Journal of Machine Learning Research 23 (2022) 1-59 Submitted 12/20; Revised 7/22; Published 8/22

Gaussian Process Parameter Estimation Using Mini-batch Stochastic Gradient Descent: Convergence Guarantees and Empirical Benefits

Hao Chen[*] HAOCHEN@STAT.WISC.EDU
Department of Statistics
University of Wisconsin-Madison
1300 University Avenue
Madison, WI 53706, USA
Lili Zheng[*†] LILI.ZHENG@RICE.EDU
Department of Electrical and Computer Engineering
Rice University
6100 Main St
Houston, TX 77005, USA
Raed Al Kontar[‡] ALKONTAR@UMICH.EDU
Department of Industrial and Operations Engineering
University of Michigan
1891 IOE Building 1205, Beal Ave
Ann Arbor, MI 48109, USA
Garvesh Raskutti RASKUTTI@STAT.WISC.EDU
Department of Statistics
University of Wisconsin-Madison
1300 University Avenue
Madison, WI 53706, USA

Editor: John Cunningham

Abstract

Stochastic gradient descent (SGD) and its variants have established themselves as the go-to algorithms for large-scale machine learning problems with independent samples due to their generalization performance and intrinsic computational advantage. However, the fact that the stochastic gradient is a biased estimator of the full gradient with correlated samples has led to the lack of theoretical understanding of how SGD behaves under correlated settings and hindered its use in such cases. In this paper, we focus on hyperparmeter estimation for the Gaussian process (GP) and take a step forward towards breaking the barrier by proving minibatch SGD converges to a critical point of the full log-likelihood loss function, and recovers model hyperparameters with rate $O(\frac{1}{K})$ for K iterations, up to a statistical error term depending on the minibatch size. Our theoretical guarantees hold provided that the kernel functions exhibit exponential or polynomial eigendecay which is satisfied by a wide range of kernels commonly used in GPs. Numerical studies on both

[*] Equal contribution.
[†] This work was done while Lili Zheng was at the University of Wisconsin-Madison.
[‡] For correspondence.

simulated and real datasets demonstrate that minibatch SGD has better generalization over state-of-the-art GP methods while reducing the computational burden and opening a new, previously unexplored, data size regime for GPs.

Keywords: Stochastic Optimization, Gaussian Processes, Convergence Rate, Computational Speed-up

1. Introduction

The Gaussian process (GP) has seen many success stories in various domains, be it in optimization (Yue and Al Kontar, 2020; Snoek et al., 2012), reinforcement learning (Srinivas et al., 2009; Krause and Ong, 2011), time series analysis (Kontar et al., 2020; Álvarez and Lawrence, 2011), control theory (Kocijan et al., 2004; Mesbah, 2016) and simulation meta-modeling (Zhou et al., 2011; Qian and Wu, 2008). One can attribute such success to its natural Bayesian interpretation, uncertainty quantification capability and highly flexible model priors. Yet its main limitation is the $O(n^3)$ computation and $O(n^2)$ storage for n training points (Rasmussen, 2003). Indeed, as mentioned in Hensman et al. (2013), a traditional large dataset for a GP is one with a few thousand data points and even those often require approximation techniques.

As a result, in the past two decades, a large body of work on GPs tackled approximate inference procedures to reduce the computational demands and numerical instabilities (mainly due to the need for matrix inversions). This push towards scalability dates back more than two decades ago. More recently, Quiñonero-Candela and Rasmussen (2005) unified previous approximation methods into a single probabilistic framework based on inducing points. Since then, many new methods have also been introduced. Most notable are: variational inference procedures that laid the theoretical foundation for the class of inducing point methods (Damianou et al., 2016; Nguyen et al., 2014; Zhao and Sun, 2016; Álvarez et al., 2010; Wilson et al., 2016), mixture of experts models (Deisenroth and Ng, 2015; Tresp, 2000), covariance tapering (Furrer et al., 2006; Kaufman et al., 2008) and kernel expansions (Le et al., 2013; Rahimi and Recht, 2008; Yang et al., 2015). On the other hand, there has been a recent push to utilize increasing computational power and GPU acceleration to solve exact GPs. This recent literature includes distributed Cholesky factorizations (Nguyen et al., 2019), preconditioned conjugate gradients (PCG) to solve linear systems (Gardner et al., 2018) and kernel matrix partitioning to perform all matrix-vector multiplications (Wang et al., 2019). Interestingly, Wang et al. (2019) was able to fit a bit more than 1 million data points using 8 GPUs in a few days.

One possible solution to extend GPs far beyond what is currently possible is through stochastic gradient decent (SGD) and its variants: *drawing $m << n$ samples at each iteration and updating model parameters following the gradient of the log-likelihood loss function on the m subsamples.* Indeed, SGD, or more generally the capability of inference via minibatches (possibly also with second order information), has been a key propeller behind the success of deep learning (LeCun et al., 2015) in its various forms and other objectives. *The caveat in GPs, however, is that, unlike empirical loss minimization, there exists correlation across all samples where any finite collection of the samples have a joint Gaussian distribution with covariance characterized by an empirical kernel matrix.* Hence the log-likelihood loss function is no longer the sum of losses evaluated at each sample, translating to the stochastic gradient being a biased estimator of the full gradient when taking expectation with respect to the random sampling. The lack of theoretical backing and understanding of how SGD behaves has long stood in the way of using SGD to conduct inference in GPs (Hensman et al., 2013) and in most settings where correlation amongst samples is high.

In this paper, we establish convergence guarantees of SGD for GPs for both the full gradient and the model parameters. Interestingly, without convexity or even Liptchitz conditions on the loss function, the structure of GP leads to an optimization error term of $O(\frac{1}{K})$ after K iterations for converging to a critical point and recovering the true noise variance up to a statistical error that vanishes as the minibatch size m tends to $+\infty$, for both RBF kernels and Matérn kernels. Our proof involves two key steps: first we concentrate the stochastic gradient to its conditional expectation using a covering argument and then we show that the latter satisfies a property similar to strong convexity by exploiting eigenvalues of the empirical kernel matrix. The proof and key findings offer standalone value beyond GPs and we hope they encourage researchers to further investigate SGD in other correlated settings such as Lévy, Itô and Markov processes.

Most importantly, our results open up a new data size regime to explore GPs. We are able to train $n \approx 1.2 \times 10^6$ data points using a single CPU core in around 30 minutes. Recall, it took the most recent advancements in exact GPs a couple of days using 8 GPUs to train when $n \approx 10^6$, and n is limited to approximately 10^4 without GPU. We find that GPs inferred using SGD offer remarkably better performance in various case studies with different dataset sizes, noise levels and input dimensions.

1.1 Main Contributions

We establish convergence guarantees for the minibatch SGD algorithm when training a GP under sampling with or without replacement and conduct numerical experiments to validate and supplement our theoretical results. Our main contributions are summarized as follows:

- **Convergence guarantees:** For a large enough minibatch size m, minibatch SGD converges to a *critical point* of the full log-likelihood loss function, and *recovers the true noise variance* up to a statistical error depending on m, when the kernel function exhibits exponential (RBF kernels) or polynomial eigendecay (Matérn kernels). To be specific, the full gradient and the estimation error of the noise variance evaluated at the Kth iterate are bounded by an *optimization error* term $O(\frac{1}{K})$ and a *statistical error* term $O(m^{-\frac{1}{2}})$, see Theorems 3.1 to 3.4.
  - **Proof techniques for statistical error:** Since the stochastic gradient is biased for estimating full gradients, we instead bound the difference from its conditional expectation given the covariates in the corresponding minibatch, *uniformly* over all possible parameter iterates. We use novel truncation and covering arguments to prove the uniform error bound, in order to avoid the dependence between past parameter iterates and the minibatch in the current iteration. This contributes to the statistical error term $O(m^{-\frac{1}{2}})$ in the convergence error bound.
  - **Proof techniques for optimization error:** To guarantee the $O(\frac{1}{K})$ optimization error bound, no convexity or even Liptchitz condition on the loss function are assumed. Instead, we prove that the conditional expectation of the loss function given covariates

$\mathbf{X}_n$ satisfies a relaxed property of strong convexity (see e.g., Lemma 3), where the "curvature" parameter is lower bounded by a constant regardless of minibatch size m. This proof relies on careful analysis for bounding the eigenvalues of empirical kernel matrices.

- **Numerical findings:** Through benchmarking with state-of-the-art methods on various datasets we show that SGD offers great value from both computational and statistical perspectives. Computationally, we scale to dataset sizes previously unexplored in GPs in a fraction of time needed for competing methods. Meanwhile statistically, we find that SGD improves generalization in GPs, specifically in large data settings.

1.2 Related Work

As mentioned earlier, there are several methods trying to tackle the *computational complexity of GPs*. Those can be roughly split into the following three categories, though it is by no means an exhaustive list (see the survey in [1]).

- **Exact inference via conjugate gradient based methods:** This recent class of literature has had the most success in scaling GPs. Initially such approaches depended on a structured kernel matrix where data lies in a regularly spaced grid (Saatçi, 2012; Wilson and Nickisch, 2015). Then with the help of GPU acceleration, conjugate gradient and distributed Cholesky factorization were applied to more general settings (Wang et al., 2019; Gardner et al., 2018; Ubaru et al., 2017). Such approaches have training complexity of $O(n^2)$ ($O(n\log n)$ possible on spaced grids), yet amenable to distributed computation and GPU acceleration.

- **Sparse approximate inference:** This class of methods is based on low rank approximation of the empirical kernel matrix where $\mathbf{K}_n \approx \mathbf{K}_{nz}\mathbf{K}_{zz}^{-1}\mathbf{K}_{zn}$ and z denotes a set of inducing points with cardinality$(z) = n_z << n$ (Kontar et al., 2018; Alvarez and Lawrence, 2009; Damianou et al., 2016; Zhao and Sun, 2016; Snelson and Ghahramani, 2006). Their time complexity is mainly $O(n_z^2 n)$ which can be reduced to $O(n + cn_z)$ for structured and regularly spaced grids. Indeed, low rank GPs have gained increased attention. Also, variational inference (VI) provided a theoretical foundation for this class of inducing points/kernel approximations (starting from the early work of Titsias (2009)).

- **Stochastic variational inference (SVI):** Following the work of Hoffman et al. (2013), SVI was introduced to GPs in Hensman et al. (2013). The key idea is to introduce a variational distribution over the inducing points so that the VI framework is amenable to stochastic optimization. This leads to a complexity of $O(n_z^3)$ at each iteration (Hoang et al., 2015; Blei et al., 2017). Unfortunately, recent results in Burt et al. (2019) show the need for at least $O(\log^D n)$ inducing points for Gaussian kernels, which implies a superlinear growth with the input dimension D. Although many of the aforementioned methods are proposed in the context of model inference (prediction), the idea can often carry over to the parameter estimation, and the complexity is multiplied by the number of iterations.

The *theoretical analysis for SGD* has also been extensively studied under various assumptions (Nemirovski et al., 2009; Rakhlin et al., 2011; Frostig et al., 2015; Bottou et al., 2018). In particular, in the context of empirical risk minimization where the objective loss function is summed over n data points and stochastic gradients are calculated for i.i.d. sampled data at all iterations, it is known that the expected squared error of SGD iterates at iteration K compared to the true minimizer is $O(1/K)$, with diminishing step size for strongly convex objectives. Furthermore, some recent literature (Hardt et al., 2016; Keskar et al., 2016) suggests that SGD has good generalization power, which has encouraged more practitioners to apply SGD to various application scenarios.

However, there are much fewer results when the stochastic gradients are *biased estimators* for the full gradients, despite the fact that unbiased estimates for the full gradient can be expensive or unavailable in certain cases. Some examples include: learning graph neural networks (Chen et al., 2018), distributed parallel optimization where sparsified stochastic gradient is applied (Stich and Karimireddy, 2019) and performing model selection for GPs. Homem-de Mello (2008); Chen and Luss (2018); Ajalloeian and Stich (2020) study the stochastic gradient algorithms under non-i.i.d. sampling or when the stochastic gradients are biased, and provide error bounds involving the bias term, or convergence guarantees built on consistency assumptions. Our paper does not make assumptions on the consistency of stochastic gradients or convexity of the full loss function, but exploits the nature of GP loss function and kernel matrices instead.

1.3 Organization

The paper is organized as follows: The problem setup is described in Section 2, and theoretical guarantees are provided in Section 3; Section 4 is devoted to a proof outline, key lemmas and the proof of some main steps; Section 5 presents practical considerations for applying minibatch SGD on GPs, while Section 6 includes our numerical results. We point out some open problems in Section 7 and conclude in Section 8.

2. Problem Setup

Notation Vectors and matrices are denoted by boldface letters, e.g., $\mathbf{K}_n$, $\boldsymbol{\theta}$, except for the full gradient $\nabla\ell(\boldsymbol{\theta})$ and stochastic gradient $g(\boldsymbol{\theta})$. For any vector $\mathbf{u} \in \mathbb{R}^p$, u_i denotes its ith entry, and $\|\mathbf{u}\|_2 = \left(\sum_{i=1}^p u_i^2\right)^{\frac{1}{2}}$ denotes its ℓ_2 norm. For any square matrix $\mathbf{A}$, $\lambda_i(\mathbf{A})$ denotes its ith largest eigenvalue. Also see a table of important notations in Appendix A.

We consider the Gaussian process model

$$\begin{aligned} f \sim \mathcal{GP}(\mu(\cdot), k(\cdot,\cdot)), \quad \mathbf{x}_1, \ldots, \mathbf{x}_n \overset{\text{i.i.d.}}{\sim} \mathbb{P}, \\ y_i = f(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma_\epsilon^2), \quad 1 \le i \le n, \end{aligned} \tag{1}$$

where $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^D$ is the input, $\mu(\cdot) : \mathcal{X} \to \mathbb{R}$ is the prior mean function, $k(\cdot,\cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the prior covariance function, and ϵ_i is the observational noise with variance σ_ϵ^2. Without loss of generality, we consider constant 0 mean function. In addition, the prior covariance function $k(\cdot,\cdot) = \sigma_f^2 k_0(\cdot,\cdot)$ involves a known kernel function $k_0(\cdot,\cdot)$ and a signal variance parameter σ_f^2. We observe data points $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ generated from (1) and organize them into

$(\mathbf{X}_n, \mathbf{y}_n) = ((\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top, (y_1, \ldots, y_n)^\top)$, from which we aim to learn the hyperparameters in order to predict outputs from new inputs based on the posterior process.

Denote by $\boldsymbol{\theta}^* = (\sigma_f^2, \sigma_\epsilon^2)^\top \in \mathbb{R}^2$ the underlying hyperparameters to be determined, and for notational convenience, we may also use θ_1^* to denote σ_f^2 and θ_2^* to denote σ_ϵ^2 in the following. One direct approach to estimate $\boldsymbol{\theta}^*$ is by applying gradient descent to minimize the scaled negative log marginal likelihood function

$$\begin{aligned} \ell(\boldsymbol{\theta}; \mathbf{X}_n, \mathbf{y}_n) &= -\frac{1}{n} \log p(\mathbf{y}_n | \mathbf{X}_n, \boldsymbol{\theta}) \\ &= \frac{1}{2n} [\mathbf{y}_n^\top \mathbf{K}_n^{-1}(\boldsymbol{\theta}) \mathbf{y}_n + \log |\mathbf{K}_n(\boldsymbol{\theta})| + n \log(2\pi)] \end{aligned} \tag{2}$$

over $\boldsymbol{\theta} \in (0, \infty)^2$, where $\mathbf{K}_n(\boldsymbol{\theta}) = \theta_1 \mathbf{K}_{f,n} + \theta_2 \mathbf{I}_n \in \mathbb{R}^{n \times n}$ is the marginal covariance matrix for noisy observations $\mathbf{y}_n$ given $\mathbf{X}_n$ and $\mathbf{K}_{f,n} \in \mathbb{R}^{n \times n}$ is the kernel matrix of k_0 evaluated at $\mathbf{X}_n$, i.e. $(\mathbf{K}_{f,n})_{i,j} = k_0(\mathbf{x}_i, \mathbf{x}_j)$. For notational convenience we will omit $\mathbf{K}_n(\boldsymbol{\theta})$ to $\mathbf{K}_n$ when $\boldsymbol{\theta}$ is clear from the context and denote $\mathbf{K}_n(\boldsymbol{\theta}^*)$ by $\mathbf{K}_n^*$. In this case, the derivative of $\ell(\boldsymbol{\theta})$ is of particular interest to us where each of its entries takes the form

$$\begin{aligned} (\nabla \ell(\boldsymbol{\theta}; \mathbf{X}_n, \mathbf{y}_n))_l &= \frac{1}{2n} \left[-\mathbf{y}_n^\top \mathbf{K}_n^{-1} \frac{\partial \mathbf{K}_n}{\partial \theta_l} \mathbf{K}_n^{-1} \mathbf{y}_n + \operatorname{tr}\left(\mathbf{K}_n^{-1} \frac{\partial \mathbf{K}_n}{\partial \theta_l} \right) \right] \\ &= \frac{1}{2n} \operatorname{tr}\left[(\mathbf{K}_n^{-1}(\mathbf{I}_n - \mathbf{y}_n \mathbf{y}_n^T \mathbf{K}_n^{-1}) \frac{\partial \mathbf{K}_n}{\partial \theta_l} \right], \end{aligned} \tag{3}$$

where θ_l is the lth element of $\boldsymbol{\theta}$ and $(\partial \mathbf{K}_n / \partial \theta_l)_{jk} = \partial (\mathbf{K}_n)_{jk} / \partial \theta_l$. For notational convenience we will suppress $\mathbf{X}_n, \mathbf{y}_n$ and use $\nabla \ell(\boldsymbol{\theta})$ instead. Notice that the computation in (3) is dominated by the calculation of $\mathbf{K}_n^{-1}$ which requires $O(n^3)$ time. In order to reduce the computational cost of training, we consider the minibatch stochastic gradient descent approach to optimize (2).

2.1 Minibatch SGD algorithm

Here we formally define the minibatch SGD algorithm considered in this paper. For any $k \geq 1$, consider the kth iteration which starts from $\boldsymbol{\theta}^{(k-1)}$, the parameter estimate after the $k - 1$th iteration ($\boldsymbol{\theta}^{(0)}$ is the initialization). We randomly sample ξ_k as a subset of $\{i\}_{i=1}^n$ of size m, then $\{(\mathbf{x}_i, y_i)\}_{i \in \xi_k}$ is the corresponding subset of data points which we organize into $(\mathbf{X}_{\xi_k}, \mathbf{y}_{\xi_k})$, where $\mathbf{X}_{\xi_k}$ is the submatrix formed by the rows of $\mathbf{X}_n$ and $\mathbf{y}_{\xi_k}$ is the subvector of $\mathbf{y}_n$, both indexed by ξ_k. Define $g(\boldsymbol{\theta}^{(k-1)}; \mathbf{X}_\xi, \mathbf{y}_\xi) \in \mathbb{R}^2$ as an approximation to $\nabla \ell(\boldsymbol{\theta}^{(k-1)}; \mathbf{X}_n, \mathbf{y}_n)$ that can be calculated from this subset, i.e.

$$\left(g(\boldsymbol{\theta}^{(k-1)}; \mathbf{X}_{\xi_k}, \mathbf{y}_{\xi_k}) \right)_l = \frac{1}{2 s_l(m)} \operatorname{tr}\left[(\mathbf{K}_{\xi_k}^{-1}(\mathbf{I}_m - \mathbf{y}_{\xi_k} \mathbf{y}_{\xi_k}^\top \mathbf{K}_{\xi_k}^{-1}) \frac{\partial \mathbf{K}_{\xi_k}}{\partial \theta_l^{(k-1)}} \right], \quad 1 \leq l \leq 2, \tag{4}$$

where $\mathbf{K}_{\xi_k}$ is the covariance matrix for $\mathbf{y}_{\xi_k}$ while also the principle submatrix formed by the rows and columns of $\mathbf{K}_n$ indexed by ξ_k. In the following we will also let $\mathbf{K}_{f,\xi_k}$ denote the $m \times m$ block of $\mathbf{K}_{f,n}$ indexed by ξ_k. A natural choice for $s_l(m)$ is m, but we will see in Section 3 that if kernels k_0 have exponential eigendecay, setting $s_1(m) \asymp \log m$ and $s_2(m) = m$ would lead to the convergence of both $\theta_1^{(k)}, \theta_2^{(k)}$ to the true hyperparameters. Algorithm 1

Algorithm 1: Minibatch SGD with uniform sampling

```
1 Input: $\boldsymbol{\theta}^{(0)} \in \mathbb{R}^2$, initial step size $\alpha_1 > 0$.
2 for $k = 1, 2, \ldots, K$ do
3     Randomly sample a subset of indices $\xi_k$ of size $m$;
4     Compute the stochastic gradient $g(\boldsymbol{\theta}^{(k-1)}; \mathbf{X}_{\xi_k}, \mathbf{y}_{\xi_k})$;
5     $\alpha_k \leftarrow \frac{\alpha_1}{k}$;
6     $\boldsymbol{\theta}^{(k)} \leftarrow \boldsymbol{\theta}^{(k-1)} - \alpha_k g(\boldsymbol{\theta}^{(k-1)}; \mathbf{X}_{\xi_k}, \mathbf{y}_{\xi_k})$;
7 end for
```

summarizes the steps of minibatch SGD, where we do not specify whether minibatches are sampled with or without replacement since our convergence guarantees will hold true under both cases, if minibatch size m is large enough (details provided in Section 3). We consider *diminishing step sizes*: the step size at the k th iteration is $\alpha_k = \frac{\alpha_1}{k}$. It is noteworthy that the time complexity of Algorithm 1 is $O(Km^3)$, compared to $O(Kn^3)$ for running gradient descent with K iterations.

2.2 Sampling Methods

In Algorithm 1 we conduct uniform sampling for each minibatch, that is, any subset of indices $\xi \subset [n]$ of size m has the same probability of being selected. An alternative to uniform sampling is to sample data points that are close to each other, which we call *nearby sampling*. One particular nearby sampling strategy is nearest neighbor search, where a minibatch consists of a uniformly sampled data point and its $m-1$ nearest neighbors within the data pool. We may construct a *k-d* tree to conduct nearest neighbor search, which finds the $m-1$ nearest neighbors for every data point in a given dataset of size n in $O(n \log n)$ time and $O(n)$ space. That is to say, the time complexity for minibatch SGD with this nearby sampling method (only line 3 in Algorithm 1 changes) is $O(Km^3 + n \log n)$ for K iterations.

Our main theoretical contribution is establishing convergence guarantees for uniform sampling SGD in Algorithm 1, but in addition to that, we will also provide some theoretical insights and numerical experiments for understanding the effect of nearby sampling.

3. Theoretical Guarantees

In this section, we present convergence guarantees for Algorithm 1, including error bounds for $\theta_l^{(k)} - \theta_l^*, l = 1, 2$ and $\nabla \ell(\boldsymbol{\theta}^{(k)})$. Two types of kernels are considered: those with exponential eigendecay (Section 3.1) and those with polynomial eigendecay (Section 3.2). For both types of kernels, the convergence of $\theta_2^{(k)}$ to the true noise variance σ_ϵ^2 and the full gradient $\nabla \ell(\boldsymbol{\theta}^{(k)})$ to 0 is guaranteed. In particular, for the former type of kernel, $\theta_1^{(k)}$ is also guaranteed to converge to the truth σ_f^2 under appropriate choice of $s_1(m)$. First we state the assumptions needed for our convergence guarantees.

Assumption 3.1 (Bounded iterates) *Both $\boldsymbol{\theta}^*$ and $\boldsymbol{\theta}^{(k)}$ for $0 \le k \le K$ lie in $[\theta_{\min}, \theta_{\max}]^2$, where $0 < \theta_{\min} < \theta_{\max}$.*

Remark 1 (Justification for Assumption 3.1) *The boundedness of parameter iterates is usually assumed in the literature of theoretical analysis for SGD (e.g., Nemirovski et al., 2009). As will be revealed by Theorem 3.1 and Theorem 3.4, $\boldsymbol{\theta}^{(k)}$ is guaranteed to be bounded within a region around $\boldsymbol{\theta}^*$ that gets smaller as k increases w.h.p., whenever the previous $\boldsymbol{\theta}^{(0)}, \dots, \boldsymbol{\theta}^{(k-1)}$ are bounded within some $[\theta_{\min}, \theta_{\max}]$. Hence, we only need to be careful about the initial steps of Algorithm 1, ensuring that the parameters $\theta_j^{(k)}, j = 1, 2$ are always positive and bounded. Moreover, our numerical experiments suggest that the iterate $\boldsymbol{\theta}^{(k)}$ of Algorithm 1 is always bounded as long as the initial step size is chosen appropriately (see Figure 3 in Section 6).*

Assumption 3.2 (Bounded stochastic gradient) *For all $0 \le k < K$,*

$$\|g(\boldsymbol{\theta}^{(k)}; \mathbf{X}_{\xi_{k+1}}, \mathbf{y}_{\xi_{k+1}})\|_2 \le G$$

for some $G > 0$.

Remark 2 (Justification for Assumption 3.2) *The boundedness assumption for stochastic gradients is also commonly seen in the literature (e.g., Hazan and Kale, 2011). Furthermore, **Assumption 3.1 can imply Assumption 3.2 with high probability**, under similar conditions to those in Theorem 3.1 or Theorem 3.3. The key idea is the stochastic gradients will be shown to be close to their conditional expectation $\mathbb{E}(g(\boldsymbol{\theta}^{(k)}; \mathbf{X}_{\xi_{k+1}}, \mathbf{y}_{\xi_{k+1}} | \mathbf{X}_{\xi_{k+1}})$, and the latter is bounded with high probability given Assumption 3.1 and the eigendecay assumptions for kernels in Theorem 3.1 or Theorem 3.3. We include the detailed explanation on this in Appendix F.*

3.1 Kernels with Exponential Eigendecay

The exponential eigendecay assumption is stated in detail as follows:

Assumption 3.3 (Exponential eigendecay) *Consider the kernel operator $\mathcal{K} : L^2(\mathbb{P}) \to L^2(\mathbb{P})$ that satisfies $\mathcal{K}\phi(\cdot) = \int \phi(\mathbf{x}) k_0(\cdot, \mathbf{x}) d\mathbb{P}(\mathbf{x})$, where $\mathbb{P}$ is the probability measure of the input as defined in (1). The eigenvalues of $\mathcal{K}$ are $\{Ce^{-bj}\}_{j=0}^{\infty}$, where $b > 0$ and $C \le 1$ are regarded as constants.*

The exponential eigendecay assumption is satisfied by the Radial Basis Function (RBF) kernels $k(\mathbf{x}, \mathbf{x}') = \exp\{-\|\mathbf{x} - \mathbf{x}'\|_2^2/(2l^2)\}$ when the probability distribution $\mathbb{P}$ of the input is Gaussian (see Section 4.3.1 of Rasmussen (2003)), which is widely seen in the GP literature. The specific decay rate b depends on the lengthscale parameter l of the corresponding kernel k_0. The requirement $C \le 1$ is only for theoretical convenience, and it suffices to have bounded C. The following theorem guarantees the convergence of the parameter iterates under the aforementioned assumptions.

Theorem 3.1 (Convergence of parameter iterates, exponential eigendecay) *Consider the output $\boldsymbol{\theta}^{(K)}$ of Algorithm 1, the minibatch SGD algorithm with diminishing step sizes. Under Assumptions 3.1 to 3.3, when $m > C$ for some constant $C > 0$, we have the following results under two corresponding conditions on $s_l(m)$:*

1. *If $s_2(m) = m$, initial step size α_1 satisfies $\frac{3}{2\gamma} \leq \alpha_1 \leq \frac{2}{\gamma}$ where $\gamma = \frac{1}{4\theta_{\max}^2}$, then for any $0 < \varepsilon < C\frac{\log\log m}{\log m}$, with probability at least $1 - CK\exp\{-cm^{2\varepsilon}\}$,*

$$(\theta_2^{(K)} - \theta_2^*)^2 \leq \frac{8G^2}{\gamma^2(K+1)} + Cm^{-\frac{1}{2}+\varepsilon}. \tag{5}$$

2. *If in addition to $s_2(m) = m$, $s_1(m)$ is set as $\tau \log m$ where $\tau > C$, $\frac{3}{2\gamma} \leq \alpha_1 \leq \frac{2}{\gamma}$ where γ depends on τ, then for any $0 < \varepsilon < \frac{1}{2}$, with probability at least $1 - CK\exp\{-c(\log m)^{2\varepsilon}\}$,*

$$(\theta_1^{(K)} - \theta_1^*)^2 + (\theta_2^{(K)} - \theta_2^*)^2 \leq \frac{8G^2}{\gamma^2(K+1)} + C(\log m)^{-\frac{1}{2}+\varepsilon}. \tag{6}$$

Here $c, C > 0$ depend only on $\theta_{\min}, \theta_{\max}, b$.

Remark 3.1 *Theorem 3.1 suggests that the noise variance parameter $\theta_2^{(K)}$ is guaranteed to converge to the truth θ_2^*, with the optimization error term $O(\frac{1}{K})$ and the statistical error term $O(m^{-\frac{1}{2}+\varepsilon})$ with high probability if $\varepsilon \log m$ is large, when the initial stepsize is appropriately chosen and $s_2(m) = m$. Furthermore, if we let $s_1(m) = \tau \log m$, then Algorithm 1 achieves convergence for both $\theta_1^{(K)}$ and $\theta_2^{(K)}$ with statistical error $O((\log m)^{-\frac{1}{2}+\varepsilon})$.*

Remark 3.2 *The different rates of statistical errors for estimating θ_1^* and θ_2^* come from the different eigenvalue structures between $\mathbf{K}_{f,\xi}$ (the $m \times m$ block of $\mathbf{K}_{f,n}$ indexed by ξ) and $\mathbf{I}_m$. One may also note that the statistical errors depend on m instead of n: this is due to the correlation among $\mathbf{y}_\xi$ from different minibatches ξ, conditioning on $\mathbf{X}$, which is different from the problems with independent samples.*

Remark 3.3 *The choice $s_1(m) \asymp \log m$ is because*

$$s_1(m)g_1(\boldsymbol{\theta}) = \operatorname{tr}\left[\mathbf{K}_\xi^{-1}(\mathbf{I}_m - \mathbf{y}_\xi\mathbf{y}_\xi^\top\mathbf{K}_\xi^{-1})\frac{\partial \mathbf{K}_\xi}{\partial \theta_1}\right] \asymp \log m, \tag{7}$$

and thus this choice of $s_1(m)$ ensures that $g_1(\boldsymbol{\theta})$ has the same scale as $g_2(\boldsymbol{\theta})$ (constant scale).

Remark 3.4 *For the second case where $s_1(m) = \tau \log m$, we need $\tau > \frac{64\theta_{\max}^4}{b\theta_{\min}^4}$ and*

$$\gamma = \min\left\{\frac{1}{32\tau b\theta_{\max}^2}, \frac{1}{4\theta_{\max}^2} - \frac{2\theta_{\max}^2}{\tau b\theta_{\min}^4}\right\}.$$

Remark 3.5 *The optimization error $O(\frac{1}{K})$ is credited to the structure of the GP loss function, which satisfies a relaxation of strong convexity (see Lemma 3 in Section 4). γ can be viewed a lower bound of an approximate "curvature" of the loss function, in the sense of a relaxed convexity. We will revisit this curvature term in Section 5.1 and illustrate the potential improvement nearby sampling brings to the curvature.*

Based on Theorem 3.1, we also derive the following convergence guarantee for the full gradient.

Theorem 3.2 (Convergence of full gradient, exponential eigendecay) *Consider the output $\boldsymbol{\theta}^{(K)}$ of Algorithm 1, the minibatch SGD algorithm with diminishing step sizes. Under Assumptions 3.1 to 3.3, if $\frac{3}{2\gamma} \leq \alpha_1 \leq \frac{2}{\gamma}$ for $\gamma = \frac{1}{4\theta_{\max}^2}$, $m > C$, $s_2(m) = m$, then for any $0 < \varepsilon < C\frac{\log\log m}{\log m}$, with probability at least $1 - CK\exp\{-cm^{2\varepsilon}\}$,*

$$\|\nabla \ell(\boldsymbol{\theta}^{(K)})\|_2^2 \leq C\left[\frac{G^2}{K+1} + m^{-\frac{1}{2}+\varepsilon}\right], \tag{8}$$

holds, where $c, C > 0$ depend only on $\theta_{\min}, \theta_{\max}, b$.

Theorem 3.2 implies that running SGD for sufficiently many iterations with large minibatch size leads to the convergence to a critical point of $\ell(\boldsymbol{\theta})$. As we will show in the proof sketch, the error bound for the full gradient is dominated by $(\theta_2^{(K)} - \theta_2^*)^2$, the estimation error of the noise variance, thus it scales the same as (5).

3.2 Kernels with Polynomial Eigendecay

Now we consider the kernels with polynomial eigendecay, which captures much stronger correlation than the kernels with exponential eigendecay, and thus broadens the applications of GP to a wider class of data sets. Due to this reason, it is of both practical and theoretical interest to investigate how SGD performs for this type of kernels. The polynomial eigendecay assumption is stated in detail as follows:

Assumption 3.4 (Polynomial eigendecay) *Consider the kernel operator $\mathcal{K} : L^2(\mathbb{P}) \to L^2(\mathbb{P})$ that satisfies $\mathcal{K}\phi(\cdot) = \int \phi(\mathbf{x})k_0(\cdot, \mathbf{x})d\mathbb{P}(\mathbf{x})$, where $\mathbb{P}$ is the probability measure of the input as defined in (1). The eigenvalues of $\mathcal{K}$ are $\{Cj^{-2b}\}_{j=0}^{\infty}$, where $b > \frac{\sqrt{21}+3}{4}$, and $C \leq 1$ are regarded as constants.*

This assumption is satisfied by the Matérn kernels (see section 2.3 in Bach (2017), Kanagawa et al. (2018)), another important kernel function class widely used in GP:

$$k_{\alpha,h}(\mathbf{x}, \mathbf{x}') = \frac{1}{2^{\alpha-1}\Gamma(\alpha)}\left(\frac{\sqrt{2\alpha\|\mathbf{x} - \mathbf{x}\|_2^2}}{h}\right)^{\alpha} B_\alpha\left(\frac{\sqrt{2\alpha\|\mathbf{x} - \mathbf{x}\|_2^2}}{h}\right), \tag{9}$$

where $B_\alpha(\cdot)$ is the modified Bessel function of the second kind of order α, and larger α leads to faster decay rate $b > 0$.

Theorem 3.3 (Convergence of parameter iterates, polynomial eigendecay) *Consider the output $\boldsymbol{\theta}^{(K)}$ of Algorithm 1, the minibatch SGD algorithm with diminishing step sizes. Under Assumptions 3.1 to 3.2 and Assumption 3.4, when $m > C$ for some constant $C > 0$, $s_2(m) = m$, $\frac{3}{2\gamma} \leq \alpha_1 \leq \frac{2}{\gamma}$ where $\gamma = \frac{1}{8\theta_{\max}^2}$, then for any $\varepsilon \in (\max\{0, f_1(b)\}, \frac{1}{2})$, with probability at least $1 - CKm^{-f_2(b)[\varepsilon - f_1(b)]} - CK\exp\{-cm^{2\varepsilon}\}$,*

$$(\theta_2^{(K)} - \theta_2^*)^2 \leq \frac{8G^2}{\gamma^2(K+1)} + Cm^{-\frac{1}{2}+\varepsilon}. \tag{10}$$

Here $c, C > 0$ depend only on $\theta_{\min}, \theta_{\max}, b$, and $f_1(b) = -\frac{2b^2-5b-3}{2b(2b-1)}$, $f_2(b) = \frac{4b(2b-1)}{4b+3}$.

Remark 3.6 (Comparison with Theorem 3.1: error bounds) *Compared with Theorem 3.1, Theorem 3.3 reflects the influence of stronger correlations (slower eigendecay of kernels) on the convergence of SGD. More specifically, $\varepsilon > f_1(b)$ is required in addition to $0 < \varepsilon < \frac{1}{2}$. That is to say, when $0 < f_1(b) < \frac{1}{2}$ ($\frac{\sqrt{21}+3}{4} < b < 3$), the statistical error scales at least as $m^{-(\frac{1}{2}-f_1(b))} > m^{-\frac{1}{2}}$, which decreases as b increases on $(\frac{\sqrt{21}+3}{4}, 3)$; while if $f_1(b) \le 0$ ($b \ge 3$) which means the correlation is not too strong, the statistical error still scales roughly as $m^{-\frac{1}{2}}$, the same as the exponential eigendecay case. Therefore, the slower eigendecay of kernels (stronger correlation structure) may lead to a slower convergence of SGD; while the good news is that, for moderately fast polynomial eigendecay ($b \ge 3$) we still have the same rate as the exponential eigendecay case.*

Remark 3.7 (Comparison with Theorem 3.1: probability terms) *Another difference between Theorem 3.1 and Theorem 3.3 lies in the probability term. When $b < 3$ and thus $f_1(b) > 0$, $CKm^{-f_2(b)[\varepsilon - f_1(b)]}$ always dominates $CK\exp\{-cm^{2\varepsilon}\}$ for $\varepsilon > f_1(b)$, which is a lower probability for the error bound in* (10) *compared to* (5)*. This is also the price we need to pay when considering kernels with slower eigendecay.*

To extend the theoretical results for kernels with exponential eigendecay to polynomial eigendecay, we develop novel upper and lower bounds for $\sum_{j=1}^n \lambda_j^l (\theta_1^{(k)}\lambda_j + \theta_2^{(k)})^{-2}, l = 0, 1, 2$ where λ_j is the jth largest eigenvalue of $K_{f,\xi}$, see Lemma 12. The proof for Lemma 12 requires careful analysis and different arguments from the proof for Lemma 11 that is established for kernels with exponential eigendecay, although they are both based on error bounds for eigenvalues of empirical kernel matrices in Braun (2006).

We briefly explain the reason behind the different scalings of statistical errors between Theorem 3.3 and the first part of Theorem 3.1 in the following. In fact, the statistical error term for $(\theta_2^{(K)} - \theta_2^*)^2$ is composed of two parts: one is $m^{-\frac{1}{2}+\varepsilon}$ for both types of kernels, another is caused by the fact that $\theta_1^{(K)}$ may not be estimated well, and the error induced by this fact depends on the eigendecay of the kernel, which scales as $m^{-(\frac{1}{2}-f_1(b))}$ for kernels with polynomial eigendecay. While for kernels with exponential eigendecay, this error term scales as $\frac{\log m}{m}$ and is dominated by the first error term $m^{-\frac{1}{2}+\varepsilon}$.

Remark 3.8 *For kernels with polynomial eigendecay, we don't have convergence guarantee for $\theta_1^{(K)}$ (signal variance of the kernel with the slowest eigendecay). This is due to that it is very hard to derive matched upper and lower bounds for the stochastic gradient $s_1(m)g_1(\boldsymbol{\theta}) = \operatorname{tr}\left[(\mathbf{K}_\xi^{-1}(\mathbf{I}_m - \mathbf{y}_\xi\mathbf{y}_\xi^\top\mathbf{K}_\xi^{-1})\frac{\partial \mathbf{K}_\xi}{\partial\theta_1}\right]$ (which scales as $\log m$ for exponential eigendecay), and thus we cannot specify the choice for $s_1(m)$ in order to make $g_1(\boldsymbol{\theta})$ scales similarly from $g_2(\boldsymbol{\theta})$.*

Theorem 3.4 (Convergence of full gradient, polynomial eigendecay) *Consider the output $\boldsymbol{\theta}^{(K)}$ of Algorithm 1, the minibatch SGD algorithm with diminishing step sizes. Under the same conditions as Theorem 3.4, for any $\varepsilon \in (\max\{0, f_1(b)\}, \frac{1}{2})$, with probability at least $1 - CK\left(m^{-f_2(b)[\varepsilon - f_1(b)]} + \exp\{-cm^{2\varepsilon}\}\right)$,*

$$\|\nabla \ell(\boldsymbol{\theta}^{(K)})\|_2^2 \le C\left[\frac{G^2}{K+1} + m^{-\frac{1}{2}+\varepsilon}\right], \tag{11}$$

holds, where $c, C > 0$ depend only on $\theta_{\min}, \theta_{\max}, b$, $f_1(b)$ and $f_2(b)$ are defined as in Theorem 3.3.

As mentioned after Theorem 3.2, the bound (11) for the full gradient scales the same as the bound (10) for $(\theta_2^{(K)} - \theta_2^*)^2$.

3.3 Extension to Summation of Multiple Kernels

So far we have assumed the kernel function $k_0(\cdot,\cdot)$ to be known. However, sometimes there might be several choices of potential kernels and it is desired to learn which kernel is the most appropriate from the data instead of manually picking one kernel. One possible approach is to let the covariance function $k(\cdot,\cdot)$ in (1) be a linear combination of all potential M kernels:

$$k(\cdot,\cdot) = \sum_{l=1}^{M} \sigma_{f,l}^2 k_l(\cdot,\cdot), \tag{12}$$

and then learn the signal variances $\sigma_{f,l}^2$ associated with the $M > 1$ kernels (Rasmussen, 2003). The kernel selection problem then translates to how well we can learn the signal variance parameters $\sigma_{f,l}^2$, $l = 1, \ldots M$.

Under this extension of the classical GP model, let the hyperparameter

$$\boldsymbol{\theta}^* = (\sigma_{f,1}^2, \ldots, \sigma_{f,M}^2, \sigma_\epsilon^2)^\top \in \mathbb{R}^{M+1}.$$

We can still write out the log-likelihood loss $\ell(\boldsymbol{\theta})$ as in (2), while the only difference lies in the formulation of $\mathbf{K}_n(\boldsymbol{\theta})$:

$$\mathbf{K}_n(\boldsymbol{\theta}) = \sum_{l=1}^{M} \theta_l \mathbf{K}_{f,n}^{(l)} + \theta_{M+1}\mathbf{I}_n,$$

where $\mathbf{K}_{f,n}^{(l)}$ is the kernel matrix of $k_l(\cdot,\cdot)$ evaluated at $\mathbf{X}_n$, i.e., $(\mathbf{K}_{f,n}^{(l)})_{ij} = k_l(\mathbf{x}_i, \mathbf{x}_j)$. Then it is straightforward to extend Algorithm 1 to this setting for learning $\boldsymbol{\theta}^*$.

In this more general setting, we can still derive convergence guarantees for some parameters and the full gradient, but under an additional technical assumption on the kernel matrices. In particular, we require that for any sample set $\{\mathbf{x}_i\}_{i=1}^n$, all kernel matrices $\mathbf{K}_{f,n}^{(l)}$, $l = 1, \ldots, M$ share the same eigenvectors. This assumption ensures that $\mathbf{K}_n(\boldsymbol{\theta})$ and $\mathbf{K}_n^*$ are simultaneously diagonizable and helps us focus on the eigenvalues of these matrices. This is a very strong assumption that would not hold in practice, but it remains an extremely challenging problem to prove the convergence results in such settings without this technical assumption. Specifically, under this additional assumption, we have convergence guarantees for $\sigma_{f,1}^2$, σ_ϵ^2, and the full gradient when $k_l(\cdot,\cdot), 1 \le l \le M$ have exponential eigendecay; for σ_ϵ^2 and the full gradient when $k_l(\cdot,\cdot), 1 \le l \le M$ have polynomial eigendecay. The detailed assumptions and theoretical results for this setting are included in Appendix B.

4. Proof Sketch

In this section, we present the proof sketch for the first part of Theorem 3.1 and Theorem 3.2 (kernels with exponential eigendecay). The proof of the second part in Theorem 3.1, Theorem 3.3 and Theorem 3.4 follows similar ideas although requiring more careful analysis. With a bit abuse of notation, we will omit $g(\boldsymbol{\theta}^{(k)}; \mathbf{X}_{\xi_{k+1}}, \mathbf{y}_{\xi_{k+1}})$ to $g(\boldsymbol{\theta}^{(k)})$ and denote its conditional expectation $\mathbb{E}(g(\boldsymbol{\theta}^{(k)})|\mathbf{X}_{\xi_{k+1}})$ by $g^*(\boldsymbol{\theta}^{(k)})$. Similarly we define $\nabla\ell^*(\boldsymbol{\theta}^{(k)}) = \mathbb{E}(\nabla\ell(\boldsymbol{\theta}^{(k)})|\mathbf{X}_n)$.

Due to the bias in the stochastic gradient, we take the followings steps instead of directly drawing the connection between $g(\boldsymbol{\theta}^{(k)})$ and $\nabla\ell(\boldsymbol{\theta}^{(k)})$:

- For proving the first part of Theorem 3.1:
  - We first show that the conditional expectation $g^*(\boldsymbol{\theta}^{(k)})$ of the stochastic gradient has a property similar to strong convexity, see Lemma 3.
  - We then prove that $g(\boldsymbol{\theta})$ is close to its conditional expectation $g^*(\boldsymbol{\theta})$ uniformly over all possible $\boldsymbol{\theta}$, and thus $g(\boldsymbol{\theta}^{(k)})$ is close to $g^*(\boldsymbol{\theta}^{(k)})$. Applying Lemma 4 to each minibatch leads to the desired result.

  These two steps lead to the $O(\frac{1}{K})$ optimization error rate for $(\theta_2^{(k)} - \theta_2^*)^2$, and a statistical error rate depending on m, as shown in Theorem 3.1.

- For proving Theorem 3.2:
  - Lemma 4 suggests that $\nabla\ell(\boldsymbol{\theta}^{(k)})$ is close to $\nabla\ell^*(\boldsymbol{\theta}^{(k)})$
  - The eigendecay of kernel matrices (see Lemma 5) ensures that $\|\nabla\ell^*(\boldsymbol{\theta}^{(k)})\|_2$ is controlled by $(\theta_2^{(k)} - \theta_2^*)^2$, which is upper bounded in Theorem 3.1.

  These steps above provide us with the same error bound of $\|\nabla\ell^*(\boldsymbol{\theta}^{(k)})\|_2$ from that of $(\theta_2^{(k)} - \theta_2^*)^2$ in Theorem 3.1.

4.1 Key Lemmas

The following two lemmas are the key building blocks of the proof: one shows that $g^*(\boldsymbol{\theta}^{(k)})$ enjoys a property similar to strong convexity, the other establishes a uniform bound for the statistical error $\nabla\ell(\boldsymbol{\theta}) - \nabla\ell^*(\boldsymbol{\theta})$ over $\boldsymbol{\theta} \in [\theta_{\min}, \theta_{\max}]^2$, and thus also bounds for $g(\boldsymbol{\theta}^{(k)}) - g^*(\boldsymbol{\theta}^{(k)})$.

Lemma 3 (Strongly convex-like property of $g^*(\boldsymbol{\theta}^{(k)})$) *Under Assumptions 3.1 to 3.3, if $s_2(m) = m$, $m > C$, then with probability at least $1 - 3Km^{-c}$, the following claim holds true for $0 \le k < K$:*

$$(\theta_2^{(k)} - \theta_2^*)(g^*(\boldsymbol{\theta}^{(k)}))_2 \ge \frac{1}{8\theta_{\max}^2}(\theta_2^{(k)} - \theta_2^*)^2 - \frac{C\log m}{m}, \tag{13}$$

Here $C > 0$ depends only on $\theta_{\min}, \theta_{\max}, b$.

Lemma 3 is a relaxation of strong convexity, but leads to similar convergence guarantees from running SGD on strongly convex objectives. To gain a further understanding for Lemma 3, define the "approximate" curvature term w.r.t. the noise variance at the $(k+1)$th iteration as

$$\gamma(\boldsymbol{\theta}^{(k)}) := \frac{\partial\mathbb{E}(g_2(\boldsymbol{\theta}^{(k)}; \mathbf{X}_{\xi_{k+1}}, \mathbf{y}_{\xi_{k+1}})|\mathbf{X}_{\xi_{k+1}})}{\partial\theta_2} = \frac{1}{2m}\sum_{j=1}^{m}(\theta_1^{(k)}\lambda_j^{(k)} + \theta_2^{(k)})^{-2}, \tag{14}$$

where $\lambda_j^{(k)}$ is the jth largest eigenvalue of $\mathbf{K}_{f,\xi_{k+1}}$. The term $\frac{1}{8\theta_{\max}^2}$ on the R.H.S of (13) can be viewed as a lower bound for $\gamma(\boldsymbol{\theta}^{(k)})$ for all k with high probability, and it remains a constant regardless of how large m is, which is a key to our proof of convergence.

To guarantee the constant "curvature", we establish novel upper and lower bounds on $\sum_{j=1}^m \lambda_j^l(\theta_1^{(k)}\lambda_j + \theta_2^{(k)})^{-2}, l = 0, 1, 2$ with high probability when m is large, where λ_j is the jth largest eigenvalue of $\mathbf{K}_{f,\xi}$ (see Lemma 5). The proof of Lemma 5 is based on established error bounds for the empirical eigenvalues in Braun (2006) and the eigendecay of the kernel $k_0(\cdot,\cdot)$.

Lemma 4 (Uniform statistical error) *Under Assumptions 3.1 to 3.2, Assumption 3.3 or 3.4, for any* $0 < \varepsilon < \frac{1}{2}$, $1 \leq i \leq 2$, *we have*

$$\mathbb{P}\left(\sup_{\boldsymbol{\theta}\in[\theta_{\min},\theta_{\max}]^2} |(\nabla\ell(\boldsymbol{\theta}))_i - (\nabla\ell^*(\boldsymbol{\theta}))_i| > Cn^{-\frac{1}{2}+\varepsilon}\right) \leq C\exp\{-cn^{2\varepsilon}\}. \tag{15}$$

Here $c, C > 0$ *only depend on* $\theta_{\min}$, $\theta_{\max}, b$.

The major difficulty in the proof of Lemma 4 is to control the error term uniformly over $\boldsymbol{\theta} \in [\theta_{\min}, \theta_{\max}]^2$. We need a uniform error bound, since $g^*(\boldsymbol{\theta}^{(k)})$ is no longer the conditional expectation of $g(\boldsymbol{\theta}^{(k)})$ if conditioning on the past iterate $\boldsymbol{\theta}^{(k)}$. Although the set $[\theta_{\min}, \theta_{\max}]^2$ has constant dimension, the kernel matrix $\mathbf{K}_n(\boldsymbol{\theta}) \in \mathbb{R}^{n\times n}$ is of high dimension and is determined by $\boldsymbol{\theta}$ in a non-linear way. Our solution is to explore the Taylor's expansion of $\nabla\ell(\boldsymbol{\theta}) - \nabla\ell^*(\boldsymbol{\theta})$, then use truncation and covering arguments.

4.2 Proof of the First Part of Theorem 3.1

Let $\widehat{e}_k = (g(\boldsymbol{\theta}^{(k)}))_2 - (g^*(\boldsymbol{\theta}^{(k)}))_2$. Due to Lemma 3 and Assumption 3.2, we have

$$\begin{aligned}\left(\theta_2^{(k)} - \theta_2^*\right)^2 &= \left(\theta_2^{(k-1)} - \theta_2^*\right)^2 - 2\alpha_k(\theta_2^{(k-1)} - \theta_2^*)(g(\boldsymbol{\theta}^{(k-1)}))_2 \\ &\quad + \alpha_k^2(g(\boldsymbol{\theta}^{(k-1)}))_2^2 \\ &\leq \left(\theta_2^{(k-1)} - \theta_2^*\right)^2(1-\alpha_k\gamma) + \alpha_k^2G^2 \\ &\quad + 2\alpha_k\left(\frac{C\log m}{m} - (\theta_2^{(k-1)} - \theta_2^*)\widehat{e}_{k-1}\right),\end{aligned} \tag{16}$$

where $\gamma = \frac{1}{4\theta_{\max}^2}$. Recall that $\frac{3}{2\gamma} \leq \alpha_1 \leq \frac{2}{\gamma}$, and $\alpha_k = \frac{\alpha_1}{k}$ for all $k \geq 1$. Now we prove the following statement for $k \geq 1$ by induction:

$$\left(\theta_2^{(k)} - \theta_2^*\right)^2 \leq \frac{2\alpha_1^2G^2}{k+1} + \sum_{i=0}^{k-1}\eta_{k,i}\left(\frac{C\log m}{m} - (\theta_2^{(k-1)} - \theta_2^*)\widehat{e}_{k-1}\right), \tag{17}$$

where $\eta_{k,i} = 2\alpha_{i+1}\prod_{j=i+2}^k(1-\alpha_j\gamma)$. When $k = 1$, by (16) and the fact that $1 - \alpha_1\gamma < 0$,

$$\left(\theta_2^{(1)} - \theta_2^*\right)^2 \leq \alpha_1^2G^2 + \eta_{1,0}\left(\frac{C\log m}{m} - (\theta_2^{(0)} - \theta_2^*)\widehat{e}_0\right). \tag{18}$$

Assuming (17) holds for $k = l \geq 1$, then due to (16) and the fact that $1 - \alpha_{l+1}\gamma \geq 0$ for $l \geq 1$, we have

$$\begin{aligned}
&\left(\theta_2^{(l+1)} - \theta_2^*\right)^2 \\
\leq &\left(\frac{2\alpha_1^2 G^2}{l+1} + \sum_{i=0}^{l-1} \eta_{l,i}\left(\frac{C\log m}{m} - (\theta_2^{(i)} - \theta_2^*)\widehat{e}_i\right)\right)(1 - \alpha_{l+1}\gamma) + \alpha_{l+1}^2 G^2 \\
&+ 2\alpha_{l+1}\left(\frac{C\log m}{m} - (\theta_2^{(l)} - \theta_2^*)\widehat{e}_l\right) \\
\leq &\frac{2\alpha_1^2 G^2(l+1-\alpha_1\gamma)}{(l+1)^2} + \frac{\alpha_1^2 G^2}{(l+1)^2} + \sum_{i=0}^{l} \eta_{l+1,i}\left(\frac{C\log m}{m} - (\theta_2^{(i)} - \theta_2^*)\widehat{e}_i\right) \\
\leq &\frac{2\alpha_1^2 G^2}{l+2} + \sum_{i=0}^{l} \eta_{l+1,i}\left(\frac{C\log m}{m} - (\theta_2^{(i)} - \theta_2^*)\widehat{e}_i\right).
\end{aligned} \tag{19}$$

Here the last two lines are due to range of α_1 and the definitions of $\eta_{l,i}$. The next step is to bound $\sum_{i=0}^{K-1} \eta_{K,i}\left(\frac{C\log m}{m} - (\theta_2^{(i)} - \theta_2^*)\widehat{e}_i\right)$. First we have

$$\begin{aligned}
&\left|\sum_{i=0}^{K-1} \eta_{K,i}\left(\frac{C\log m}{m} - (\theta_2^{(i)} - \theta_2^*)\widehat{e}_i\right)\right| \\
\leq &\frac{2\alpha_1}{K}\sum_{i=0}^{K-1}\left|\theta_2^{(i)} - \theta_2^*\right||\widehat{e}_i| + \frac{C\alpha_1\log m}{m} \\
\leq &C\left(\max_{0\leq i\leq K-1}|\widehat{e}_i| + \frac{\log m}{m}\right).
\end{aligned} \tag{20}$$

Note that the distribution of each minibatch $\{\mathbf{X}_{\xi_{k+1}}, \mathbf{y}_{\xi_{k+1}}\}_{i=1}^m$ is the same as sampling m independent $\mathbf{x}_i$ from $\mathbb{P}$, and then sampling $\mathbf{y}_{\xi_{k+1}} \sim \mathcal{N}(0, \mathbf{K}_{\xi_{k+1}}^*)$, thus we can apply the Lemma 4 on each $\widehat{e}_i = g(\boldsymbol{\theta}^{(i)})_2 - g^*(\boldsymbol{\theta}^{(i)})_2$ and take a union bound over $0 \leq i \leq K-1$:

$$\mathbb{P}\left(\max_{0\leq i\leq K-1}|\widehat{e}_i| > Cm^{-\frac{1}{2}+\varepsilon}\right) \leq CK\exp\{-cm^{2\varepsilon}\}, \tag{21}$$

for any $\varepsilon > 0$. Therefore,

$$\left(\theta_2^{(k)} - \theta_2^*\right)^2 \leq \frac{2\alpha_1^2 G^2}{k+1} + Cm^{-\frac{1}{2}+\varepsilon}, \tag{22}$$

with probability at least

$$1 - CK\exp\{-c\min\{\log m, m^{2\varepsilon}\} \geq 1 - CK\exp\{-cm^{2\varepsilon}\}, \tag{23}$$

for any $0 < \varepsilon < C\frac{\log\log m}{\log m}$, when $m > C$ for some $C > 0$ depending on $\theta_{\min}, \theta_{\max}, b$.

4.3 Proof of Theorem 3.2

We start from bounding $\nabla\ell^*(\boldsymbol{\theta}^{(k)})$, the conditional expectation of $\nabla\ell(\boldsymbol{\theta}^{(k)})$ given $\mathbf{x}_1, \ldots, \mathbf{x}_n$, then control the statistical error $\nabla\ell(\boldsymbol{\theta}^{(k)}) - \nabla\ell^*(\boldsymbol{\theta}^{(k)})$. By the definition of $\nabla\ell^*(\boldsymbol{\theta}^{(k)})$, for $1 \leq i \leq 2$,

$$\begin{aligned}\left(\nabla\ell^*(\boldsymbol{\theta}^{(k)})\right)_i =& \frac{1}{2n}\mathrm{tr}\left[\mathbf{K}_n(\boldsymbol{\theta}^{(k)})^{-1}(\mathbf{I}_n - \mathbf{K}_n^*\mathbf{K}_n(\boldsymbol{\theta}^{(k)})^{-1})\frac{\partial\mathbf{K}_n(\boldsymbol{\theta}^{(k)})}{\partial\theta_i^{(k)}}\right] \\ =& \frac{1}{2n}\sum_{j=1}^{n}\frac{(\theta_1^{(k)} - \theta_1^*)\lambda_j^{1+\mathbb{1}_{\{i=1\}}} + (\theta_2^{(k)} - \theta_2^*)\lambda_j^{\mathbb{1}_{\{i=1\}}}}{\left(\theta_1^{(k)}\lambda_j + \theta_2^{(k)}\right)^2},\end{aligned} \tag{24}$$

where λ_j is the jth largest eigenvalue of $\mathbf{K}_{f,n}$. The following lemma provides bounds for $\sum_{j=1}^{n}\frac{\lambda_j^l}{\left(\theta_1^{(k)}\lambda_j+\theta_2^{(k)}\right)^2}$ for all $l = 0, 1, 2$.

Lemma 5 *Under Assumption 3.3, for any $\alpha > 0$, if $n > C$ for $C > 0$ depending on b, then with probability at least $1 - 3n^{-\alpha}$,*

$$\begin{aligned}\text{if } l = 1 \text{ or } 2, \sum_{j=1}^{n}\frac{\lambda_j^l}{(\theta_1\lambda_j + \theta_2)^2} \leq& \frac{2(2+\alpha)}{b\theta_{\min}^2}\log n, \\ \sum_{j=1}^{n}\frac{1}{(\theta_1\lambda_j + \theta_2)^2} \leq& \frac{n}{\theta_{\min}^2},\end{aligned} \tag{25}$$

holds for any $\boldsymbol{\theta} \in [\theta_{\min}, \theta_{\max}]^2$.

We prove Lemma 5 by exploiting the error bounds for eigenvalues of empirical kernel matrices from the population eigenvalues of the kernel operator. A detailed version of Lemma 5 including also lower bounds for $\frac{\lambda_j^l}{(\theta_1\lambda_j+\theta_2)^2}$ is presented in the Appendix, which is a key result for proving Lemma 3.

For any constant $c > 0$, apply Lemma 5 with $\alpha = c$, then (25) holds with probability at least $1 - 3n^{-c}$, if $n > C$ for C depending on b. Combining this result and (24) together implies that

$$\left|\left(\nabla\ell^*(\boldsymbol{\theta}^{(k)})\right)_1\right| \leq \frac{C\log n}{n} \tag{26}$$

where $C > 0$ depends on $\theta_{\min}, \theta_{\max}, b$. Meanwhile,

$$\left|\left(\nabla\ell^*(\boldsymbol{\theta}^{(k)})\right)_2\right| \leq C\left(|\theta_2^{(k)} - \theta_2^*| + \frac{\log n}{n}\right). \tag{27}$$

Thus we have

$$\|\nabla\ell^*(\boldsymbol{\theta}^{(k)})\|_2^2 \leq C\left[\left(\frac{\log n}{n}\right)^2 + (\theta_2^{(k)} - \theta_2^*)^2\right]. \tag{28}$$

For bounding $\nabla\ell(\boldsymbol{\theta}^{(k)}) - \nabla\ell^*(\boldsymbol{\theta}^{(k)})$, we can apply Lemma 4.

By (15) and Theorem 3.1, for any $0 < \varepsilon < C\frac{\log\log m}{\log m}$, if $m > C$, then with probability at least $1 - CK\exp\{-cm^{2\varepsilon}\}$, we have

$$\|\nabla\ell(\boldsymbol{\theta}^{(K)})\|_2^2 \le C\left[\frac{G^2}{K+1} + m^{-\frac{1}{2}+\varepsilon}\right], \tag{29}$$

where $c, C > 0$ depend only on $\theta_{\min}, \theta_{\max}, b$.

5. Practical Considerations for Applying SGD on GP

5.1 Sampling Scheme

As discussed in Section 2, one may consider both uniform and nearby sampling. Although our theoretical guarantees are all derived for uniform sampling, some empirical evidence suggests that sampling nearby points for each minibatch can lead to lower errors for learning the noise variance $\theta_2^* = \sigma_\epsilon^2$ and sometimes improved prediction performance (some comparisons are provided in Section 6.2). Below we highlight the Algorithm for nearby sampling which is a simple extension of Algorithm 1.

```
Algorithm 2: Minibatch SGD with nearby sampling
Input: $\boldsymbol{\theta}^{(0)} \in \mathbb{R}^2$, initial step size $\alpha_1 > 0$.
for $k = 1, 2, \ldots, K$ do
    Sample a data point uniformly from the data pool, and then select its $m-1$
      nearest neighbors, which forms $(\mathbf{X}_{\xi_k}, \mathbf{y}_{\xi_k})$ of size $m$;
    Compute the stochastic gradient $g(\boldsymbol{\theta}^{(k-1)}; \mathbf{X}_{\xi_k^{(2)}}, \mathbf{y}_{\xi_k^{(2)}})$;
    $\alpha_k \leftarrow \frac{\alpha_1}{k}$;
    $\boldsymbol{\theta}^{(k)} \leftarrow \boldsymbol{\theta}^{(k-1)} - \alpha_k g(\boldsymbol{\theta}^{(k-1)}; \mathbf{X}_{\xi_k}, \mathbf{y}_{\xi_k})$;
end for
```

In the following, we present some theoretical insights and numerical evidence to understand why and how nearby sampling helps, in terms of learning θ_2^*. Note that we focus on θ_2^* instead of θ_1^*, since $\theta_2^{(k)} - \theta_2^*$ dominates the convergence of the upper bound for $\|\nabla\ell(\boldsymbol{\theta}^{(k)}\|_2$ at the kth iteration.

In the following, we investigate the following hypothesis: *the approximate "curvature" for $\theta_2^{(k)}$ is improved when the points within each minibatch are closer to each other, so that nearby sampling leads to larger curvature and hence faster convergence.* Recall that we have defined the "approximate" curvature term w.r.t. the noise variance at the $(k+1)$th iteration in (14). As discussed in Section 4, our convergence rate results critically depends on a lower bound for (14).

Numerical evidence: Figure 1 demonstrates an example of the "approximate" curvature term (14) with various minibatch sizes m under the uniform sampling and nearby sampling schemes, where nearby sampling leads to larger curvatures.

The detailed numerical experiments for generating Figure 1 is as follows. We randomly generate a full data pool including $n = 2048$ data points: $x_1, \ldots, x_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 10^2)$. For each minibatch size m, we perform uniform sampling and nearby sampling for 50 replicates; for

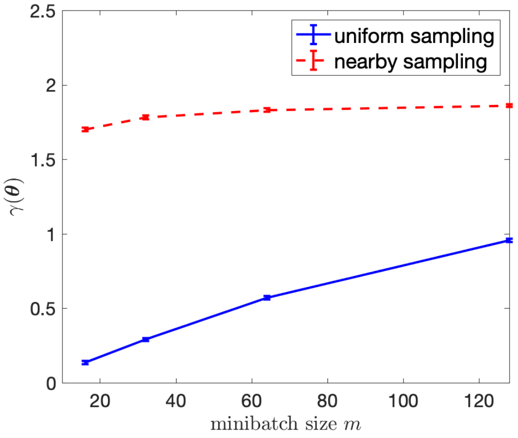

Figure 1: Curvature $\gamma(\boldsymbol{\theta})$, defined in (14) under different minibatch sizes m and sampling schemes. For each minibatch size m and sampling scheme, the mean value of $\gamma(\boldsymbol{\theta})$ over 50 replicates is taken and the standard deviation is marked as the error bar.

uniform sampling the minibatch ξ of size m is sampled from $1, \ldots, n$ uniformly at random; for nearby sampling, the first index ξ_1 in the minibatch ξ is sampled uniformly at random from $1, \ldots, n$, and the rest $m-1$ indices correspond to the nearest neighbors of $\mathbf{x}_{\xi_1}$. Then for each replicate, we calculate $\gamma(\boldsymbol{\theta}) = \frac{1}{2m}\sum_{j=1}^{m}(\theta_1\lambda_j + \theta_2)^{-2}$, where $\boldsymbol{\theta} = (4,1)^\top$ and λ_j is the jth eigenvalue of $\mathbf{K}_{f,\xi}$, the kernel matrix formed by $\mathbf{x}_i, i \in \xi$. The kernel function $k_0(\cdot)$ is set as the RBF kernel with lengthscale $l = 0.5$. We then take the mean of $\gamma(\boldsymbol{\theta})$ over the 50 replicates.

Theoretical insights: Now we provide more theoretical insights into the influence of nearby sampling upon $\gamma(\boldsymbol{\theta})$. For simplicity of analysis, we fix our focus on the $D = 1$ case, and let the kernel k_0 be the RBF kernel: $k_0(x, x') = \exp\{-(x-x')^2/(2l^2)\}$, $x_1, \ldots, x_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$. First note that $\frac{\lambda_j^{(k)}}{m}$ converges to λ_j^* (the jth largest eigenvalue of kernel $k_0(\cdot,\cdot)$) under uniform sampling, when minibatch size m tends to ∞. Define $\widetilde{\gamma}$ by substituting $\lambda_j^{(k)}$ with $m\lambda_j^*$ (λ_j^* is the jth eigenvalue of kernel k) in the definition (14) of $\gamma^{(k)}$:

$$\widetilde{\gamma}(\boldsymbol{\theta}) = \frac{1}{m}\sum_{j=1}^{m}\left(\theta_1\lambda_j^* m + \theta_2\right)^{-2}, \tag{30}$$

then $\widetilde{\gamma}(\boldsymbol{\theta}^{(k)})$ is a reasonable approximation for $\gamma^{(k)}$ when m is large enough under uniform sampling. *If we increase the length scale l under uniform sampling, then it is equivalent to decreasing the distance between points, and thus can serve as an approximation to nearby sampling.*

The following lemma illustrates how the length scale l influences $\widetilde{\gamma}(\boldsymbol{\theta})$:

Lemma 6 *For any $l_0 > 0$, there exists $m_0 > 0$ depending on $\theta_1, \theta_2, \sigma, l_0$ such that as long as $m > m_0$, $\widetilde{\gamma}(\boldsymbol{\theta})$ is an increasing function of $l \geq l_0$.*

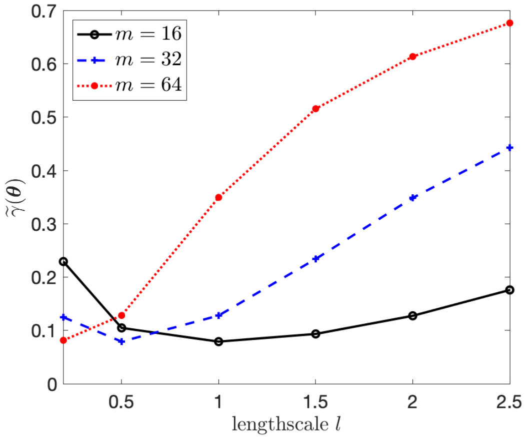

Figure 2: Curvature $\widetilde{\gamma}(\boldsymbol{\theta})$, defined in (30) under different lengthscale parameters l and minibatch sizes m. The other parameters are chosen as $\theta_1 = 4$, $\theta_2 = 1$, and $\sigma = 10$. As an illustration for Lemma 6, we can see that $\widetilde{\gamma}(\boldsymbol{\theta})$ is an increasing function of l when m and l are large enough.

Lemma 6 suggests that, for large enough minibatches, larger length scale leads to faster convergence for $\theta_2^{(k)}$ to $\theta_2^* = \sigma_\epsilon^2$. An example is plotted in Figure 2 as a simple illustration for Lemma 6. Therefore, our hypothesis that “nearby sampling may improve the curvature” is plausible. More numerical support will be provided in Section 6.

5.2 Optimizing other hyperparameters

In practice, we may also need to determine other hyperparameters of the kernel function besides signal variance and noise variance. For example, when considering the RBF kernel $k(\mathbf{x}, \mathbf{x}') = \exp\{-\sum_{j=1}^d \frac{(x_j - x_j')^2}{2l_j^2}\}$, we need to estimate the lengthscale parameters $l_j, 1 \leq j \leq d$; when considering the Matèrn kernel (9), the hyperparameters α and h are also unknown and require estimation. Similar to Algorithm 1 and Algorithm 2, we can update these parameters alongside the variance parameters using minibatch SGD. Our numerical experiments in Section 6 suggest that nearby sampling method may also be a good option for the lengthscale parameters.

5.3 Prediction

Although our main focus in this paper is estimating the hyperparameters the last step when applying GP in real applications is always prediction. Following the parameter estimation process from which we obtain optimal hyperparameters, various strategies can be applied to calculate the predictive mean for $\mathbf{x}_*$ and the predictive covariance between $\mathbf{x}_*$ and $\mathbf{x}_*'$ using the well known predictive equation below

$$\mu_{\text{pred}}(\mathbf{x}_*) = \mathbf{k}_{\mathbf{X}_n\mathbf{x}_*}^\top \mathbf{K}_n^{-1}\mathbf{y}_n, \qquad k_{\text{pred}}(\mathbf{x}_*, \mathbf{x}_*') = k(\mathbf{x}_*, \mathbf{x}_*') - \mathbf{k}_{\mathbf{X}_n\mathbf{x}_*}^\top \mathbf{K}_n^{-1}\mathbf{k}_{\mathbf{X}_n\mathbf{x}_*'}, \tag{31}$$

where $\mathbf{k}_{\mathbf{X}_n\mathbf{x}_*} = (k(\mathbf{x}_1, \mathbf{x}_*) \ldots, k(\mathbf{x}_n, \mathbf{x}_*))^\top$. The main computational cost of the predictive mean and the predictive covariance come from $\mathbf{K}_n^{-1}$. In general, for $n < 10^4$, they can be

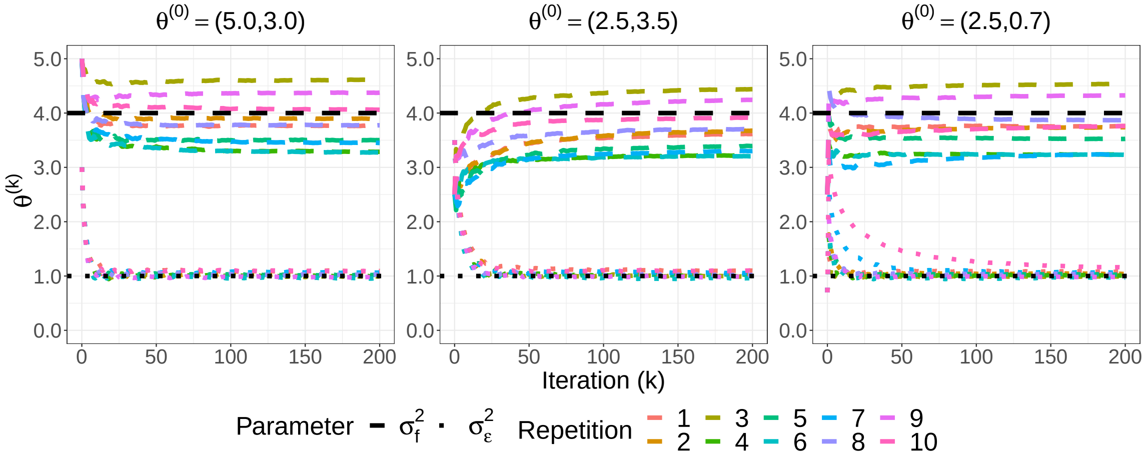

Figure 3: Illustration of the convergence of parameters under uniform sampling. We consider $m = 128$, and demonstrate three cases with varying initial points where initial stepsizes are $\alpha_1 = 9, 9$ and 6, respectively. Lines in black denote the true parameters.

computed via Cholesky decomposition; for $n < 10^5$, preconditioned conjugate gradient (PCG) (Gardner et al., 2018) can be applied for acceleration; for $n < 10^6$, PCG with partitioned kernel (Wang et al., 2019) could provide further speed up, if distributed computational resources are available. Another practical but less ideal strategy when predicting with extremely large n is to follow the same approach as nearby sampling and utilize only $\tilde{n}$ nearest neighbors of $\mathbf{x}_*$ within the observed data to solve (31), where $\tilde{n} < n$ is determined by the available computational resource. Fortunately, prediction is a one-shot process compared to the iterative training process.

6. Numerical Results

6.1 Numerical Illustration of Theory

In this section, we conduct simulation studies to verify our theoretical results.

We consider $n = 1,024$, $\mathbf{x}_i \overset{i.i.d.}{\sim} \mathcal{N}(0, 5^2)$ and $\mathbf{y}_n \sim \mathcal{N}(\mathbf{0}, \sigma_f^2 \mathbf{K}_{f,n} + \sigma_\epsilon^2 \mathbf{I}_n)$, where $\mathbf{K}_{f,n}$ is an RBF kernel matrix with known lengthscale $l = 0.5$. The underlying true parameters are outputscale $\sigma_f^2 = 4$ and noise variance $\sigma_\epsilon^2 = 1$. In each experiment, we perform 25 epochs of minibatch SGD with diminishing step sizes $\alpha_k = \alpha_1/k$. We set scaling factors to $s_1(m) = 3\log m$ for σ_f^2 and $s_2(m) = m$ for σ_ϵ^2. Recall m is the mini-batch size. Each experiment is repeated 10 times with independent data pools for different repetitions.

Fig. 3 shows the convergence of parameters under uniform sampling, varying initializations and step sizes. All, the curves display $O(\frac{1}{K})$ convergence rates which are consistent with our results in Theorem 3.1. Moreover, the locations where the updates of σ_ϵ^2 converges to, are significantly more concentrated around the truth compared to that of σ_f^2, which is consistent with the $O((\log m)^{-\frac{1}{2}})$ statistical error for σ_f^2 and $O(m^{-\frac{1}{2}})$ statistical error for σ_ϵ^2, also stated in Theorem 3.1.

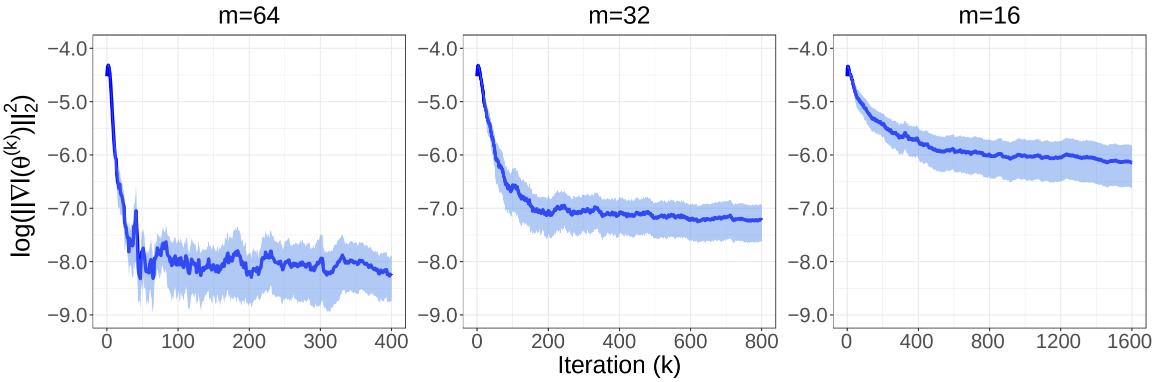

Figure 4: Comparison of the convergence of the full gradient under uniform sampling with varying minibatch sizes. The mean of $\|\nabla \ell(\boldsymbol{\theta}^{(k)})\|_2^2$ is shown in blue and the region within its one standard error over 10 repetitions is shown in light blue, both under log scale. The three experiments share initial point $\boldsymbol{\theta}^{(0)} = (5.0, 3.0)$ and inital step size $\alpha_1 = 9$.

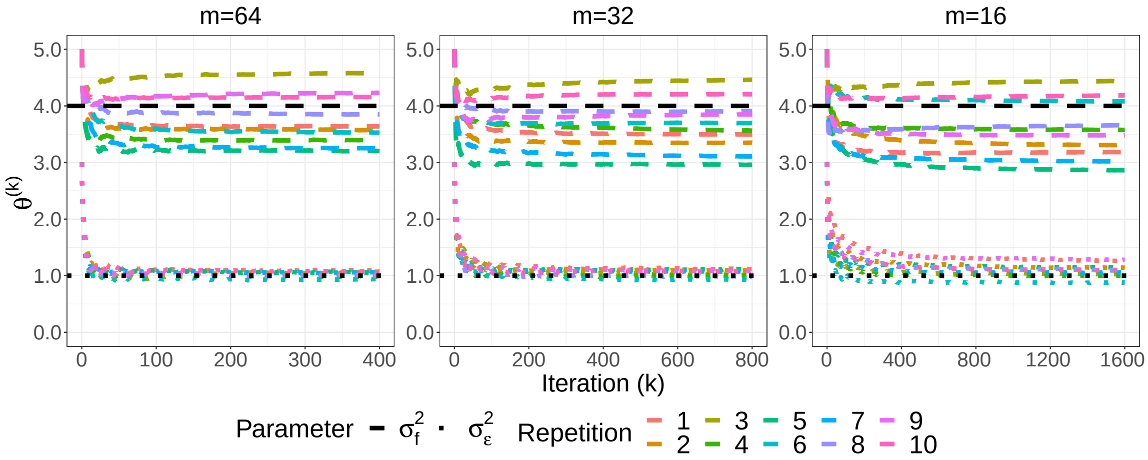

Figure 5: Comparison of the convergence of parameters with varying minibatch sizes. Lines in black denote the true parameters. The three experiments share initial point $\boldsymbol{\theta}^{(0)} = (5.0, 3.0)$ and inital step size $\alpha_1 = 9$.

Fig. 4 displays the effect of minibatch size on the convergence of the full gradient. To start with, the curves flatten slower and converge to larger values as minibatch size decreases, suggesting that a larger minibatch leads to faster convergence of the full gradient, as well as a full gradient with smaller statistical error. In addition, the convergence points of $\log(\|\nabla \ell(\boldsymbol{\theta}^{(k)})\|_2^2)$ scale linearly with minibatch size m, indicating a $O(m^{-\frac{1}{2}})$ statistical error for $\|\nabla \ell(\boldsymbol{\theta}^{(k)})\|_2^2$. The above observations confirm our statements in Theorem 3.2.

We also investigate how minibatch size m influences the convergence of parameters, which is illustrated in Fig. 5. As highlighted in our theory, we find that a larger mini-batch size results in faster convergence and smaller statistical error (more concentrated curves) for the parameters. Here we note that similar results hold for the Matérn kernel and hence we omit the figures.

6.2 Case Studies

In this section, we test our model's performance on publicly available real and simulated datasets. Our benchmarked models are: (i) Exact inference using matrix vector multiplication denoted as EGP (Gardner et al., 2018; Wang et al., 2019), (ii) Vecchia's GP approximation denoted as Vecchia (Guinness, 2018; Katzfuss et al., 2020), (iii) sparse GP regression denoted as SGPR (Titsias, 2009) and (iv) stochastic variational GP denoted as SVGP (Hensman et al., 2013). Our stochastic gradient-based GP approach is denoted as sgGP.

All models are tested on real datasets from the UCI repository (Dua and Graff, 2017) and simulated datasets from the Virtual Library of Simulation Experiments (Surjanovic and Bingham).The real datasets are: Bike, Energy, PM2.5, Protein and Query. The simulated dataset are Levy, Greiwank and Borehole. We also use two other simulated datasets from the Virtual Library Simulation Experiments that represent real-life systems. The OTL circuit models an output transformerless push-pull circuit while the Wing Weight models a light aircraft wing.

Throughout all experiments, we consider constant zero prior mean function and the scaled RBF covariance function with a separate lengthscale for each input dimension. We run Adam to learn the signal variance, noise variance, and also lengthscales, as an extension from our problem set-up stated in Section 2. We conduct 10 independent trials on each dataset. In each trial, we randomly split the dataset into 60% training set and 40% test set. In addition, the training set is normalized to 0 mean and 1 standard deviation, and the test set is scaled accordingly.

During parameter estimation, the hyperparameters and variational parameters are learned through minimizing the negative log marginal likelihood or its surrogate. (i) For sgGP, we consider both uniform and nearest neighbor sampling schemes, where we perform 100 epochs of Adam with minibatch size $m = 16$ and a learning rate of 0.01. (ii) For Vecchia, we order the observed data following the maximum minimum distance (MMD) ordering (Guinness, 2018). MMD ordering works by first selecting a center point, and then sequentially selecting the next point to have maximum minimum distance to all previously selected points. We let each observed response condition on its $m = 16$ nearest neighbors within its predecessors from the ordered set. We also carry out 100 iterations of the Fisher scoring (Guinness, 2021) algorithm. (iii) For other methods, we follow the theoretical recommendations in Burt et al. (2019) and the practical recommendations in Wang et al. (2019). Further, for EGP, we perform 100 iterations of Adam with a learning rate of 0.1. For SGPR, we use $m = 512$ inducing points and carry out 100 iterations of Adam with a learning rate of 0.1. For SVGP, we use $m = 1,024$ inducing points and perform 100 epochs of Adam with a minibatch size of $1,024$ and a learning rate of 0.01. To ensure fairness of comparison, we do not perform any pretraining or fine-tuning, and we let different methods share a common but randomly selected starting point in each trial.

Regarding the prediction of sgGP, we adopt the PCG algorithm in EGP to approximate (31). While for prediction in Vecchia, we order the inputs to be predicted using MMD ordering and append them to the ordered observed inputs. We set the conditioning-set size to $m = 64$.

Parameter estimation of sgGP is coded with R, where RANN package (Arya et al., 2019) is used for finding nearest neighbors. Vecchia is coded using R, where we utilize GpGp package (Guinness, 2018) to find ordered nearest neighbors and implement Fisher's scoring algorithm. The prediction of sgGP, together with EGP, SGPR and SVGP are implemented through GPyTorch (Gardner et al., 2018). Each experiment is performed on a single core from a 2.50GHz CPU workstation released in 2014. For simulated datasets, we manually inject noise to the response. For query dataset, we constrain the learned noise to be at least 0.1 to regularize the ill-conditioned kernel matrix. Due to memory limit, for Borehole, OTL Circuit and Wing Weight datasets, we use PCG algorithm for prediction in sgGP yet using only 60,000 nearest neighbors of each test point.

The results of our experiments are shown in Tables 1 - 5. We start first by analyzing Tables 1 - 3. Table 1 summarizes the root mean squared error of all benchmarked methods, Table 2 shows the negative log-likelihood while Table 3 highlights the accuracy of the learned noise variance $\hat{\sigma}_\epsilon^2$ on simulated datasets where we know the underlying truth σ_ϵ^2. Based on the results, one can derive many insights.

First, we find that sgGP equipped with nearest-neighbor sampling (sgGP (nn)) exhibits the best predictive performance among the various methods on datasets with varying sizes, input dimensions, and noise levels. In addition, its learned noise variance is accurate and superior to all other GP models expect Vecchia which achieves relatively similar performance in noise variance accuracy. Second, while sgGP (uni) can sometimes achieve good performance, it performs poorly in comparison to sgGP (nn), Vecchia, and EGP. This supports our numerical and theoretical evidence of the advantages of nearby sampling in Section 5. Third, Vecchia does perform well overall in terms of prediction performance and accuracy of learned parameters, which is contrary to the finding in Jankowiak and Pleiss (2021). Most likely, the heuristic MMD ordering we adopted offers significant improvement in model approximation over the default coordinate-based ordering (Guinness, 2018). Here, it should be noted that the ordering of observations is crucial for the quality of Vecchia's approximation, and therefore, extensive effort towards dataset-specific tuning may be required, yet there lacks heuristic guidance and theoretical support for datasets of higher dimensions. Fourth, EGP exhibits inferior prediction accuracy compared to sgGP (nn). This highlights the ability of sgGP (nn) to learn parameters that generalize better as both sgGP and EGP aim at exact inference. Yet, it should be noted that while EGP tackles exact inference, it features many approximations within. Finally, we find that SGPR and SVGP both do poorly overall and yield twice the prediction errors of sgGP (nn) on datasets like Levy, PM 2.5 and Query. Also, SGPR and SVGP (especially) tend to exaggerate the noise level (Bauer et al., 2016; Jankowiak et al., 2020), as seen in Table 3. Similar to Wang et al. (2019), this finding sheds light on the ability of exact GPs to significantly benefit from the increase in the number of training points. It is important to note here that the NLL measure is not necessarily convex and hence a smaller NLL may not imply a better estimate of the hyperparameters. This is indeed shown in the comparison between Tables 2 and 3.

Table 4 summarizes the training time of all competing methods. The results exhibit the overwhelming time advantage of sgGP in training, which significantly scales with dataset size. Not only does sgGP achieve better generalization, but it also does that in a fraction of the training time needed for competing methods. This result is again confirmed by our test of the application-driven simulated datasets of size 2×10^6 in Table 5. Remarkably, it takes around 30 minutes to perform parameter estimation for OTL Circuit dataset using a single core with R functions that are not designed for fast execution. In addition, sgGP enjoys superior memory efficiency due to the use of minibatches. These experiments justify that SGD does open up a new data size regime for exploring GPs. Here we note that we are aware that EGP is designed to leverage multiple GPU parallelization; however, much like regular SGD, sgGP can be readily extended to a batch version where the gradient estimate in each update is the average of M gradient estimates from M mini-batches. This allows sgGP to take advantage of parallel computing when the hardware is available.

			RMSE					
Dataset	**Size**	***D***	**sgGP (uni)**	**sgGP (nn)**	**Vecchia**	**EGP**	**SGPR**	**SVGP**
Levy	10,000	4	0.594 ± 0.002	**0.264 ± 0.002**	0.309 ± 0.003	0.313 ± 0.004	0.564 ± 0.010	0.582 ± 0.013
Griewank	10,000	6	0.149 ± 0.003	0.070 ± 0.000	0.081 ± 0.007	**0.063 ± 0.000**	0.132 ± 0.003	0.093 ± 0.005
Bike	17,379	17	0.226 ± 0.002	**0.221 ± 0.002**	0.223 ± 0.002	0.228 ± 0.002	0.276 ± 0.004	0.249 ± 0.009
Energy	19,735	27	**0.711 ± 0.005**	0.786 ± 0.008	0.741 ± 0.004	0.802 ± 0.007	0.843 ± 0.006	0.795 ± 0.005
PM2.5	41,757	15	0.570 ± 0.003	**0.286 ± 0.002**	0.385 ± 0.004	0.287 ± 0.003	0.638 ± 0.005	0.540 ± 0.010
Protein	45,730	9	0.829 ± 0.002	0.659 ± 0.004	**0.597 ± 0.001**	0.696 ± 0.004	0.715 ± 0.003	0.676 ± 0.004
Query	100,000	4	0.128 ± 0.000	0.054 ± 0.004	**0.024 ± 0.000**	—	0.058 ± 0.002	0.049 ± 0.001
Borehole	1,000,000	8	0.173 ± 0.000	**0.172 ± 0.000**	0.174 ± 0.000	—	0.178 ± 0.002	0.173 ± 0.000

Table 1: We summarize the RMSE of sgGP and other GPs on benchmark datasets. Here and elsewhere, we report the averages ± standard errors over 10 dataset splits. Best results are in bold (lower is better). sgGP (unif) utilizes uniform minibatches for training and sgGP (nn) utilizes nearest neighbor minibatches for training. For query and borehole datasets, we are unable to train with EGP due to memory limit.

			NLL					
Dataset	**Size**	***D***	**sgGP (uni)**	**sgGP (nn)**	**Vecchia**	**EGP**	**SGPR**	**SVGP**
Levy	10,000	4	0.891 ± 0.004	**0.034 ± 0.009**	0.205 ± 0.005	0.576 ± 0.018	0.688 ± 0.027	0.868 ± 0.022
Griewank	10,000	6	0.317 ± 0.009	−1.251 ± 0.006	−1.054 ± 0.092	**−1.292 ± 0.005**	−0.350 ± 0.019	−0.430 ± 0.161
Bike	17,379	17	0.238 ± 0.048	−0.038 ± 0.013	−0.103 ± 0.006	−0.107 ± 0.012	**−0.150 ± 0.018**	0.063 ± 0.047
Energy	19,735	27	1.165 ± 0.046	**0.876 ± 0.010**	0.958 ± 0.030	0.884 ± 0.008	0.920 ± 0.013	1.182 ± 0.006
PM2.5	41,757	15	0.864 ± 0.005	**0.144 ± 0.008**	0.244 ± 0.009	0.382 ± 0.005	0.748 ± 0.009	0.819 ± 0.016
Protein	45,730	9	1.241 ± 0.001	0.884 ± 0.009	**0.693 ± 0.003**	0.902 ± 0.006	0.948 ± 0.005	1.026 ± 0.006
Query	100,000	4	−0.493 ± 0.024	−1.436 ± 0.087	**−2.377 ± 0.002**	—	−0.920 ± 0.006	−1.575 ± 0.010
Borehole	1,000,000	8	−0.323 ± 0.004	**−0.335 ± 0.001**	−0.227 ± 0.001	—	−0.299 ± 0.011	**−0.335 ± 0.002**

Table 2: We summarize the NLL of sgGP and other GPs on benchmark datasets. Results follow the experiments in Table 1.

		Learned Parameters					
Dataset	**Parameter**	**sgGP (uni)**	**sgGP (nn)**	**Vecchia**	**EGP**	**SGPR**	**SVGP**
Levy	σ_f	0.886 ± 0.005	0.968 ± 0.006	1.706 ± 0.025	2.113 ± 0.123	1.773 ± 0.049	1.615 ± 0.128
	$\sigma_\epsilon = 0.174$	0.549 ± 0.005	0.200 ± 0.004	0.176 ± 0.002	0.064 ± 0.002	0.265 ± 0.007	0.577 ± 0.012
Griewank	σ_f	0.993 ± 0.003	1.533 ± 0.007	2.473 ± 0.192	2.228 ± 0.065	2.096 ± 0.065	1.657 ± 0.094
	$\sigma_\epsilon = 0.061$	0.010 ± 0.000	0.071 ± 0.001	0.054 ± 0.001	0.042 ± 0.001	0.183 ± 0.005	0.256 ± 0.044
Borehole	σ_f	2.159 ± 0.006	1.091 ± 0.011	5.733 ± 0.093	—	0.684 ± 0.069	0.479 ± 0.017
	$\sigma_\epsilon = 0.172$	0.168 ± 0.001	0.171 ± 0.001	0.172 ± 0.000	—	0.157 ± 0.009	0.241 ± 0.020

Table 3: We summarize the learned signal variance and noise variance of sgGP and other GPs on simulated datasets. Results follow the experiments in Table 1.

			Training Time (min)					
Dataset	**Size**	***D***	**sgGP (uni)**	**sgGP (nn)**	**Vecchia**	**EGP**	**SGPR**	**SVGP**
Levy	10,000	4	0.41 ± 0.02	0.35 ± 0.03	3.15 ± 0.12	11.12 ± 0.71	3.55 ± 0.22	14.74 ± 0.69
Griewank	10,000	6	0.54 ± 0.02	0.48 ± 0.03	4.24 ± 0.09	13.37 ± 1.18	1.76 ± 0.12	14.60 ± 0.65
Bike	17,379	17	1.50 ± 0.12	1.55 ± 0.10	6.70 ± 0.47	29.48 ± 3.96	5.31 ± 2.05	25.26 ± 3.97
Energy	19,735	27	3.03 ± 0.02	2.58 ± 0.17	10.44 ± 1.29	53.25 ± 2.47	5.41 ± 0.73	25.09 ± 5.50
PM2.5	41,757	15	4.90 ± 0.33	4.24 ± 0.29	13.69 ± 0.70	372.88 ± 16.78	13.59 ± 2.30	52.46 ± 10.08
Protein	45,730	9	3.12 ± 0.01	2.63 ± 0.17	13.06 ± 0.12	453.40 ± 21.31	19.55 ± 1.66	55.27 ± 13.09
Query	100,000	4	4.73 ± 0.36	5.03 ± 0.36	30.86 ± 1.69	—	20.73 ± 1.63	124.73 ± 22.25
Borehole	1,000,000	8	54.82 ± 2.69	65.19 ± 2.98	235.74 ± 18.00	—	857.60 ± 76.02	1380.86 ± 11.32

Table 4: We summarize the training time of sgGP and other GPs on benchmark dataset. Results follow the experiments in Table 1.

Dataset	**Size**	***D***	**RMSE**	**Training Time (min)**	**Memory Usage (GB)**
OTL Circuit	2,000,000	6	0.401 ± 0.000	33.43 ± 4.40	0.99 ± 0.00
Wing Weight	2,000,000	10	0.072 ± 0.004	78.78 ± 9.26	1.22 ± 0.00

Table 5: We summarize the results of sgGP (nn) on simulated application-driven datasets. We follow similar setups of the experiments in Table 1.

7. Open Problems

There still exist some open problems that are worth future investigations.

1. The extension to convergence guarantees for learning the lengthscale parameter in RBF kernel is an interesting but extremely challenging problem: our case studies suggest that SGD may still be used for estimating the lengthscale in practice, but the proof for both Lemma 3 and Lemma 4 presents additional challenges if looking at the lengthscale. This is due to it being wrapped within the exponential term as a denominator, which translates to different eigenvectors for $\frac{\partial \mathbf{K}_\xi}{\partial l}$, $\mathbf{K}_\xi$ and $\mathbf{K}_\xi^*$. To see the difficulty for proving Lemma 3, note that the curvature term for estimating l involves

$$\operatorname{tr}\left(\mathbf{K}_\xi(\boldsymbol{\theta}^*)\mathbf{K}_\xi(\boldsymbol{\theta})^{-1}\frac{\partial \mathbf{K}_\xi(\boldsymbol{\theta})}{\partial l}\mathbf{K}_\xi(\boldsymbol{\theta})^{-1}\frac{\partial \mathbf{K}_\xi(\boldsymbol{\theta})}{\partial l}\mathbf{K}_\xi(\boldsymbol{\theta})^{-1}\right) \tag{32}$$

   and cannot be expressed as a function of the eigenvalues of kernel matrices due to their different eigenvectors. It is also very hard to upper bound the statistical error in Lemma 4 due to similar reasons.

2. It would also be interesting to extend theoretical guarantees from $M = 1$ to $M > 1$ without Assumption B.2. The technical challenge for this part is similar to the previous point: without Assumption B.2, K_n and K_n^* can not be simultaneously diagonalized, which hinders the proofs of Lemma 3 and Lemma 4.

3. Another open problem is to establish convergence guarantees for running SGD with nearby sampling and to explore different techniques for nearby sampling upon the choice of the kernel.

8. Conclusion

In this paper, we provide theoretical guarantees for minibatch SGD used to estimate Gaussian process paremeters. In particular, we prove that the iterates of SGD converge to the true hyperparameters and the critical point of the full loss function, with rate $O(\frac{1}{K})$ up to a statistical error term depending on minibatch size. Given the correlation structure of GPs, the challenge lies in the bias of stochastic gradient when taking expectation w.r.t. random sampling. Numerical studies support our theoretical results and show that minibatch SGD has better performance than state-of-the-art methods on various datasets while enjoying huge computational benefits.

References

Ahmad Ajalloeian and Sebastian U Stich. Analysis of sgd with biased gradient estimators. *arXiv preprint arXiv:2008.00051*, 2020. Cited on page 5.

M. Álvarez and N. D. Lawrence. Computationally efficient convolved multiple output gaussian processes. *Journal of Machine Learning Research*, 12(May):1459–1500, 2011. Cited on page 2.

Mauricio Alvarez and Neil D Lawrence. Sparse convolved gaussian processes for multi-output regression. In *Advances in neural information processing systems*, pages 57–64, 2009. Cited on page 4.

Mauricio Álvarez, David Luengo, Michalis Titsias, and Neil D Lawrence. Efficient multioutput gaussian processes through variational inducing kernels. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 25–32, 2010. Cited on page 2.

Sunil Arya, David Mount, Samuel E. Kemp, and Gregory Jefferis. *RANN: Fast Nearest Neighbour Search (Wraps ANN Library) Using L2 Metric*, 2019. URL `https://CRAN.R-project.org/package=RANN`. R package version 2.6.1. Cited on page 22.

Francis Bach. On the equivalence between kernel quadrature rules and random feature expansions. *The Journal of Machine Learning Research*, 18(1):714–751, 2017. Cited on page 10.

Matthias Bauer, Mark van der Wilk, and Carl Edward Rasmussen. Understanding probabilistic sparse gaussian process approximations. In *Advances in neural information processing systems*, pages 1533–1541, 2016. Cited on page 23.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017. Cited on page 4.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018. Cited on page 5.

Mikio L Braun. Accurate error bounds for the eigenvalues of the kernel matrix. *Journal of Machine Learning Research*, 7(Nov):2303–2328, 2006. Cited on pages 11, 14, and 48.

David R Burt, Carl E Rasmussen, and Mark Van Der Wilk. Rates of convergence for sparse variational gaussian process regression. *roceedings of the 36 th International Conference on Machine Learning*, 2019. Cited on pages 4 and 22.

Jie Chen and Ronny Luss. Stochastic gradient descent with biased but consistent gradient estimators. *arXiv preprint arXiv:1807.11880*, 2018. Cited on page 5.

Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247*, 2018. Cited on page 5.

Andreas C Damianou, Michalis K Titsias, and Neil D Lawrence. Variational inference for latent variables and uncertain inputs in gaussian processes. *The Journal of Machine Learning Research*, 17(1):1425–1486, 2016. Cited on pages 2 and 4.

Marc Peter Deisenroth and Jun Wei Ng. Distributed gaussian processes. *ICML*, 2015. Cited on page 2.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`. Cited on page 22.

Roy Frostig, Rong Ge, Sham M Kakade, and Aaron Sidford. Competing with the empirical risk minimizer in a single pass. In *Conference on learning theory*, pages 728–763, 2015. Cited on page 5.

Reinhard Furrer, Marc G Genton, and Douglas Nychka. Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics*, 15(3):502–523, 2006. Cited on page 2.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, pages 7576–7586, 2018. Cited on pages 2, 4, 20, 22, and 23.

Joseph Guinness. Permutation and grouping methods for sharpening gaussian process approximations. *Technometrics*, 60(4):415–429, 2018. Cited on pages 22 and 23.

Joseph Guinness. Gaussian process learning via fisher scoring of vecchia's approximation. *Statistics and Computing*, 31(3):1–8, 2021. Cited on page 22.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234. PMLR, 2016. Cited on page 5.

Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 421–436. JMLR Workshop and Conference Proceedings, 2011. Cited on page 8.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *UAI*, 2013. Cited on pages 2, 3, 4, and 22.

Trong Nghia Hoang, Quang Minh Hoang, and Bryan Kian Hsiang Low. A unifying framework of anytime sparse gaussian process regression models with stochastic variational inference for big data. In *ICML*, pages 569–578, 2015. Cited on page 4.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013. Cited on page 4.

Tito Homem-de Mello. On rates of convergence for stochastic optimization problems under non–independent and identically distributed sampling. *SIAM Journal on Optimization*, 19 (2):524–551, 2008. Cited on page 5.

Martin Jankowiak and Geoff Pleiss. Scalable cross validation losses for gaussian process models. *arXiv preprint arXiv:2105.11535*, 2021. Cited on page 23.

Martin Jankowiak, Geoff Pleiss, and Jacob Gardner. Parametric gaussian process regressors. In *International Conference on Machine Learning*, pages 4702–4712. PMLR, 2020. Cited on page 23.

Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic, and Bharath K Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018. Cited on page 10.

Matthias Katzfuss, Joseph Guinness, Wenlong Gong, and Daniel Zilber. Vecchia approximations of gaussian-process predictions. *Journal of Agricultural, Biological and Environmental Statistics*, 25(3):383–414, 2020. Cited on page 22.

Cari G Kaufman, Mark J Schervish, and Douglas W Nychka. Covariance tapering for likelihood-based estimation in large spatial data sets. *Journal of the American Statistical Association*, 103(484):1545–1555, 2008. Cited on page 2.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016. Cited on page 5.

Juš Kocijan, Roderick Murray-Smith, Carl Edward Rasmussen, and Agathe Girard. Gaussian process model based predictive control. In *Proceedings of the 2004 American control conference*, volume 3, pages 2214–2219. IEEE, 2004. Cited on page 2.

Raed Kontar, Shiyu Zhou, Chaitanya Sankavaram, Xinyu Du, and Yilu Zhang. Nonparametric modeling and prognosis of condition monitoring signals using multivariate gaussian convolution processes. *Technometrics*, 60(4):484–496, 2018. Cited on page 4.

Raed Kontar, Garvesh Raskutti, and Shiyu Zhou. Minimizing negative transfer of knowledge in multivariate gaussian processes: A scalable and regularized approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. Cited on page 2.

Andreas Krause and Cheng S Ong. Contextual gaussian process bandit optimization. In *Advances in neural information processing systems*, pages 2447–2455, 2011. Cited on page 2.

Quoc Le, Tamás Sarlós, and Alex Smola. Fastfood-approximating kernel expansions in loglinear time. In *Proceedings of the international conference on machine learning*, volume 85, 2013. Cited on page 2.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. Cited on page 2.

Ali Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, 36(6):30–44, 2016. Cited on page 2.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009. Cited on pages 5 and 8.

Duc-Trung Nguyen, Maurizio Filippone, and Pietro Michiardi. Exact gaussian process regression with distributed computations. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 1286–1295, 2019. Cited on page 2.

Trung V Nguyen, Edwin V Bonilla, et al. Collaborative multi-output gaussian processes. In *UAI*, pages 643–652, 2014. Cited on page 2.

Peter ZG Qian and CF Jeff Wu. Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments. *Technometrics*, 50(2):192–204, 2008. Cited on page 2.

Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec): 1939–1959, 2005. Cited on page 2.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008. Cited on page 2.

Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. *arXiv preprint arXiv:1109.5647*, 2011. Cited on page 5.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003. Cited on pages 2, 8, and 12.

Yunus Saatçi. *Scalable inference for structured Gaussian process models*. PhD thesis, Citeseer, 2012. Cited on page 4.

Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2006. Cited on page 4.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012. Cited on page 2.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009. Cited on page 2.

Sebastian U Stich and Sai Praneeth Karimireddy. The error-feedback framework: Better rates for sgd with delayed gradients and compressed communication. *arXiv preprint arXiv:1909.05350*, 2019. Cited on page 5.

S. Surjanovic and D. Bingham. Virtual library of simulation experiments: Test functions and datasets. Retrieved May 25, 2020, from `http://www.sfu.ca/~ssurjano`. Cited on page 22.

Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009. Cited on pages 4 and 22.

Volker Tresp. A bayesian committee machine. *Neural computation*, 12(11):2719–2741, 2000. Cited on page 2.

Shashanka Ubaru, Jie Chen, and Yousef Saad. Fast estimation of tr(f(a)) via stochastic lanczos quadrature. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1075–1099, 2017. Cited on page 4.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010. Cited on page 44.

Ke Wang, Geoff Pleiss, Jacob Gardner, Stephen Tyree, Kilian Q Weinberger, and Andrew Gordon Wilson. Exact gaussian processes on a million data points. In *Advances in Neural Information Processing Systems*, pages 14622–14632, 2019. Cited on pages 2, 4, 20, 22, and 23.

Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *International Conference on Machine Learning*, pages 1775–1784, 2015. Cited on page 4.

Andrew G Wilson, Zhiting Hu, Russ R Salakhutdinov, and Eric P Xing. Stochastic variational deep kernel learning. In *Advances in Neural Information Processing Systems*, pages 2586–2594, 2016. Cited on page 2.

Zichao Yang, Andrew Wilson, Alex Smola, and Le Song. A la carte–learning fast kernels. In *Artificial Intelligence and Statistics*, pages 1098–1106, 2015. Cited on page 2.

Xubo Yue and Raed Al Kontar. Why non-myopic bayesian optimization is promising and how far should we look-ahead? a study via rollout. *AISTATS*, 2020. Cited on page 2.

Jing Zhao and Shiliang Sun. Variational dependent multi-output gaussian process dynamical systems. *The Journal of Machine Learning Research*, 17(1):4134–4169, 2016. Cited on pages 2 and 4.

Qiang Zhou, Peter ZG Qian, and Shiyu Zhou. A simple approach to emulation for computer models with qualitative and quantitative factors. *Technometrics*, 53(3):266–273, 2011. Cited on page 2.

Huaiyu Zhu, Christopher KI Williams, Richard Rohwer, and Michal Morciniec. Gaussian regression and optimal finite dimensional linear models. 1997. Cited on page 57.

Appendix A. Table of Notations

Notations	Description
n	number of data points in the full data set
m	number of data points in a minibatch
K	number of iterations of minibatch SGD
G	An upper bound for $\|g(\boldsymbol{\theta}^{(k)})\|_2$, specified in Assumption 3.2
$\sigma_f^2 = \theta_1^*$	true signal variance parameter
$\sigma_\epsilon^2 = \theta_2^*$	true noise variance parameter
$\boldsymbol{\theta}^{(k)}$	output of minibatch SGD at the kth iteration, as an estimate of $\boldsymbol{\theta}^*$
$\mathbf{K}_n(\boldsymbol{\theta}) = \theta_1\mathbf{K}_{f,n} + \theta_2\mathbf{I}_n$	covariance of $\mathbf{y}_n$ given $\mathbf{X}_n$, if the hyperparameter is $\boldsymbol{\theta}$
$\mathbf{K}_\xi(\boldsymbol{\theta})$	submatrix of $\mathbf{K}_n(\boldsymbol{\theta})$ with rows and columns both indexed by ξ
$\mathbf{K}_{f,n}$	kernel matrix evaluated at $\mathbf{X}_n$
$\nabla\ell(\boldsymbol{\theta}; \mathbf{X}_n, \mathbf{y}_n)$ or $\nabla\ell(\boldsymbol{\theta})$	full gradient evaluated at $\boldsymbol{\theta}$ and full data $\mathbf{X}_n$, $\mathbf{y}_n$
$g(\boldsymbol{\theta}; \mathbf{X}_\xi, \mathbf{y}_\xi)$	stochastic gradient evaluated at $\boldsymbol{\theta}$ and minibatch $\mathbf{X}_\xi$, $\mathbf{y}_\xi$
$\alpha_k = \frac{\alpha_1}{k}$	step size at the kth iteration
$g^*(\boldsymbol{\theta}^{(k)}; \mathbf{X}_{\xi_{k+1}})$ or $g^*(\boldsymbol{\theta}^{(k)})$	conditional expectation of $g(\boldsymbol{\theta}^{(k)}; \mathbf{X}_{\xi_{k+1}}, \mathbf{y}_{\xi_{k+1}})$ at the kth iteration given $\mathbf{X}_{\xi_{k+1}}$
$\lambda_j^{(k)}$	the jth largest eigenvalue of $\mathbf{K}_{f,\xi_{k+1}}$
λ_j^*	the jth largest eigenvalue of $\mathbf{K}_{f,n}$

Table 6: Important notations used throughout the paper

Appendix B. Theoretical Guarantees for Section 3.3

Before presenting the theoretical guarantees under this setting, we first provide a formal definition for the considered minibatch SGD algorithm. With sampled indices $\xi \subset [n]$, let the stochastic gradient $g(\boldsymbol{\theta}; \mathbf{X}_\xi, \mathbf{y}_\xi) \in \mathbb{R}^{M+1}$ be defined as follows:

$$(g(\boldsymbol{\theta}; \mathbf{X}_\xi, \mathbf{y}_\xi))_l = \frac{1}{2s_l(m)}\mathrm{tr}\left[(\mathbf{K}_\xi^{-1}(\mathbf{I}_m - \mathbf{y}_\xi\mathbf{y}_\xi^\top\mathbf{K}_\xi^{-1})\frac{\partial\mathbf{K}_\xi}{\partial\theta_l}\right], \quad 1 \le l \le M+1, \tag{33}$$

where $\mathbf{K}_\xi$ is the principle submatrix formed by the rows and columns of $\mathbf{K}_n$ indexed by ξ. In the following we will also let $\mathbf{K}_{f,\xi}^{(l)}$ denote the $m \times m$ block of $\mathbf{K}_{f,n}^{(l)}$ indexed by ξ. Algorithm 3 summarizes the steps of minibatch SGD.

In the following, we present convergence guarantees for Algorithm 3 when kernels exhibit exponential or polynomial eigendecay. The assumptions are similar to the ones presented in Section 3.

Assumption B.1 (Bounded iterates) *Both $\boldsymbol{\theta}^*$ and $\boldsymbol{\theta}^{(k)}$ for $0 \le k \le K$ lie in $[\theta_{\min}, \theta_{\max}]^{M+1}$, where $0 < \theta_{\min} < \theta_{\max}$.*

Assumption B.2 *For any $n > 0$ and sample $\{\mathbf{x}_i\}_{i=1}^n$, the kernel matrices $\mathbf{K}_{f,n}^{(1)}, \dots, \mathbf{K}_{f,n}^{(M)}$ share the same eigenvectors.*

Algorithm 3: Minibatch SGD with uniform sampling when the covariance function is the sum of multiple kernels

```
Input: $\boldsymbol{\theta}^{(0)} \in \mathbb{R}^{M+1}$, initial step size $\alpha_1 > 0$.
for $k = 1, 2, \dots, K$ do
    Randomly sample a subset of indices $\xi_k$ of size $m$;
    Compute the stochastic gradient $g(\boldsymbol{\theta}^{(k-1)}; \mathbf{X}_{\xi_k}, \mathbf{y}_{\xi_k}) \in \mathbb{R}^{M+1}$;
    $\alpha_k \leftarrow \frac{\alpha_1}{k}$;
    $\boldsymbol{\theta}^{(k)} \leftarrow \boldsymbol{\theta}^{(k-1)} - \alpha_k g(\boldsymbol{\theta}^{(k-1)}; \mathbf{X}_{\xi_k}, \mathbf{y}_{\xi_k})$;
end for
```

Remark 7 (Explanation for Assumption B.2) *When extending the theoretical guarantees from $M = 1$ to $M > 1$, we find it extremely challenging without Assumption B.2, which ensures that matrix $\mathbf{K}_n(\boldsymbol{\theta})$ and $\mathbf{K}_n^*$ are simultaneously diagonalizable, and thus facilitates the analysis for the gradient. It remains an open question to establish theoretical results without this assumption. We believe that if the eigenvectors of the kernel matrices are "close" our results should still hold.*

Assumption B.3 (Bounded stochastic gradient) *For all $0 \le k < K$,*

$$\|g(\boldsymbol{\theta}^{(k)}; \mathbf{X}_{\xi_{k+1}}, \mathbf{y}_{\xi_{k+1}})\|_2 \le G$$

for some $G > 0$.

B.1 Kernels with Exponential Eigendecay

Assumption B.4 (Exponential eigendecay) *For $1 \le i \le M$, the eigenvalues of kernel function k_i w.r.t. probability measure $\mathbb{P}$ are $\{C_i e^{-b_i j}\}_{j=0}^{\infty}$, where $0 < b_1 < b_2 < \cdots < b_M$, and $C_i \le 1$ are regarded as constants.*

Theorem B.1 (Convergence of parameter iterates, exponential eigendecay) *Under Assumptions B.1 to B.4, when $m > C$ for some constant $C > 0$, we have the following results under two corresponding conditions on $s_l(m)$:*

1. *If $s_{M+1}(m) = m$, initial step size α_1 satisfies $\frac{3}{2\gamma} \le \alpha_1 \le \frac{2}{\gamma}$ where $\gamma = \frac{1}{4\theta_{\max}^2}$, then for any $0 < \varepsilon < C\frac{\log\log m}{\log m}$, with probability at least $1 - CK\exp\{-cm^{2\varepsilon}\}$,*

$$(\theta_{M+1}^{(K)} - \theta_{M+1}^*)^2 \le \frac{8G^2}{\gamma^2(K+1)} + Cm^{-\frac{1}{2}+\varepsilon}. \tag{34}$$

2. *If in addition to $s_{M+1}(m) = m$, $s_l(m)$ is set as $\tau \log m$ for $1 \le l \le M$ where $\tau > C$, the eigendecay rates $b_2 > 2b_1$ when $M \ge 2$, $\frac{3}{2\gamma} \le \alpha_1 \le \frac{2}{\gamma}$ where γ depends on τ, then for any $0 < \varepsilon < \frac{1}{2}$, with probability at least $1 - CK\exp\{-c(\log m)^{2\varepsilon}\}$,*

$$(\theta_1^{(K)} - \theta_1^*)^2 + (\theta_{M+1}^{(K)} - \theta_{M+1}^*)^2 \le \frac{8G^2}{\gamma^2(K+1)} + C(\log m)^{-\frac{1}{2}+\varepsilon}. \tag{35}$$

Here $c, C > 0$ depend only on $M, \theta_{\min}, \theta_{\max}, b_1, \ldots, b_M$.

Remark B.1 *Here we do not provide estimation error bounds for θ_l^* or $\sigma_{f,l}^2$, for $2 \leq l \leq M$, since they are associated with kernels with faster eigendecay than k_1 and thus are not identifiable. The technical condition on decay rates $b_2 > 2b_1$ is to ensure the convergence of $\theta_1^{(K)}$ although $|\theta_l^{(K)} - \theta_l^*|$ does not converge to 0 for $2 \leq l \leq M$.*

Remark B.2 *For the second case where $s_l(m) = \tau \log m$, τ and γ need to satisfy*

$$\tau > \frac{8b_2(b_2 - b_1)(M+1)^2\theta_{\max}^4}{3b_1^2(b_2 - 2b_1)\theta_{\min}^4}, \tag{36}$$

$$\gamma = \min\left\{\frac{3(b_2 - 2b_1)}{8\tau b_2(b_2 - b_1)(M+1)^2\theta_{\max}^2}, \frac{1}{4\theta_{\max}^2} - \frac{2b_2(b_2 - b_1)(M+1)^2\theta_{\max}^2}{3\tau b_1^2(b_2 - 2b_1)\theta_{\min}^4}\right\}. \tag{37}$$

Theorem B.2 (Convergence of full gradient, exponential eigendecay) *Under Assumptions B.1 to B.4, if $\frac{3}{2\gamma} \leq \alpha_1 \leq \frac{2}{\gamma}$ for $\gamma = \frac{1}{4\theta_{\max}^2}$, $m > C$, $s_{M+1}(m) = m$, then for any $0 < \varepsilon < C\frac{\log\log m}{\log m}$, with probability at least $1 - CK\exp\{-cm^{2\varepsilon}\}$,*

$$\|\nabla \ell(\boldsymbol{\theta}^{(K)})\|_2^2 \leq C\left[\frac{G^2}{K+1} + m^{-\frac{1}{2}+\varepsilon}\right], \tag{38}$$

holds, where $c, C > 0$ depend only on $M, \theta_{\min}, \theta_{\max}, b_1, \ldots, b_M$.

B.2 Kernels with Polynomial Eigendecay

Assumption B.5 (Polynomial eigendecay) *For $1 \leq l \leq M$, the eigenvalues of kernel function k_l w.r.t. probability measure $\mathbb{P}$ are $\{C_l j^{-2b_l}\}_{j=0}^{\infty}$, where $\frac{\sqrt{21}+3}{4} < b_1 < b_2 < \cdots < b_M$, and $C_l \leq 1$ are regarded as constants.*

Theorem B.3 (Convergence of parameter iterates, polynomial eigendecay) *Under Assumptions B.1 to B.3 and Assumption B.5, when $m > C$ for some constant $C > 0$, $s_{M+1}(m) = m$, $\frac{3}{2\gamma} \leq \alpha_1 \leq \frac{2}{\gamma}$ where $\gamma = \frac{1}{8\theta_{\max}^2}$, then for any $\varepsilon \in (\max\{0, f_1(b_1)\}, \frac{1}{2})$, with probability at least $1 - CKm^{-f_2(b_1)[\varepsilon - f_1(b_1)]} - CK\exp\{-cm^{2\varepsilon}\}$,*

$$(\theta_{M+1}^{(K)} - \theta_{M+1}^*)^2 \leq \frac{8G^2}{\gamma^2(K+1)} + Cm^{-\frac{1}{2}+\varepsilon}. \tag{39}$$

Here $c, C > 0$ depend only on $M, \theta_{\min}, \theta_{\max}, b_1, \ldots, b_M$, and $f_1(\cdot)$, $f_2(\cdot)$ are defined as in Theorem 3.3.

Theorem B.4 (Convergence of full gradient, polynomial eigendecay) *Under the same conditions as Theorem B.3, for any $\varepsilon \in (\max\{0, f_1(b_1)\}, \frac{1}{2})$, with probability at least $1 - CK\left(m^{-f_2(b_1)[\varepsilon - f_1(b_1)]} + \exp\{-cm^{2\varepsilon}\}\right)$,*

$$\|\nabla \ell(\boldsymbol{\theta}^{(K)})\|_2^2 \leq C\left[\frac{G^2}{K+1} + m^{-\frac{1}{2}+\varepsilon}\right], \tag{40}$$

holds, where $c, C > 0$ depend only on $M, \theta_{\min}, \theta_{\max}, b_1, \ldots, b_M$, $f_1(\cdot)$ and $f_2(\cdot)$ are defined as in Theorem 3.3.

Appendix C. Detailed Versions of Key Lemmas

First we present the detailed versions of the key lemmas (Lemma 3, 4 and 5) discussed in Section 4, which are useful for proving the second part of Theorem 3.1, Theorem 3.3 to Theorem B.4, and other supporting lemmas. We will focus on the general case where the covariance function is a linear combination of multiple kernels, introduced in Section 3.3. The model introduced in Section 2 can be viewed a special case of this general model, with $M = 1$. Since we are considering the general setting in our proofs, the notations are consistent with the ones introduced in Section 3.3: when $M = 1$, the only kernel function is referred to as $k_1(\cdot,\cdot)$ instead of $k_0(\cdot,\cdot)$, we use b_1 to denote its eigendecay rate instead of b.

Lemma 8 (Strongly convex-like property of $g^*(\boldsymbol{\theta}^{(k)})$, exponential eigendecay) *Under Assumptions B.1 to B.4,*

1. *if $s_{M+1}(m) = m$, $m > C$ for some $C > 0$, then with probability at least $1 - 3MKm^{-c}$, the following claim holds true for $0 \le k < K$:*

$$\langle \widetilde{\boldsymbol{\theta}}^{(k)} - \widetilde{\boldsymbol{\theta}}^*, \widetilde{g}_k^* \rangle \ge \frac{\gamma}{2} \|\widetilde{\boldsymbol{\theta}}^{(k)} - \widetilde{\boldsymbol{\theta}}^*\|_2^2 - \varepsilon, \tag{41}$$

where $\widetilde{\boldsymbol{\theta}}^{(k)} = \theta_{M+1}^{(k)}$, $\widetilde{\boldsymbol{\theta}}^* = \theta_{M+1}^*$, $\widetilde{g}_k^* = (g^*(\boldsymbol{\theta}^{(k)}))_{M+1}$, $\gamma = \frac{1}{4\theta_{\max}^2}$, $\varepsilon = \frac{C\log m}{m}$;

2. *if $M \ge 2$, in addition to $s_{M+1}(m) = m$, we also have $s_i(m) = \tau \log m$ for $1 \le i \le M$, and τ satisfies (36), $b_2 > 2b_1$, then for any $0 < \alpha < \min\{\frac{2b_1+b_2}{2b_1}, \frac{2b_2-4b_1}{14b_1+b_2}\}$, with probability at least $1 - 3MKm^{-\alpha}$, (41) holds for $\widetilde{\boldsymbol{\theta}}^{(k)} = (\theta_1^{(k)}, \theta_{M+1}^{(k)})$, $\widetilde{\boldsymbol{\theta}}^* = (\theta_1^*, \theta_{M+1}^*)$, $\widetilde{g}_k^* = ((g^*(\boldsymbol{\theta}^{(k)}))_1, (g^*(\boldsymbol{\theta}^{(k)}))_{M+1})^\top$,*

$$\gamma = \min\left\{\frac{3(b_2 - 2b_1)}{8\tau b_2(b_2 - b_1)(M+1)^2\theta_{\max}^2}, \frac{1}{4\theta_{\max}^2} - \frac{2b_2(b_2 - b_1)(M+1)^2\theta_{\max}^2}{3\tau b_1^2(b_2 - 2b_1)\theta_{\min}^4}\right\}; \tag{42}$$

and $\varepsilon = C(\alpha + (\log m)^{-1})$;

3. *if $M = 1$, in addition to $s_{M+1}(m) = m$, we also have $s_1(m) = \tau \log m$ where $\tau > \frac{64\theta_{\max}^4}{b_1\theta_{\min}^4}$, then with probability at least $1-2Km^{-c}$, (41) holds for $\widetilde{\boldsymbol{\theta}}^{(k)} = \boldsymbol{\theta}^{(k)}$, $\widetilde{\boldsymbol{\theta}}^* = \boldsymbol{\theta}^*$, $\widetilde{g}_k^* = g^*(\boldsymbol{\theta})$,*

$$\gamma = \min\left\{\frac{1}{32\tau b_1\theta_{\max}^2}, \frac{1}{4\theta_{\max}^2} - \frac{2\theta_{\max}^2}{\tau b_1\theta_{\min}^4}\right\}. \tag{43}$$

and $\varepsilon = C\frac{\log m}{m}$.

Here $C > 0$ depends only on $M, \theta_{\min}, \theta_{\max}, b_1, \ldots, b_M$.

Lemma 9 (Strongly convex-like property of $g^*(\boldsymbol{\theta}^{(k)})$, polynomial eigendecay) *If $s_{M+1}(m) = m$, $m > C$ for some $C > 0$, then for any $0 < \alpha < \frac{8b_1^2-12b_1-6}{4b_1+3}$, with probability at least $1 - MKm^{-\alpha}$, the following claim holds true for $0 \le k < K$:*

$$(g^*(\boldsymbol{\theta}^{(k)}))_{M+1}(\theta_{M+1}^{(k)} - \theta_{M+1}^*) \ge \frac{\gamma}{2}(\theta_{M+1}^{(k)} - \theta_{M+1}^*)^2 - \epsilon, \tag{44}$$

where $\gamma = \frac{1}{8\theta_{\max}^2}$, $\epsilon = Cm^{-\frac{8b_1^2-12b_1-6-\alpha(4b_1+3)}{4b_1(2b_1-1)}}$. Here $C > 0$ depends only on $M, \theta_{\min}, \theta_{\max}, b_1, \ldots, b_M$.

Lemma 8 and Lemma 9 are detailed versions of Lemma 3.

Lemma 10 (Uniform statistical error) *For any $x > 0$, $1 \le i \le M+1$, we have*

$$\mathbb{P}\left(\sup_{\boldsymbol{\theta}\in[\theta_{\min},\theta_{\max}]^{M+1}} \frac{n}{s_i(n)}\left|(\nabla \ell(\boldsymbol{\theta}))_i - (\nabla \ell^*(\boldsymbol{\theta}))_i\right| > Cx\right) \le \delta(x). \tag{45}$$

If Assumption B.4 holds, $s_i(n) = \tau \log n$ for τ satisfying (36)*, $n > C$ for some $C > 0$, then*

$$\delta(x) \le Cn^{-c} + C(\log x)^{2M+2}\exp\{-c\log n \min\{x^2, x\}\}.$$

If Assumption B.4 or B.5 hold, $s_i(n) = n$,

$$\delta(x) \le C(\log x)^{2M+2}\exp\{-cn\min\{x^2, x\}\}.$$

Here $c, C > 0$ only depend on $M, \theta_{\min}, \theta_{\max}, b_1, \ldots, b_M$.

Lemma 10 is a detailed version of Lemma 4, including results for kernels with exponential or polynomial eigendecay.

Lemma 11 (Detailed version of Lemma 5, exponential eigendecay) *Under Assumption B.4, for any $\alpha > 0$, if $n > C$ for $C > 0$ depending on $M, b_1, \ldots, b_M$, then with probability at least $1 - 3Mn^{-\alpha}$,*

$$\begin{aligned}
\text{if } l \wedge l' \le M, \sum_{j=1}^{n} \frac{\lambda_{lj}\lambda_{l'j}}{\left(\sum_{h=1}^{M+1}\theta_h\lambda_{hj}\right)^2} &\le \frac{2(2+\alpha)}{b_1\theta_{\min}^2}\log n,\\
\frac{n - C(\alpha)\log n}{4\theta_{\max}^2} \le \sum_{j=1}^{n} \frac{\lambda_{M+1,j}^2}{\left(\sum_{h=1}^{M+1}\theta_h\lambda_{hj}\right)^2} &\le \frac{n}{\theta_{\min}^2},\\
\sum_{j=1}^{n} \frac{\lambda_{1j}\lambda_{M+1,j}}{\left(\sum_{h=1}^{M+1}\theta_h\lambda_{hj}\right)^2} &\le \frac{5+2\alpha}{7b_1\theta_{\min}^2}\log n,
\end{aligned} \tag{46}$$

holds for any $\boldsymbol{\theta} \in [\theta_{\min}, \theta_{\max}]^{M+1}$, where $C(\alpha) > 0$ depends only on α, b_1. Furthermore,

- *if $M = 1$, then for any $0 < \alpha, \epsilon < 1$, with probability at least $1 - 2n^{-\alpha}$, in addition to* (46) *we have*

$$\sum_{j=1}^{n} \frac{\lambda_{1j}^2}{\left(\sum_{h=1}^{2}\theta_h\lambda_{hj}\right)^2} \ge \frac{\epsilon\log n}{8b\theta_{\max}^2}; \tag{47}$$

- *if $b_2 > 2b_1$ holds, then for any $\frac{2b_1}{b_2} < \epsilon < 1$, $0 < \alpha < \min\left\{\epsilon, \frac{2\epsilon b_2 - 4b_1}{6b_1 + \epsilon b_2}\right\}$, with probability at least $1 - 3Mn^{-\alpha}$, in addition to* (46) *we have*

$$\begin{aligned}
\sum_{j=1}^{n} \frac{\lambda_{1j}^2}{\left(\sum_{h=1}^{M+1}\theta_h\lambda_{hj}\right)^2} &\ge \frac{\epsilon(b_2 - 2b_1)}{2b_1(b_2-b_1)(M+1)^2\theta_{\max}^2}\log n,\\
\text{for } 1 < i \le M,\ l \in S_i,\ \sum_{j=1}^{n} \frac{\lambda_{lj}\lambda_{ij}}{\left(\sum_{h=1}^{M+1}\theta_h\lambda_{hj}\right)^2} &\le \frac{(6b_1 + b_2)\alpha\log n}{2b_1(4b_l - b_2 - 6b_1)\theta_{\min}^2} + \frac{C(\epsilon)}{\theta_{\min}^2};
\end{aligned} \tag{48}$$

as long as $n > C(\epsilon)$, where $C(\epsilon) > 0$ depends on $M, b_1, \ldots, b_M$ and ϵ. Here $S_i = \{1, i, i+1, \ldots, M+1\}$.

Lemma 12 (Detailed version of Lemma 5, polynomial eigendecay) *Under Assumption B.5, for any $0 < \alpha < \frac{8b_1^2 - 12b_1 - 6}{4b_1+3}$, with probability at least $1 - Mn^{-\alpha}$, for any $\boldsymbol{\theta} \in [\theta_{\min}, \theta_{\max}]^M$,*

$$\begin{aligned} \textit{if } l \leq l' \leq M, \sum_{j=1}^n \frac{\lambda_{lj}\lambda_{l'j}}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} &\leq n^{\frac{(2+\alpha)(4b_l+3)}{4b_l(2b_l-1)}} \left(\frac{1}{\theta_{\min}^2} + \frac{a_l(4b_l+3)}{\theta_{\min}^2(4b_l^2 - 6b_l - 3)} \right), \\ \frac{n - M \max_l a_l n^{\frac{(2+\alpha)(4b_1+3)}{4b_1(2b_1-1)}}}{4\theta_{\max}^2} \leq \sum_{j=1}^n \frac{\lambda_{M+1,j}^2}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} &\leq \frac{n}{\theta_{\min}^2}, \end{aligned} \tag{49}$$

where $a_l = 2\sqrt{2C_l} + \sqrt{\frac{2C_l}{2b_l-1}} + \frac{C_l}{2b_l-1}$.

Appendix D. Proofs of Theorem B.1, B.3 and B.4

Proof [Proof of Theorem B.1] First we apply Lemma 8 under both cases of $s_i(m)$: for the first case ($s_{M+1}(m) = m$) discussed in Lemma 8, define $\widetilde{g}(\boldsymbol{\theta}^{(k)}) = (g(\boldsymbol{\theta}^{(k)}))_{M+1}$, and for the second case ($s_i(m) = \tau \log m$ and $s_{M+1}(m) = m$), define $\widetilde{g}(\boldsymbol{\theta}^{(k)}) = ((g(\boldsymbol{\theta}^{(k)}))_1, (g(\boldsymbol{\theta}^{(k)})_{M+1})^\top$. Then let $\widehat{\mathbf{e}}_k = \widetilde{g}(\boldsymbol{\theta}^{(k)}) - \widetilde{g}_k^*$. Due to Lemma 8 and Assumption 3.2, we have

$$\begin{aligned} \|\widetilde{\boldsymbol{\theta}}^{(k)} - \widetilde{\boldsymbol{\theta}}^*\|_2^2 =& \|\widetilde{\boldsymbol{\theta}}^{(k-1)} - \widetilde{\boldsymbol{\theta}}^*\|_2^2 - 2\alpha_k \langle \widetilde{\boldsymbol{\theta}}^{(k-1)} - \widetilde{\boldsymbol{\theta}}^*, \widetilde{g}(\boldsymbol{\theta}^{(k-1)}) \rangle + \alpha_k^2 \|\widetilde{g}(\boldsymbol{\theta}^{(k-1)})\|_2^2 \\ \leq& \|\widetilde{\boldsymbol{\theta}}^{(k-1)} - \widetilde{\boldsymbol{\theta}}^*\|_2^2 (1 - \alpha_k \gamma) + \alpha_k^2 G^2 + 2\alpha_k \left(\varepsilon - \langle \widetilde{\boldsymbol{\theta}}^{(k-1)} - \widetilde{\boldsymbol{\theta}}^*, \widehat{\mathbf{e}}_{k-1} \rangle \right). \end{aligned} \tag{50}$$

Recall that $\frac{3}{2\gamma} \leq \alpha_1 \leq \frac{2}{\gamma}$, and $\alpha_k = \frac{\alpha_1}{k}$ for all $k \geq 1$. Now we prove the following statement for $k \geq 1$ by induction:

$$\|\widetilde{\boldsymbol{\theta}}^{(k)} - \widetilde{\boldsymbol{\theta}}^*\|_2^2 \leq \frac{2\alpha_1^2 G^2}{k+1} + \sum_{i=0}^{k-1} \eta_{k,i} \left(\varepsilon - \langle \widetilde{\boldsymbol{\theta}}^{(i)} - \widetilde{\boldsymbol{\theta}}^*, \widehat{\mathbf{e}}_i \rangle \right), \tag{51}$$

where $\eta_{k,i} = 2\alpha_{i+1} \prod_{j=i+2}^k (1 - \alpha_j \gamma)$. When $k = 1$, by (50) and the fact that $1 - \alpha_1 \gamma < 0$,

$$\|\widetilde{\boldsymbol{\theta}}^{(1)} - \widetilde{\boldsymbol{\theta}}^*\|_2^2 \leq \alpha_1^2 G^2 + \eta_{1,0} \left(\varepsilon - \langle \widetilde{\boldsymbol{\theta}}^{(0)} - \widetilde{\boldsymbol{\theta}}^*, \widehat{\mathbf{e}}_0 \rangle \right). \tag{52}$$

Assuming (51) holds for $k = l \geq 1$, then due to (50) and the fact that $1 - \alpha_{l+1}\gamma \geq 0$ for $l \geq 1$, we have

$$\begin{aligned}
&\|\widetilde{\boldsymbol{\theta}}^{(l+1)} - \widetilde{\boldsymbol{\theta}}^*\|_2^2 \\
\leq &\left(\frac{2\alpha_1^2 G^2}{l+1} + \sum_{i=0}^{l-1} \eta_{l,i} \left(\varepsilon - \langle \widetilde{\boldsymbol{\theta}}^{(i)} - \widetilde{\boldsymbol{\theta}}^*, \widehat{\mathbf{e}}_i \rangle \right) \right) (1 - \alpha_{l+1}\gamma) + \alpha_{l+1}^2 G^2 \\
&+ 2\alpha_{l+1} \left(\varepsilon - \langle \widetilde{\boldsymbol{\theta}}^{(l)} - \widetilde{\boldsymbol{\theta}}^*, \widehat{\mathbf{e}}_l \rangle \right) \\
\leq &\frac{2\alpha_1^2 G^2 (l+1-\alpha_1\gamma)}{(l+1)^2} + \frac{\alpha_1^2 G^2}{(l+1)^2} + \sum_{i=0}^{l} \eta_{l+1,i} \left(\varepsilon - \langle \widetilde{\boldsymbol{\theta}}^{(i)} - \widetilde{\boldsymbol{\theta}}^*, \widehat{\mathbf{e}}_i \rangle \right) \\
\leq &\frac{2\alpha_1^2 G^2}{l+2} + \sum_{i=0}^{l} \eta_{l+1,i} \left(\varepsilon - \langle \widetilde{\boldsymbol{\theta}}^{(i)} - \widetilde{\boldsymbol{\theta}}^*, \widehat{\mathbf{e}}_i \rangle \right).
\end{aligned} \tag{53}$$

Here the last two lines are due to range of α_1 and the definitions of $\eta_{l,i}$. The next step is to bound $\sum_{i=0}^{K-1} \eta_{K,i} \left(\varepsilon - \langle \widetilde{\boldsymbol{\theta}}^{(i)} - \widetilde{\boldsymbol{\theta}}^*, \widehat{\mathbf{e}}_i \rangle \right)$. First we have

$$\begin{aligned}
&\left| \sum_{i=0}^{K-1} \eta_{K,i} \left(\varepsilon - \langle \widetilde{\boldsymbol{\theta}}^{(i)} - \widetilde{\boldsymbol{\theta}}^*, \widehat{\mathbf{e}}_i \rangle \right) \right| \\
\leq &\frac{2\alpha_1}{K} \sum_{i=0}^{K-1} \|\widetilde{\boldsymbol{\theta}}^{(i)} - \widetilde{\boldsymbol{\theta}}^*\|_2 \|\widehat{\mathbf{e}}_i\|_2 + 2\alpha_1\varepsilon \\
\leq &C \left(\max_{0 \leq i \leq K-1} \|\widehat{\mathbf{e}}_i\|_2 + \varepsilon \right).
\end{aligned} \tag{54}$$

Note that the distribution of each minibatch $\{\mathbf{X}_{\xi_{k+1}}, \mathbf{y}_{\xi_{k+1}}\}_{i=1}^m$ is the same as sampling m independent $\mathbf{x}_i$ from $\mathbb{P}$, and then sampling $\mathbf{y}_{\xi_{k+1}} \sim \mathcal{N}(0, \mathbf{K}^*_{\xi_{k+1}})$, thus we can apply Lemma 10 on $\widetilde{g}(\boldsymbol{\theta}^{(k)})$ and $\widetilde{g}_k^*$. Combining Lemma 8, Lemma 10 and (51) leads to the following conclusion.

1. If $s_{M+1}(m) = m$, $m > C$, then for any $0 < \varepsilon < \frac{1}{2}$, with probability at least $1 - CKm^{-c} - CK\exp\{-cm^{2\varepsilon}\}$,

   $$(\theta_{M+1}^{(K)} - \theta_{M+1}^*)^2 \leq \frac{8G^2}{\gamma^2(K+1)} + Cm^{-\frac{1}{2}+\varepsilon}, \tag{55}$$

   where $\gamma = \frac{1}{4\theta_{\max}^2}$. Let $\varepsilon < C\frac{\log\log m}{\log m}$, then $K\exp\{-cm^{2\varepsilon}\} \geq CKm^{-c}$, thus the probability term is $1 - CK\exp\{-cm^{2\varepsilon}\}$.

2. If $M = 1$, $s_1(m) = \tau \log m$, $s_2(m) = m$, $m > C$, then for any $0 < \varepsilon < \frac{1}{2}$, with probability at least

   $$1 - CK\exp\{-c(\log m)^{2\varepsilon}\},$$

   we have

   $$(\theta_1^{(K)} - \theta_1^*)^2 + (\theta_2^{(K)} - \theta_2^*)^2 \leq \frac{8G^2}{\gamma^2(K+1)} + C(\log m)^{-\frac{1}{2}+\varepsilon}, \tag{56}$$

where $\gamma = \min\left\{\frac{1}{32\tau b\theta_{\max}^2}, \frac{1}{4\theta_{\max}^2} - \frac{2\theta_{\max}^2}{\tau b\theta_{\min}^4}\right\}$.

3. If $s_i(m) = \tau\log m$ for $1 \leq i \leq M$, $s_{M+1}(m) = m$, $m > C$, $b_2 > 2b_1$, then for any $0 < \alpha < \min\{\frac{2b_1+b_2}{2b_1}, \frac{2b_2-4b_1}{14b_1+b_2}\}$,$0 < \varepsilon < \frac{1}{2}$, with probability at least

$$1 - CKm^{-\alpha} - CK\exp\{-c(\log m)^{2\varepsilon}\},$$

we have

$$(\theta_1^{(K)} - \theta_1^*)^2 + (\theta_{M+1}^{(K)} - \theta_{M+1}^*)^2 \leq \frac{8G^2}{\gamma^2(K+1)} + C(\log m)^{-\frac{1}{2}+\varepsilon} + C\alpha, \tag{57}$$

where γ is defined in (43). Let $c(\log m)^{-1+2\varepsilon)}\alpha < C(\log m)^{-\frac{1}{2}+\varepsilon}$, then $Km^{-\alpha} \geq \exp\{-c(\log m)^{2\varepsilon}\}$ and thus the probability term can be written as $1 - CK\exp\{-c(\log m)^{2\varepsilon}\}$ and the error bound is $\frac{8G^2}{\gamma^2(K+1)} + C(\log m)^{-\frac{1}{2}+\varepsilon}$.

Here $c, C > 0$ depend only on $M, \theta_{\min}, \theta_{\max}, b_1, \ldots, b_M$. ■

Proof [Proof of Theorem B.3] Define $\widehat{e}_k = (g(\boldsymbol{\theta}^{(k)}))_{M+1} - (g^*(\boldsymbol{\theta}^{(k)}))_{M+1}$. Following similar arguments from the proof of Theorem B.1 and applying Lemma 9, one can show that

$$(\theta_{M+1}^{(k)} - \theta_{M+1}^*)^2 \leq \frac{2\alpha_1^2 G^2}{k+1} + \sum_{i=0}^{k-1}\eta_{k,i}\left(\epsilon - (\theta_{M+1}^{(i)} - \theta_{M+1}^*)\widehat{e}_i\right), \tag{58}$$

where $\eta_{k,i} = 2\alpha_{i+1}\prod_{j=i+2}^{k}(1-\alpha_j\gamma)$, $\gamma = \frac{1}{8\theta_{\max}^2}$ and $\varepsilon = Cm^{-\frac{8b_1^2-12b_1-6-\alpha(4b_1+3)}{4b_1(2b_1-1)}}$. Also note that

$$\begin{aligned}&\left|\sum_{i=0}^{K-1}\eta_{K,i}\left(\epsilon - (\theta_{M+1}^{(i)} - \theta_{M+1}^*)\widehat{e}_i\right)\right| \\ \leq & 2\alpha_1(\theta_{\max} - \theta_{\min})\max_{0\leq i<K}\widehat{e}_i| + 2\alpha_1\epsilon.\end{aligned} \tag{59}$$

Similarly from the proof of Theorem B.2, we can apply Lemma 10 on $(g(\boldsymbol{\theta}^{(k)}))_{M+1}$ and $(g^*(\boldsymbol{\theta}^{(k)}))_{M+1}$. Therefore, combining (58) and Lemma 10 leads to the following result: If $s_{M+1}(m) = m$, $m > C$, then for any $0 < \alpha < \frac{8b_1^2-12b_1-6}{4b_1+3}$, $0 < \varepsilon < \frac{1}{2}$, with probability at least $1 - MKm^{-\alpha} - CK\exp\{-cm^{2\varepsilon}\}$,

$$(\theta_{M+1}^{(K)} - \theta_{M+1}^*)^2 \leq \frac{8G^2}{\gamma^2(K+1)} + Cm^{-\frac{8b_1^2-12b_1-6-\alpha(4b_1+3)}{4b_1(2b_1-1)}} + Cm^{-\frac{1}{2}+\varepsilon}, \tag{60}$$

where $\gamma = \frac{1}{8\theta_{\max}^2}$. Here $c, C > 0$ depend only on $M, \theta_{\min}, \theta_{\max}, b_1, \ldots, b_M$. Let

$$\frac{8b_1^2 - 12b_1 - 6 - \alpha(4b_1+3)}{4b_1(2b_1-1)} = \frac{1}{2} - \varepsilon,$$

we arrive at the final conclusion. ■

Proof [Proof of Theorem B.4] Similarly from the proof of Theorem B.2, we utilize (24) and let λ_{lj} be the jth largest eigenvalue of $\mathbf{K}_{f,n}^{(l)}$, $\lambda_{M+1,j} = 1$. By (24) and Lemma 12, for any $0 < \alpha < \frac{8b_1^2-12b_1-6}{4b_1+3}$, with probability at least $1 - Mn^{-\alpha}$,

$$\left|\left(\nabla \ell^*(\boldsymbol{\theta}^{(k)})\right)_i\right| \leq C n^{-\frac{8b_1^2-12b_1-6-\alpha(4b_1+3)}{4b_1(2b_1-1)}}, \tag{61}$$

for $1 \leq i \leq M$, where $C > 0$ depends on $M, \theta_{\min}, \theta_{\max}, b_1, \ldots, b_M$. Meanwhile,

$$\left|\left(\nabla \ell^*(\boldsymbol{\theta}^{(k)})\right)_{M+1}\right| \leq C\left(|\theta_{M+1}^{(k)} - \theta_{M+1}^*| + n^{-\frac{8b_1^2-12b_1-6-\alpha(4b_1+3)}{4b_1(2b_1-1)}}\right). \tag{62}$$

Thus we have

$$\|\nabla \ell^*(\boldsymbol{\theta}^{(k)})\|_2^2 \leq C\left[n^{-\frac{8b_1^2-12b_1-6-\alpha(4b_1+3)}{2b_1(2b_1-1)}} + (\theta_{M+1}^{(k)} - \theta_{M+1}^*)^2\right]. \tag{63}$$

By (45), Theorem B.3 and Lemma 10, for any $\varepsilon \in (\max\{0, f_1(b_1)\}, \frac{1}{2})$, if $m > C$, then with probability at least $1 - CK\left(m^{-f_2(b_1)[\varepsilon - f_1(b_1)]} + \exp\{-cm^{2\varepsilon}\}\right)$, we have

$$\|\nabla \ell(\boldsymbol{\theta}^{(K)})\|_2^2 \leq C\left[\frac{G^2}{K+1} + m^{-\frac{1}{2}+\varepsilon}\right], \tag{64}$$

where $c, C > 0$ depend only on $M, \theta_{\min}, \theta_{\max}, b_1, \ldots, b_M$. ■

Appendix E. Proofs of Supporting Lemmas

Proof [proof of Lemma 8] Let $\lambda_{lj}^{(k)}$ be the jth eigenvalue of $\mathbf{K}_{f,\xi_{k+1}}^{(l)}$ for $1 \leq l \leq M$, and $\lambda_{M+1,j}^{(k)} = 1$, then by the definition of $g^*(\boldsymbol{\theta}^{(k)})$, we have

$$\begin{aligned}
(g^*(\boldsymbol{\theta}^{(k)}))_1 =& \frac{1}{2s_1(m)} \mathrm{tr}\left[\mathbf{K}_{\xi_{k+1}}(\boldsymbol{\theta}^{(k)})^{-1}\left(\mathbf{I}_m - \mathbf{K}_{\xi_{k+1}}(\boldsymbol{\theta}^*)\mathbf{K}_{\xi_{k+1}}(\boldsymbol{\theta}^{(k)})^{-1}\right)\mathbf{K}_{f,\xi_{k+1}}^{(1)}\right] \\
=& \frac{1}{2s_1(m)} \mathrm{tr}\left[\mathbf{K}_{\xi_{k+1}}(\boldsymbol{\theta}^{(k)})^{-1}\left(\sum_{l=1}^{M}(\theta_l^{(k)} - \theta_l^*)\mathbf{K}_{f,\xi_{k+1}}^{(l)} + (\theta_{M+1}^{(k)} - \theta_{M+1}^*)\mathbf{I}_m\right)\right. \\
& \left.\mathbf{K}_{\xi_{k+1}}(\boldsymbol{\theta}^{(k)})^{-1}\mathbf{K}_{f,\xi_{k+1}}^{(1)}\right] \\
=& \frac{1}{2s_1(m)} \sum_{l=1}^{M+1}(\theta_l^{(k)} - \theta_l^*) \sum_{j=1}^{m} \frac{\lambda_{lj}^{(k)}\lambda_{1j}^{(k)}}{\left(\sum_{l=1}^{M+1}\theta_l^{(k)}\lambda_{lj}^{(k)}\right)^2},
\end{aligned} \tag{65}$$

and

$$\begin{aligned}(g^*(\boldsymbol{\theta}^{(k)}))_{M+1} =&\frac{1}{2m}\mathrm{tr}\left[\mathbf{K}_{\xi_{k+1}}(\boldsymbol{\theta}^{(k)})^{-1}\left(\mathbf{I}_m - \mathbf{K}_{\xi_{k+1}}(\boldsymbol{\theta}^*)\mathbf{K}_{\xi_{k+1}}(\boldsymbol{\theta}^{(k)})^{-1}\right)\right]\\ =&\frac{1}{2m}\sum_{l=1}^{M+1}(\theta_l^{(k)}-\theta_l^*)\sum_{j=1}^{m}\frac{\lambda_{lj}^{(k)}}{\left(\sum_{l=1}^{M+1}\theta_l^{(k)}\lambda_{lj}^{(k)}\right)^2}.\end{aligned} \tag{66}$$

We prove Lemma 8 under two cases separately.

1. $s_i(m)=\tau\log m$ for $1\le i\le M$, $s_{M+1}(m)=m$, $\widetilde{\boldsymbol{\theta}}^{(k)}=(\theta_1^{(k)},\theta_{M+1}^{(k)})^\top$, $\widetilde{\boldsymbol{\theta}}^*=(\theta_1^*,\theta_{M+1}^*)^\top$ and $\widetilde{g}_k^*=((g^*(\boldsymbol{\theta}^{(k)}))_1,(g^*(\boldsymbol{\theta}^{(k)})_{M+1})^\top$.
Under this case, we can write $\langle\widetilde{\boldsymbol{\theta}}^{(k)}-\widetilde{\boldsymbol{\theta}}^*,\widetilde{g}_k^*\rangle$ as

$$\langle\widetilde{\boldsymbol{\theta}}^{(k)}-\widetilde{\boldsymbol{\theta}}^*,\widetilde{g}_k^*\rangle=(\widetilde{\boldsymbol{\theta}}^{(k)}-\widetilde{\boldsymbol{\theta}}^*)^\top\mathbf{A}(\widetilde{\boldsymbol{\theta}}^{(k)}-\widetilde{\boldsymbol{\theta}}^*)+\varepsilon,$$

where each entry A_{ij} of $\mathbf{A}\in\mathbb{R}^{2\times 2}$ is defined as follows:

$$\begin{aligned}A_{11}=&\frac{1}{2\tau\log m}\sum_{j=1}^{m}\frac{\lambda_{1j}^{(k)2}}{\left(\sum_{l=1}^{M+1}\theta_l^{(k)}\lambda_{lj}^{(k)}\right)^2},\\ A_{12}=&\frac{1}{2\tau\log m}\sum_{j=1}^{m}\frac{\lambda_{1j}^{(k)}}{\left(\sum_{l=1}^{M+1}\theta_l^{(k)}\lambda_{lj}^{(k)}\right)^2},\\ A_{21}=&\frac{1}{2m}\sum_{j=1}^{m}\frac{\lambda_{1j}^{(k)}}{\left(\sum_{l=1}^{M+1}\theta_l^{(k)}\lambda_{lj}^{(k)}\right)^2},\\ A_{22}=&\frac{1}{2m}\sum_{j=1}^{m}\frac{1}{\left(\sum_{l=1}^{M+1}\theta_l^{(k)}\lambda_{lj}^{(k)}\right)^2},\end{aligned}$$

and

$$\begin{aligned}\varepsilon=&\frac{1}{2\tau\log m}\sum_{l=2}^{M}(\theta_l^{(k)}-\theta_l^*)(\theta_1^{(k)}-\theta_1^*)\sum_{j=1}^{m}\frac{\lambda_{lj}^{(k)}\lambda_{1j}^{(k)}}{\left(\sum_{l=1}^{M+1}\theta_l^{(k)}\lambda_{lj}^{(k)}\right)^2}\\ &+\frac{1}{2m}\sum_{l=2}^{M}(\theta_l^{(k)}-\theta_l^*)(\theta_{M+1}^{(k)}-\theta_{M+1}^*)\sum_{j=1}^{m}\frac{\lambda_{lj}^{(k)}}{\left(\sum_{l=1}^{M+1}\theta_l^{(k)}\lambda_{lj}^{(k)}\right)^2},\end{aligned} \tag{67}$$

for $M\ge 2$ and $\varepsilon=0$ for $M=1$. Note that the distribution of each minibatch $\mathbf{X}_{\xi_{k+1}}$ can be seen as m independent samples from $\mathbb{P}$, thus we can still apply Lemma 11, but substituting n by m.

- When $M = 1$, apply (47) in Lemma 11 with $\epsilon = \frac{1}{2}$, then for any $0 < \alpha < 1$, with probability at least $1 - 2m^{-\alpha}$, we have

$$\begin{aligned} A_{11} \geq \frac{1}{32\tau b_1 \theta_{\max}^2}, \quad A_{22} \geq \frac{1}{8\theta_{\max}^2}\left(1 - \frac{C\log m}{m}\right), \\ A_{12} \leq \frac{1}{2\tau b_1 \theta_{\min}^2}, \quad A_{21} \leq \frac{\log m}{2b_1\theta_{\min}^2 m}. \end{aligned} \tag{68}$$

Also note that for any $\omega > 0$,

$$\begin{aligned} &(\widetilde{\boldsymbol{\theta}}^{(k)} - \widetilde{\boldsymbol{\theta}}^*)^\top \mathbf{A}(\widetilde{\boldsymbol{\theta}}^{(k)} - \widetilde{\boldsymbol{\theta}}^*) \\ \geq &\left(A_{11} - \frac{(A_{12} + A_{21})\omega}{2}\right)(\theta_1^{(k)} - \theta_1^*)^2 + \left(A_{22} - \frac{(A_{12} + A_{21})}{2\omega}\right)(\theta_2^{(k)} - \theta_2^*)^2 \end{aligned} \tag{69}$$

Let $\omega = \frac{\theta_{\min}^2}{16\theta_{\max}^2}$, then by (68) and (69), one can show that

$$\begin{aligned} &\langle \widetilde{\boldsymbol{\theta}}^{(k)} - \widetilde{\boldsymbol{\theta}}^*, \widetilde{g}_k^* \rangle \\ \geq &\frac{1}{64\tau b\theta_{\max}^2}(\theta_1^{(k)} - \theta_1^*)^2 + \left(\frac{1}{8\theta_{\max}^2} - \frac{4\theta_{\max}^2}{\tau b\theta_{\min}^4}\right)(\theta_2^{(k)} - \theta_2^*)^2 - C\frac{\log m}{m} \\ \geq &\frac{\gamma}{2}\|\widetilde{\boldsymbol{\theta}}^{(k)} - \widetilde{\boldsymbol{\theta}}^*\|_2^2 - C\frac{\log m}{m}, \end{aligned} \tag{70}$$

where

$$\gamma = \min\left\{\frac{1}{32\tau b\theta_{\max}^2}, \frac{1}{4\theta_{\max}^2} - \frac{2\theta_{\max}^2}{\tau b\theta_{\min}^4}\right\}, \tag{71}$$

and $C > 0$ depends on $\theta_{\min}, \theta_{\max}, b_1$. It is guaranteed that $\gamma > 0$ Since we have assumed

$$\tau > \frac{64\theta_{\max}^4}{b\theta_{\min}^4}.$$

Therefore, if $m > C$, for any $0 < \alpha < 1$, with probability $1 - 2m^{-\alpha}$, the following claims holds true:

$$\langle \widetilde{\boldsymbol{\theta}}^{(k)} - \widetilde{\boldsymbol{\theta}}^*, \widetilde{g}_k^* \rangle \geq \frac{\gamma}{2}\|\widetilde{\boldsymbol{\theta}}^{(k)} - \widetilde{\boldsymbol{\theta}}^*\|_2^2 - \varepsilon, \tag{72}$$

where $\varepsilon = C\frac{\log m}{m}$ for some constant $C > 0$ depending on $\theta_{\min}, \theta_{\max}, b_1$.

- When $M \geq 2$ and $b_2 > 2b_1$, apply (48) in Lemma 11 with $\epsilon = \frac{2b_1 + b_2}{2b_2}$, then for any $0 < \alpha < \min\{\frac{2b_1+b_2}{2b_2}, \frac{2b_2-4b_1}{14b_1+b_2}\}$, with probability at least $1 - 3Mm^{-\alpha}$, we have

$$|\varepsilon| \leq CM(\theta_{\max} - \theta_{\min})^2\left(\frac{\alpha + 1/\log m}{\tau\theta_{\min}^2} + \frac{\log m}{m}\right), \tag{73}$$

$$\begin{aligned} A_{11} \geq \frac{(2b_1 + b_2)(b_2 - 2b_1)}{8\tau b_1 b_2 (b_2 - b_1)(M+1)^2\theta_{\max}^2}, \quad A_{22} \geq \frac{1}{8\theta_{\max}^2}\left(1 - \frac{C\log m}{m}\right), \\ A_{12} \leq \frac{1}{2\tau b_1\theta_{\min}^2}, \quad A_{21} \leq \frac{\log m}{2b_1\theta_{\min}^2 m}. \end{aligned} \tag{74}$$

Also note that for any $\omega > 0$,

$$
\begin{aligned}
&(\widetilde{\boldsymbol{\theta}}^{(k)} - \widetilde{\boldsymbol{\theta}}^{*})^\top \mathbf{A} (\widetilde{\boldsymbol{\theta}}^{(k)} - \widetilde{\boldsymbol{\theta}}^{*}) \\
\geq &\left(A_{11} - \frac{(A_{12} + A_{21})\omega}{2} \right) (\theta_1^{(k)} - \theta_1^*)^2 + \left(A_{22} - \frac{(A_{12} + A_{21})}{2\omega} \right) (\theta_{M+1}^{(k)} - \theta_{M+1}^*)^2
\end{aligned} \tag{75}
$$

Let

$$
\omega = \frac{(2b_1 + b_2)(b_2 - 2b_1)\theta_{\min}^2}{4b_2(b_2 - b_1)(M+1)^2\theta_{\max}^2},
$$

then by (73), (74) and (75), one can show that

$$
\begin{aligned}
&\langle \widetilde{\boldsymbol{\theta}}^{(k)} - \widetilde{\boldsymbol{\theta}}^{*}, \widetilde{g}_k^* \rangle \\
\geq &\frac{(2b_1 + b_2)(b_2 - 2b_1)}{16\tau b_1 b_2 (b_2 - b_1)(M+1)^2 \theta_{\max}^2} (\theta_1^{(k)} - \theta_1^*)^2 \\
&+ \left(\frac{1}{8\theta_{\max}^2} - \frac{b_2(b_2 - b_1)(M+1)^2\theta_{\max}^2}{\tau b_1 (2b_1 + b_2)(b_2 - 2b_1)\theta_{\min}^4} \right) (\theta_{M+1}^{(k)} - \theta_{M+1}^*)^2 \\
&- \frac{CM(\theta_{\max} - \theta_{\min})^2}{\tau \theta_{\min}^2} (\alpha + (\log m)^{-1}) \\
&- \frac{C(M+1)^2\theta_{\max}^4(\theta_{\max} - \theta_{\min})^2}{\theta_{\min}^4} \frac{\log m}{m} \\
\geq &\frac{\gamma}{2} \| \widetilde{\boldsymbol{\theta}}^{(k)} - \widetilde{\boldsymbol{\theta}}^{*} \|_2^2 - C(\alpha + (\log m)^{-1}),
\end{aligned} \tag{76}
$$

where

$$
\gamma = \min \left\{ \frac{3(b_2 - 2b_1)}{8\tau b_2 (b_2 - b_1)(M+1)^2 \theta_{\max}^2}, \frac{1}{4\theta_{\max}^2} - \frac{2b_2(b_2 - b_1)(M+1)^2\theta_{\max}^2}{3\tau b_1^2 (b_2 - 2b_1)\theta_{\min}^4} \right\}, \tag{77}
$$

and $C > 0$ depends on $M, \theta_{\min}, \theta_{\max}, b_1, \ldots, b_M$. C does not depend on τ since we have assumed

$$
\tau > \frac{8b_2(b_2 - b_1)(M+1)^2\theta_{\max}^4}{3b_1^2(b_2 - 2b_1)\theta_{\min}^4},
$$

which implies $\gamma > 0$.
Therefore, if $m > C$, for any $0 < \alpha < \min\{\frac{2b_1 + b_2}{2b_2}, \frac{2b_2 - 4b_1}{14b_1 + b_2}\}$, with probability $1 - 3Mm^{-\alpha}$, the following claims holds true:

$$
\langle \widetilde{\boldsymbol{\theta}}^{(k)} - \widetilde{\boldsymbol{\theta}}^{*}, \widetilde{g}_k^* \rangle \geq \frac{\gamma}{2} \| \widetilde{\boldsymbol{\theta}}^{(k)} - \widetilde{\boldsymbol{\theta}}^{*} \|_2^2 - \varepsilon, \tag{78}
$$

where $\varepsilon = C(\alpha + (\log m)^{-1})$ for some constant $C > 0$ depending on $M, \theta_{\min}, \theta_{\max}, b_1, \ldots, b_M$.

2. $s_{M+1}(m) = m$, $\widetilde{\boldsymbol{\theta}}^{(k)} = \theta_{M+1}^{(k)}$, $\widetilde{\boldsymbol{\theta}}^{*} = \theta_{M+1}^*$ and $\widetilde{g}_k^* = (g^*(\boldsymbol{\theta}^{(k)}))_{M+1}$
Under this case, we can apply (46) in Lemma 11. Following similar arguments from the first case, one can show that with probability at least $1 - 3Mm^{-c}$,

$$
\langle \widetilde{\boldsymbol{\theta}}^{(k)} - \widetilde{\boldsymbol{\theta}}^{*}, \widetilde{g}_k^* \rangle \geq \frac{\gamma}{2} (\theta_{M+1}^{(k)} - \theta_{M+1}^*)^2 - \varepsilon, \tag{79}
$$

where $\gamma = \frac{1}{4\theta_{\max}^2}$, $\varepsilon = \frac{C\log m}{m}$, if $m > C$. Here $C > 0$ depends only on $\theta_{\min}, \theta_{\max}, M, b_1, \ldots, b_M$. ■

Proof [proof of Lemma 10] Without loss of generality, we start from bounding $(\nabla\ell(\boldsymbol{\theta}))_i - (\nabla\ell^*(\boldsymbol{\theta}))_i$ for an arbitrary $1 \le i \le M+1$. By the definition of $\nabla\ell(\boldsymbol{\theta})$ and $\nabla\ell^*(\boldsymbol{\theta})$, we have

$$\begin{aligned}
&(\nabla\ell(\boldsymbol{\theta}))_i - (\nabla\ell^*(\boldsymbol{\theta}))_i \\
=& -\frac{1}{2n}\left[\mathbf{y}_n^\top \mathbf{K}_n^{-1}(\boldsymbol{\theta})\mathbf{K}_{f,n}^{(i)}\mathbf{K}_n^{-1}(\boldsymbol{\theta})\mathbf{y}_n - \operatorname{tr}\left(\mathbf{K}_n(\boldsymbol{\theta})^{-1}\mathbf{K}_{f,n}^{(i)}\mathbf{K}_n(\boldsymbol{\theta})^{-1}\mathbf{K}_n^*\right)\right] \\
=& -\left((\mathbf{K}_n^*)^{-\frac{1}{2}}\mathbf{y}_n\right)^\top \mathbf{A}(\boldsymbol{\theta})\left((\mathbf{K}_n^*)^{-\frac{1}{2}}\mathbf{y}_n\right) + \operatorname{tr}(\mathbf{A}(\boldsymbol{\theta})),
\end{aligned} \tag{80}$$

where $\mathbf{A}(\boldsymbol{\theta}) = \frac{1}{2n}\mathbf{K}_n^{*\frac{1}{2}}\mathbf{K}_n^{-1}(\boldsymbol{\theta})\mathbf{K}_{f,n}^{(i)}\mathbf{K}_n^{-1}(\boldsymbol{\theta})\mathbf{K}_n^{*\frac{1}{2}}$. Since we have assumed that the eigenvectors of $\mathbf{K}_{f,n}^{(1)}, \ldots, \mathbf{K}_{f,n}^{(M)}$ are all the same in Assumption B.2, we can write $\mathbf{K}_{f,n}^{(j)} = \mathbf{P}^\top\boldsymbol{\Lambda}_j\mathbf{P}$ for all j, where $\mathbf{P}$ is an orthogonal matrix and $\boldsymbol{\Lambda}_j$ is a diagonal matrix consisting of the eigenvalues of $\mathbf{K}_{f,n}^{(j)}$. Also let $\boldsymbol{\Lambda}_{M+1} = \mathbf{I}_n$, then we have

$$\mathbf{A}(\boldsymbol{\theta}^{(k)}) = \mathbf{P}^\top \frac{1}{2n}\left(\sum_{l=1}^{M+1}\theta_l^*\boldsymbol{\Lambda}_l\right)\left(\sum_{l=1}^{M+1}\theta_l\boldsymbol{\Lambda}_l\right)^{-2}\boldsymbol{\Lambda}_i\mathbf{P}. \tag{81}$$

Let $\mathbf{z}_n = \mathbf{P}(\mathbf{K}_n^*)^{-\frac{1}{2}}\mathbf{y}_n$, and $\boldsymbol{\Lambda}(\boldsymbol{\theta}) = \left(\sum_{l=1}^{M+1}\theta_l^*\boldsymbol{\Lambda}_l\right)\left(\sum_{l=1}^{M+1}\theta_l\boldsymbol{\Lambda}_l\right)^{-2}\boldsymbol{\Lambda}_i$, where θ_l is the lth entry of $\boldsymbol{\theta}$, then our goal is to derive a bound for

$$\sup_{\boldsymbol{\theta}\in[\theta_{\min},\theta_{\max}]^{M+1}} \frac{1}{2n}\left|\mathbf{z}_n^\top\boldsymbol{\Lambda}(\boldsymbol{\theta})\mathbf{z}_n - \operatorname{tr}(\boldsymbol{\Lambda}(\boldsymbol{\theta}))\right|.$$

We claim that there exists an ε-net $\{\boldsymbol{\theta}_\varepsilon^{(1)}, \ldots, \boldsymbol{\theta}_\varepsilon^{(N)}\}$ of $[\theta_{\min}, \theta_{\max}]^{M+1}$ under $\|\cdot\|_\infty$, with size $N = (1 + \frac{(\theta_{\max}-\theta_{\min})}{\varepsilon})^{M+1}$. That is to say, for any $\boldsymbol{\theta} \in [\theta_{\min}, \theta_{\max}]^{M+1}$, $\exists\widetilde{\boldsymbol{\theta}} \in \{\boldsymbol{\theta}_\varepsilon^{(1)}, \ldots, \boldsymbol{\theta}_\varepsilon^{(N)}\}$ such that $\boldsymbol{\Delta} = \boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}$ satisfies$\|\boldsymbol{\Delta}\|_\infty \le \varepsilon$. The following proof of this claim is very similar to the proof of Lemma 5.2 in Vershynin (2010).

Define $\boldsymbol{\theta}_c = (\frac{\theta_{\min}+\theta_{\max}}{2}, \ldots, \frac{\theta_{\min}+\theta_{\max}}{2}) \in \mathbb{R}^{M+1}$, then an alternative way to represent $[\theta_{\min}, \theta_{\max}]^{M+1}$ is $\boldsymbol{\theta}_c + B_\infty(\frac{\theta_{\max}-\theta_{\min}}{2})$. Let $\{\boldsymbol{\theta}_\varepsilon^{(1)}, \ldots, \boldsymbol{\theta}_\varepsilon^{(N)}\}$ be a maximal ε-separated subset of $\boldsymbol{\theta}_c + B_\infty(\frac{\theta_{\max}-\theta_{\min}}{2})$ (not the iterates of the SGD algorithm), which means that it is an ε-net of $\boldsymbol{\theta}_c + B_\infty(\frac{\theta_{\max}-\theta_{\min}}{2})$, and $\forall 1 \le i \ne j \le N$, $\|\boldsymbol{\theta}_\varepsilon^{(i)} - \boldsymbol{\theta}_\varepsilon^{(j)}\|_\infty \ge \varepsilon$. Consider the ℓ_∞ balls with centers $\{\boldsymbol{\theta}_\varepsilon^{(i)}\}_{i=1}^N$ and radius $\frac{\varepsilon}{2}$, then these balls are disjoint and are subsets of $\boldsymbol{\theta}_c + B_\infty(\frac{\theta_{\max}-\theta_{\min}+\varepsilon}{2})$. Thus the sum of volumes of these balls is bounded by that of $\boldsymbol{\theta}_c + B_\infty(\frac{\theta_{\max}-\theta_{\min}+\varepsilon}{2})$, which finishes the proof of

$$N \le \left(1 + \frac{\theta_{\max}-\theta_{\min}}{\varepsilon}\right)^{M+1}, \tag{82}$$

In the following we linearize $\boldsymbol{\Lambda}(\boldsymbol{\theta}) = \boldsymbol{\Lambda}(\widetilde{\boldsymbol{\theta}} + \boldsymbol{\Delta})$ based on the Taylor series expression of each of its diagonal entries, so that the upper bound for $\left|\mathbf{z}_n^\top\boldsymbol{\Lambda}(\boldsymbol{\theta})\mathbf{z}_n - \operatorname{tr}(\boldsymbol{\Lambda}(\boldsymbol{\theta}))\right|$ can be implied by some bounds related to $\widetilde{\boldsymbol{\theta}}$. For any $1 \leq j \leq m$, denote the jth diagonal entry of $\boldsymbol{\Lambda}_l$ by λ_{lj} which is independent of $\boldsymbol{\theta}$, then the jth diagonal entry of $\boldsymbol{\Lambda}(\boldsymbol{\theta})$ can be written as follows:

$$\boldsymbol{\Lambda}_{jj}(\boldsymbol{\theta}) = \frac{\sum_{l=1}^{M+1} \lambda_{lj}\lambda_{ij}\theta_l^*}{\left(\sum_{l=1}^{M+1} \lambda_{lj}\theta_l\right)^2}. \tag{83}$$

Meanwhile, let Δ_l and $\widetilde{\theta}_l$ be the lth entry of $\boldsymbol{\Delta}$ and $\boldsymbol{\theta}$, then one can show that

$$\begin{aligned}
\frac{1}{\left(\sum_{l=1}^{M+1} \lambda_{lj}\theta_l\right)^2} =& \frac{1}{\left(\sum_{l=1}^{M+1} \lambda_{lj}\widetilde{\theta}_l + \sum_{l=1}^{M+1} \lambda_{lj}\Delta_l\right)^2} \\
=& \left(\sum_{l=1}^{M+1} \lambda_{lj}\widetilde{\theta}_l\right)^{-2} \left(1 + \frac{\sum_{l=1}^{M+1} \lambda_{lj}\Delta_l}{\sum_{l=1}^{M+1} \lambda_{lj}\widetilde{\theta}_l}\right)^{-2} \\
=& \left(\sum_{l=1}^{M+1} \lambda_{lj}\widetilde{\theta}_l\right)^{-2} \sum_{h=0}^{H-1} \frac{h+1}{\left(-\sum_{l=1}^{M+1} \lambda_{lj}\widetilde{\theta}_l\right)^h} \left(\sum_{l=1}^{M+1} \lambda_{lj}\Delta_l\right)^h \\
&+ \frac{H+1}{(1+\xi)^{H+2}\left(-\sum_{l=1}^{M+1} \lambda_{lj}\widetilde{\theta}_l\right)^{H+2}} \left(\sum_{l=1}^{M+1} \lambda_{lj}\Delta_l\right)^H \\
=& \left(\sum_{l=1}^{M+1} \lambda_{lj}\widetilde{\theta}_l\right)^{-2} \left(\sum_{h_1+\cdots+h_{M+1}\leq H-1} \alpha^{(j)}_{h_1,\ldots,h_{M+1}} \prod_{l=1}^{M+1} \Delta_l^{h_l} + \mathrm{RES}_H^{(j)}(\boldsymbol{\theta})\right),
\end{aligned} \tag{84}$$

where the third equality holds if $\left|\sum_{l=1}^{M+1} \lambda_{lj}\Delta_l\right| < \sum_{l=1}^{M+1} \lambda_{lj}\widetilde{\theta}_l$, which is implied by $\|\boldsymbol{\Delta}\|_\infty \leq \theta_{\min}$, and we will choose ε small enough to satisfy this. Here ξ lies between 0 and $\frac{\sum_{l=1}^{M+1} \lambda_{lj}\Delta_l}{\sum_{l=1}^{M+1} \lambda_{lj}\widetilde{\theta}_l}$,

$$\begin{aligned}
\alpha^{(j)}_{h_1,\ldots,h_{M+1}} =& \frac{(\sum_{l=1}^{M+1} h_l + 1)! \prod_{l=1}^{M+1} \lambda_{lj}^{h_l}}{h_1! \cdots h_{M+1}! (-\sum_{l=1}^{M+1} \lambda_{lj}\widetilde{\theta}_l)^{\sum_{l=1}^{M+1} h_l}}, \\
\mathrm{RES}_H^{(j)}(\boldsymbol{\theta}) =& \sum_{h_1+\cdots+h_{M+1}=H} \frac{(H+1)! \prod_{l=1}^{M+1} \lambda_{lj}^{h_l}\Delta_l^{h_l}}{h_1! \cdots h_{M+1}! (1+\xi)^{H+2} (-\sum_{l=1}^{M+1} \lambda_{lj}\widetilde{\theta}_l)^H}.
\end{aligned} \tag{85}$$

The quantities above satisfy

$$|\alpha^{(j)}_{h_1,\ldots,h_{M+1}}| \leq \left(\sum_{l=1}^{M+1} h_l + 1\right) \left(\frac{M+1}{\theta_{\min}}\right)^{\sum_{l=1}^{M+1} h_l}, |\mathrm{RES}_H^{(j)}(\boldsymbol{\theta})| \leq (H+1) \left(\frac{\varepsilon(M+1)}{\theta_{\min}}\right)^H, \tag{86}$$

since $\sum_{h_1+\cdots+h_{M+1}=h} \frac{h!}{h_1!\cdots h_{M+1}!} = (M+1)^h$. Define the following diagonal matrices: $\boldsymbol{\Lambda}^{(h_1,\ldots,h_{M+1})}(\widetilde{\boldsymbol{\theta}}), \boldsymbol{\Lambda}^{(H)}(\boldsymbol{\theta}) \in \mathbb{R}^{n\times n}$ are with diagonal entries

$$\boldsymbol{\Lambda}_{jj}^{(h_1,\ldots,h_{M+1})}(\widetilde{\boldsymbol{\theta}}) = \alpha^{(j)}_{h_1,\ldots,h_{M+1}} \boldsymbol{\Lambda}_{jj}(\widetilde{\boldsymbol{\theta}}), \boldsymbol{\Lambda}_{jj}^{(H)}(\boldsymbol{\theta}) = \mathrm{RES}_H^{(j)}(\boldsymbol{\theta}) \boldsymbol{\Lambda}_{jj}(\widetilde{\boldsymbol{\theta}}). \tag{87}$$

Then we can write

$$\boldsymbol{\Lambda}(\boldsymbol{\theta}) = \sum_{h_1+\cdots+h_{M+1}\le H-1} \boldsymbol{\Lambda}^{(h_1,\ldots,h_{M+1})}(\widetilde{\boldsymbol{\theta}}) \prod_{l=1}^{M+1} \Delta_l^{h_l} + \boldsymbol{\Lambda}^{(H)}(\boldsymbol{\theta}),$$

and thus

$$\begin{aligned}
&\left|\mathbf{z}_n^\top \boldsymbol{\Lambda}(\boldsymbol{\theta})\mathbf{z}_n - \operatorname{tr}(\boldsymbol{\Lambda}(\boldsymbol{\theta}))\right| \\
\le & \max_{1\le k\le N} \sum_{h_1+\cdots+h_{M+1}\le H-1} \varepsilon^{\sum_{l=1}^{M+1} h_l} \left|\mathbf{z}_n^\top \boldsymbol{\Lambda}^{(h_1,\ldots,h_{M+1})}(\boldsymbol{\theta}_\varepsilon^{(k)})\mathbf{z}_n - \operatorname{tr}(\boldsymbol{\Lambda}^{(h_1,\ldots,h_{M+1})}(\boldsymbol{\theta}_\varepsilon^{(k)}))\right| \\
&+ \left|\mathbf{z}_n^\top \boldsymbol{\Lambda}^{(H)}(\boldsymbol{\theta})\mathbf{z}_n - \operatorname{tr}(\boldsymbol{\Lambda}^{(H)}(\boldsymbol{\theta}))\right|.
\end{aligned} \tag{88}$$

In order to provide an upper bound for the first term above, we first bound

$$\left|\mathbf{z}_n^\top \boldsymbol{\Lambda}^{(h_1,\ldots,h_{M+1})}(\boldsymbol{\theta}_\varepsilon^{(k)})\mathbf{z}_n - \operatorname{tr}(\boldsymbol{\Lambda}^{(h_1,\ldots,h_{M+1})}(\boldsymbol{\theta}_\varepsilon^{(k)}))\right|$$

for an arbitrary k. First note that for any $1 \le k \le N$,

$$\|\boldsymbol{\Lambda}(\boldsymbol{\theta}_\varepsilon^{(k)})\|_2 = \max_j \frac{\sum_{l=1}^{M+1} \lambda_{lj}\lambda_{ij}\theta_l^*}{\left(\sum_{l=1}^{M+1} \lambda_{lj}\theta_l^{(k)}\right)^2} \le \frac{\theta_{\max}}{\theta_{\min}^2}. \tag{89}$$

While for $\|\boldsymbol{\Lambda}(\boldsymbol{\theta}_\varepsilon^{(k)})\|_F^2$, one can show that

$$\begin{aligned}
\|\boldsymbol{\Lambda}(\boldsymbol{\theta}_\varepsilon^{(k)})\|_F^2 \le & \theta_{\max}^2 \sum_{j=1}^n \frac{(\sum_{l=1}^{M+1} \lambda_{lj}\lambda_{ij})^2}{(\sum_{l=1}^{M+1} \lambda_{lj}\theta_l^{(k)})^4} \\
\le & C \sum_{j=1}^n \frac{\sum_{l=1}^{M+1} \lambda_{lj}\lambda_{ij}}{(\sum_{l=1}^{M+1} \lambda_{lj}\theta_l)^2}.
\end{aligned} \tag{90}$$

Let

$$t_i(n) = \sum_{j=1}^n \frac{\sum_{l=1}^{M+1} \lambda_{lj}\lambda_{ij}}{(\sum_{l=1}^{M+1} \lambda_{lj}\theta_l)^2},$$

then a deterministic bound for $t_i(n)$ is

$$t_i(n) \le Cn. \tag{91}$$

If Assumption B.4 holds, applying Lemma 11 without the condition $b_2 > 2b_1$ leads to

$$t_i(n) \le C \log n \text{ for } 1 \le i \le M, \text{ and } t_{M+1}(n) \le Cn, \tag{92}$$

with probability at least $1 - 3Mn^{-c}$ for any constant $c > 0$, if $n > C$. Here $C > 0$ depends only on $M, b_1, \ldots, b_M, \theta_{\min}, \theta_{\max}$. Therefore, by the definition of $\boldsymbol{\Lambda}^{(h_1,\ldots,h_{M+1})}(\boldsymbol{\theta}_\varepsilon^{(k)})$, for any $\boldsymbol{\theta}_\varepsilon^{(k)}$,

$$\begin{aligned}
\|\boldsymbol{\Lambda}^{(h_1,\ldots,h_{M+1})}(\boldsymbol{\theta}_\varepsilon^{(k)})\|_2 \le & C\left(\sum_{l=1}^{M+1} h_l + 1\right)\left(\frac{M+1}{\theta_{\min}}\right)^{\sum_{l=1}^{M+1} h_l}, \\
\|\boldsymbol{\Lambda}^{(h_1,\ldots,h_{M+1})}(\boldsymbol{\theta}_\varepsilon^{(k)})\|_F^2 \le & C\left(\sum_{l=1}^{M+1} h_l + 1\right)^2\left(\frac{M+1}{\theta_{\min}}\right)^{2\sum_{l=1}^{M+1} h_l} t_i(n).
\end{aligned} \tag{93}$$

Let $\varepsilon = \frac{\theta_{\min}}{e(M+1)H}$, then by applying Hanson-wright's inequality, one can show that with probability at least $1 - 2\exp\{-c\min\{t, \frac{t^2}{t_i(n)}\}\}$,

$$\left|\mathbf{z}_n^\top \mathbf{\Lambda}^{(h_1,\ldots,h_{M+1})}(\boldsymbol{\theta}_\varepsilon^{(k)})\mathbf{z}_n - \mathrm{tr}(\mathbf{\Lambda}^{(h_1,\ldots,h_{M+1})}(\boldsymbol{\theta}_\varepsilon^{(k)}))\right| \le (e\varepsilon)^{-\sum_{l=1}^{M+1} h_l} t, \tag{94}$$

where $c > 0$ depends on $M, \theta_{\min}, \theta_{\max}, b_1, \ldots, b_M$. Meanwhile, the following lemma provides an upper bound for the residual term:

Lemma 13 *$\exists\ c, C > 0$ depending only on $M, \theta_{\min}, \theta_{\max}, b_1, \ldots, b_M$ such that,*

$$\begin{aligned}&\mathbb{P}\left[\sup_{\boldsymbol{\theta}\in[\theta_{\min},\theta_{\max}]^{M+1}} |\mathbf{z}_n^\top \mathbf{\Lambda}^{(H)}(\boldsymbol{\theta})\mathbf{z}_n - \mathrm{tr}(\mathbf{\Lambda}^{(H)}(\boldsymbol{\theta}))| > e^{-H}(t + Ct_i(n))\right] \\ \le& 2\exp\left\{-c\min\left\{\frac{t^2}{t_i(n)}, t\right\}\right\}.\end{aligned} \tag{95}$$

Now we take a union bound for each term in (88), then with probability at least

$$\begin{aligned}&1 - 2\left(\binom{H+M+1}{M+1}N + 1\right)\exp\left\{-c\min\left\{\frac{t^2}{t_i(n)}, t\right\}\right\} \\ \ge& 1 - CH^{2M+2}\exp\left\{-c\min\left\{\frac{t^2}{t_i(n)}, t\right\}\right\},\end{aligned} \tag{96}$$

we have

$$\begin{aligned}&\sup_{\boldsymbol{\theta}\in[\theta_{\min},\theta_{\max}]^{M+1}} \left|\mathbf{z}_n^\top \mathbf{\Lambda}(\boldsymbol{\theta})\mathbf{z}_n - \mathrm{tr}(\mathbf{\Lambda}(\boldsymbol{\theta}))\right| \\ \le& t\left[\sum_{h=0}^{H-1}\binom{h+M+1}{M+1}e^{-h} + e^{-H}\right] + Ce^{-H}t_i(n) \\ \le& C(t + e^{-H}t_i(n)).\end{aligned} \tag{97}$$

If Assumption B.4 holds, $s_i(n) = \tau\log n$ for some τ satisfying (36), we apply the probabilistic bound (92) on $t_i(n)$. For any $x > 0$, let $H = \log\frac{1}{x}$ and $t = x\log n$, then with probability at least

$$1 - Cn^{-c} - C(\log x)^{2(M+1)}\exp\left\{-c\log n\min\left\{x^2, x\right\}\right\},$$

we have

$$\sup_{\boldsymbol{\theta}\in[\theta_{\min},\theta_{\max}]^{M+1}} \left|\mathbf{z}_n^\top \mathbf{\Lambda}(\boldsymbol{\theta})\mathbf{z}_n - \mathrm{tr}(\mathbf{\Lambda}(\boldsymbol{\theta}))\right| \le Cx\log n, \tag{98}$$

which implies

$$\sup_{\boldsymbol{\theta}\in[\theta_{\min},\theta_{\max}]^{M+1}} \frac{n}{s_i(n)}\left|(\nabla\ell(\boldsymbol{\theta}))_i - (\nabla\ell^*(\boldsymbol{\theta}))_i\right| \le Cx. \tag{99}$$

Otherwise, if $s_i(n) = n$, we apply the deterministic bound (91) on $t_i(n)$. For any $x > 0$, let $H = \log\frac{1}{x}$ and $t = xn$, then with probability at least

$$1 - C(\log x)^{2(M+1)}\exp\left\{-cn\min\left\{x^2, x\right\}\right\},$$

we have

$$\sup_{\boldsymbol{\theta}\in[\theta_{\min},\theta_{\max}]^{M+1}} \left|\mathbf{z}_n^\top \boldsymbol{\Lambda}(\boldsymbol{\theta})\mathbf{z}_n - \mathrm{tr}(\boldsymbol{\Lambda}(\boldsymbol{\theta}))\right| \leq Cxn, \tag{100}$$

which implies

$$\frac{n}{s_i(n)} \left|(\nabla \ell(\boldsymbol{\theta}))_i - (\nabla \ell^*(\boldsymbol{\theta}))_i\right| \leq Cx. \tag{101}$$

■

Proof [proof of Lemma 11] In order to prove Lemma 11, we need to derive upper and lower bounds for $\lambda_{lj}, 1 \leq l \leq M$ w.h.p. First we restate Theorem 1 and Theorem 4 in Braun (2006) on the bounds for λ_{lj} in the following:

Lemma 14 *Let k be a Mercer kernel on a probability space $\mathcal{X}$ with probability measure $\mathbb{P}$, satisfying $k(x,x) \leq 1$ for all $x \in \mathcal{X}$, with eigenvalues $\{\lambda_i^*\}_{i=1}^\infty$. Let $\mathbf{K}_{f,n} \in \mathbb{R}^{n\times n}$ be the empirical kernel matrix evaluated on data $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ i.i.d. sampled from $\mathbb{P}$, then the eigenvalues $\lambda_i(\mathbf{K}_{f,n})$ satisfies the following bound for $1 \leq j, r \leq n$:*

$$\left|\frac{\lambda_j(\mathbf{K}_{f,n})}{n} - \lambda_j^*\right| \leq \lambda_j^* C(r,n) + E(r,n),$$

and for any $1 \leq r \leq n$, there are two bounds for $C(r,n), E(r,n)$:

(i) With probability at least $1-\delta$,

$$\begin{aligned} C(r,n) <& r\sqrt{\frac{2}{n\lambda_r^*}\log\frac{2r(r+1)}{\delta}} + \frac{4r}{3n\lambda_r^*}\log\frac{2r(r+1)}{\delta}, \\ E(r,n) <& \lambda_r^* + \sum_{i=r+1}^\infty \lambda_i^* + \sqrt{\frac{2\sum_{i=r+1}^\infty \lambda_i^*}{n}\log\frac{2}{\delta}} + \frac{2}{3n}\log\frac{2}{\delta}; \end{aligned} \tag{102}$$

(ii) With probability at least $1-\delta$,

$$C(r,n) < r\sqrt{\frac{r(r+1)}{n\delta\lambda_r^*}}, \quad E(r,n) < \lambda_r^* + \sum_{i=r+1}^\infty \lambda_i^* + \sqrt{\frac{2\sum_{i=r+1}^\infty \lambda_i^*}{n\delta}}. \tag{103}$$

We consider two different upper bounds for λ_{lj} that could be useful in later arguments. First we apply Lemma 14 on $\mathbf{K}_{f,n}^{(l)}$. In particular, plug $r = j$ for each $1 \leq j \leq n$ into (103) and let $\delta = n^{-(1+\alpha)}$ for some $\alpha > 0$. Then with probability at least $1 - n^{-\alpha}$, for all $1 \leq j \leq n$,

$$\begin{aligned} 1 + C(j,n) <& C_l^{-\frac{1}{2}} j^2 n^{\frac{\alpha}{2}} e^{\frac{b_l j}{2}} \sqrt{\frac{j+1}{j}} + 1 < Cj^2 n^{\frac{\alpha}{2}} e^{\frac{b_l j}{2}}, \\ E(r,n) <& \frac{C_l e^{-b_l j}}{1 - e^{-b_l}} + \sqrt{\frac{2C_l e^{-b_l}}{1-e^{-b_l}}} e^{-\frac{b_l j}{2}} n^{\frac{\alpha}{2}}. \end{aligned}$$

Thus we have

$$\begin{aligned}\lambda_{lj} \leq & \left(Cj^2 + \frac{C_l n^{-\frac{\alpha}{2}} e^{-\frac{b_l}{2}j}}{1-e^{-b_l}} + \sqrt{\frac{2C_l e^{-b_l}}{1-e^{-b_l}}} \right) n^{1+\frac{\alpha}{2}} e^{-\frac{b_l j}{2}} \\ \leq & C(\eta) n^{1+\frac{\alpha}{2}} e^{-\frac{b_l j}{2\eta}},\end{aligned} \tag{104}$$

where the last line holds for any $\eta > 1$, and $C(\eta) > 0$ depends on b_l, η. We will specify η later to suit our needs.

The second upper bound for λ_{lj} requires applying (103) with $r = \frac{1+\alpha}{b_l} \log n$, and $\delta = n^{-\alpha}$. Then with probability at least $1 - n^{-\alpha}$,

$$1 + C(r,n) < C_l^{-\frac{1}{2}} \sqrt{r^3(r+1)n^{2\alpha}} + 1 \leq C(\log n)^2 n^{\alpha},$$

$$E(r,n) < \frac{n^{-(1+\alpha)}}{1-e^{-b_l}} + \sqrt{\frac{2e^{-b_l} n^{-(1+\alpha)}}{(1-e^{-b_l})n^{(1-\alpha)}}} \leq \frac{C}{n},$$

where C depends on b_l. Thus $\lambda_{lj} \leq C(\log n)^2 n^{1+\alpha} e^{-b_l j} + C$. Thus for any $\alpha > 0$, with probability at least $1 - 2Mn^{-\alpha}$,

$$\lambda_{lj} \leq \min \left\{ C(\eta) n^{1+\frac{\alpha}{2}} e^{-\frac{b_l j}{2\eta}}, C(\log n)^2 n^{1+\alpha} e^{-b_l j} + C \right\}, \tag{105}$$

holds for $\eta > 1$, $1 \leq l \leq M$, $1 \leq j \leq n$, where $C > 0$ depends on $b_1, \dots, b_M$, $C(\eta)$ depends on $b_1, \dots, b_M, \eta$.

While for lower bounding λ_{lj}, we apply (102) in Lemma 14 with $r = \frac{\epsilon}{b_l} \log n$ for some $0 < \epsilon < 1$, and $\delta = n^{-\alpha}$ for some $0 < \alpha < 1$. Then when $n > C(\epsilon)$ for some constant $C(\epsilon) > 0$ depending on b_l, ϵ, with probability at least $1 - n^{-\alpha}$,

$$\begin{aligned} C(r,n) <& r\sqrt{\frac{2\log\left[2r(r+1)n^{\alpha}\right]}{C_l n^{1-\epsilon}}} + \frac{4r \log\left[2r(r+1)n^{\alpha}\right]}{3C_l n^{1-\epsilon}} < \frac{1}{2}, \\ E(r,n) <& \frac{C_l}{(1-e^{-b_l})n^{\epsilon}} + \sqrt{\frac{2C_l e^{-b_l}}{1-e^{-b_l}}} \sqrt{\frac{\log 2n^{\alpha}}{n^{1+\epsilon}}} + \frac{2}{3n} \log 2n^{\alpha} < Cn^{-\epsilon}, \end{aligned}$$

thus $\lambda_{lj} \geq \frac{C_l}{2} n e^{-b_l j} - Cn^{1-\epsilon}$ for $C > 0$ depending on b_l.

Therefore, for any $0 < \epsilon, \alpha < 1$, if $n > C(\epsilon)$ for $C(\epsilon) > 0$ depending on $b_1, \dots, b_M, \epsilon$, then with probability at least $1 - Mn^{-\alpha}$,

$$\lambda_{lj} \geq \frac{C_l}{2} n e^{-b_l j} - Cn^{1-\epsilon}, \tag{106}$$

holds for $1 \leq l \leq M$, $1 \leq j \leq n$, where $C > 0$ depends on $b_1, \dots, b_M$. Now we are ready to prove the bounds for $\sum_{j=1}^{n} \frac{\lambda_{lj}\lambda_{l'j}}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2}$ for $1 \leq l, l' \leq M+1$.

1. $l = l' = 1$

   First we derive an upper bound. Let $\eta = \frac{3}{2}$ in (105), then we have

$$\begin{aligned}\sum_{j=1}^{n} \frac{\lambda_{1j}^2}{\left(\sum_{h=1}^{M+1}\theta_h\lambda_{hj}\right)^2} \le& \frac{|\{n^{1+\frac{\alpha}{2}}e^{-\frac{b_1 j}{3}} > 1\}|}{\theta_{\min}^2} + \sum_{n^{1+\frac{\alpha}{2}}e^{-\frac{b_1 j}{3}} \le 1} \frac{\lambda_{1j}^2}{\theta_{\min}^2} \\ \le& \frac{|\{n^{1+\frac{\alpha}{2}}e^{-\frac{b_1 j}{3}} > 1\}|}{\theta_{\min}^2} + \frac{Cn^{2+\alpha}}{\theta_{\min}^2} \sum_{e^{-\frac{2b_1 j}{3}} \le n^{-2-\alpha}} e^{-\frac{2b_1}{3}j}.\end{aligned} \tag{107}$$

   Since

$$n^{1+\frac{\alpha}{2}}e^{-\frac{b_1 j}{3}} > 1 \Rightarrow j < \frac{6+3\alpha}{2b_1}\log n, \tag{108}$$

   one can show that

$$\sum_{j=1}^{n} \frac{\lambda_{1j}^2}{\left(\sum_{h=1}^{M+1}\theta_h\lambda_{hj}\right)^2} \le \frac{6+3\alpha}{2b_1\theta_{\min}^2}\log n + \frac{C}{\theta_{\min}^2} \le \frac{4+2\alpha}{b_1\theta_{\min}^2}\log n, \tag{109}$$

   when $n > C$ for C depending on b_1. In terms of the lower bound, we discuss the proof for two cases separately:

   - When $M = 1$, first note that

$$\begin{aligned}\sum_{j=1}^{n} \frac{\lambda_{1j}^2}{\left(\sum_{h=1}^{2}\theta_h\lambda_{hj}\right)^2} \ge& \sum_{j=1}^{n} \frac{\lambda_{1j}^2}{4\theta_{\max}^2 \max_h \lambda_{hj}^2} \\ \ge& \frac{|\{j : \lambda_{1j} = \max_h \lambda_{hj}\}|}{4\theta_{\max}^2}.\end{aligned} \tag{110}$$

     Due to (106), we have

$$\begin{aligned}\lambda_{1j} = \max_h \lambda_{hj} \Leftarrow& \frac{C_1}{2}ne^{-b_1 j} \ge Cn^{1-\epsilon} + C \\ \Leftarrow& e^{-b_1 j} \ge Cn^{-\epsilon} \\ \Leftarrow& j \le \frac{\epsilon}{b_1}\log n - C,\end{aligned} \tag{111}$$

     when $n > C$ for some $C > 0$ depending on b, which implies

$$\left|\{j : \lambda_{1j} = \max_h \lambda_{hj}\}\right| \ge \frac{\epsilon}{b_1}\log n - C \ge \frac{\epsilon}{2b_1}\log n,$$

     if $n > C$. Thus we have

$$\sum_{j=1}^{n} \frac{\lambda_{1j}^2}{\left(\sum_{h=1}^{2}\theta_h\lambda_{hj}\right)^2} \ge \frac{\epsilon\log n}{8b_1\theta_{\max}^2},$$

- When $M \geq 2$ and $b_2 > 2b_1$, first note that

$$\begin{aligned}\sum_{j=1}^{n} \frac{\lambda_{1j}^2}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \geq & \sum_{j=1}^{n} \frac{\lambda_{1j}^2}{(M+1)^2 \theta_{\max}^2 \max_h \lambda_{hj}^2} \\ \geq & \frac{|\{j : \lambda_{1j} = \max_h \lambda_{hj}\}|}{(M+1)^2 \theta_{\max}^2}.\end{aligned} \tag{112}$$

Due to (106), we have

$$\begin{aligned}\lambda_{1j} = \max_h \lambda_{hj} \Leftarrow & \frac{C_1}{2} n e^{-b_1 j} \geq C n^{1-\epsilon} + C(\log n)^2 n^{1+\alpha} e^{-b_2 j} + C \\ \Leftarrow & e^{-b_1 j} \geq C n^{-\epsilon} + C(\log n)^2 n^{\alpha} e^{-b_2 j} \\ \Leftarrow & \frac{\alpha}{b_2 - b_1} \log n + \frac{3}{b_2 - b_1} \log\log n \leq j \leq \frac{\epsilon}{b_1} \log n - C,\end{aligned} \tag{113}$$

when $n > C$ for some $C > 0$ depending on $b_1, \ldots, b_M$, which implies

$$\left|\{j : \lambda_{1j} = \max_h \lambda_{hj}\}\right| \geq \frac{\epsilon b_2 - (\epsilon + \alpha) b_1}{b_1(b_2 - b_1)} \log n - \frac{3 \log\log n}{b_2 - b_1} - C.$$

Thus we have

$$\begin{aligned}& \sum_{j=1}^{n} \frac{\lambda_{1j}^2}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \\ \geq & \frac{\epsilon b_2 - (\epsilon + \alpha) b_1}{b_1(b_2 - b_1)(M+1)^2 \theta_{\max}^2} \log n - \frac{3 \log\log n}{(b_2 - b_1)(M+1)^2 \theta_{\max}^2} - C \\ \geq & \frac{\epsilon(b_2 - 2b_1) \log n}{2b_1(b_2 - b_1)(M+1)^2 \theta_{\max}^2},\end{aligned}$$

if $n > C$ and $\alpha \leq \epsilon$.

2. $l = l' = M + 1$

The upper bound for $\sum_{j=1}^{n} \frac{\lambda_{M+1,j}^2}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2}$ in Lemma 11 is straightforward, since $\sum_{h=1}^{M+1} \theta_h \lambda_{hj} \geq \theta_{\min} \lambda_{M+1,j}$. While for the lower bound, note that $\lambda_{M+1,j} = 1$, and thus

$$\sum_{j=1}^{n} \frac{\lambda_{M+1,j}^2}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \geq \frac{|\{j : \sum_{h=1}^{M+1} \theta_h \lambda_{hj} \leq 2\theta_{\max}\}|}{4\theta_{\max}^2}. \tag{114}$$

Meanwhile, let $\eta = \frac{3}{2}$ in (105), then one can show that

$$\begin{aligned}\sum_{h=1}^{M+1} \theta_h \lambda_{hj} \leq 2\theta_{\max} \Leftarrow & C M \theta_{\max} n^{1+\frac{\alpha}{2}} e^{-\frac{b_1 j}{3}} \leq \theta_{\max} \\ \Leftarrow & j \geq \frac{6 + 3\alpha}{2b_1} \log n + C \\ \Leftarrow & j \geq \frac{6 + 3\alpha}{b_1} \log n\end{aligned} \tag{115}$$

when $n > C$ for $C > 0$ depending on $M, b_1, \dots, b_M$. Therefore,

$$\sum_{j=1}^{n} \frac{\lambda_{M+1,j}^2}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \geq \frac{n - \frac{6+3\alpha}{b_1} \log n}{4\theta_{\max}^2}. \tag{116}$$

3. $1 < l \leq l' \leq M$

   First note that by similar arguments from the first case where $l = l' = 1$, one can show that

$$\sum_{j=1}^{n} \frac{\lambda_{lj}\lambda_{l'j}}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \leq \frac{4+2\alpha}{b_l \theta_{\min}^2} \log n. \tag{117}$$

   Furthermore, if $M \geq 2$ and $b_2 > 2b_1$ hold, then we can utilize the following upper bound for each term $\frac{\lambda_{lj}\lambda_{l'j}}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2}$:

$$\frac{\lambda_{lj}\lambda_{l'j}}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \leq \min\left\{ \frac{1}{\theta_{\min}^2}, \frac{C(\eta) n^{2+\alpha} e^{-\frac{b_l}{\eta} j}}{\theta_{\min}^2 \left(\lambda_{1j} + 1\right)^2} \right\}, \tag{118}$$

   where $\lambda_{1j} \geq \frac{C_1}{2} n e^{-b_1 j} - C n^{1-\epsilon}$. When $j < \frac{\epsilon}{b_1} \log n - C$ for some C depending on $b_1, \dots, b_M$, we have $C n^{1-\epsilon} < \frac{C_1}{4} n e^{-b_1 j}$, and thus

$$\frac{\lambda_{lj}\lambda_{l'j}}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \leq \frac{C(\eta) n^{\alpha}}{\theta_{\min}^2} e^{(2b_1 - \frac{b_l}{\eta}) j};$$

   while for $j \geq \frac{\epsilon}{b_1} \log n - C$, we have the bound

$$\frac{\lambda_{lj}\lambda_{l'j}}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \leq \frac{C(\eta) n^{2+\alpha} e^{-\frac{b_l}{\eta} j}}{\theta_{\min}^2}.$$

   Let $\frac{2b_1}{b_2} < \epsilon < 1$, $\eta = \frac{6b_1 + \epsilon b_2}{8b_1}$, $\alpha \leq \frac{2\epsilon b_2 - 4b_1}{6b_1 + \epsilon b_2}$, then one can show that

$$\begin{aligned} \sum_{j=1}^{n} \frac{\lambda_{lj}\lambda_{l'j}}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \leq & \frac{\alpha \log n}{\theta_{\min}^2 (\frac{b_l}{\eta} - 2b_1)} + \sum_{j=\lceil \frac{\alpha}{b_l/\eta - 2b_1} \log n \rceil}^{\lfloor \frac{\epsilon}{b_1} \log n - C \rfloor} \frac{C(\eta) n^{\alpha} e^{-(\frac{b_l}{\eta} - 2b_1) j}}{\theta_{\min}^2} \\ & + \sum_{j=\lceil \frac{\epsilon}{b_1} \log n - C \rceil}^{n} \frac{C(\eta) n^{2+\alpha} e^{-\frac{b_l j}{\eta}}}{\theta_{\min}^2} \\ \leq & \frac{(6b_1 + b_2)\alpha}{2b_1(4b_l - b_2 - 6b_1)\theta_{\min}^2} \log n + \frac{C(\epsilon)}{\theta_{\min}^2}, \end{aligned} \tag{119}$$

   for $C(\epsilon) > 0$ depending on $\epsilon, b_1, \dots, b_M$. Here the last line is due to that when $n > C(\epsilon)$, we have $\frac{\epsilon b_2}{b_1} - \frac{C b_2}{\log n} \geq \frac{2b_1 + \epsilon b_2}{2b_1}$, and thus

$$n^{2+\alpha} \exp\left\{ -\frac{b_l}{\eta} \left(\frac{\epsilon}{b_1} \log n - C \right) \right\} \leq n^{\frac{4b_1 - 2\epsilon b_2}{6b_1 + \epsilon b_2} + \alpha} \leq 1.$$

4. $1 = l < l' \leq M$
   Similarly from the previous case, we first have the bound

$$\sum_{j=1}^{n} \frac{\lambda_{lj}\lambda_{l'j}}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \leq \frac{4+2\alpha}{b_1 \theta_{\min}^2} \log n, \tag{120}$$

   which holds with as long as $0 < b_1 < b_2 < \cdots < b_M$. If $M \geq 2$ and $b_2 > 2b_1$ hold, then we bound each term in the summation as follows

$$\frac{\lambda_{1j}\lambda_{l'j}}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \leq \min\left\{\frac{1}{\theta_{\min}^2}, \frac{C(\eta) n^{1+\frac{\alpha}{2}} e^{-\frac{b_{l'}}{2\eta} j}}{\theta_{\min}^2 (\lambda_{1j}+1)}\right\}. \tag{121}$$

   Let $\frac{2b_1}{b_2} < \epsilon < 1$, $\eta = \frac{6b_1 + \epsilon b_2}{8b_1}$, $\alpha \leq \frac{2\epsilon b_2 - 4b_1}{6b_1 + \epsilon b_2}$, then one can show that

$$\begin{aligned}\sum_{j=1}^{n} \frac{\lambda_{1j}\lambda_{l'j}}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \leq& \frac{\alpha \log n}{\theta_{\min}^2 (\frac{b_{l'}}{\eta} - 2b_1)} + \sum_{j=\lceil \frac{\alpha}{b_{l'}/\eta - 2b_1} \log n \rceil}^{\lfloor \frac{\epsilon}{b_1} \log n - C \rfloor} \frac{C(\eta) n^{\frac{\alpha}{2}} e^{-(\frac{b_{l'}}{2\eta} - b_1)j}}{\theta_{\min}^2} \\ &+ \sum_{j=\lceil \frac{\epsilon}{b_1} \log n - C \rceil}^{n} \frac{C(\eta) n^{1+\frac{\alpha}{2}} e^{-\frac{b_{l'} j}{2\eta}}}{\theta_{\min}^2} \\ \leq& \frac{(6b_1 + b_2)\alpha}{2b_1(4b_{l'} - b_2 - 6b_1)\theta_{\min}^2} \log n + \frac{C(\epsilon)}{\theta_{\min}^2},\end{aligned} \tag{122}$$

   for $C(\epsilon) > 0$ depending on $\epsilon, b_1, \ldots, b_M$.

5. $1 < l < l' = M+1$
   Note that

$$\frac{\lambda_{lj}\lambda_{M+1,j}}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \leq \min\left\{\frac{1}{\theta_{\min}^2}, \frac{C(\eta) n^{1+\frac{\alpha}{2}} e^{-\frac{b_l}{2\eta} j}}{\theta_{\min}^2 (\lambda_{1j}+1)}\right\}, \tag{123}$$

   thus based on the same argument as the previous case, we have

$$\sum_{j=1}^{n} \frac{\lambda_{lj}\lambda_{M+1,j}}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \leq \frac{4+2\alpha}{b_1 \theta_{\min}^2} \log n, \tag{124}$$

   and when $M \geq 2$, $b_2 > 2b_1$,

$$\sum_{j=1}^{n} \frac{\lambda_{lj}\lambda_{M+1,j}}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \leq \frac{(6b_1 + b_2)\alpha}{2b_1(4b_l - b_2 - 6b_1)\theta_{\min}^2} \log n + \frac{C(\epsilon)}{\theta_{\min}^2}. \tag{125}$$

6. $l = 1, l' = M+1$
   Since

$$\frac{\lambda_{1j}\lambda_{M+1,j}}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \leq \min\left\{\frac{1}{4\theta_{\min}^2}, \frac{C(\eta) n^{1+\frac{\alpha}{2}} e^{-\frac{b_1}{2\eta} j}}{\theta_{\min}^2}\right\}, \tag{126}$$

one can show that

$$\begin{aligned}\sum_{j=1}^{n}\frac{\lambda_{1j}\lambda_{M+1,j}}{\left(\sum_{h=1}^{M+1}\theta_h\lambda_{hj}\right)^2}\leq&\frac{(2+\alpha)\eta\log n}{4b_1\theta_{\min}^2}+\frac{C(\eta)n^{1+\frac{\alpha}{2}}}{\theta_{\min}^2}\sum_{j=\lceil\frac{(2+\alpha)\eta}{b_1}\log n\rceil}^{n}e^{-\frac{b_1}{2\eta}j}\\\leq&\left(\frac{(2+\alpha)\eta}{4}+\frac{C(\eta)}{\log n}\right)\frac{\log n}{b_1\theta_{\min}^2}.\end{aligned}\tag{127}$$

Let $\eta=\frac{8}{7}$, then when $n>C$ for some $C>0$ depending on $b_1,\ldots,b_M$,

$$\sum_{j=1}^{n}\frac{\lambda_{1j}\lambda_{Mj}}{\left(\sum_{h=1}^{M+1}\theta_h\lambda_{hj}\right)^2}\leq\frac{(5+2\alpha)\log n}{7b_1\theta_{\min}^2}.$$

Therefore, for any $\alpha>0$, if $n>C$ for $C>0$ depending on $M,b_1,\ldots,b_M$, then with probability at least $1-3Mn^{-\alpha}$, (46) holds.

If $M=1$, then for any $0<\alpha,\epsilon<1$, with probability at least $1-2n^{-\alpha}$, (47) holds in addition to (46); If $M\geq 2$ and $b_2>2b_1$ hold, then any $\frac{2b_1}{b_2}<\epsilon<1$, $0<\alpha<\min\left\{\frac{2\epsilon b_2-4b_1}{6b_1+\epsilon b_2},1\right\}$, if $n>C(\epsilon)$ for $C(\epsilon)>0$ depending on $M,b_1,\ldots,b_M,\epsilon$, with probability at least $1-3Mn^{-\alpha}$, (48) holds in addition to (46). ■

Proof [Proof of Lemma 13] First note that

$$\begin{aligned}\left|z_n^\top\mathbf{\Lambda}^{(H)}(\boldsymbol{\theta})z_n-\operatorname{tr}(\mathbf{\Lambda}^{(H)}(\boldsymbol{\theta}))\right|=&\left|\sum_{j=1}^{n}\mathbf{\Lambda}_{jj}^{(H)}(\boldsymbol{\theta})(z_{nj}^2-1)\right|\\\leq&\sum_{j=1}^{n}\mathbf{\Lambda}_{jj}^{(H)}(\boldsymbol{\theta})|z_{nj}^2-1|.\end{aligned}\tag{128}$$

By the definition of $\mathbf{\Lambda}^{(H)}(\boldsymbol{\theta})$, ε, $t_i(n)$, (89) and (90),

$$\begin{aligned}\|\mathbf{\Lambda}^{(H)}(\boldsymbol{\theta})\|_2\leq&\|\mathbf{\Lambda}(\widetilde{\boldsymbol{\theta}})\|_2(H+1)\left(\frac{\varepsilon(M+1)}{\theta_{\min}}\right)^H\leq Ce^{-H},\\\|\mathbf{\Lambda}^{(H)}(\boldsymbol{\theta})\|_F^2\leq&e^{-2H}\|\mathbf{\Lambda}(\widetilde{\boldsymbol{\theta}})\|_F^2\leq Ce^{-2H}t_i(n).\end{aligned}$$

Also note that following similar arguments for bounding $\|\mathbf{\Lambda}(\boldsymbol{\theta}_\varepsilon^{(k)})\|_F^2$, we have

$$\sum_{j=1}^{n}\left|\mathbf{\Lambda}_{jj}(\widetilde{\boldsymbol{\theta}})\right|\leq Ct_i(n),\tag{129}$$

and thus

$$\sum_{j=1}^{n}\left|\mathbf{\Lambda}_{jj}^{(H)}(\boldsymbol{\theta})\right|\leq e^{-H}\sum_{j=1}^{n}\left|\mathbf{\Lambda}_{jj}(\widetilde{\boldsymbol{\theta}})\right|\leq Ce^{-H}t_i(n).\tag{130}$$

Therefore,

$$\begin{aligned}&\mathbb{P}\left(\left|z_n^\top \mathbf{\Lambda}^{(H)}(\boldsymbol{\theta})z_n - \operatorname{tr}(\mathbf{\Lambda}^{(H)}(\boldsymbol{\theta}))\right| > e^{-H}\left(t + Ct_i(n)\right)\right)\\ \le&\mathbb{P}\left(\sum_{i=1}^m \mathbf{\Lambda}_{jj}^{(H)}(\boldsymbol{\theta})(|z_{nj}^2-1| - \mathbb{E}(|z_{nj}^2-1|)) > e^{-H}t\right),\end{aligned}\tag{131}$$

where $C > 0$ depends on $M, \theta_{\min}, \theta_{\max}, b_1, \ldots, b_M$. Since $|z_{nj}^2 - 1|$ is sub-exponential with constant parameter,

$$\mathbb{P}\left(\sum_{j=1}^n \mathbf{\Lambda}_{jj}^{(H)}(\boldsymbol{\theta})\left(|z_{nj}^2-1| - \mathbb{E}|z_{nj}^2-1|\right) > e^{-H}t\right) \le 2\exp\left\{-c\min\left\{\frac{t^2}{t_i(n)}, t\right\}\right\}. \tag{132}$$

■

Proof [proof of Lemma 9] Following the calculations in the proof of Lemma 8, one can show that

$$\begin{aligned}&(g^*(\boldsymbol{\theta}^{(k)}))_{M+1}(\theta_{M+1}^{(k)} - \theta_{M+1}^*)\\ =&\frac{1}{2m}\sum_{l=1}^{M+1}(\theta_l^{(k)} - \theta_l^*)(\theta_{M+1}^{(k)} - \theta_{M+1}^*)\sum_{j=1}^m \frac{\lambda_{lj}^{(k)}}{\left(\sum_{l=1}^{M+1}\theta_l\lambda_{lj}^{(k)}\right)^2}\\ \ge&\frac{1}{2m}(\theta_{M+1}^{(k)} - \theta_{M+1}^*)^2\sum_{j=1}^m \frac{1}{\left(\sum_{l=1}^{M+1}\theta_l\lambda_{lj}^{(k)}\right)^2} - \frac{1}{2m}(\theta_{\max}-\theta_{\min})^2\sum_{l=1}^M\sum_{j=1}^m \frac{\lambda_{lj}^{(k)}}{\left(\sum_{l=1}^{M+1}\theta_l\lambda_{lj}^{(k)}\right)^2}\\ \ge&(\theta_{M+1}^{(k)} - \theta_{M+1}^*)^2\frac{1 - M\max_l a_l m^{\frac{(2+\alpha)(4b_1+3)}{4b_1(2b_1-1)}-1}}{8\theta_{\max}^2}\\ &-(\theta_{\max}-\theta_{\min})^2 M\left(\frac{1}{2\theta_{\min}^2} + \frac{\max_l a_l(4b_1+3)}{2\theta_{\min}^2(4b_1^2-6b_1-3)}\right)m^{\frac{(2+\alpha)(4b_1+3)}{4b_1(2b_1-1)}-1},\end{aligned}\tag{133}$$

with probability at least $1 - Mm^{-\alpha}$ for any $0 < \alpha < \frac{8b_1^2-12b_1-6}{4b_1+3}$, where the last line is due to the following Lemma 12.

Therefore,

$$(g^*(\boldsymbol{\theta}^{(k)}))_{M+1}(\theta_{M+1}^{(k)} - \theta_{M+1}^*) \ge \frac{\gamma}{2}(\theta_{M+1}^{(k)} - \theta_{M+1}^*)^2 - \varepsilon, \tag{134}$$

where $\gamma = \frac{1}{8\theta_{\max}^2}$, $\varepsilon = Cm^{\frac{(2+\alpha)(4b_1+3)}{4b_1(2b_1-1)}-1}$, if $m > C$. Here $C > 0$ depends only on $\theta_{\min}$,$\theta_{\max}$,M,$b_1, \ldots, b_M$. ■

Proof [proof of Lemma 12] Similarly from the proof of Lemma 11, we apply Lemma 14 on $\mathbf{K}_{f,n}^{(l)}$ to derive upper bounds for $\lambda_{lj}, 1 \le l \le M-1$ w.h.p. In particular, plug $r = j^{\frac{4b_l}{4b_l+3}}$ for each $1 \leq j \leq n$ into (103) and let $\delta = n^{-(\alpha+1)}$ for $0 < \alpha < \frac{8b_1^2-12b_1-6}{4b_1+3}$, then with probability at least $1 - n^{-(\alpha+1)}$,

$$\begin{aligned}
C(r,n) <& \sqrt{\frac{2}{C_l}} r^{b_l+2} n^{\frac{\alpha}{2}} \leq \sqrt{\frac{2}{C_l}} j^{\frac{4b_l(b_l+2)}{4b_l+3}} n^{\frac{\alpha}{2}}, \\
E(r_j,n) <& \frac{C_l}{2b_l-1} r^{-(2b_l-1)} + \sqrt{\frac{2C_l}{2b_l-1}} r^{-(b_l-\frac{1}{2})} n^{\frac{\alpha}{2}} \\
\leq& \left(\frac{C_l}{2b_l-1} + \sqrt{\frac{2C_l}{2b_l-1}} \right) j^{-\frac{2b_l(2b_l-1)}{4b_l+3}} n^{\frac{\alpha}{2}},
\end{aligned}$$

Thus we have

$$\begin{aligned}
\lambda_{lj} \leq& C_l j^{-2b_l} \left(n + \sqrt{\frac{2}{C_l}} j^{\frac{4b_l(b_l+2)}{4b_l+3}} n^{1+\frac{\alpha}{2}} \right) + \left(\frac{C_l}{2b_l-1} + \sqrt{\frac{2C_l}{2b_l-1}} \right) j^{-\frac{2b_l(2b_l-1)}{4b_l+3}} n^{1+\frac{\alpha}{2}} \\
\leq& \left(2\sqrt{2C_l} + \sqrt{\frac{2C_l}{2b_l-1}} + \frac{C_l}{2b_l-1} \right) j^{-\frac{2b_l(2b_l-1)}{4b_l+3}} n^{1+\frac{\alpha}{2}} \qquad (135) \\
:=& a_l j^{-\frac{2b_l(2b_l-1)}{4b_l+3}} n^{1+\frac{\alpha}{2}},
\end{aligned}$$

for $1 \leq j \leq n, 1 \leq l \leq M$, with probability at least $1 - Mn^{-\alpha}$. Now we are ready to prove the bounds for $\sum_{j=1}^n \frac{\lambda_{lj}\lambda_{l'j}}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2}$ for $1 \leq l, l' \leq M+1$.

1. $1 \leq l \leq l' \leq M$
   For any $0 < L \leq n$, one can show that

$$\begin{aligned}
\sum_{j=1}^n \frac{\lambda_{lj}\lambda_{l'j}}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \leq& \frac{L}{\theta_{\min}^2} + \frac{a_l a_{l'}}{\theta_{\min}^2} \sum_{j=L}^{\infty} j^{-\frac{4b_l(2b_l-1)}{4b_l+3}} n^{2+\alpha} \\
\leq& \frac{L}{\theta_{\min}^2} + \frac{a_l a_{l'} L^{1-\frac{4b_l(2b_l-1)}{4b_l+3}}}{\theta_{\min}^2 (\frac{4b_l(2b_l-1)}{4b_l+3} - 1)} n^{2+\alpha}
\end{aligned} \qquad (136)$$

   Let $L = n^{\frac{(2+\alpha)(4b_l+3)}{4b_l(2b_l-1)}}$, then we have

$$\sum_{j=1}^n \frac{\lambda_{lj}\lambda_{l'j}}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \leq n^{\frac{(2+\alpha)(4b_l+3)}{4b_l(2b_l-1)}} \left(\frac{1}{\theta_{\min}^2} + \frac{a_l a_{l'}(4b_l+3)}{\theta_{\min}^2 (8b_l^2 - 8b_l - 3)} \right) \qquad (137)$$

2. $l = l' = M+1$
   The upper bound for $\sum_{j=1}^n \frac{\lambda_{M+1,j}^2}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2}$ in Lemma 12 is straightforward, since $\sum_{h=1}^{M+1} \theta_h \lambda_{hj} \geq \theta_{\min} \lambda_{M+1,j}$. While for the lower bound, note that $\lambda_{M+1,j} = 1$, and

$$\sum_{j=1}^n \frac{\lambda_{M+1,j}^2}{\left(\sum_{h=1}^{M+1} \theta_h \lambda_{hj}\right)^2} \geq \frac{|\{j : \sum_{h=1}^{M+1} \theta_h \lambda_{hj} \leq 2\theta_{\max}\}|}{4\theta_{\max}^2}. \qquad (138)$$

Meanwhile, by (135) one can show that

$$\begin{aligned}\sum_{h=1}^{M+1}\theta_h\lambda_{hj}\le 2\theta_{\max} \Leftarrow & M\max_l a_l\theta_{\max} j^{-\frac{2b_1(2b_1-1)}{4b_1+3}} n^{1+\frac{\alpha}{2}} \le \theta_{\max} \\ \Leftarrow & j \ge M\max_l a_l n^{\frac{(2+\alpha)(4b_1+3)}{4b_1(2b_1-1)}}.\end{aligned} \tag{139}$$

Therefore,

$$\sum_{j=1}^{n}\frac{\lambda_{M+1,j}^2}{\left(\sum_{h=1}^{M+1}\theta_h\lambda_{hj}\right)^2}\ge\frac{n-M\max_l a_l n^{\frac{(2+\alpha)(4b_1+3)}{4b_1(2b_1-1)}}}{4\theta_{\max}^2}. \tag{140}$$

3. $1\le l\le M$, $l'=M+1$
   First note that by similar arguments from the first case where $1\le l\le l'\le M$, one can show that for any $L>0$,

$$\begin{aligned}\sum_{j=1}^{n}\frac{\lambda_{lj}\lambda_{l'j}}{\left(\sum_{h=1}^{M+1}\theta_h\lambda_{hj}\right)^2}\le&\frac{L}{\theta_{\min}^2}+\frac{a_l}{\theta_{\min}^2}\sum_{j=L}^{\infty}j^{-\frac{2b_l(2b_l-1)}{4b_l+3}}n^{1+\frac{\alpha}{2}} \\ \le&\frac{L}{\theta_{\min}^2}+\frac{a_l L^{1-\frac{2b_l(2b_l-1)}{4b_l+3}}}{\theta_{\min}^2(\frac{2b_l(2b_l-1)}{4b_l+3}-1)}n^{1+\frac{\alpha}{2}},\end{aligned} \tag{141}$$

Let $L=n^{\frac{(2+\alpha)(4b_l+3)}{4b_l(2b_l-1)}}$, then we have

$$\sum_{j=1}^{n}\frac{\lambda_{lj}\lambda_{l'j}}{\left(\sum_{h=1}^{M+1}\theta_h\lambda_{hj}\right)^2}\le n^{\frac{(2+\alpha)(4b_l+3)}{4b_l(2b_l-1)}}\left(\frac{1}{\theta_{\min}^2}+\frac{a_l(4b_l+3)}{\theta_{\min}^2(4b_l^2-6b_l-3)}\right) \tag{142}$$

Therefore, for any $0<\alpha<\frac{8b_1^2-12b_1-6}{4b_1+3}$, with probability at least $1-Mn^{-\alpha}$, (49) holds. ∎

Proof [proof for Lemma 6] Consider the case where $x_i\sim\mathcal{N}(0,\sigma^2)$, $k(x,x')=\exp\{-(x-x')^2/2l^2\}$, then λ_j takes the following analytical form (Zhu et al., 1997, see):

$$\lambda_j=(1-\beta)\beta^{j-1}, \tag{143}$$

where $\beta = \frac{2\sigma^2}{2\sigma^2+l^2+l\sqrt{l^2+4\sigma^2}}$ is a decreasing function of positive l. We want to see if $\widetilde{\gamma}_\epsilon$ is a decreasing function of β. First note that

$$\begin{aligned}\frac{\partial\widetilde{\gamma}_\epsilon^{(k)}(\beta,m)}{\partial\beta} =& -2\sum_{j=1}^{m}\frac{(j-1)\beta^{j-2}-j\beta^{j-1}}{\left(\theta_1^{(k)}m(1-\beta)\beta^{j-1}+\theta_2^{(k)}\right)^3}\\ =&2\sum_{j=1}^{m}\frac{j\beta^{j-1}}{\left(\theta_1^{(k)}m(1-\beta)\beta^{j-1}+\theta_2^{(k)}\right)^3}-2\sum_{j=1}^{m-1}\frac{j\beta^{j-1}}{\left(\theta_1^{(k)}m(1-\beta)\beta^{j}+\theta_2^{(k)}\right)^3}\\ =&2\sum_{j=1}^{m}j\beta^{j-1}\left[\left(\theta_1^{(k)}m(1-\beta)\beta^{j-1}+\theta_2^{(k)}\right)^{-3}-\left(\theta_1^{(k)}m(1-\beta)\beta^{j}+\theta_2^{(k)}\right)^{-3}\right]\\ &+2\frac{m\beta^{m-1}}{\left(\theta_1^{(k)}m(1-\beta)\beta^{m}+\theta_2^{(k)}\right)^3}.\end{aligned} \tag{144}$$

Let $a=\frac{\theta_1^{(k)}(1-\beta)}{\theta_2^{(k)}}$, and we provide an upper bound for $\frac{\theta_2^{(k)3}m}{2\log m}\frac{\partial\widetilde{\gamma}_\epsilon^{(k)}(\beta,m)}{\partial\beta}$ in the following:

$$\begin{aligned}\frac{\theta_2^{(k)3}m}{2\log m}\frac{\partial\widetilde{\gamma}_\epsilon^{(k)}(\beta,m)}{\partial\beta} =& \frac{m}{\log m}\sum_{j=1}^{m}j\beta^{j-1}\left[\left(am\beta^{j-1}+1\right)^{-3}-\left(am\beta^{j}+1\right)^{-3}\right]\\ &+\frac{m^2\beta^{m-1}}{\left(am\beta^m+1\right)^3\log m}\\ \le& -\frac{3a(1-\beta)m^2}{\log m}\sum_{j=1}^{m}\frac{j\beta^{2j-2}}{\left(am\beta^{j-1}+1\right)^4}+\frac{m^2\beta^{m-1}}{\left(am\beta^m+1\right)^3\log m}\\ \le& -\frac{3a(1-\beta)}{\log(\beta^{-1})\beta^2\left(a/\beta+1\right)^4}+\frac{m^2\beta^{m-1}}{\log m},\end{aligned} \tag{145}$$

where we let $j=\frac{\log m}{\log(\beta^{-1})}$ on the last line. Since

$$\lim_{m\to\infty}\frac{m^2\beta^{m-1}}{\log m}=0, \tag{146}$$

(145) implies that for any $l_0>0$, there exists a $m_0>0$ depending on $\theta_1^{(k)},\theta_2^{(k)},\sigma,l_0$ such that as long as $m>m_0$, $\widetilde{\gamma}_\epsilon$ is a increasing function of $l\ge l_0$. That is to say, for large enough minibatch, larger length scale leads to faster convergence for σ_ϵ^2, which suggests the potential benefit of nearby sampling. ■

Appendix F. Explanation on the connection between Assumption 3.1 and Assumption 3.2

We explain the how Assumption 3.2 can be proved with Assumption 3.1 under the exponential eigendecay and polynomial eigendecay cases separately.

- **Exponential eigendecay (Assumption 3.3)**: Consider Lemma 10 with $M = 1$, $s_1(m) = \tau \log m$ and $s_2(m) = m$, then we have

$$\|g(\boldsymbol{\theta}^{(k)}; \mathbf{X}_{\xi_{k+1}}, \mathbf{y}_{\xi_{k+1}}) - g^*(\boldsymbol{\theta}^{(k)})\|_2 \leq C(\log m)^{-\frac{1}{2}+\varepsilon},$$

with probability at least $1 - C\exp\{-c(\log m)^{2\varepsilon}\}$. Hence it suffices to show that $\|g^*(\boldsymbol{\theta}^{(k)})\|_2$ is bounded with high probability. Meanwhile, some calculation suggests

$$\begin{aligned}\mathbb{E}[g_1(\boldsymbol{\theta}^{(k)}; \mathbf{X}_{\xi_{k+1}}, \mathbf{y}_{\xi_{k+1}})|\mathbf{X}_{\xi_{k+1}}] &= \frac{1}{2s_1(m)}\sum_{l=1}^{2}(\theta_l^{(k)} - \theta_l^*)\sum_{j=1}^{m}\frac{(\lambda_j^{(k)})^{1+\mathbb{1}_{\{l=1\}}}}{(\theta_1^{(k)}\lambda_j^{(k)} + \theta_2^{(k)})^2},\\ \mathbb{E}[g_2(\boldsymbol{\theta}^{(k)}; \mathbf{X}_{\xi_{k+1}}, \mathbf{y}_{\xi_{k+1}})|\mathbf{X}_{\xi_{k+1}}] &= \frac{1}{2s_2(m)}\sum_{l=1}^{2}(\theta_l^{(k)} - \theta_l^*)\sum_{j=1}^{m}\frac{(\lambda_j^{(k)})^{\mathbb{1}_{\{l=1\}}}}{(\theta_1^{(k)}\lambda_j^{(k)} + \theta_2^{(k)})^2}.\end{aligned} \tag{147}$$

Under the exponential eigendecay assumption, Lemma 11 suggests that

$$\begin{aligned}\sum_{j=1}^{m}\frac{(\lambda_j^{(k)})^{1+\mathbb{1}_{\{l=1\}}}}{(\theta_1^{(k)}\lambda_j^{(k)} + \theta_2^{(k)})^2} &\leq C\log m,\\ \sum_{j=1}^{m}\frac{1}{(\theta_1^{(k)}\lambda_j^{(k)} + \theta_2^{(k)})^2} &\leq Cm.\end{aligned} \tag{148}$$

By the boundedness of $\boldsymbol{\theta}^{(k)}$ (Assumption 3.1), (148) and (147), we have

$$\|g(\boldsymbol{\theta}^{(\mathbf{k})}; \mathbf{X}_{\xi_{k+1}}, \mathbf{y}_{\xi_{k+1}})\|_2 \leq C$$

when $s_1(m) \geq c\log m$ and $s_2(m) = m$ and m is large enough.

- **Polynomial eigendecay (Assumption 3.4)**: Consider Lemma 10 with $M = 1$ and Assumption 3.4, then when $s_1(m) = s_2(m) = m$,

$$\|g(\boldsymbol{\theta}^{(k)}; \mathbf{X}_{\xi_{k+1}}, \mathbf{y}_{\xi_{k+1}}) - g^*(\boldsymbol{\theta}^{(k)})\|_2 \leq Cm^{-\frac{1}{2}+\varepsilon},$$

with probability at least $1 - C\exp\{-cm^{2\varepsilon}\}$. Meanwhile, note that Lemma 11 suggests that for $i = 0, 1$,

$$\sum_{j=1}^{m}\frac{(\lambda_j^{(k)})^i}{(\theta_1^{(k)}\lambda_j^{(k)} + \theta_2^{(k)})^2} \leq Cm, \tag{149}$$

and since (147) still holds, we have $\|g^*(\boldsymbol{\theta}^{(k)})\|_2 \leq C$ when $s_1(m) = s_2(m) = m$. Therefore, by the boundedness of $\boldsymbol{\theta}^{(k)}$ (Assumption 3.1), $\|g(\boldsymbol{\theta}^{(k)}; \mathbf{X}_{\xi_{k+1}}, \mathbf{y}_{\xi_{k+1}})\|_2 \leq C$ when $s_1(m) = s_2(m) = m$ and m is large enough.