{(K,s)}} \otimes \cdots \otimes \widetilde{\mathbf{B}}_{p_{(1,s)}}$ for any $l \in [L]$ and we have the following:

$$\begin{aligned}
\|\underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}\|_F^2 &= \frac{\varepsilon^2}{\widetilde{r}} \left\| \sum_{s=1}^{S} \left( \bigotimes_{k=K}^{k=1} \widetilde{\mathbf{B}}_{p_{(k,s)}} \right) \widetilde{\mathbf{g}}_{(f,1)} - \sum_{s=1}^{S} \left( \bigotimes_{k=K}^{k=1} \widetilde{\mathbf{B}}_{p'_{(k,s)}} \right) \widetilde{\mathbf{g}}_{(f',1)} \right\|_2^2 \\
&= \frac{\varepsilon^2}{\widetilde{r}} \left( \left\| \sum_{s=1}^{S} \left( \bigotimes_{k=K}^{k=1} \widetilde{\mathbf{B}}_{p_{(k,s)}} \right) \widetilde{\mathbf{g}}_{(f,1)} \right\|_2^2 + \left\| \sum_{s=1}^{S} \left( \bigotimes_{k=K}^{k=1} \widetilde{\mathbf{B}}_{p'_{(k,s)}} \right) \widetilde{\mathbf{g}}_{(f',1)} \right\|_2^2 \right. \\
&\qquad \left. -2 \left\langle \sum_{s=1}^{S} \left( \bigotimes_{k=K}^{k=1} \widetilde{\mathbf{B}}_{p_{(k,s)}} \right) \widetilde{\mathbf{g}}_{(f,1)}, \sum_{s=1}^{S} \left( \bigotimes_{k=K}^{k=1} \widetilde{\mathbf{B}}_{p'_{(k,s)}} \right) \widetilde{\mathbf{g}}_{(f',1)} \right\rangle \right).
\end{aligned}$$

Define $\mathbf{T}_s = \bigotimes_{k=K}^{k=1} \widetilde{\mathbf{B}}_{p_{(k,s)}}$, $\mathbf{V}_s = \bigotimes_{k=K}^{k=1} \widetilde{\mathbf{B}}_{p'_{(k,s)}}$ for any $s \in [S]$, and $\mathbf{T}_s^{(j)}$, $\mathbf{V}_s^{(j)}$ as the $j^{th}$ column of $\mathbf{T}_s$ and $\mathbf{V}_s$, respectively, then we have

$$
\begin{aligned}
&\|\underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}\|_F^2 = \frac{\varepsilon^2}{\widetilde{r}} \left( \left\| \sum_{s=1}^{S} \mathbf{T}_s \widetilde{\mathbf{g}}_{(f,1)} \right\|_2^2 + \left\| \sum_{s=1}^{S} \mathbf{V}_s \widetilde{\mathbf{g}}_{(f',1)} \right\|_2^2 - 2 \left\langle \sum_{s=1}^{S} \mathbf{T}_s \widetilde{\mathbf{g}}_{(f,1)}, \sum_{s=1}^{S} \mathbf{V}_s \mathbf{g}_{(f',1)} \right\rangle \right) \\
&= \frac{\varepsilon^2}{\widetilde{r}} \left( \left( \sum_{s=1}^{S} \mathbf{T}_s^{T(1)} \widetilde{\mathbf{g}}_{(f,1)} \right)^2 + \cdots + \left( \sum_{s=1}^{S} \mathbf{T}_s^{T(m)} \widetilde{\mathbf{g}}_{(f,1)} \right)^2 + \left( \sum_{s=1}^{S} \mathbf{V}_s^{T(1)} \widetilde{\mathbf{g}}_{(f',1)} \right)^2 + \cdots + \left( \sum_{s=1}^{S} \mathbf{V}_s^{T(m)} \widetilde{\mathbf{g}}_{(f',1)} \right)^2 \right. \\
&\quad \left. -2 \left( \left( \sum_{s=1}^{S} \mathbf{T}_1^{T(1)} \widetilde{\mathbf{g}}_{(f,1)} \right) \left( \sum_{s=1}^{S} \mathbf{V}_s^{T(1)} \widetilde{\mathbf{g}}_{(f',1)} \right) + \cdots + \left( \sum_{s=1}^{S} \mathbf{T}_s^{T(m)} \widetilde{\mathbf{g}}_{(f,1)} \right) \left( \sum_{s=1}^{S} \mathbf{V}_s^{T(m)} \widetilde{\mathbf{g}}_{(f',1)} \right) \right) \right)
\end{aligned}
$$

We group every $\left(\sum_{s=1}^{S} \mathbf{T}_s^{T(j)} \widetilde{\mathbf{g}}_{(f,1)}\right)^2 + \left(\sum_{s=1}^{S} \mathbf{V}_s^{T(j)} \widetilde{\mathbf{g}}_{(f',1)}\right)^2 - 2\left(\sum_{s=1}^{S} \mathbf{T}_1^{T(j)} \widetilde{\mathbf{g}}_{(f,1)}\right)\left(\sum_{s=1}^{S} \mathbf{V}_s^{T(j)} \widetilde{\mathbf{g}}_{(f',1)}\right)$, for $j \in [m]$. We get

$$
\|\underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}\|_F^2 \geq \frac{\varepsilon^2}{\widetilde{r}} \sum_{i=1}^{m} \left( \left| \sum_{s=1}^{S} \mathbf{T}_s^{T(i)} \widetilde{\mathbf{g}}_{(f,1)} \right| - \left| \sum_{s=1}^{S} \mathbf{V}_s^{T(i)} \widetilde{\mathbf{g}}_{(f',1)} \right| \right)^2 . \tag{73}
$$

The expression in (73) contains inner products. Specifically, $\left|\sum_{s=1}^{S} \mathbf{T}_s^{T(i)} \widetilde{\mathbf{g}}_{(f,1)}\right| = \left|\left\langle \sum_{s=1}^{S} \mathbf{T}_s^{T(i)}, \widetilde{\mathbf{g}}_{(f,1)} \right\rangle\right|$. Denote $\lambda_i = \left|\cos\angle\left(\sum_{s=1}^{S} \mathbf{T}_s^{T(i)}, \widetilde{\mathbf{g}}_{(f,1)}\right)\right|$, then we have

$$
\begin{aligned}
\|\underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}\|_F^2 &\overset{(f)}{\geq} \frac{\varepsilon^2}{\widetilde{r}} \sum_{i=1}^{m} \left( \lambda_i \left\| \sum_{s=1}^{S} \mathbf{T}_s^{T(i)} \right\|_2 \|\widetilde{\mathbf{g}}_{(f,1)}\|_2 - \left\| \sum_{s=1}^{S} \mathbf{V}_s^{T(i)} \right\|_2 \|\widetilde{\mathbf{g}}_{(f',1)}\|_2 \right)^2 \\
&\overset{(g)}{=} \frac{\widetilde{r}\varepsilon^2}{\widetilde{r}(\widetilde{r}-1)} \sum_{i=1}^{m} \left( \lambda_i \left\| \sum_{s=1}^{S} \mathbf{T}_s^{T(i)} \right\| - \left\| \sum_{s=1}^{S} \mathbf{V}_s^{T(i)} \right\| \right)^2 \\
&\overset{(h)}{\geq} \frac{\widetilde{r}\varepsilon^2}{\widetilde{r}(\widetilde{r}-1)} \sum_{i=1}^{m} S^2(\lambda_i + 1)^2 \\
&\overset{(i)}{\geq} \frac{\widetilde{r}\varepsilon^2}{\widetilde{r}(\widetilde{r}-1)} \sum_{i=1}^{m} S^2(1 + \frac{1}{\sqrt{m}})^2 \geq \frac{S^2\widetilde{r}}{\widetilde{r}-1}\varepsilon^2,
\end{aligned} \tag{74}
$$

where $(d)$ is due to applying Cauchy-Schwartz inequality to $\left|\sum_{s=1}^{S} \mathbf{V}_s^{T(i)} \widetilde{\mathbf{g}}_{(f',1)}\right|$, $(g)$ is due to (59), $(h)$ is due to the fact that $\left\|\sum_{s=1}^{S} \mathbf{T}_s^{T(i)}\right\|$ and $\left\|\sum_{s=1}^{S} \mathbf{T}_s^{V(i)}\right\|$ are lower and upper bounded by $-S$ abd $S$, respectively. Finally, $(i)$ is from the result in Lemma 9. Finally, for finding upper bounds on $\|\underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}\|_F^2$, we have:

$$
\begin{aligned}
&\|\underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}\|_F^2 \\
&\overset{(j)}{\leq} \left( \left\| \sum_{s=1}^{S} \left( \mathbf{B}_{p_{(K,s)}} \otimes \cdots \otimes \mathbf{B}_{p_{(1,s)}} \right) \mathbf{g}_f \right\|_F + \left\| \sum_{s=1}^{S} \left( \mathbf{B}_{p'_{(K,s)}} \otimes \cdots \otimes \mathbf{B}_{p'_{(1,s)}} \right) \mathbf{g}_{f'} \right\|_F \right)^2 \\
&\leq \left( \sum_{s=1}^{S} \prod_{k} \left\|\mathbf{B}_{p_{(k,s)}}\right\|_F \|\mathbf{g}_f\|_2 + \sum_{s=1}^{S} \prod_{k} \left\|\mathbf{B}_{p'_{(k,s)}}\right\|_F \|\mathbf{g}_{f'}\|_2 \right)^2 \\
&\overset{(k)}{=} \left( 2 \sum_{s=1}^{S} \prod_{k} \left\|\mathbf{B}_{p_{(k,s)}}\right\|_F \|\mathbf{g}_f\|_2 \right)^2 \\
&= \frac{4S^2\widetilde{r}}{\widetilde{r}-1}\varepsilon^2,
\end{aligned} \tag{75}
$$

where $(j)$ follows from the triangle inequality, $(k)$ follows from the fact that $\left\|\mathbf{B}_{p_{(k,s)}}\right\|_F = \left\|\mathbf{B}_{p'_{(k,s)}}\right\|_F$ and that $\|\mathbf{g}_f\|_2 = \|\mathbf{g}_{f'}\|_2$. $\square$

*Proof of Lemma 7.* Consider the set $\mathcal{B}_L$ from Lemma 6, where the bounds in (74) and (75) hold. For the LSR-TGLM model in (13), consider $n$ i.i.d samples, with covariate tensors $\underline{\mathbf{X}}_i \in \mathbb{R}^{m_1 \times \cdots \times m_K}, \forall i \in [n]$, where $\text{vec}(\underline{\mathbf{X}}_i) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_x^2)$. According to (13), observations $y_i$ follow an exponential family distribution when conditioned on $\underline{\mathbf{X}}_i$, $\forall i \in [n]$. Consider the vector of $n$ observations, $\mathbf{y}$, and the tensor of $n$ samples, $\mathbf{X}$. Define also $\mathbb{I}(\mathbf{y}; l|\mathbf{X})$ as the mutual information between observations $\mathbf{y}$ and index $l$ conditioned on side-information $\mathbf{X}$. From (Cover & Thomas, 2012; Wainwright, 2009), we have,

$$\mathbb{I}(\mathbf{y}; l|\mathbf{X}) \le \frac{1}{L^2} \sum_{l,l'} \mathbb{E}_{\mathbf{X}} D_{KL}(f_l(\mathbf{y}|\mathbf{X})||f_{l'}(\mathbf{y}|\mathbf{X}), \tag{76}$$

where $D_{KL}(f_l(\mathbf{y}|\mathbf{X})||f_{l'}(\mathbf{y}|\mathbf{X})$ is the Kullback-Leibler (KL) divergence of probability distribution $f_l(\mathbf{y}|\mathbf{X})$ and

$$f_{l'}(\mathbf{y}|\mathbf{X})$$

of $\mathbf{y}$ given $\mathbf{X}$ for some $\underline{\mathbf{B}}_l, \underline{\mathbf{B}}_{l'} \in \mathcal{B}_L$. Denote $\eta_{l_i}$ and $\eta_{l'_i}$ as the link functions associated with $f_l(y_i|\underline{\mathbf{X}}_i)$ and $f_{l'}(y_i|\underline{\mathbf{X}}_i)$, respectively. Also denote $\mu_{l_i}$ as the expectation of sufficient statistic $T(y_i)$ conditioned on $\underline{\mathbf{X}}_i$ under model $\underline{\mathbf{B}}_l$ (otherwise known as the canonical parameter). We evaluate the KL divergence which is defined as follows (Nielsen, 2022):

$$D_{KL}(f_l(\mathbf{y}|\mathbf{X})||f_{l'}(\mathbf{y}|\mathbf{X})) = \sum_{i\in[n]} (\eta_{l_i} - \eta_{l'_i})\mu_l - a(\eta_{l_i}) + a(\eta_{l'_i}). \tag{77}$$

Now, we take the expectation of (77) with respect to the side-information $\mathbf{X}$. We have $\mathbb{E}_{\mathbf{X}}\left[(\eta_{l_i} - \eta_{l'_i})\mu_l - a(\eta_{l_i}) + a(\eta_{l'_i})\right] = \mathbb{E}_{\mathbf{X}}\left[(\eta_{l_i} - \eta_{l'_i})\mu_l\right]$, due to the fact that $\mathbb{E}_{\mathbf{X}}[a(\eta_{l_i})] = \mathbb{E}_{\mathbf{X}}[a(\eta_{l'_i})]$. We now have:

$$\begin{aligned}
\mathbb{E}_{\mathbf{X}} D_{KL}(f_l(\mathbf{y}|\mathbf{X})||f_{l'}(\mathbf{y}|\mathbf{X}) &= \sum_{i\in[n]} \mathbb{E}_{\underline{\mathbf{X}}}\big[(\langle \underline{\mathbf{B}}_l, \underline{\mathbf{X}}_i\rangle - \langle \underline{\mathbf{B}}_{l'}, \underline{\mathbf{X}}_i\rangle)\,\mathbb{E}[T(y_i)|\underline{\mathbf{X}}_i, l]\big] \\
&\overset{(l)}{=} \sum_{i\in[n]} \mathbb{E}_{\underline{\mathbf{X}}}\big[(\langle \underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}, \underline{\mathbf{X}}_i\rangle)\,\frac{\partial a(\eta_{l_i})}{\partial \eta_{l_i}}\big] \\
&\le \sum_{i\in[n]} \sqrt{\mathbb{E}_{\underline{\mathbf{X}}}[\langle \underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}, \underline{\mathbf{X}}_i\rangle]\mathbb{E}_{\underline{\mathbf{X}}}\Big[\Big(\frac{\partial a(\eta_{l_i})}{\partial \eta_{l_i}}\Big]\Big)^2} &(78)\\
&\le n\|\Sigma_x\|_2 \|\underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}\|_F^2 M, &(79)
\end{aligned}$$

Where $(l)$ follows from the fact that $\mu_{l_i} = \mathbb{E}[T(y_i)|\underline{\mathbf{X}}_i, l] = \frac{\partial a(\eta_{l_i})}{\partial \eta_{l_i}}$. We achieve (78) through Cauchy-Schwartz inequality. We make some remarks regarding (79): First, we replace the summation over $n$ samples with $n$ since each sample $\underline{\mathbf{X}}_i$ is independent. Secondly, Assumption 2 allows us to bound $\frac{\partial a(\eta_{l_i})}{\partial \eta_{l_i}}$ with $M$. Thirdly, the conditions on $\varepsilon$ in (67) means $\|\underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}\|_F > 1$ thus $\|\underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}\|_F^2 > \|\underline{\mathbf{B}}_l - \underline{\mathbf{B}}_{l'}\|_F$. Plugging in (79) into (76) gives us,

$$\mathbb{I}(\mathbf{y}; l|\underline{\mathbf{X}}^c) \le n\|\mathbf{\Sigma}_x\|_2 \|\mathbf{B}_l - \mathbf{B}_{l'}\|_F^2 M \overset{(e)}{\le} 4S^2Mn\|\Sigma_x\|_2 \frac{\widetilde{r}}{\widetilde{r}-1}\varepsilon^2,$$

where $(e)$ follows from (75). $\square$