Multi-fidelity Uncertainty Quantification and Optimization

by

Nuojin Cheng

B.S., University of Minnesota, Twin Cities, 2020

M.S., University of Colorado, Boulder, 2023

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Applied Mathematics

2025

Committee Members:

Alireza Doostan, Chair

Prof. Stephen Becker

Prof. Akil Narayan

Prof. Ian Grooms

Prof. Will Kleiber

Cheng, Nuojin (Ph.D., Applied Mathematics)

Multi-fidelity Uncertainty Quantification and Optimization

Thesis directed by Prof. Alireza Doostan

This thesis presents four novel bi-fidelity modeling approaches designed to enhance computational efficiency and accuracy in uncertainty quantification and optimization. First, Bi-fidelity Boosting (BFB) introduces an effective sketching-based subsampling method, accompanied by theoretical analysis of how inter-model correlation impacts performance. Second, the Bi-fidelity Variational Auto-encoder (BF-VAE) leverages deep generative models and transfer learning to achieve high performance with minimal high-fidelity data, also revealing connections between multi-fidelity learning and information bottleneck theory. Third, Langevin Bi-fidelity Importance Sampling (L-BF-IS) develops an efficient score-based Metropolis-Hastings importance sampling estimator for uncertainty quantification, whose effectiveness is linked to the discrepancy between model failure probability measures. Finally, a bi-fidelity zero-order optimization framework employs local multi-fidelity surrogates and an Armijo-based line search for optimal step sizes, demonstrating strong empirical performance supported by theoretical convergence guarantees under specific conditions. Collectively, these contributions advance multi-fidelity modeling by providing efficient, theoretically grounded methods for tackling complex computational challenges.[1]

[1] There are more contributed researches addressing optimization under uncertainty, including Bayesian optimization, information-theoretic interpretations of expected improvement [97], and exploration strategies [417].

Dedication

To my loving parents, Jianwen and Ling, my partner, Wanchen, and all the friends who have accompanied me on this journey.

Acknowledgements

This work was supported by the AFOSR awards FA9550-20-1-0138,FA9550-20-1-0188 with Dr. Fariba Fahroo as the program manager, and US Department of Energy's Wind Energy Technologies Office.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Quadrature Sampling of Parametric Models with Bi-fidelity Boosting

1.1 Abstract

Least squares regression is a ubiquitous tool for building emulators (a.k.a. surrogate models) of problems across science and engineering for purposes such as design space exploration and uncertainty quantification. When the regression data are generated using an experimental design process (e.g., a quadrature grid) involving computationally expensive models, or when the data size is large, sketching techniques have shown promise to reduce the cost of the construction of the regression model while ensuring accuracy comparable to that of the full data. However, random sketching strategies, such as those based on leverage scores, lead to regression errors that are random and may exhibit large variability. To mitigate this issue, we present a novel boosting approach that leverages cheaper, lower-fidelity data of the problem at hand to identify the *best* sketch among a set of candidate sketches. This in turn specifies the sketch of the intended high-fidelity model and the associated data. We provide theoretical analyses of this bi-fidelity boosting (BFB) approach and discuss the conditions the low- and high-fidelity data must satisfy for a successful boosting. In doing so, we derive a bound on the residual norm of the BFB sketched solution relating it to its ideal, but computationally expensive, high-fidelity boosted counterpart. Empirical results on both manufactured and PDE data corroborate the theoretical analyses and illustrate the efficacy of the BFB solution in reducing the regression error, as compared to the non-boosted solution. [1]

[1] The original version of this work is presented in [96], co-authored with Y. Xu, O. Malik, S. Becker, A. Doostan, and A. Narayan.

1.2 Introduction

Computational models are becoming central tools in analysis, design, and prediction. In these models, input parameters are often modeled as a random vector $\boldsymbol{p}$ to account for either uncertainty in precise values of these parameters, or as a means to model variability of parameters in order to assess robustness of an output [301, 481]. We consider such types of models given a (possibly non-linear) parameter-to-output map,

$$b = \mathcal{T}(\boldsymbol{p}), \qquad \mathcal{T} : \mathbb{R}^q \to \mathbb{R}.$$

A canonical example is when $\mathcal{T}$ is a measurement functional (e.g., the spatial average) operating on the solution to an elliptic partial differential equation (PDE) whose formulation contains random variables that, e.g., parameterize the diffusion coefficient. Hence, $\mathcal{T}$ is the composition of a measurement functional with the solution map of a parametric PDE. By placing a probability distribution on $\boldsymbol{p}$ that reflects a model of uncertainty, the goal of forward uncertainty quantification (UQ) is to quantify the resulting randomness in $b(\boldsymbol{p})$, frequently via statistics. Since explicit formulas revealing the dependence of b on $\boldsymbol{p}$ are typically not available, one resorts to approximations. One such sampling-based approach that we focus on is that of polynomial chaos (PC) methods [185, 565] using variants of stochastic collocation [564].

In this paper we consider building emulators for forward UQ via a non-intrusive least squares-based PC strategy. More precisely, we assume an *a priori* form for an emulator b_V:

$$b(\boldsymbol{p}) \approx b_V(\boldsymbol{p}) \coloneqq \sum_{j=1}^{d} x_j^* \psi_j(\boldsymbol{p}), \qquad V \coloneqq \operatorname{span}\{\psi_1, \ldots, \psi_d\}, \tag{1.1}$$

where ψ_j are fixed, known functions (in PC approaches they are multivariate polynomial functions of $\boldsymbol{p}$), and the coefficients x_j^* must be determined. We identify these coefficients through data collected from evaluating b on a prescribed quadrature rule $\{(\boldsymbol{p}_n, w_n)\}_{n=1}^N$, with quadrature nodes $\boldsymbol{p}_n$ and positive weights w_n. The coefficients x_j^* are then chosen as the solution to a quadrature-based least squares problem,

$$\boldsymbol{x}^* = \operatorname*{arg\,min}_{\boldsymbol{x} \in \mathbb{R}^d} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2, \qquad \boldsymbol{A}(n,j) = \sqrt{w}_n \psi_j(\boldsymbol{p}_n), \qquad \boldsymbol{b}(n) = \sqrt{w}_n b(\boldsymbol{p}_n), \tag{1.2}$$

where $\boldsymbol{A} \in \mathbb{R}^{N\times d}$ is referred to as the **design matrix** of the problem. Once $\boldsymbol{x}^*$ is computed, the emulator b_V is easily manipulated and computationally analyzed to compute (approximate) statistics for b or the sensitivity of b to each entry of $\boldsymbol{p}$. The challenge with this approach is that when $\dim \boldsymbol{p} = q \gg 1$, then designing an appropriately accurate quadrature rule requires $N \gg 1$ samples of b, which is prohibitively expensive when such evaluations amount to PDE solutions. (For example a q-dimensional tensorized Gaussian quadrature rule with n points per dimension requires $N = n^q$ points.)

In this paper, we describe one strategy to mitigate this cost via a procedure that combines statistical boosting ideas from theoretical computer science (see, e.g., [337, Sec. 7.2] and [557, Sec. 2.3]) with bi-fidelity strategies in UQ. More precisely, our approach boosts on the randomness of a *sketching* operator $\boldsymbol{S} \in \mathbb{R}^{m\times N}$ that is used to approximately solve (1.2):

$$\boldsymbol{x}^{**} = \underset{\boldsymbol{x}\in\mathbb{R}^d}{\arg\min} \|\boldsymbol{SAx} - \boldsymbol{Sb}\|_2^2. \tag{1.3}$$

Without *a priori* knowledge of $\boldsymbol{b}$, a deterministic sketch with $m < N$ generally is not robust to adversarial vectors $\boldsymbol{b}$ that result in a large residual for $\boldsymbol{x}^{**}$ relative to the residual for $\boldsymbol{x}^*$. However, in general scenarios one can identify constructive *probabilistic* models for $\boldsymbol{S}$ where sketches of near-optimal size, $m \gtrsim d \log d/(\epsilon\delta)$, ensure

$$\|\boldsymbol{Ax}^{**} - \boldsymbol{b}\|^2 \le (1+\epsilon)\|\boldsymbol{Ax}^* - \boldsymbol{b}\|^2 \text{ with probability } \ge 1-\delta.$$

We provide a more detailed discussion of existing sketching guarantees in section 1.3.2, in particular for *row sketches*, for which computing $\boldsymbol{Sb}$ requires knowledge of only m entries of $\boldsymbol{b}$, rather than all N entries. While random sketching provides attractive guarantees when $m \ll N$, it is still random and hence is subject to randomness in performance, and "failure" events can occur with nonzero probability δ. Naive statistical boosting mitigates this issue by generating several (say L) sketches and choosing the one that yields the smallest residual. However, this requires generating Lm entries of $\boldsymbol{b}$, which can be computationally expensive when each evaluation is an expensive PDE solve. Our approach attacks this problem in the sketch selection boosting phase by replacing

$\boldsymbol{b}$ with an approximate, low-fidelity version from which collecting Lm samples is computationally feasible. Once a "good" sketch is identified in the boosting phase, we solve the sketched least squares problem using the corresponding sketch of the original data $\boldsymbol{b}$.

Thus, we assume availability of and leverage a *low-fidelity* model $\widetilde{b}(\boldsymbol{p})$. For example, $\widetilde{b}$ may correspond to using a discretized PDE solver with a mesh coarser than the one which produces accurate realizations of b, or to model approximations such as Reynolds-averaged Navier Stokes solvers, or to solutions computed with arithmetic in lower precision compared to samples for b. Although $\widetilde{b}$ may be untrusted as a replacement for b, it can be used to extract some useful information about b, as is done in by-now standard multi-fidelity approaches [423]. Throughout this paper, we assume the bi-fidelity setup, i.e., two levels of fidelity, and also that the cost of evaluating $\widetilde{b}$ is much less than the corresponding cost for b; both of these are common practical assumptions [148, 380, 598, 172, 387].

1.2.1 Contributions of this article

The contributions of this article are as follows:

- We propose a new bi-fidelity boosting (BFB) algorithm to compute an approximation to $\boldsymbol{x}^*$. The procedure, given in Algorithm 2, computes the solution of a *sketched* least squares problem, where the sketch matrix is identified by a boosting procedure on a low-fidelity data vector $\widetilde{\boldsymbol{b}}$. The sketching approach reduces the required sample complexity from N evaluations of b to $\sim d \log d$ samples of b, which can be a significant saving. The boosting procedure requires $\sim Ld \log d$ evaluations of the low-fidelity model $\widetilde{b}$, where, in the language of statistical learning, L is the number of weak learners used in the boosting procedure. When $\widetilde{b}$ costs substantially less than b, this cost for collecting the boosting data is negligible.

- We provide a theoretical analysis of BFB under certain assumptions, which provides quantitative bounds on the residual of the BFB solution $\hat{\boldsymbol{x}}_{\mathsf{BFB}}$ relative to the full, computationally expensive solution $\boldsymbol{x}^*$ (see Theorems 1.4.2 and 1.4.4). We also provide some asymptotic

bounds on the correlation between the low- and high-fidelity solutions in a certain sense (see Theorem 1.4.5). Finally, we provide concrete computational strategies to ensure that the required assumptions of BFB hold (see Theorem 1.4.11).

- We investigate the numerical performance of BFB when combined with several different sampling strategies and compare the performance to the corresponding sampling strategies without boosting. We also demonstrate using real-world problems that the assumptions required for BFB's theoretical analysis frequently hold in practice.

The idea of sketching for least squares solutions has a substantial history in the computer science and numerical linear algebra communities [337, 557]. Our use of sparse row sketches of size $\sim d$ is identical to existing methods for leverage score-based [337], Gaussian-sketch based [357], and volume-maximizing sketching [137, 138]. In addition, boosting for least squares problems is also not a new idea [206]. However our combination of these approaches in a bi-fidelity setting is new to our knowledge, and our analysis in this bi-fidelity context provides novel, non-trivial insight into the algorithm performance.

The rest of this manuscript is organized as follows. Section 1.3 introduces the notation we use and provides some background material on various sketching approaches in least squares approximation. Section 1.4 presents the BFB algorithm along with its theoretical analysis. Section 1.5 contains numerical experiments which illustrate various aspects of the BFB approach. We conclude the present study in Section 1.6. The paper also contains several appendices. Appendix A.1 provides a brief introduction to the sampling approach that we proposed in [349] and which we make use of in this paper. Appendices A.2 and A.3 contain some proofs that have been left out of the main text.

1.3 Preliminaries

For the interest of clarity and completeness, we next introduce the notation used throughout the manuscript and introduce four sampling strategies to sketch the least squares problem (1.2),

namely, sampling via column-pivoted QR, leverage scores, volume maximization, and Gaussian distribution.

1.3.1 Notation

Matrices are denoted by bold upper-case letters (e.g., $\boldsymbol{A}$), vectors are denoted by bold lower-case letters (e.g., $\boldsymbol{x}$) and scalars by lower case regular and Greek letters (e.g., a and α). Entries of matrices and vectors are indicated in parentheses. For example, $\boldsymbol{A}(i,j)$ is the entry on position (i,j) in $\boldsymbol{A}$ and $\boldsymbol{a}(i)$ is the ith entry in $\boldsymbol{a}$. A colon is used to denote all entries along a mode of a matrix. For example, $\boldsymbol{A}(i,:)$ is the ith row of $\boldsymbol{A}$ represented as a row vector. For a set of indices $\mathcal{J}$, $\boldsymbol{A}(\mathcal{J},:)$ denotes the submatrix $(\boldsymbol{A}(j,:))_{j\in\mathcal{J}}$ and $\boldsymbol{a}(\mathcal{J})$ denotes the subvector $(\boldsymbol{a}(j))_{j\in\mathcal{J}}$.

The **compact SVD** of a matrix $\boldsymbol{A}$ takes the form $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, where $\boldsymbol{U}$ and $\boldsymbol{V}$ have $\operatorname{rank}(\boldsymbol{A})$ columns and $\boldsymbol{\Sigma}$ is of size $\operatorname{rank}(\boldsymbol{A}) \times \operatorname{rank}(\boldsymbol{A})$. The **pseudoinverse** of $\boldsymbol{A}$ is denoted by $\boldsymbol{A}^\dagger \coloneqq \boldsymbol{V}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^\top$. For a matrix $\boldsymbol{U}$ with orthonormal columns, we use $\boldsymbol{U}_\perp$ to denote an orthonormal complement of $\boldsymbol{U}$, i.e., $\boldsymbol{U}_\perp$ is any matrix such that $[\boldsymbol{U}, \quad \boldsymbol{U}_\perp]$ is square and has orthonormal columns. We use $\boldsymbol{P}_{\boldsymbol{A}} \coloneqq \boldsymbol{A}\boldsymbol{A}^\dagger = \boldsymbol{U}\boldsymbol{U}^\top$ to denote the **orthogonal projection** onto $\operatorname{range}(\boldsymbol{A})$, where $\boldsymbol{U} = \operatorname{orth}(\boldsymbol{A})$ is a(ny) matrix whose columns are an orthonormal basis for $\operatorname{range}(\boldsymbol{A})$, e.g., via the compact SVD or QR decomposition of $\boldsymbol{A}$. The determinant of $\boldsymbol{A}$ is denoted by $\det(\boldsymbol{A})$. For a positive integer n, we use the notation $[n] \coloneqq \{1, 2, \ldots, n\}$. We use $\boldsymbol{a}_{\mathcal{P}}$ to denote a vector $\boldsymbol{a} \neq \boldsymbol{0}$ rescaled to unit length:

$$\boldsymbol{a}_{\mathcal{P}} = \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|_2}. \tag{1.4}$$

We also introduce two notions of correlation: for given deterministic vectors $\boldsymbol{a}, \boldsymbol{b} \neq \boldsymbol{0}$, we define the correlation between them as the cosine of the angle separating them:

$$\operatorname{corr}(\boldsymbol{a}, \boldsymbol{b}) \coloneqq \frac{\langle \boldsymbol{a}, \boldsymbol{b} \rangle}{\|\boldsymbol{a}\|_2 \|\boldsymbol{b}\|_2},$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product. We will also require Pearson's correlation coefficient, which is widely used in statistics. For two (non-constant) random variables X and Y with

bounded second moments defined on the same probability space, their correlation is defined as

$$\operatorname{corr}(X, Y) \coloneqq \frac{\varepsilon[(X - \varepsilon[X])(Y - \varepsilon[Y])]}{\sqrt{\mathbb{V}[X]\mathbb{V}[Y]}}, \tag{1.5}$$

where $\varepsilon[\cdot]$ and $\mathbb{V}[\cdot]$ are, respectively, the mathematical expectation and variance operators. Note that our notation $\operatorname{corr}(\cdot, \cdot)$ is overloaded, operating differently on vectors and (random) scalars. The context of use in what follows should make it clear which definition above is used.

We will use the following notation to denote the minimum of the least squares objective in (1.2):

$$r(\boldsymbol{A}, \boldsymbol{b}) \coloneqq \min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 = \|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|_2, \tag{1.6}$$

where $\boldsymbol{x}^*$ is defined as in (1.2).

1.3.2 Sketching of least squares problems

Solving the problem (1.2) using standard methods (e.g., via the QR decomposition) costs $\mathcal{O}(Nd^2)$[2]. When N is large, this may be prohibitively expensive. A popular approach to address this issue is to apply a **sketch operator** $\boldsymbol{S} \in \mathbb{R}^{m \times N}$ where $m \ll N$ to both $\boldsymbol{A}$ and $\boldsymbol{b}$ in (1.2) in order to reduce the size of the problem:

$$\hat{\boldsymbol{x}} \coloneqq \arg\min_{\boldsymbol{x} \in \mathbb{R}^d} \|\boldsymbol{S}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{S}\boldsymbol{b}\|_2 . \tag{1.7}$$

This approach has two benefits: (i) If $\boldsymbol{S}$ is a row-sketch, i.e., has only a small number of non-zero columns, then $\boldsymbol{S}\boldsymbol{b}$ requires knowledge of only a small number of entries of $\boldsymbol{b}$, and (ii) the cost of solving this smaller problem is $\mathcal{O}(md^2)$, a substantial reduction from $\mathcal{O}(Nd^2)$ when $m \ll N$. Analogously to (1.6), we will use the following to denote the least squares objective value for the approximate solution:

$$r_{\boldsymbol{S}}(\boldsymbol{A}, \boldsymbol{b}) \coloneqq \|\boldsymbol{A}\hat{\boldsymbol{x}} - \boldsymbol{b}\|_2. \tag{1.8}$$

The goal is for the approximation $\hat{\boldsymbol{x}}$ to yield a residual "close" to the optimal residual of the full problem (1.2),

$$r(\boldsymbol{A}, \boldsymbol{b}) \approx r_{\boldsymbol{S}}(\boldsymbol{A}, \boldsymbol{b}), \tag{1.9}$$

[2] In our context, we have $N > d$; see Assumption 1.4.1.

which is typically achieved if m is "large enough". The following definition makes this more precise.

Definition 1.3.1 ((ε, δ) pair condition)**.** Let $\boldsymbol{S} \in \mathbb{R}^{m \times N}$ be a random matrix. Given $\boldsymbol{A} \in \mathbb{R}^{N \times d}$, $\boldsymbol{b} \in \mathbb{R}^N$, and $\varepsilon, \delta > 0$, the distribution of $\boldsymbol{S}$ is said to satisfy an (ε, δ) pair condition for $(\boldsymbol{A}, \boldsymbol{b})$ if, with probability at least $1 - \delta$, both conditions,

$$\operatorname{rank}(\boldsymbol{S}\boldsymbol{A}) = \operatorname{rank}(\boldsymbol{A}) \quad \text{and} \quad r_{\boldsymbol{S}}(\boldsymbol{A}, \boldsymbol{b}) \leq (1 + \varepsilon)\, r(\boldsymbol{A}, \boldsymbol{b}), \tag{1.10}$$

hold simultaneously, where $r(\boldsymbol{A}, \boldsymbol{b})$ and $r_{\boldsymbol{S}}(\boldsymbol{A}, \boldsymbol{b})$ are defined as in (1.6) and (1.8), respectively.

Note that one can only ask for the above condition with probability less than 1: For any sketch with $m < N$, there are vectors $\boldsymbol{b}$ for which the residual bound condition in (1.10) can be violated. Such a condition can be satisfied with $m < N$ samples; see sections 1.3.2.2, 1.3.2.3, and 1.3.2.4. Sketching operators $\boldsymbol{S}$ that sample a subset of the rows are of particular interest in UQ since $\boldsymbol{S}\boldsymbol{b}$ in (1.7) then requires knowledge of only a subset of entries in the vector $\boldsymbol{b}$, meaning that fewer samples need to be collected. In this paper, we consider three different sketching operators of this type, one of which is deterministic and two of which are random. These are described in Sections 1.3.2.1–1.3.2.3. Another popular sketching operator is the Gaussian sketching operator whose entries are appropriately scaled i.i.d. normal random variables. Applying such a random matrix to $\boldsymbol{b}$ requires knowledge of all entries in $\boldsymbol{b}$. While this makes the Gaussian sketch unsuitable for use in practice for quadrature sampling, we still consider it in some of our theoretical results since it is easier to analyze than the sampling-based sketches. Furthermore, since it is known to have excellent guarantees, it provides a nice baseline. We introduce the Gaussian sketch in Section 1.3.2.4.

Much research has been conducted over the last two decades on randomized algorithms in numerical linear algebra, including the problem of solving least squares problems. We only cover the basics that are relevant for this paper. For a more in-depth discussion, we refer the reader to the surveys in [208, 337, 557, 357] and the references therein.

1.3.2.1 Sampling via column-pivoted QR decomposition

Let $\boldsymbol{A}^\top \boldsymbol{P} = \boldsymbol{A}(\mathcal{J},:)^\top = \boldsymbol{Q}\boldsymbol{R}$ be a column-pivoted QR (CPQR) decomposition where $\mathcal{J}$ is a length-N permutation vector. A simple deterministic heuristic for sampling m rows from $\boldsymbol{A}$ is to simply choose those rows corresponding to the first m entries in $\mathcal{J}$, i.e., $\boldsymbol{A}(\mathcal{J}(1:m),:)$. This corresponds to applying a sketch $\boldsymbol{S} = (\boldsymbol{P}(:,1:m))^\top$ to $\boldsymbol{A}$. Such an approach has been used to subsample points from either tensor product quadratures [466] or from random samples (approximate D-optimal design) [207, 146, 203] in the context of least squares polynomial approximation.

Recall that $\boldsymbol{A}$ is an $N \times d$ tall-and-skinny matrix. When $m \leq d$, the subsample is straightforward and just takes the first m entries in $\mathcal{J}$ since the list $\mathcal{J}$ contains the entries in decreasing order of importance (as approximated by the column-pivoting algorithm). When $m > d$, the situation is more subtle since the remaining entries $\mathcal{J}(d+1:N)$ have no particular meaning and will not be useful in our row-sampling procedure. To get around this, we use the heuristic in Algorithm 1 in order to sample $m > d$ rows. The heuristic chooses the first d rows indices to be the entries in $\mathcal{J}(1:m)$ where $\mathcal{J}$ comes from the column-pivoted QR decomposition of $\boldsymbol{A}^\top$. The rows with indices in $\mathcal{J}(1:m)$ are then removed from $\boldsymbol{A}$. Another column-pivoted QR decomposition is then computed for the updated $\boldsymbol{A}^\top$, and the next set of d rows is chosen to be the rows of $\boldsymbol{A}$ corresponding to the top-d entries in the new permutation vector $\mathcal{J}$. Once again, the chosen rows are removed from $\boldsymbol{A}$. This procedure is repeated until m rows have been chosen. It is straightforward to formulate a sampling matrix $\boldsymbol{S}$ such that $\boldsymbol{S}\boldsymbol{A} = \boldsymbol{A}_s$, where $\boldsymbol{A}_s$ is the output of Algorithm 1.

Algorithm 1: Heuristic for sampling via column-pivoted QR decomposition

Input: $\boldsymbol{A}$: design matrix; m: desired number of row samples
Output: $\boldsymbol{A}_s$: matrix containing m rows of $\boldsymbol{A}$

Initialize $\boldsymbol{A}_s$ to an empty matrix: $\boldsymbol{A}_s = [\,]$
while $m > 0$ **do**
  Compute column-pivoted QR of $\boldsymbol{A}^\top$: $\boldsymbol{A}(\mathcal{J},:)^\top = \boldsymbol{Q}\boldsymbol{R}$
  Let $k = \min(d, m)$
  Append top-k rows from $\boldsymbol{A}$ to $\boldsymbol{A}_s$: $\boldsymbol{A}_s = [\boldsymbol{A}_s;\, \boldsymbol{A}(\mathcal{J}(1:k),:)]$
  Remove top-k rows from $\boldsymbol{A}$: $\boldsymbol{A} = \boldsymbol{A}(\mathcal{J}(k+1:\text{end}),:)$
  $m = m - k$
end while
return $\boldsymbol{A}_s$

Since the approach in Algorithm 1 is deterministic, it cannot satisfy guarantees of the form in Definition 1.3.1. However, for the case $m = d$ it is possible to prove bounds on the condition number of $\boldsymbol{A}(\mathcal{J}(1:d),:)$; see Lemma 2.1 in [466] for details.

1.3.2.2 Leverage score sampling

Let $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ be a compact SVD. The **leverage scores** of $\boldsymbol{A}$ are defined as

$$\ell_i(\boldsymbol{A}) \coloneqq \|\boldsymbol{U}(i,:)\|_2^2 \quad \text{for } i \in [N]. \tag{1.11}$$

They take values in the range $\ell_i(\boldsymbol{A}) \in [d/N, 1]$ and indicate how important each row of $\boldsymbol{A}$ is in a certain sense. The matrix $\boldsymbol{U}$ can be replaced with any matrix whose columns form an orthonormal basis for range($\boldsymbol{A}$) without impacting the definition in (1.11) [557, Sec. 2.4]. The **coherence** of $\boldsymbol{A}$ is defined as

$$\gamma(\boldsymbol{A}) \coloneqq \max_{i\in[N]} \ell_i(\boldsymbol{A}). \tag{1.12}$$

It takes values in the range $\gamma(\boldsymbol{A}) \in [d/N, 1]$; it is maximal when one of the leverage scores is 1 and minimal when all leverage scores are equal to d/N. Let $r \coloneqq \sum_i \ell_i(\boldsymbol{A})$. The leverage score sampling distribution of $\boldsymbol{A}$ is defined as

$$p_i(\boldsymbol{A}) \coloneqq \frac{\ell_i(\boldsymbol{A})}{r} \quad \text{for } i \in [N], \tag{1.13}$$

which is indeed a probability distribution as $\ell_i(\boldsymbol{A}) > 0$. Let $f : [m] \to [N]$ be a random map such that each $f(j)$ is independent and $\mathcal{P}\{f(j) = i\} = p_i(\boldsymbol{A})$ for each $j \in [m]$. The leverage score sampling sketch $\boldsymbol{S} \in \mathbb{R}^{m\times N}$ is defined elementwise via

$$\boldsymbol{S}_{ji} = \frac{\mathrm{Ind}\{f(j) = i\}}{\sqrt{m p_{f(j)}(\boldsymbol{A})}} \quad \text{for } (j,i) \in [m] \times [N], \tag{1.14}$$

where $\mathrm{Ind}\{A\}$ is the indicator function which is 1 if the random event A occurs and zero otherwise. Algorithms and theory for leverage score sampling have been developed in a number of papers; see e.g., [157, 156, 158, 337, 297] and references therein. The distribution for the leverage score sketch in (1.14) satisfies an (ε, δ) condition for $(\boldsymbol{A}, \boldsymbol{b})$ if

$$m \gtrsim d\log(d/\delta) + d/(\varepsilon\delta); \tag{1.15}$$

see Theorem 1.4.11 for a more detailed and slightly stronger statement.

Choosing $p_i(\boldsymbol{A}) = 1/N$ results in **uniform sampling**. For general matrices, there are no useful guarantees when sampling uniformly in this fashion. However, if $\boldsymbol{A}$ has low coherence, then uniform sampling will be close to the leverage score sampling distribution and guarantees similar to those for leverage score sampling hold. More precisely, if $\ell_i(\boldsymbol{A}) \leq Cd/N$ for some constant $C \geq 1$, then uniform sampling satisfies an (ε, δ) condition for $(\boldsymbol{A}, \boldsymbol{b})$ if m is chosen as in (1.15) (this is a direct consequence of, e.g., Theorem 6 in [297]). Notice that the difference from sampling according to the exact leverage scores is that there now is an additional constant C hidden in the lower bound on m.

In addition to a parsimonious sampling of $\boldsymbol{b}$, the computational complexity of the sketched least squares approach in (1.7) is a consideration. Direct sampling of the leverage score distribution via the formula (1.11) requires a matrix decomposition (e.g., QR or SVD), which costs $\mathcal{O}(Nd^2)$ effort, the same effort required to solve the original least squares problem. **(author?)** [153] propose a procedure for computing leverage score estimates with cost $\mathcal{O}(Nd \log N)$ for any matrix $\boldsymbol{A}$. When $\boldsymbol{A}$ has particular structure it is possible to improve this considerably. **(author?)** [349] propose such a method for the case when the multivariate basis functions ψ_j in (1.1) are certain products of one-dimensional functions, which corresponds to impose certain structure on the subspace V. In the polynomial approximation setting, those structural conditions are satisfied by a large family of subspaces, including the popular tensor product, total degree, and hyperbolic cross spaces. For example, if the multivariate basis polynomials for q-dimensional inputs correspond to polynomials of at most degree k in each dimension and use n grid points per dimension (in which case $\boldsymbol{A}$ has $N = n^q$ rows), then the total cost of our method is at most $\mathcal{O}(qnk^2 + mq)$ for drawing m samples. This sampling approach is an ingredient in our method, so we describe the key aspects of how this sampling approach works in Appendix A.1 and refer the reader to [349] for a more comprehensive treatment.

1.3.2.3 Leveraged volume sampling

Volume sampling is a technique that samples a set $\mathcal{J} \subset [N]$ of m row indices of $\boldsymbol{A}$ with probability proportional to the squared volume of the parallelepiped spanned by the columns of the submatrix $\boldsymbol{A}(\mathcal{J},:)$, i.e., $\mathcal{P}(\mathcal{J}) \propto \det\big(\boldsymbol{A}(\mathcal{J},:)^\top \boldsymbol{A}(\mathcal{J},:)\big)$. This means that, unlike for leverage score sampling, the rows are not sampled independently. This has several benefits, including that the sketched least square solution $\boldsymbol{A}(\mathcal{J},:)^\dagger \boldsymbol{b}(\mathcal{J})$ is correct in expectation [136, Prop. 7]: $\varepsilon[\boldsymbol{A}(\mathcal{J},:)^\dagger \boldsymbol{b}(\mathcal{J})] = \boldsymbol{A}^\dagger \boldsymbol{b}$. Leverage score sampling, by contrast, may produce a biased estimate of the solution vector. Despite the apparent issue of sampling from a combinatorial number of subsets of $[N]$, there are algorithms for volume sampling that run in polynomial time. **(author?)** [137] propose two such algorithms, RegVol and FastRegVol. RegVol runs in $\mathcal{O}((N-m+d)Nd)$ time, and FastRegVol runs in $\mathcal{O}((N + \log(N/d)\log(1/\delta))d^2)$ time with probability at least $1-\delta$. The dependence on N can be prohibitive in quadrature sampling since the number of (tensor-product) quadrature points N is exponential in the number of variables.

(author?) [138] propose **leveraged** volume sampling which improves on standard volume sampling in several ways. Importantly, it still retains the correctness in expectation but allows for more efficient sampling. In particular, the cost of sampling does not depend on N. Unlike standard volume sampling, the sketch distribution satisfies an (ε,δ) condition for $(\boldsymbol{A},\boldsymbol{y})$ if $m \gtrsim d\log(d/\delta) + d/(\varepsilon\delta)$, which is on par with what leverage score sampling requires for such guarantees. Leveraged volume sampling has two stages. In the first stage, $\mathcal{O}(d^2)$ rows are chosen from $\boldsymbol{A}$ using a combination of leverage score sampling and rejection sampling. After that, the $\mathcal{O}(d^2)$ subset is further reduced to $\mathcal{O}(d\log(d/\delta) + d/(\varepsilon\delta))$ via standard volume sampling. In the experiments, we use FastRegVol from [137] for the second step. When FastRegVol is used, the cost of leveraged volume sampling is $\mathcal{O}(((d^2+m)d^2 + mC_{\text{samp}})\log(1/\delta))$, where C_{samp} is the cost of drawing one row index of $\boldsymbol{A}$ using leverage score sampling. As discussed in Section 1.3.2.2, the the cost C_{samp} of leverage score sampling can be reduced drastically in our setting by using the structured sampling techniques from [349].

1.3.2.4 Gaussian sketching operator

The Gaussian sketching operator $\boldsymbol{S} \in \mathbb{R}^{m\times N}$ has entries that are i.i.d. Gaussian random variables with mean zero and variance $1/m$. The Gaussian sketch satisfies an (ε, δ) condition if $m \gtrsim (d/\varepsilon)\log(d/\delta)$. These results also extend to the case when the entries of $\boldsymbol{S}$ are sub-Gaussian; see Theorem 1.4.11 for further details.

The main benefit of the Gaussian sketching operator is that it allows for simple and precise theoretical analysis of procedures that use sketching as a subroutine [357, Remark 8.2]. This is our motivation for considering the Gaussian sketch in this paper. Computationally, it is not efficient to use Gaussian sketching for least squares problems. The reason is that computing $\boldsymbol{SA}$ costs $\mathcal{O}(mNd)$ which is more than the $\mathcal{O}(Nd^2)$ cost of solving the original least squares problem (recall that $m > d$). As discussed earlier, an additional issue in bi-fidelity estimation is that computing $\boldsymbol{Sb}$ requires knowledge of all elements of $\boldsymbol{b}$ which is prohibitively expensive when that vector contains high-fidelity data.

1.3.3 Bi-fidelity problems

The main goal of this paper is to propose a strategy that improves the accuracy of sketching via a boosting procedure that employs a full vector $\tilde{\boldsymbol{b}}$ corresponding to an inexpensive low-fidelity approximation to $\boldsymbol{b}$.

Bi-fidelity frameworks assume the availability of a low-fidelity simulation $\widetilde{\mathcal{T}}$; that is, a map $\widetilde{\mathcal{T}} : \mathbb{R}^q \to \mathbb{R}$ such that $\widetilde{\mathcal{T}}$ is parameterically correlated with $\mathcal{T}$ in some sense, but need not be close to $\mathcal{T}$ in terms of sampled values. Such properties arise, for example, in parametric PDE contexts when $\widetilde{\mathcal{T}}$ arises as the discretized PDE solution operator on a spatial mesh that is coarser (and hence less trusted) than the mesh corresponding to $\mathcal{T}$. The decreased accuracy/trustworthiness of $\widetilde{\mathcal{T}}$ is balanced by its decreased cost, so that employment of $\widetilde{\mathcal{T}}$ may not furnish precise high-fidelity information, but may provide useful knowledge in terms of dependence on the parameter $\boldsymbol{p}$ with substantially reduced cost.

In the context of constructing our emulator (1.2), our core assumption is that the low-fidelity operator $\widetilde{\mathcal{T}}$ is cheap enough so that full exploration of the response over the sampled parameter set $\{\boldsymbol{p}_i\}_{i\in[N]}$ is more computationally feasible, resulting in a vector $\tilde{\boldsymbol{b}} \in \mathbb{R}^N$ with low-fidelity entries

$$\tilde{\boldsymbol{b}}(n) = \sqrt{w_n}\widetilde{\mathcal{T}}(\boldsymbol{p}_n). \tag{1.16}$$

Of course, one may propose constructing the emulator $\mathcal{T}$ in (1.2) by simply replacing $\boldsymbol{b}$ by $\tilde{\boldsymbol{b}}$, but this restricts the accuracy of the emulator $\mathcal{T}$ to the potentially bad accuracy of $\widetilde{\mathcal{T}}$. In this paper, we propose a more sophisticated use of $\tilde{\boldsymbol{b}}$, in conjunction with a single sparse sketch of $\boldsymbol{b}$, that retains some accuracy characteristics of $\boldsymbol{x}^*$.

1.4 Bi-fidelity boosting (BFB) in sketched least squares problems

In practice, one often requires the probability of successfully obtaining a good approximation $\boldsymbol{x}^*$ associated with a random sketch from section 1.3.2 to be sufficiently close to 1, and one way to achieve this with fixed sketch size is through a boosting procedure. Assuming the availability of a collection of sketching matrices $\{\boldsymbol{S}_\ell \in \mathbb{R}^{m\times N}\}_{\ell\in[L]}$, one computes the residual for the $\boldsymbol{S}_\ell$-sketched solution (i.e., $\|\boldsymbol{A}(\boldsymbol{S}_\ell\boldsymbol{A})^\dagger(\boldsymbol{S}_\ell\boldsymbol{b}) - \boldsymbol{b}\|_2$) for each $\boldsymbol{S}_\ell$ and then selects the one that yields the smallest residual for use. Even if each sketch is sparse, this straightforward procedure inflates the required sampling cost of the forward model $\mathcal{T}$ by the factor L, which may be computationally prohibitive. To ameliorate this boosting cost, we employ a bi-fidelity strategy.

In Section 1.4.1 we present our proposed algorithm for quadrature sampling which leverages sketching BFB. Sections 1.4.2 and 1.4.3 give our pre-asymptotic and asymptotic analysis results, respectively. We collect some preliminary technical results in section 1.4.4, and prove our pre-asymptotic results in section 1.4.5. The asymptotic result is proven in Appendix A.2. We end with section 1.4.6 that provides results for random sketches achieving the (ϵ, δ) condition in Definition 1.3.1.

1.4.1 Proposed algorithm

A distinguishing feature of the least squares problem in our setup is that full information of the high-fidelity data $\boldsymbol{b}$ is unaffordable due to computational restrictions; instead, we can only afford to generate a small number of entries of $\boldsymbol{b}$. Meanwhile, the low-fidelity data vector $\tilde{\boldsymbol{b}} \in \mathbb{R}^N$ that exhibits some type of correlation with $\boldsymbol{b}$ is readily available for repeated use. (This correlation-like condition is quantifying through the parameter ν introduced in Theorem 1.4.2.) We propose a modified boosting procedure, where the boosting phase of a sketched least squares problem replaces high-fidelity data with low-fidelity data to find the "best" sketching operator and then employs this best sketch directly with high-fidelity data to compute an approximate least squares solution. This procedure is outlined in Algorithm 2.

Algorithm 2: Bi-Fidelity Quadrature Boosting (BFB)

Input: design matrix $\boldsymbol{A}$, low-fidelity vector $\tilde{\boldsymbol{b}}$, method for computing entries of the high-fidelity vector $\boldsymbol{b}$, collection of sketches for boosting $\{\boldsymbol{S}_\ell\}_{\ell\in[L]}$

Output: an approximate solution $\hat{\boldsymbol{x}}_{\mathsf{BFB}}$ to (1.2)

1: **for** $\ell \in [L]$ **do**

2: compute the ℓ-th sketched solution $\hat{\boldsymbol{x}}_\ell$ using the low-fidelity data:

$$\hat{\boldsymbol{x}}_\ell = \underset{\boldsymbol{x}\in\mathbb{R}^d}{\arg\min} \left\| \boldsymbol{S}_\ell \boldsymbol{A}\boldsymbol{x} - \boldsymbol{S}_\ell \tilde{\boldsymbol{b}} \right\|_2 \tag{1.17}$$

3: **end for**

4: find the best low-fidelity sketch index ℓ^* using boosting:

$$\ell^* = \underset{\ell\in[L]}{\arg\min} \|\boldsymbol{A}\hat{\boldsymbol{x}}_\ell - \tilde{\boldsymbol{b}}\|_2 \tag{1.18}$$

5: use sketch $\boldsymbol{S}_{\ell^*}$ to compute an approximate solution to (1.2):

$$\hat{\boldsymbol{x}}_{\mathsf{BFB}} = \underset{\boldsymbol{x}\in\mathbb{R}^d}{\arg\min} \|\boldsymbol{S}_{\ell^*}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{S}_{\ell^*}\boldsymbol{b}\|_2 \quad \texttt{// Requires computing } m \texttt{ entries of } \boldsymbol{b} \tag{1.19}$$

The oracle sketch in this scenario is the one identified by the boosting strategy operating directly on the high-fidelity least squares problem, which is computationally unaffordable:

$$\ell^{**} = \underset{\ell\in[L]}{\arg\min} \|\boldsymbol{A}\boldsymbol{x}_\ell^{**} - \boldsymbol{b}\|_2^2, \quad \text{where } \boldsymbol{x}_\ell^{**} = \underset{\boldsymbol{x}\in\mathbb{R}^d}{\arg\min} \|\boldsymbol{S}_\ell\boldsymbol{A}\boldsymbol{x} - \boldsymbol{S}_\ell\boldsymbol{b}\|_2. \tag{1.20}$$

In the coming sections we will theoretically investigate the sketch transferability between high- and

low-fidelity boosting, i.e., when the residual associated to $\hat{\boldsymbol{x}}_{\mathsf{BFB}}$, the solution produced by Algorithm 2, is comparable to the residual associated to $\hat{\boldsymbol{x}}_{\ell^{**}}$.

We divide our analysis into two cases: Our first analysis frames performance of Algorithm 2 in terms of an *optimality coefficient*, defined in (1.21), which measures the quality of the least squares residual for a particular sketch $\boldsymbol{S}$; we provide pre-asymptotic analysis with quantitative results that provides qualitative guidance on how the BFB algorithm behaves in terms of the tradeoff in the number of sketches L versus the optimality coefficient (see the discussion following Theorem 1.4.4). Our second theoretical result is an asymptotic analysis with Gaussian sketches that confirms the intuition that the probabilistic correlations between the low- and high-fidelity random sketches is high when $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$ have high vector correlations (see the discussion around Theorem 1.4.5).

For analysis purposes we make the following assumption.

Assumption 1.4.1. Assume that neither $\tilde{\boldsymbol{b}}$ nor $\boldsymbol{b}$ lie in range$(\boldsymbol{A})$, i.e., we assume $\tilde{\boldsymbol{b}}, \boldsymbol{b} \notin \text{range}(\boldsymbol{A})$.

This is a reasonable assumption. If $\boldsymbol{b} \in \text{range}(\boldsymbol{A})$, then it would be possible to solve the high-fidelity least squares problem exactly by sampling $m = d$ linearly independent rows of $\boldsymbol{A}$ and the corresponding rows of $\boldsymbol{b}$. In this case, it is therefore easy to solve (1.2) and only requires accessing d rows of $\boldsymbol{b}$. Similarly, if $\tilde{\boldsymbol{b}} \in \text{range}(\boldsymbol{A})$ then it would be easy to compute a sketch $\boldsymbol{S}_\ell$ which only samples $m = d$ rows and achieves zero error in Line 4 of Algorithm 2, therefore making the boosting procedure vacuous.

1.4.2 Pre-asymptotic analysis via optimality coefficients

We introduce the following measure of relative error difference between the sketched and optimal solutions:

$$\mu_{\boldsymbol{A}}(\boldsymbol{b}, \boldsymbol{S}) \coloneqq \sqrt{\frac{r_{\boldsymbol{S}}^2(\boldsymbol{A}, \boldsymbol{b}) - r^2(\boldsymbol{A}, \boldsymbol{b})}{r^2(\boldsymbol{A}, \boldsymbol{b})}} \overset{(*)}{=} \frac{\|(\boldsymbol{S}\boldsymbol{Q})^\dagger \boldsymbol{S}\boldsymbol{Q}_\perp \boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2}{\|\boldsymbol{Q}_\perp \boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2}, \tag{1.21}$$

where $\boldsymbol{Q} = \text{orth}(\boldsymbol{A})$, and the second equality marked $(*)$ is valid if $\text{rank}(\boldsymbol{S}\boldsymbol{A}) = \text{rank}(\boldsymbol{A})$, which we establish in Lemma 1.4.7. For notational simplicity we usually drop the subscript and write $\mu(\boldsymbol{b}, \boldsymbol{S})$ when $\boldsymbol{A}$ is clear from context, but we emphasize that μ does depend on $\boldsymbol{A}$. Note that

$r(\boldsymbol{A}, \boldsymbol{b}) = \|\boldsymbol{Q}_\perp \boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2 > 0$ due to Assumption 1.4.1, so the denominator in (1.21) is nonzero. We call μ the **optimality coefficient**. Smaller values of μ are better in practice: $\mu = 0$ implies the sketch achieves perfect reconstruction of the data relative to the full least squares solution.

We provide two main theoretical results which shed light on the performance of Algorithm 2 from two different perspectives. The first result shows that with an appropriate choice of the sketches $\{\boldsymbol{S}_\ell\}_{\ell \in [L]}$, Algorithm 2 produces a solution whose relative error is close to that of the oracle sketch solution in (1.20). Note that it would be straightforward to provide such guarantees if $r_{\boldsymbol{S}}(\boldsymbol{A}, \tilde{\boldsymbol{b}}) \leq r_{\boldsymbol{S}'}(\boldsymbol{A}, \tilde{\boldsymbol{b}})$ implied $r_{\boldsymbol{S}}(\boldsymbol{A}, \boldsymbol{b}) \leq r_{\boldsymbol{S}'}(\boldsymbol{A}, \boldsymbol{b})$, in which case $\ell^* = \ell^{**}$. This may happen, for instance, when $\tilde{\boldsymbol{b}}$ and $\boldsymbol{b}$ differ by a scaling. This monotone property of r when replacing $\boldsymbol{b}$ with $\tilde{\boldsymbol{b}}$ is unfortunately unlikely to hold in practice. Our result, which appears in Theorem 1.4.2, identifies alternative conditions that ensure $\boldsymbol{S}_{\ell^*}$ is a "good" sketch for the high-fidelity data.

Theorem 1.4.2. *Fix a positive integer L and suppose $\delta, \varepsilon \in (0, 1]$. If $\{\boldsymbol{S}_\ell\}_{\ell \in [L]}$ is a sequence of i.i.d. random matrices whose distribution is an $(\varepsilon, \frac{\delta}{L})$ pair for $(\boldsymbol{Q}, \boldsymbol{h})$, where*

$$\boldsymbol{h} \coloneqq \left((\boldsymbol{P}_{\boldsymbol{Q}_\perp} \boldsymbol{b})_{\mathcal{P}} - (\boldsymbol{P}_{\boldsymbol{Q}_\perp} \tilde{\boldsymbol{b}})_{\mathcal{P}} \right)_{\mathcal{P}} \quad \text{and} \quad \boldsymbol{Q} \coloneqq \operatorname{orth}(\boldsymbol{A}), \tag{1.22}$$

then with probability at least $1 - \delta$,

$$\mu(\boldsymbol{b}, \boldsymbol{S}_{\ell^*}) \leq \mu(\boldsymbol{b}, \boldsymbol{S}_{\ell^{**}}) + 2\sqrt{6(1 - \nu)\varepsilon}, \tag{1.23}$$

where ν denotes the absolute correlation coefficient between $\boldsymbol{P}_{\boldsymbol{Q}_\perp} \boldsymbol{b}$ and $\boldsymbol{P}_{\boldsymbol{Q}_\perp} \tilde{\boldsymbol{b}}$:

$$\nu \coloneqq \left| \operatorname{corr}(\boldsymbol{P}_{\boldsymbol{Q}_\perp} \boldsymbol{b}, \boldsymbol{P}_{\boldsymbol{Q}_\perp} \tilde{\boldsymbol{b}}) \right|. \tag{1.24}$$

In addition, on the event where (1.23) is true, we also have that (1.10) holds with $\boldsymbol{S} = \boldsymbol{S}_\ell$ for every $\ell \in [L]$.

Theorem 1.4.2 shows that if a sketch satisfies an $(\varepsilon, \delta/L)$ condition for the pair $\boldsymbol{Q}$ and an element $\boldsymbol{h}$ of $\operatorname{range}(\boldsymbol{Q}_\perp)$, then we are able to prove bounds on the low-fidelity boosted optimality coefficient $\mu(\boldsymbol{b}, \boldsymbol{S}_{\ell^*})$ relative to the oracle high-fidelity boosted optimality coefficient $\mu(\boldsymbol{b}, \boldsymbol{S}_{\ell^{**}})$. This is quite a general statement that accommodates a wide range of sketching operators. The condition

on the operators $\{\boldsymbol{S}_\ell\}_{\ell\in[L]}$ is, for example, satisfied by all sketching operators in Sections 1.3.2.2–1.3.2.4 when the embedding dimension m is sufficiently large. More precise statements for the leverage score and Gaussian sketches are provided in Theorem 1.4.11.

In order to achieve a good approximate solution when applying sketching techniques in least squares problems the sketching operator must preserve the relevant geometry of the problem. In particular, it is key that $\boldsymbol{Q}$ and $\boldsymbol{P}_{\boldsymbol{Q}_\perp}\boldsymbol{b}$ remain roughly orthogonal after the sketching operator has been applied. This importance of preserving $\boldsymbol{P}_{\boldsymbol{Q}_\perp}\boldsymbol{b}$ in the sketching phase when $\boldsymbol{b}$ is replaced by low-fidelity data $\tilde{\boldsymbol{b}}$ manifests in Theorem 1.4.2 through the correlation parameter ν.

Remark 1.4.3. Equation (1.23) suggests that $\boldsymbol{S}_{\ell^*}$ is "good" when ν is large. This explicitly requires high parametric correlation between the portions of $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$ that lie orthogonal to the range of $\boldsymbol{A}$. A more subtle sufficient condition ensuring large ν is furnished by our discussion following Proposition 1.4.8, which provides a lower bound for ν in terms of other parameters.

Theorem 1.4.2 does not provide a concrete strategy for how the sketches used in boosting are chosen or constructed. However, near-optimal sketches (in particular satisfying our required (ϵ, δ) pair condition) are known to be produced through the well-known randomized approaches discussed in sections 1.3.2.2-1.3.2.4. Precise statements for such sketch estimates are given later in by Theorem 1.4.11 in section 1.4.6, but it is appropriate for us to establish here that combining Theorem 1.4.2 with good sketching techniques results in explicit and illuminating theory for Algorithm 2. In particular, one expects a tradeoff between the values of ν and L: boosting with a large number L of sketches should work up to a threshold determined by the amount of correlation between $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$. I.e., any accuracy gained by BFB should be limited by how correlated the low- and high-fidelity models are, and one expects this to manifest in a relationship between L and ν. The theory we develop below reveals this tradeoff. We focus on generating the sketches $\{\boldsymbol{S}_\ell\}_{\ell\in[L]}$ through leverage score sampling, as described explicitly by (1.14) in section 1.3.2.2. We briefly discuss afterward that one could generalize the result to more general sketches.

Theorem 1.4.4. *Let $\delta, \epsilon \in (0, 1/2)$ and $L \in \mathbb{N}$ be chosen, and assume*

$$d \leq \frac{\delta}{4} \exp\left(\frac{2}{35\epsilon\delta}\right). \tag{1.25}$$

Now consider Algorithm 2, where $\{\boldsymbol{S}_\ell\}_{\ell \in [L]}$ are iid samples of a leverage score sketching operator defined in (1.14), with the sampling requirement

$$m \geq \frac{4dL}{\epsilon\delta}. \tag{1.26}$$

Then each $\boldsymbol{S}_\ell$ satisfies an $(\epsilon/L, \delta/2)$ condition for the pair $(\boldsymbol{Q}, \boldsymbol{h})$, and with probability at least $1-\delta$, we have

$$r^2_{\boldsymbol{S}_{\ell^*}}(\boldsymbol{A}, \boldsymbol{b}) \leq \left[1 + \frac{\epsilon}{L}\tau\right] r^2(\boldsymbol{A}, \boldsymbol{b}), \tag{1.27}$$

where

$$\tau = \tau(\epsilon, \delta, \nu, L) = 24L(1-\nu) + \frac{\delta}{2}\left(1 + 4\sqrt{6(1-\nu)\epsilon}\right).$$

The results above give explicit behavior of the BFB residual via a concrete sketching strategy for Algorithm 2. Note in particular that the sampling requirement $m = \mathcal{O}(L/\epsilon)$ in (1.26) means that *without* boosting and simply generating one sketch $\boldsymbol{S}$ according to (1.26), which requires m high-fidelity samples (equivalent to the number from BFB), we expect that the residual from this one sketch behaves like

$$r^2_{\boldsymbol{S}}(\boldsymbol{A}, \boldsymbol{b}) \sim \left(1 + \frac{\epsilon}{L}\right) r^2(\boldsymbol{A}, \boldsymbol{b}).$$

Comparing the above to (1.27), note that the only difference is the appearance of τ, and hence we expect BFB to be useful (compared to an equivalent number of high-fidelity samples devoted to a non-boosting strategy) when $\tau \leq 1$, which requires,

$$L \lesssim \frac{1}{1-\nu}.$$

I.e., boosting with L sketches is useful in BFB up to a threshold $\sim 1/(1-\nu)$. Boosting with *more* than this threshold level of sketches causes the error bound to saturate at a level determined by

$1-\nu$. Since ν is the correlation between the range($\boldsymbol{A}$)-orthogonal components of $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$, we conclude that highly correlated range-orthogonal residuals (large values of ν very close to 1) are optimal for BFB in the sense that sketching with large L will be effective.

A second observation we make is that the the $m \sim L$ requirement (1.26) is theoretically suboptimal. In particular, we show in Theorem 1.4.11 that stronger coherence-like conditions on the matrix $\boldsymbol{A}$ imply that leverage score sketching with $m \sim \log L$ is sufficient to achieve the requisite $(\epsilon/L, \delta)$ condition, see (1.57) in Theorem 1.4.11. We also note that Gaussian sketches only require $m \sim \log L$ samples (see (1.54)), and one can achieve the (ϵ, δ) condition *on average* using $m \sim \log L$ samples (see, e.g., [349, Equation (2.18)]. Finally, if (1.25) is violated, then indeed $m \sim \log L$ (see (1.55) and the intermediate computation in (1.28)) for leverage score sketches. Thus, we expect in practice that $m \sim \log L$ samples are sufficient.

We give the proof of theorem 1.4.4 below to demonstrate how it relies on Theorem 1.4.2; we will prove Theorem 1.4.2 in the coming sections.

Proof of Theorem 1.4.4. We start by making two conclusions from the conditions (1.25) and (1.26). First, under these conditions,

$$35 \log\left(\frac{4d}{\delta}\right) \leq \frac{2}{\epsilon\delta} \quad \Longrightarrow \quad m \geq d \max\left\{35 \log\left(\frac{4dL}{(\delta/2)}\right), \frac{2L}{\epsilon(\delta/2)}\right\}, \tag{1.28}$$

implying that condition (1.55) holds, so that result 2 from Theorem 1.4.11 guarantees that the distribution from which the $\boldsymbol{S}_\ell$ sketches are drawn satisfies and $(\epsilon, \frac{\delta}{2L})$ condition. Thus, theorem 1.4.2 states that there is an event E_1 such that

$$\Pr(E_1) \geq 1 - \delta/2, \quad \text{On event } E_1, \text{ then (1.23) holds.} \tag{1.29}$$

The above is our first conclusion. For our second conclusion, we note that (1.26) and (1.25) imply,

$$m \geq \frac{2d}{\left[\frac{\epsilon}{L}\left(\frac{\delta}{2}\right)^{1-1/L}\right]\left(\frac{\delta}{2}\right)^{1/L}},$$

so that again we satisfy (1.55) (employing a variation of the argument (1.28)), and so by Theorem

1.4.11, the distribution from which $\boldsymbol{S}_\ell$ is drawn satisfies an $(\tilde{\epsilon}, \tilde{\delta})$ condition for $(\boldsymbol{A}, \boldsymbol{b})$, where,

$$\tilde{\epsilon} := \frac{\epsilon}{L}\left(\frac{\delta}{2}\right)^{1-1/L}, \qquad \tilde{\delta} := \left(\frac{\delta}{2}\right)^{1/L}.$$

Therefore with probability at least $1 - \tilde{\delta}$,

$$r^2_{\boldsymbol{S}_\ell}(\boldsymbol{A}, \boldsymbol{b}) \leq (1 + \tilde{\epsilon}) r^2(\boldsymbol{A}, \boldsymbol{b}),$$

so that a union bound implies that there is an event E_2 on which our second conclusion holds:

$$\Pr(E_2) \geq 1 - \left(\tilde{\delta}\right)^L = 1 - \delta/2 \quad \text{On event } E_2, \text{ then } \min_{\ell \in [L]} r^2_{\boldsymbol{S}_\ell}(\boldsymbol{A}, \boldsymbol{b}) \leq (1 + \tilde{\epsilon}) r^2_{\boldsymbol{S}_{\ell^{**}}}(\boldsymbol{A}, \boldsymbol{b}). \tag{1.30}$$

We now observe that for any $\eta > 0$, the bound

$$|\mu(\boldsymbol{b}, \boldsymbol{S}_{\ell^*}) - \mu(\boldsymbol{b}, \boldsymbol{S}_{\ell^{**}})| \leq \eta \tag{1.31}$$

implies that

$$r^2_{\boldsymbol{S}_{\ell^*}}(\boldsymbol{A}, \boldsymbol{b}) \leq r^2_{\boldsymbol{S}_{\ell^{**}}}(\boldsymbol{A}, \boldsymbol{b}) + r^2(\boldsymbol{A}, \boldsymbol{b})\big(\eta^2 + 2\eta\mu(\boldsymbol{b}, \boldsymbol{S}_{\ell^{**}})\big). \tag{1.32}$$

Thus, $E_1 \cap E_2$ occurs with probability at least $1 - \delta$, and on this event (1.29) ensures that η is given by the right-hand side of (1.23). Also, on this event (1.30) implies that $\mu(\boldsymbol{b}, \boldsymbol{S}_{\ell^{**}}) = \tilde{\epsilon}$, i.e., $r^2_{\boldsymbol{S}_{\ell^{**}}}(\boldsymbol{A}, \boldsymbol{b}) \leq (1 + \tilde{\epsilon}) r^2(\boldsymbol{A}, \boldsymbol{b})$. Using these expressions in the above inequality, simplifying, and using $(\delta/2)^{1-1/L} \leq \delta/2$ yields the result (1.27). □

We emphasize that the proof above shows how Theorem 1.4.2 can be used to prove results like Theorem 1.4.4 for more general sketches.

1.4.3 Asymptotic analysis via probabilistic correlation

We provide alternative analysis of Algorithm 2 motivated by the following intuition: If $\mu(\boldsymbol{b}, \boldsymbol{S})$ and $\mu(\tilde{\boldsymbol{b}}, \boldsymbol{S})$ are probabilistically correlated in some sense, then we expect that Algorithm 2 should produce a sketching operator $\boldsymbol{S}_{\ell^*}$ that is close to the oracle sketch $\boldsymbol{S}_{\ell^{**}}$. We give a technical verification of this intuition below in Theorem 1.4.5, providing an asymptotic lower bound on a certain measure of correlation between the two optimality coefficients when $\boldsymbol{S}$ is a Gaussian sketching operator.

Theorem 1.4.5. *If $\boldsymbol{S}$ is a Gaussian sketch, then*

$$\liminf_{m\to\infty} \operatorname{corr}(\mu^2(\boldsymbol{b},\boldsymbol{S}),\mu^2(\tilde{\boldsymbol{b}},\boldsymbol{S})) \geq \frac{\|\boldsymbol{P}_{\boldsymbol{Q}_\perp}\boldsymbol{b}_{\mathcal{P}}\|_2^2 - \sqrt{6}\min\{\|\boldsymbol{P}_{\boldsymbol{Q}_\perp}(\boldsymbol{b}_{\mathcal{P}} \pm \tilde{\boldsymbol{b}}_{\mathcal{P}})\|_2\}}{\|\boldsymbol{P}_{\boldsymbol{Q}_\perp}\tilde{\boldsymbol{b}}_{\mathcal{P}}\|_2^2}, \tag{1.33}$$

where $\boldsymbol{b}_{\mathcal{P}}, \tilde{\boldsymbol{b}}_{\mathcal{P}}$ are normalized versions of $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$, respectively, and the minimum is taken over the two $\pm$ options. Moreover, if

$$\varphi := \frac{|\langle \boldsymbol{b}, \tilde{\boldsymbol{b}}\rangle|}{\|\boldsymbol{b}\|_2\|\tilde{\boldsymbol{b}}\|_2} \geq \frac{\|\boldsymbol{P}_{\boldsymbol{Q}}\boldsymbol{b}\|_2}{\|\boldsymbol{b}\|_2} := \kappa, \tag{1.34}$$

then we further have that

$$\liminf_{m\to\infty} \operatorname{corr}(\mu^2(\boldsymbol{b},\boldsymbol{S}),\mu^2(\tilde{\boldsymbol{b}},\boldsymbol{S})) \geq (1-\kappa^2) - \frac{\sqrt{12(1-\varphi)}}{(\varphi-\kappa)^2}. \tag{1.35}$$

In Theorem 1.4.5 we restrict to Gaussian sketches and consider $\operatorname{corr}(\mu^2(\boldsymbol{b},\boldsymbol{S}),\mu^2(\tilde{\boldsymbol{b}},\boldsymbol{S}))$ (rather than the more natural quantity $\operatorname{corr}(\mu(\boldsymbol{b},\boldsymbol{S}),\mu(\tilde{\boldsymbol{b}},\boldsymbol{S}))$) in order to make analysis tractable. In general $\operatorname{corr}(\mu(\boldsymbol{b},\boldsymbol{S}),\mu(\tilde{\boldsymbol{b}},\boldsymbol{S}))$ and $\operatorname{corr}(\mu^2(\boldsymbol{b},\boldsymbol{S}),\mu^2(\tilde{\boldsymbol{b}},\boldsymbol{S}))$ may have significantly different statistical properties. However, if either of them is close to 1, then that would indicate a monotonically increasing (although not necessarily linear) relationship between $\mu(\boldsymbol{b},\boldsymbol{S})$ and $\mu(\tilde{\boldsymbol{b}},\boldsymbol{S})$, and when such a relationship holds we expect the boosting procedure in Algorithm 2 to work well. While we restrict to Gaussian sketches, this probabilistic model is usually a good indicator of how other sketches perform [357, Remark 8.2]. I.e., we expect the result to carry over to the random sampling-based sketches (e.g., leverage scores) that we consider. We verify this numerically in Section 1.5.

Remark 1.4.6. The lower bound in (1.35) is useful only when the right-hand side is close to 1, which roughly requires φ to be large and κ to be small. See Remark 1.4.9 for how this condition relates to Theorem 1.4.2.

The rest of this section is organized as follows. Section 1.4.4 derives some preliminary technical results. Section 1.4.5 then proves Theorem 1.4.2. Section 1.4.6 provides theoretical guarantees for when various sketches satisfy the (ε,δ) pair condition in Definition 1.3.1 and discuss how this condition in turn ensures that those sketching operators satisfy the requirements in Theorem 1.4.2. The proof of Theorem 1.4.5 is given in Appendix A.2.

1.4.4 Preliminary technical results

Our first task is to understand how the optimal residual $r(\boldsymbol{A}, \boldsymbol{b})$ compares to $r_{\boldsymbol{S}}(\boldsymbol{A}, \boldsymbol{b})$. Throughout this section let $\boldsymbol{Q} = \operatorname{orth}(\boldsymbol{A})$.

Lemma 1.4.7. *Given a sketch matrix $\boldsymbol{S}$, assume $\ker(\boldsymbol{S}) \cap \operatorname{range}(\boldsymbol{A}) = \{\boldsymbol{0}\}$, or, equivalently, $\operatorname{rank}(\boldsymbol{SA}) = \operatorname{rank}(\boldsymbol{A})$. Then we have,*

$$r_{\boldsymbol{S}}^2(\boldsymbol{A}, \boldsymbol{b}) = r^2(\boldsymbol{A}, \boldsymbol{b}) + \|(\boldsymbol{SQ})^\dagger \boldsymbol{SQ}_\perp \boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2^2. \tag{1.36}$$

Proof. Under the assumption $\ker(\boldsymbol{S}) \cap \operatorname{range}(\boldsymbol{A}) = \{\boldsymbol{0}\}$, the sketched least squares problem reproduces elements of $\operatorname{range}(\boldsymbol{A})$: For any $\boldsymbol{c} \in \operatorname{range}(\boldsymbol{A})$,

$$\boldsymbol{A}(\boldsymbol{SA})^\dagger \boldsymbol{Sc} = \boldsymbol{c}. \tag{1.37}$$

The solution to the sketched least squares problem (1.7) is $(\boldsymbol{SA})^\dagger \boldsymbol{Sb}$. Combining this fact with (1.8) and (1.37) yields

$$r_{\boldsymbol{S}}^2(\boldsymbol{A}, \boldsymbol{b}) = \|\boldsymbol{b} - \boldsymbol{A}(\boldsymbol{SA})^\dagger \boldsymbol{Sb}\|_2^2 = \|\boldsymbol{b} - \boldsymbol{A}(\boldsymbol{SA})^\dagger \boldsymbol{S}(\boldsymbol{QQ}^T + \boldsymbol{Q}_\perp \boldsymbol{Q}_\perp^T)\boldsymbol{b}\|_2^2 = r^2(\boldsymbol{A}, \boldsymbol{b}) + \|(\boldsymbol{SQ})^\dagger \boldsymbol{SQ}_\perp \boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2^2. \tag{1.38}$$

□

We conclude that $r_{\boldsymbol{S}}(\boldsymbol{A}, \boldsymbol{b})$ is comparable to $r(\boldsymbol{A}, \boldsymbol{b})$ if and only if $\|(\boldsymbol{SQ})^\dagger \boldsymbol{SQ}_\perp \boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2^2$ is small.

The quantities ν, φ and κ defined in (1.24) and (1.34) are related by the following inequality.

Proposition 1.4.8. *Assume $\varphi \geq \kappa$. Then we have the two inequalities,*

$$\nu \geq \varphi - \kappa \min\left\{1, \sqrt{2(1 - \varphi + \kappa)}\right\}. \tag{1.39}$$

$$\nu \geq \varphi - (\varphi\tilde{\kappa} + \sqrt{1 - \varphi^2}) \min\left\{1, \sqrt{2(1 - \varphi + \varphi\tilde{\kappa} + \sqrt{1 - \varphi^2})}\right\}. \tag{1.40}$$

where

$$\tilde{\kappa} := \frac{\|\boldsymbol{P}_{\boldsymbol{Q}} \tilde{\boldsymbol{b}}\|_2}{\|\tilde{\boldsymbol{b}}\|_2}, \tag{1.41}$$

measures the relative energy of the low-fidelity vector in the range of $\boldsymbol{A}$.

Proof. We first prove (1.39). Since correlation coefficients are scale-invariant, without loss of generality we assume $\|\boldsymbol{b}\|_2 = \|\tilde{\boldsymbol{b}}\|_2 = 1$. Write down the orthogonal decomposition of $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$ in $\boldsymbol{Q} \oplus \boldsymbol{Q}_\perp$ as follows:

$$\begin{aligned} \boldsymbol{b} &= \underbrace{\boldsymbol{P}_{\boldsymbol{Q}}\boldsymbol{b}}_{\boldsymbol{b}_1} + \underbrace{\boldsymbol{P}_{\boldsymbol{Q}_\perp}\boldsymbol{b}}_{\boldsymbol{b}_2}, \\ \tilde{\boldsymbol{b}} &= \underbrace{\boldsymbol{P}_{\boldsymbol{Q}}\tilde{\boldsymbol{b}}}_{\tilde{\boldsymbol{b}}_1} + \underbrace{\boldsymbol{P}_{\boldsymbol{Q}_\perp}\tilde{\boldsymbol{b}}}_{\tilde{\boldsymbol{b}}_2}. \end{aligned} \tag{1.42}$$

Notice that $\|\boldsymbol{b}_1\|_2^2 + \|\boldsymbol{b}_2\|_2^2 = \|\tilde{\boldsymbol{b}}_1\|_2^2 + \|\tilde{\boldsymbol{b}}_2\|_2^2 = 1$. It follows from the Cauchy–Schwarz inequality and the definitions in (1.24) and (1.34) that

$$\nu = \frac{|\langle \boldsymbol{b}_2, \tilde{\boldsymbol{b}}_2\rangle|}{\|\boldsymbol{b}_2\|_2\|\tilde{\boldsymbol{b}}_2\|_2} \geq |\langle \boldsymbol{b}, \tilde{\boldsymbol{b}}\rangle - \langle \boldsymbol{b}_1, \tilde{\boldsymbol{b}}_1\rangle| \geq \varphi - \|\boldsymbol{b}_1\|_2\|\tilde{\boldsymbol{b}}_1\|_2 = \varphi - \kappa\|\tilde{\boldsymbol{b}}_1\|_2 \geq \varphi - \kappa. \tag{1.43}$$

The last inequality can be replaced by a more accurate estimate for $\|\tilde{\boldsymbol{b}}_1\|_2$:

$$\varphi = |\langle \boldsymbol{b}, \tilde{\boldsymbol{b}}\rangle| = |\langle \boldsymbol{b}_1, \tilde{\boldsymbol{b}}_1\rangle + \langle \boldsymbol{b}_2, \tilde{\boldsymbol{b}}_2\rangle| \leq \|\boldsymbol{b}_1\|_2\|\tilde{\boldsymbol{b}}_1\|_2 + \|\boldsymbol{b}_2\|_2\|\tilde{\boldsymbol{b}}_2\|_2 \leq \kappa + \|\tilde{\boldsymbol{b}}_2\|_2 = \sqrt{1 - \|\tilde{\boldsymbol{b}}_1\|_2^2} + \kappa, \tag{1.44}$$

which can be reorganized as

$$\|\tilde{\boldsymbol{b}}_1\|_2 \leq \sqrt{1 - (\varphi - \kappa)^2} = \sqrt{(1 - \varphi + \kappa)(1 + \varphi - \kappa)} \leq \sqrt{2(1 - \varphi + \kappa)}. \tag{1.45}$$

Combining (1.43) and (1.45) finishes the proof of (1.39).

To show (1.40), we again assume $\|\boldsymbol{b}\|_2 = \|\tilde{\boldsymbol{b}}\|_2 = 1$, so that,

$$\kappa = \|\boldsymbol{P}_{\boldsymbol{Q}}\boldsymbol{b}\|_2 = \|\boldsymbol{P}_{\boldsymbol{Q}}(\boldsymbol{P}_{\tilde{\boldsymbol{b}}}\boldsymbol{b} + \boldsymbol{b} - \boldsymbol{P}_{\tilde{\boldsymbol{b}}}\boldsymbol{b})\|_2 \leq \varphi\|\boldsymbol{P}_{\boldsymbol{Q}}\tilde{\boldsymbol{b}}\|_2 + \|\boldsymbol{b} - \boldsymbol{P}_{\tilde{\boldsymbol{b}}}\boldsymbol{b}\|_2 = \varphi\tilde{\kappa} + \sqrt{1 - \varphi^2}. \tag{1.46}$$

Plugging this into (1.39) and noting that the right-hand side of (1.39) is decreasing in κ yields (1.40). □

The main appeal of (1.40) is that the quantity $\tilde{\kappa}$ involves only low-fidelity data, and hence can be estimated. I.e., (1.40) gives a more practically computable lower bound for ν, involving one quantity $\tilde{\kappa}$ that depends only on low-fidelity data $\tilde{\boldsymbol{b}}$, and the correlation φ between $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$.

Remark 1.4.9. Recall that our main convergence result, Theorem 1.4.2, has more attractive bounds when ν is large. By (1.39), ν is large if $\varphi \approx 1$ and $\varphi \gg \kappa$, which coincides with sufficient

conditions to ensure attractive bounds in (1.35) in Theorem 1.4.5. (Cf. Remark 1.4.6.) Thus, $\varphi \gg \kappa$ is a unifying condition under which both of our main theoretical results, Theorem 1.4.2 and Theorem 1.4.5, provide useful bounds. The condition $\varphi \gg \kappa$ means that the correlation between $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$ is high and strongly dominates the relative energy of $\boldsymbol{b}$ in range($\boldsymbol{A}$). This condition may seem counterintuitive as it requires the high-fidelity solution to have a relatively large residual. Since μ is defined relative to $r(\boldsymbol{A}, \boldsymbol{b})$, a small $r_{\boldsymbol{S}_{\ell^*}}(\boldsymbol{A}, \boldsymbol{b})$ may still result in a large $\mu(\boldsymbol{b}, \boldsymbol{S}_{\ell^*})$ even if $r_{\boldsymbol{S}_{\ell^*}}(\boldsymbol{A}, \boldsymbol{b})$ is small but relatively large compared to $r(\boldsymbol{A}, \boldsymbol{b})$.

1.4.5 Proof of Theorem 1.4.2

We first consider the case $\operatorname{corr}(\boldsymbol{P}_{\boldsymbol{Q}_\perp}\boldsymbol{b}, \boldsymbol{P}_{\boldsymbol{Q}_\perp}\tilde{\boldsymbol{b}}) \geq 0$. Fixing $\ell \in [L]$, $\boldsymbol{S} = \boldsymbol{S}_\ell$, consider the event E of probability at least $1 - \delta/L$ where the rank condition in (1.10) holds. On this event, this rank condition with Lemma 1.4.7 implies that,

$$r^2_{\boldsymbol{S}}(\boldsymbol{A}, \boldsymbol{b}) - r^2(\boldsymbol{A}, \boldsymbol{b}) = \|(\boldsymbol{SQ})^\dagger \boldsymbol{SQ}_\perp \boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2^2,$$

allowing us to directly estimate the difference between $\mu(\boldsymbol{b}, \boldsymbol{S})$ and $\mu(\tilde{\boldsymbol{b}}, \boldsymbol{S})$ as follows:

$$\begin{aligned}
|\mu(\boldsymbol{b}, \boldsymbol{S}) - \mu(\tilde{\boldsymbol{b}}, \boldsymbol{S})| &= \left| \frac{\|(\boldsymbol{SQ})^\dagger \boldsymbol{SQ}_\perp \boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2}{\|\boldsymbol{Q}_\perp \boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2} - \frac{\|(\boldsymbol{SQ})^\dagger \boldsymbol{SQ}_\perp \boldsymbol{Q}_\perp^T \tilde{\boldsymbol{b}}\|_2}{\|\boldsymbol{Q}_\perp \boldsymbol{Q}_\perp^T \tilde{\boldsymbol{b}}\|_2} \right| \\
&\leq \left\| (\boldsymbol{SQ})^\dagger \boldsymbol{S} \left((\boldsymbol{P}_{\boldsymbol{Q}_\perp}\boldsymbol{b})_{\mathcal{P}} - (\boldsymbol{P}_{\boldsymbol{Q}_\perp}\tilde{\boldsymbol{b}})_{\mathcal{P}} \right) \right\|_2 \\
&= \|(\boldsymbol{P}_{\boldsymbol{Q}_\perp}\boldsymbol{b})_{\mathcal{P}} - (\boldsymbol{P}_{\boldsymbol{Q}_\perp}\tilde{\boldsymbol{b}})_{\mathcal{P}}\|_2 \|(\boldsymbol{SQ})^\dagger \boldsymbol{Sh}\|_2 \\
&= \sqrt{\|(\boldsymbol{P}_{\boldsymbol{Q}_\perp}\boldsymbol{b})_{\mathcal{P}}\|_2^2 + \|(\boldsymbol{P}_{\boldsymbol{Q}_\perp}\tilde{\boldsymbol{b}})_{\mathcal{P}}\|_2^2 - 2\langle (\boldsymbol{P}_{\boldsymbol{Q}_\perp}\boldsymbol{b})_{\mathcal{P}}, (\boldsymbol{P}_{\boldsymbol{Q}_\perp}\tilde{\boldsymbol{b}})_{\mathcal{P}} \rangle}\, \|(\boldsymbol{SQ})^\dagger \boldsymbol{Sh}\|_2 \\
&= \sqrt{2 - 2\nu} \cdot \|(\boldsymbol{SQ})^\dagger \boldsymbol{Sh}\|_2 \\
&= \sqrt{2 - 2\nu} \cdot \|\boldsymbol{Q}(\boldsymbol{SQ})^\dagger \boldsymbol{Sh}\|_2,
\end{aligned} \tag{1.47}$$

where the first inequality follows from the reverse triangle inequality, the second to last equality follows (1.24), and the final equality follows from unitary invariance of the operator norm. The case $\operatorname{corr}(\boldsymbol{P}_{\boldsymbol{Q}_\perp}\boldsymbol{b}, \boldsymbol{P}_{\boldsymbol{Q}_\perp}\tilde{\boldsymbol{b}}) < 0$ can be treated similarly by noting that the inequality on the second line of (1.47) still holds if the minus sign on the right-hand side is changed to a plus sign. The rest of the computation is then done similarly to the case with non-negative correlation.

Note that $(\boldsymbol{SQ})^\dagger \boldsymbol{Sh}$ is the $\boldsymbol{S}$-sketched least squares solution to $\min_{\boldsymbol{x}} \|\boldsymbol{Qx} - \boldsymbol{h}\|_2$. Also, note that $\boldsymbol{h} \in \text{range}(\boldsymbol{Q}_\perp)$. Using the residual bound in (1.10), the following also holds on our probabilistic event E:

$$\|\boldsymbol{Q}(\boldsymbol{SQ})^\dagger \boldsymbol{Sh}\|_2^2 + \|\boldsymbol{h}\|_2^2 = \|\boldsymbol{Q}(\boldsymbol{SQ})^\dagger \boldsymbol{Sh} - \boldsymbol{h}\|_2^2 \leq (1+\varepsilon)^2 \min_{\boldsymbol{x}\in\mathbb{R}^d} \|\boldsymbol{Qx} - \boldsymbol{h}\|_2^2 = (1+\varepsilon)^2 \|\boldsymbol{h}\|_2^2. \quad (1.48)$$

Rearranging terms and noting $\|\boldsymbol{h}\|_2 = 1$ yields $\|\boldsymbol{Q}(\boldsymbol{SQ})^\dagger \boldsymbol{Sh}\| \leq \sqrt{3\varepsilon}$, which is substituted into (1.47), implying that on an event E with probability at least $1 - \delta/L$, we have

$$|\mu(\boldsymbol{b}, \boldsymbol{S}) - \mu(\tilde{\boldsymbol{b}}, \boldsymbol{S})| \leq \sqrt{6(1-\nu)\varepsilon}. \quad (1.49)$$

Taking a union bound over $\ell \in [L]$ yields that, with probability at least $1 - \delta$,

$$\max_{\ell\in[L]} |\mu(\boldsymbol{b}, \boldsymbol{S}_\ell) - \mu(\tilde{\boldsymbol{b}}, \boldsymbol{S}_\ell)| \leq \sqrt{6(1-\nu)\varepsilon}. \quad (1.50)$$

Conditioning on the probabilistic event in (1.50) and using the definition of ℓ^* and ℓ^{**} finishes the proof:

$$\mu(\boldsymbol{b}, \boldsymbol{S}_{\ell^*}) \leq \mu(\tilde{\boldsymbol{b}}, \boldsymbol{S}_{\ell^*}) + \sqrt{6(1-\nu)\varepsilon} \leq \mu(\tilde{\boldsymbol{b}}, \boldsymbol{S}_{\ell^{**}}) + \sqrt{6(1-\nu)\varepsilon} \leq \mu(\boldsymbol{b}, \boldsymbol{S}_{\ell^{**}}) + 2\sqrt{6(1-\nu)\varepsilon}. \quad (1.51)$$

1.4.6 Achieving the (ε, δ) pair condition

We next show that, for a variety of random sketches of interest, the $(\varepsilon, \frac{\delta}{L})$ pair condition for $(\boldsymbol{Q}, \boldsymbol{h})$ in Theorem 1.4.2 holds for sufficiently large m. We begin with a lemma that gives a sufficient condition for verification of the $(\varepsilon, \frac{\delta}{L})$ pair condition for $(\boldsymbol{Q}, \boldsymbol{h})$, which can be deduced as a special case from [158, Lemma 1]:

Lemma 1.4.10 ((author?) [158]). *Let $\boldsymbol{Q}$ and $\boldsymbol{h}$ be defined as in Theorem 1.4.2. The distribution of $\boldsymbol{S}$ is an $(\varepsilon, \frac{\delta}{L})$ pair for $(\boldsymbol{Q}, \boldsymbol{h})$ if the following two conditions hold simultaneously with probability at least $1 - \delta/L$:*

$$\sigma_{\min}^2(\boldsymbol{SQ}) \geq \frac{\sqrt{2}}{2} \quad \text{and} \quad \|\boldsymbol{Q}^T \boldsymbol{S}^T \boldsymbol{Sh}\|_2^2 \leq \frac{\varepsilon}{2}, \quad (1.52)$$

where $\sigma_{\min}(\cdot)$ denotes the smallest singular value of a matrix.

When the conditions in Lemma 1.4.10 hold, one can directly bound (1.47) using the sub-multiplicativity of operator norms instead of resorting to an (ε, δ) argument as in the proof of Theorem 1.4.2, although the latter is more general. Theorem 1.4.11 presents constructive strategies for generating sketch distributions – based on sub-Gaussian random variables and leverage scores – that achieve appropriate (ε, δ) pair conditions. We recall that a random variable X is called sub-Gaussian if, for some $K > 0$ we have $\varepsilon \exp\left(X^2/K^2\right) \leq 2$ [528, Def. 2.5.6]. The sub-Gaussian norm of X is defined as $\|X\|_{\psi_2} \coloneqq \inf\left\{K > 0 \,:\, \mathbb{E}\exp\left(X^2/K^2\right) \leq 2\right\}$ [528]. A proof of Theorem 1.4.11 is give in Appendix A.3. Variants of these results have appeared previously in the literature [157, 156, 158, 297].

Theorem 1.4.11. *Let $\boldsymbol{Q}$ and $\boldsymbol{h}$ be defined as in Theorem 1.4.2. Write $\boldsymbol{Q}$ and $\boldsymbol{S}^T$ as column vectors:*

$$\boldsymbol{Q} = [\boldsymbol{q}_1, \cdots, \boldsymbol{q}_d], \quad \boldsymbol{S}^T = [\boldsymbol{s}_1, \cdots, \boldsymbol{s}_m], \tag{1.53}$$

and denote by $q_{ij} \coloneqq \boldsymbol{q}_i(j)$ and $h_j \coloneqq \boldsymbol{h}(j)$ the j-th component of $\boldsymbol{q}_i$ and $\boldsymbol{h}$, respectively.

(1) Suppose $\boldsymbol{S} \in \mathbb{R}^{m \times N}$ is a dense sketch whose entries are i.i.d. sub-Gaussian random variables with mean 0 and variance $1/m$. Assume the sub-Gaussian norm of each entry of $\sqrt{m}\boldsymbol{S}$ is bounded by $K \geq 1$. Then the distribution of $\boldsymbol{S}$ is an $(\varepsilon, \frac{\delta}{L})$ pair for $(\boldsymbol{Q}, \boldsymbol{h})$ if

$$m \geq \frac{CK^4}{\varepsilon} d \log\left(\frac{4dL}{\delta}\right), \tag{1.54}$$

where C is an absolute constant.

(2) Suppose $\boldsymbol{S} \in \mathbb{R}^{m \times N}$ is a row sketch based on the leverage scores of $\boldsymbol{A}$, and $0 < \varepsilon, \delta < 1/2$; see Equation (1.14). Then the distribution of $\boldsymbol{S}$ is an $(\varepsilon, \frac{\delta}{L})$ pair for $(\boldsymbol{Q}, \boldsymbol{h})$ if

$$m \geq \max\left\{35d \log\left(\frac{4dL}{\delta}\right), \frac{2dL}{\varepsilon\delta}\right\}. \tag{1.55}$$

Moreover, if

$$\max_{i \in [d]} \max_{j \in [N] : \ell_j > 0} \frac{d|q_{ij}h_j|}{\ell_j} \leq C, \quad \ell_j = \sum_{k \in [d]} q_{kj}^2 \tag{1.56}$$

for some constant $C > 0$, then the distribution of $\boldsymbol{S}$ is an $(\varepsilon, \frac{\delta}{L})$ pair for $(\boldsymbol{Q}, \boldsymbol{h})$ if

$$m \geq \max\left\{35, \frac{4C^2}{\varepsilon}\right\} d \log\left(\frac{4dL}{\delta}\right). \tag{1.57}$$

The scalar ℓ_j in (1.56) is the leverage score associated to row j of $\boldsymbol{A}$, and $(\ell_j)_{j\in[N]}$ defines a (discrete) probability distribution over the row indices $[N]$ of $\boldsymbol{A}$; see (1.14).

Remark 1.4.12. When $\boldsymbol{Q}$ is incoherent, i.e., when its leverage scores satisfy $\ell_i = \mathcal{O}(d/N)$, the entries q_{ij} satisfy $q_{ij} = \mathcal{O}(1/\sqrt{N})$. For any $\boldsymbol{h}$ such that $\max_{j\in[N]} |h_j| \lesssim \mathcal{O}(1/\sqrt{N})$, the condition in (1.56) is satisfied with $C = \mathcal{O}(1)$:

$$\max_{i\in[d]} \max_{j\in[N]:\ell_j>0} \frac{d|q_{ij}h_j|}{\ell_j} \lesssim \frac{d \cdot \frac{1}{\sqrt{N}} \cdot \frac{1}{\sqrt{N}}}{\frac{d}{N}} = 1.$$

Remark 1.4.13. As noted in Section 1.3.2.3, leveraged volume sampling requires $m \gtrsim d\log(d/\delta) + d/(\varepsilon\delta)$ samples to satisfy the (ε, δ) pair condition. This result appears in Corollary 10 of [138].

1.5 Numerical experiments

In this section we illustrate various aspects of the BFB approach using both manufactured data as well as data obtained from PDE solutions. The codes used to generate the results of this section are available from the GitHub repository `https://github.com/CU-UQ/BF-Boosted-Quadrature-Sampling`.

1.5.1 Verification of theoretical results on synthetic data

We first verify the theoretical results in Theorems 1.4.2 and 1.4.5. We do this by simulating different values for $\boldsymbol{S}$, $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$. We generate a design matrix $\boldsymbol{A} \in \mathbb{R}^{1000\times 50}$ (i.e., $N = 1000$ and $d = 50$) with i.i.d. standard normal entries and fix it in the rest of the simulations. For sketching matrices $\boldsymbol{S}$, we choose the embedding dimension to be $m = 100$ and consider both the Gaussian and leverage score sampling sketches. We generate multiple different versions of the vectors $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$ that correspond to different values of κ and φ. Recall that these parameters control how much of $\boldsymbol{b}$ is in the range of $\boldsymbol{A}$ and the absolute value of the correlation between $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$, respectively.

The vectors are generated via

$$\begin{aligned} \boldsymbol{b} &= \kappa \boldsymbol{Q}\boldsymbol{z}_1 + \sqrt{1-\kappa^2}\boldsymbol{Q}_\perp \boldsymbol{z}_2, \\ \tilde{\boldsymbol{b}} &= \varphi \boldsymbol{b} + \sqrt{1-\varphi^2}\boldsymbol{b}_\perp \boldsymbol{z}_3, \end{aligned} \tag{1.58}$$

where $\boldsymbol{Q} = \text{orth}(\boldsymbol{A})$, and $\boldsymbol{z}_1 \in \mathbb{R}^{d-1}$, $\boldsymbol{z}_2 \in \mathbb{R}^{N-d-1}$ and $\boldsymbol{z}_3 \in \mathbb{R}^{N-2}$ are generated by normalizing random vectors of appropriate length whose entries are i.i.d. standard normal. In the experiment, the vectors $\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3$ are drawn once and then kept fixed for the different choices of κ and φ.

To check the upper bound in Theorem 1.4.2, we generate $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$ using 9 equi-spaced values for φ and κ between 0 and 1, which will provide 81 plots for each sketching strategy. We use a sequence of $L = 10$ independent sketching operators in our BFB approach. After computing values of ν for every case, we evaluate the optimality coefficient difference $\mu(\boldsymbol{b}, \boldsymbol{S}_{\ell^*}) - \mu(\boldsymbol{b}, \boldsymbol{S}_{\ell^{**}})$. Figure 1.1 illustrates the relation between $\mu(\boldsymbol{b}, \boldsymbol{S}_{\ell^*}) - \mu(\boldsymbol{b}, \boldsymbol{S}_{\ell^{**}})$ and the bound $2\sqrt{6(1-\nu)\varepsilon}$. Due to the unknown constants in (1.54) and (1.55), an exact value of ε corresponding to $m = 100$ is unavailable. Instead, we choose ε to be 0.01 heuristically. We chose this particular value of ε since it illustrates how the green curve's shape, which is independent with the scalar ε, separates most of the scatter plots from the rest of the area. The result shows our purposed BFB bound in Theorem 1.4.2 is effective and non-vacuous for both Gaussian and leverage score sketchings. It is noticeable that all the dots out of our proposed bound (green) are leverage score sketch spots (blue). The reason is because we set $m = 100$ for both sketch strategies, while leverage score sketch requires a higher m to satisfy the (ε, δ) pair condition, which leads to a higher deviation in μ with fixed m; see details in Theorem 1.4.11.

To further validate our theoretical results in Theorem 1.4.5, we consider four combinations of κ and φ as listed in Table 1.1. For both the Gaussian and leverage score sketches we draw 100 sketches randomly. The same set of sketches are used for each pair of the vectors $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$. Figure 1.2 shows scatter plots of the squared optimality coefficients for the four different pairs of $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$ and two different sketch types.

Table 1.1 provides the estimated correlations between $\mu^2(\boldsymbol{b}, \boldsymbol{S})$ and $\mu^2(\tilde{\boldsymbol{b}}, \boldsymbol{S})$ for each of the eight setups based on the data points in Figure 1.2. For both sketches, a small value of κ and a

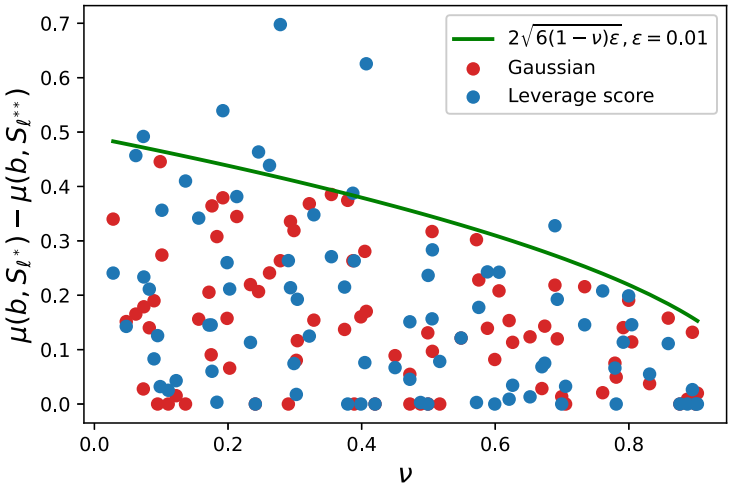


Figure 1.1: Scatter plots of $\mu(\boldsymbol{b}, \boldsymbol{S}_{\ell^*}) - \mu(\boldsymbol{b}, \boldsymbol{S}_{\ell^{**}})$ based on given values of ν for Gaussian sketch (red) and leverage score sketch (blue). The green curve is the bound we provide in Theorem 1.4.2 with $\varepsilon = 0.01$.

large value of φ together yield the highest positive correlation between $\mu^2(\boldsymbol{b}, \boldsymbol{S})$ and $\mu^2(\tilde{\boldsymbol{b}}, \boldsymbol{S})$. In this case, the sketch that attains the smallest residual on the low-fidelity data also attains a near-minimal residual on the high-fidelity data. This is indicative of the desired sketch transferability between the low- and high-fidelity regression problems. These observations are consistent with the upper bound in (1.23) and the lower bound in (1.35), supporting the idea of BFB.

In this section we verify the accuracy of Algorithm 2 on two PDE problems: Thermally-driven cavity fluid flow (Section 1.5.1.1) and simulation of a composite beam (Section 1.5.1.2). In doing so, we consider three random sketching strategies based on uniform, leverage score (Section 1.3.2.2), and leveraged volume (Section 1.3.2.3) sampling. As a baseline, we also present results based on deterministic sketching via column-pivoted QR decomposition (Section 1.3.2.1).

In both experiments, the high-fidelity solution operator takes uniformly distributed inputs $\boldsymbol{p} \in [-1, 1]^q$. We therefore consider approximations of the form in (1.1) with $\psi_j : [-1, 1]^q \mapsto \mathbb{R}$ chosen to be products of q univariate (normalized) Legendre polynomials. Specifically, let $\boldsymbol{j} = (j_1, \ldots, j_q)$, $j_k \in \mathbb{N} \cup \{0\}$, be a vector of non-negative indices and $\psi_{j_k}(p_k)$ denote the Legendre polynomial of degree j_k in p_k such that $\mathbb{E}[\psi^2_{j_k}(p_k)] = 1$. The multivariate Legendre polynomials are

Table 1.1: Empirical correlation between $\mu^2(\boldsymbol{A}, \boldsymbol{b})$ and $\mu^2(\boldsymbol{A}, \tilde{\boldsymbol{b}})$ for four different parameters setups and two different sketch types.

κ	φ	Sketch type	Correlation
0.2	0.3	Gaussian	0.21
0.2	0.95	Gaussian	0.88
0.95	0.3	Gaussian	0.17
0.95	0.95	Gaussian	0.48
0.2	0.3	Leverage score	0.19
0.2	0.95	Leverage score	0.91
0.95	0.3	Leverage score	0.08
0.95	0.95	Leverage score	0.56

given by

$$\psi_{\boldsymbol{j}}(\boldsymbol{p}) = \prod_{k=1}^{q} \psi_{j_k}(p_k). \tag{1.59}$$

The set of polynomials $\{\psi_{\boldsymbol{j}}\}$ is chosen so that it spans either a total degree or hyperbolic cross space. In the former case this means all polynomials satisfying $\sum_{k=1}^{q} j_k \leq \zeta$, while in the latter case $\boldsymbol{j}$ is limited to multi-indices with $\prod_{k=1}^{q}(j_k + 1) \leq \zeta + 1$, for some predefined $\zeta \in \mathbb{N} \cup \{0\}$.

In order to construct a design matrix $\boldsymbol{A}$ as in (1.2), and data vectors $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$ in (1.2) and (1.16), respectively, we also need to choose pairs of quadrature points and weights $(\boldsymbol{p}_n, w_n)_{n\in[N]}$. While both deterministic and random rules are possible, we here choose these quantities to be deterministic and of the form

$$\begin{aligned} \boldsymbol{p}_n &= (p_{1,n_1}, p_{2,n_2}, \ldots, p_{q,n_q}), \\ w_n &= \prod_{k=1}^{q} w_{k,n_k}, \end{aligned} \tag{1.60}$$

where each sequence $(p_{k,n_k}, w_{k,n_k})_{n_k\in[N_k]}$ consists of node-weight pairs in the N_k-point Gauss–Legendre quadrature on $[-1, 1]$. The resulting sequence $(\boldsymbol{p}_n, w_n)_{n\in[N]}$ contains $N = \prod_{k=1}^{q} N_k$ pairs. When $\boldsymbol{A}$ is constructed in this fashion, it is possible to sample rows of that matrix according to the exact leverage score using the efficient method by [349]. Please see Appendix A.1 for details on how this is done.

To measure the final performance, we use the relative error defined as

$$E := \frac{\|\boldsymbol{A}\hat{\boldsymbol{x}}_{\mathsf{BFB}} - \boldsymbol{b}\|_2}{\|\boldsymbol{b}\|_2}, \tag{1.61}$$

where $\hat{\boldsymbol{x}}_{\mathsf{BFB}}$ is the output from Algorithm 2.

1.5.1.1 Cavity fluid flow

Here we consider the case of temperature-driven fluid flow in a 2D cavity [29, 429, 212, 210, 170, 214], with the quantity of interest being the heat flux averaged along the hot wall as Figure 1.3 shows. The wall on the left hand side is the hot wall with random temperature T_h, and the cold wall at the right hand side has temperature $T_c < T_h$. $\bar{T}_c$ is the constant mean of T_c. The horizontal walls are adiabatic. The reference temperature and the temperature difference are given by $T_{\text{ref}} = (T_h + \bar{T}_c)/2$ and $\Delta T_{\text{ref}} = T_h - \bar{T}_c$, respectively. The normalized governing equations are given by

$$\begin{aligned} &\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u} = -\nabla p + \frac{\text{Pr}}{\sqrt{\text{Ra}}}\nabla^2\boldsymbol{u} + \text{Pr}\Theta\mathbf{e}_y, \\ &\nabla\cdot\boldsymbol{u} = 0, \\ &\frac{\partial \Theta}{\partial t} + \nabla\cdot(\boldsymbol{u}\Theta) = \frac{1}{\sqrt{\text{Ra}}}\nabla^2\Theta, \end{aligned} \tag{1.62}$$

where $\mathbf{e}_y$ is the unit vector $(0,1)$, $\boldsymbol{u} = (u,v)$ is the velocity vector field, $\Theta = (T - T_{\text{ref}})/\Delta T_{\text{ref}}$ is normalized temperature, p is pressure, and t is time. We assume no-slip boundary conditions on the walls. The dimensionless Prandtl and Rayleigh numbers are defined as $\text{Pr} = \nu_{\text{visc}}/\alpha$ and $\text{Ra} = g\tau\Delta T_{\text{ref}}W^3/(\nu_{\text{visc}}\alpha)$, respectively, where W is the width of the cavity, g is gravitational acceleration, ν_{visc} is kinematic viscosity, α is thermal diffusivity, and τ is the coefficient of thermal expansion. We set $g = 10$, $W = 1$, $\tau = 0.5$, $\Delta T_{\text{ref}} = 100$, $\text{Ra} = 10^6$, and $\text{Pr} = 0.71$. On the cold wall, we apply a temperature distribution with stochastic fluctuations as

$$T(x = 1, y) = \bar{T}_c + \sigma_T \sum_{i=1}^{q} \sqrt{\lambda_i}\phi_i(y)\mu_i, \tag{1.63}$$

where $\bar{T}_c = 100$ is a constant, $\{\lambda_i\}_{i\in[q]}$ and $\{\phi_i(y)\}_{i\in[q]}$ are the q largest eigenvalues and corresponding eigenfunctions of the kernel $k(y_1, y_2) = \exp(-|y_1 - y_2|/0.15)$, and each $\mu_i \overset{\text{i.i.d.}}{\sim} U[-1,1]$.

We let $q = 2$ (though in general, this does not need to match the physical dimension) and $\sigma_T = 2$. The vector $\boldsymbol{p} = (\mu_1, \mu_2)$ is the uncertain input of the model.

In order to solve (1.62) we use the finite volume method with two different grid resolutions: a finer grid of size 128×128 to produce the high-fidelity solution and a coarser grid of size 16×16 to produce the low-fidelity solution. For our surrogate model, we choose the basis set $\{\psi_j\}_{j\in[d]}$ based on the total degree and hyperbolic cross spaces of maximum order $\zeta = 4$. The corresponding spaces have $d = 15$ and $d = 10$ basis functions, respectively. The quadrature pairs $(\boldsymbol{p}_n, w_n)$ used to construct $\boldsymbol{A}$, $\boldsymbol{b}$, and $\tilde{\boldsymbol{b}}$ are defined as in (1.60) and are based on the nodes and weights from a 10-point Gauss–Legendre rule, i.e., $N_1 = N_2 = 10$.

We first repeat the test we ran on synthetic data in Section 1.5.1. Figure 1.4 shows the scatter plots of $(\mu^2(\tilde{\boldsymbol{b}}, \boldsymbol{S}), \mu^2(\boldsymbol{b}, \boldsymbol{S}))$ for the two different polynomial spaces and three different random sampling approaches. Each plot is based on 100 sketches with $m = 30$ and $m = 20$ samples used for the total degree and hyperbolic cross spaces, respectively. Table 1.2 presents the correlation coefficients between $\mu^2(\boldsymbol{b}, \boldsymbol{S})$ and $\mu^2(\tilde{\boldsymbol{b}}, \boldsymbol{S})$ based on the points in Figure 1.4. There is a discrepancy between the correlation observed for the total degree and hyperbolic cross spaces. One possible explanation for this is that a greater portion of $\boldsymbol{b}$ is in the range of $\boldsymbol{A}$ for the total degree space than for the hyperbolic cross space, i.e., κ (see (1.34)) is larger for the former space. Theorem 1.4.5 indicates that a larger κ should be associated with lower correlation.

Next, we run Algorithm 2 with $L = 10$ sketches and the number of samples $m = 1.2d$ and $m = 2d$. Figure 1.5 shows the relative error E in (1.61) from running the algorithm 1000 times for each of the different choices of polynomial space, sketch size m, and random sampling approach. We observe that in all cases the BFB approach improves the error as compared to the non-boosted case. In particular, the improvement is more considerable in the case of the hyperbolic cross basis, which is explained by the higher correlation between $\mu^2(\boldsymbol{A}, \boldsymbol{b})$ and $\mu^2(\boldsymbol{A}, \tilde{\boldsymbol{b}})$, as reported in Table 1.2. Additionally, for the case of hyperbolic space, the BFB results is comparable or better performance as compared to the column-pivoted QR decomposition (blue line in Figure 1.5). Note that the computational cost of column-pivoted QR is higher than the BFB as it requires the QR

decomposition of the entire matrix $\boldsymbol{A}$.

Table 1.2: Correlation coefficients between $\mu^2(\boldsymbol{A}, \boldsymbol{b})$ and $\mu^2(\boldsymbol{A}, \tilde{\boldsymbol{b}})$ for different sampling methods under total degree or hyperbolic cross space. The correlation is computed based on the points shown in Figure 1.4.

Polynomial Space	Uniform Sampling	Leverage Score Sampling	Leveraged Volume Sampling
Total Degree	0.66	0.57	0.18
Hyperbolic Cross	0.99	0.98	0.98

1.5.1.2 Composite beam

Following [219, 123, 124], we consider a plane-stress, cantilever beam with composite cross section and hollow web as shown in Figure 1.6. The quantity of interest in this case is the maximum displacement of the top cord. The uncertain parameters of the model are E_1, E_2, E_3, f, where E_1, E_2 and E_3 are the Young's moduli of the three components of the cross section and f is the intensity of the applied distributed force on the beam; see Figure 1.6. These are assumed to be statistically independent and uniformly distributed. The dimension of the input parameter is therefore $q = 4$. Table 1.3 shows the range of the input parameters as well as the other deterministic parameters.

Table 1.3: The values of the parameters in the composite cantilever beam model. The center of the holes are at $x = \{5, 15, 25, 35, 45\}$. The parameters f, E_1, E_2 and E_3 are drawn independently and uniformly at random from the specified intervals.

H	h_1	h_2	h_3	w	r	f	E_1	E_2	E_3
50	0.1	0.1	5	1	1.5	$[9, 11]$	$[0.9e6, 1.1e6]$	$[0.9e6, 1.1e6]$	$[0.9e4, 1.1e4]$

For the cavity fluid flow problem in Section 1.5.1.1, we created high- and low-fidelity solutions by changing the resolution of the grid used in the numerical solver. For the present problem, we instead use two different models. The high-fidelity model is based on a finite element discretization of the beam using a triangle mesh, as Figure 1.7 shows. The low-fidelity model is derived from Euler–Bernoulli beam theory in which the vertical cross sections are assumed to remain planes throughout the deformation. The low-fidelity model ignores the shear deformation of the web and

does not take the circular holes into account. Considering the Euler-Bernoulli theorem, the vertical displacement u is

$$EI\frac{d^4u(x)}{dx^4} = -f, \tag{1.64}$$

where E and I are, respectively, the Young's modulus and the moment of inertia of an equivalent cross section consisting of a single material. We let $E = E_3$, and the width of the top and bottom sections are $w_1 = (E_1/E_3)w$ and $w_2 = (E_2/E_3)w$, while all other dimensions are the same, as Figure 1.6 shows. The solution of (1.64) is

$$u(x) = -\frac{qH^4}{24EI}\left(\left(\frac{x}{H}\right)^4 - 4\left(\frac{x}{H}\right)^3 + 6\left(\frac{x}{H}\right)^2\right). \tag{1.65}$$

The surrogate model is based on multivariate Legendre polynomials of maximum degree $\zeta = 2$ with total degree and hyperbolic cross truncation. The corresponding spaces have $d = 15$ and $d = 9$ basis functions, respectively. As in the case of the cavity flow problem, the quadrature pairs $(\boldsymbol{p}_n, w_n)$ used to construct $\boldsymbol{A}$, $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$ are based on the nodes and weights from 10-point Gauss–Legendre rule appropriately mapped into the ranges given in Table 1.3.

Figure 1.8 shows the scatter plots of $(\mu^2(\tilde{\boldsymbol{b}}, \boldsymbol{S}), \mu^2(\boldsymbol{b}, \boldsymbol{S}))$ when repeating the experiment in Section 1.5.1 for the two different polynomial spaces and three different random sampling approaches. Each plot is based on 100 sketches with $m = 2d$, i.e., $m = 30$ and $m = 18$ samples used for the total degree and hyperbolic cross spaces, respectively. Table 1.4 reports the correlation coefficient between $\mu^2(\boldsymbol{b}, \boldsymbol{S})$ and $\mu^2(\tilde{\boldsymbol{b}}, \boldsymbol{S})$, indicating an overall high correlation in all cases.

Table 1.4: Correlation coefficient between $\mu^2(\boldsymbol{A}, \boldsymbol{b})$ and $\mu^2(\boldsymbol{A}, \tilde{\boldsymbol{b}})$ for different sampling methods under total degree or hyperbolic cross space. The correlation is computed based on the points shown in Figure 1.8.

Polynomial Space	Uniform Sampling	Leverage Score Sampling	Leveraged Volume Sampling
Total Degree	0.77	0.69	0.84
Hyperbolic Cross	0.72	0.73	0.82

Next, we run Algorithm 2 with $L = 10$ sketches and m chosen to be $m = 1.2d$ and $m = 2d$. Figure 1.9 shows the results from running the algorithm 1000 times for each of the different choices

of polynomial space, number of samples m, and random sampling approach. We observe that the BFB performance is superior to that of the non-boosted implementation as it leads to smaller variance of the error and fewer outliers with smaller deviation from the mean performance. In this example, the BFB leads to comparable accuracy as the column-pivoted QR sketch, but with smaller sketching cost. As in the case of the cavity flow, the results corroborate the discussion below Theorem 1.4.5, in that the BFB improves the regression accuracy when $\mathrm{corr}(\mu^2(\boldsymbol{b}, \boldsymbol{S}), \mu(\tilde{\boldsymbol{b}}, \boldsymbol{S}))$ is large.

1.6 Conclusion

This work was concerned with the construction of (polynomial) emulators of parameter-to-solution maps of PDE problems via sketched least-squares regression. Sketching is a design of experiments approach that aims to improve the cost of building a least squares solution in terms of reducing the number of samples needed — when the cost of generating data is high — or the cost of generating a least squares solution —- when data size is substantial. Focusing on the former case, we have proposed a new boosting algorithm to compute a sketched least squares solution.

The procedure consisted in identifying the best sketch from a set of candidates used to construct least squares regression of the low-fidelity data and applying this *optimal* sketch to the regression of high-fidelity data. The bi-fidelity boosting (BFB) approach limits the required sample complexity to $\sim d \log d$ high-fidelity data, where d is the size of the (polynomial) basis. We have provided theoretical analysis of the BFB approach identifying assumptions on the low- and high-fidelity data under which the BFB leads to improvement of the solution relative to non-boosted regression of the high-fidelity data. We have also provided quantitative bounds on the residual of the BFB solution relative to the full, computationally expensive solution. We have investigated the performance of BFB on manufactured and PDE data from fluid and solid mechanics. These cover sketching strategies based on leverage score and leveraged volume sampling, for truncated Legendre polynomials of both total degree and hyperbolic cross type. All tests illustrated the efficacy of BFB in reducing the residual — as compared to the non-boosted implementation —- and validate the

theoretical results.

The present study was focused on the case of (weighted) least squares polynomial regression. When the regression coefficients are sparse, methods based on compressive sampling have proven efficient in reducing the sample complexity below the size of the polynomial basis; see, e.g., [150, 6]. As interesting future research direction is to extend the BFB strategy to such under-determined cases, for instance, using the approach of [146].

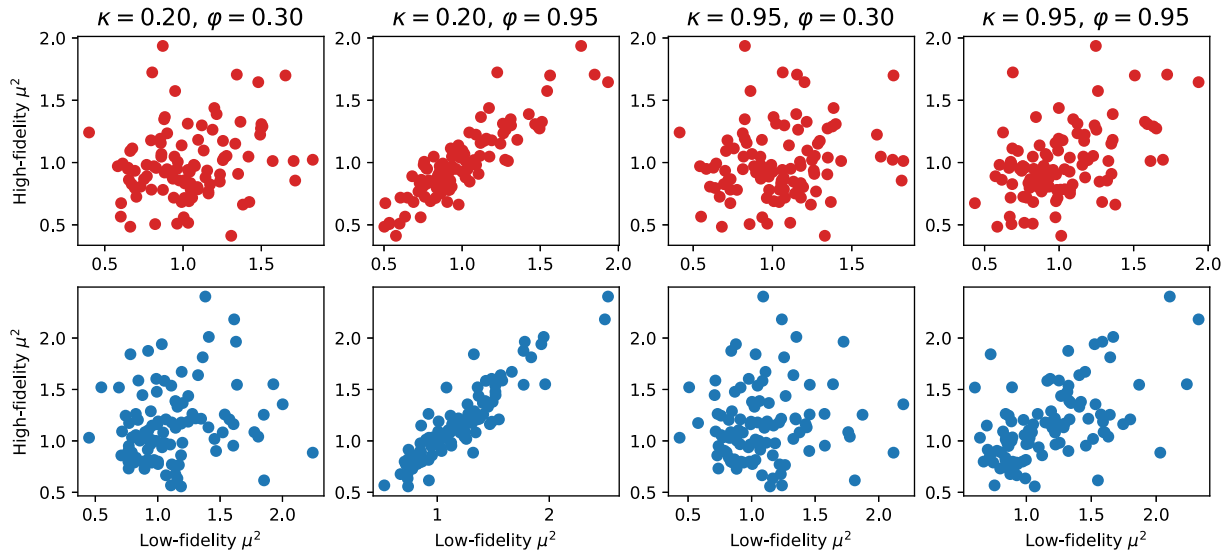


Figure 1.2: Scatter plots of the square of the optimality coefficient for high- and low-fidelity data for each of 100 different sketches. Each point is equal to $(\mu^2(\tilde{\boldsymbol{b}}, \boldsymbol{S}), \mu^2(\boldsymbol{b}, \boldsymbol{S}))$ for one realization of the sketch $\boldsymbol{S}$. The top and bottom panels correspond to the sketches constructed using Gaussian and leverage score sampling sketches, respectively.

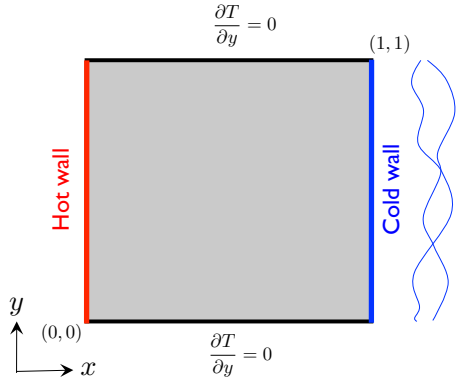


Figure 1.3: A figure of the temperature driven cavity flow problem, reproduced from Figure 5 of [170].

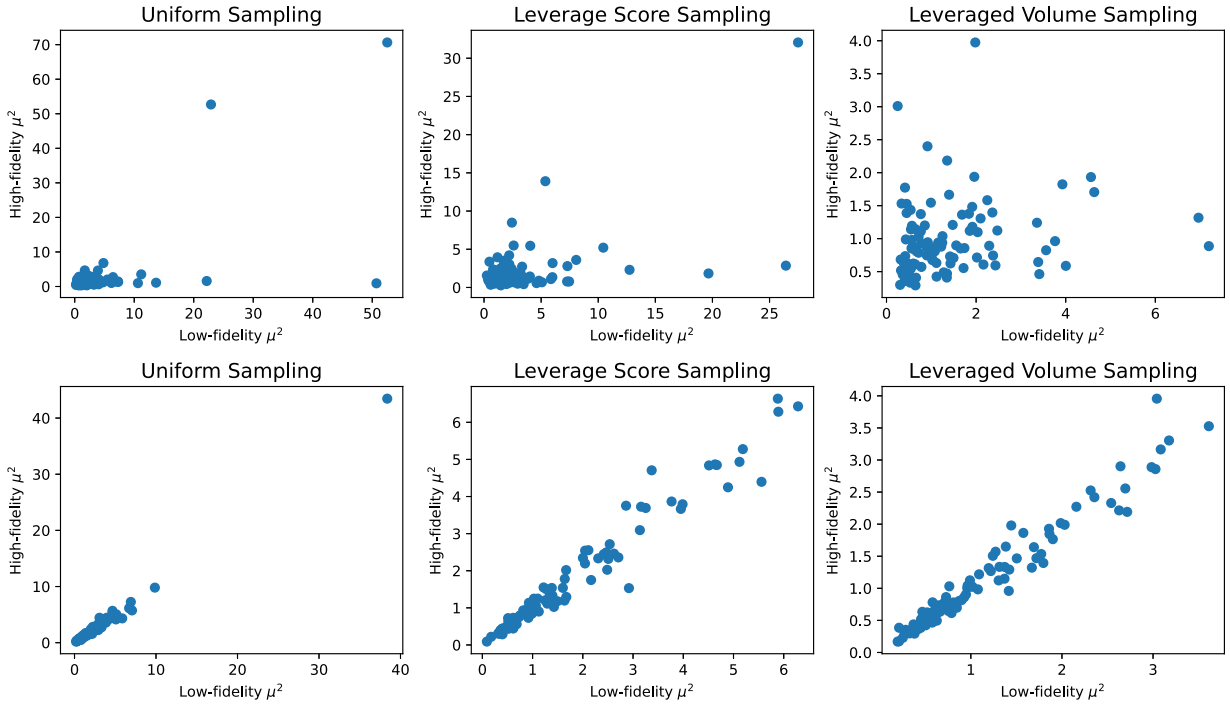


Figure 1.4: Scatter plots of the square of the optimality coefficient for high- and low-fidelity data from the cavity fluid flow problem for different polynomial spaces (top: total degree; bottom: hyperbolic cross) and types of sampling. Each point is equal to $(\mu^2(\tilde{\boldsymbol{b}}, \boldsymbol{S}), \mu^2(\boldsymbol{b}, \boldsymbol{S}))$ for one realization of the sketch $\boldsymbol{S}$, and each subplot contains 100 points (i.e., is based on 100 sketch realizations). For the total degree space $m = 30$ samples are used and for the hyperbolic cross space $m = 20$ samples are used. The corresponding correlation coefficients are presented in Table 1.2.

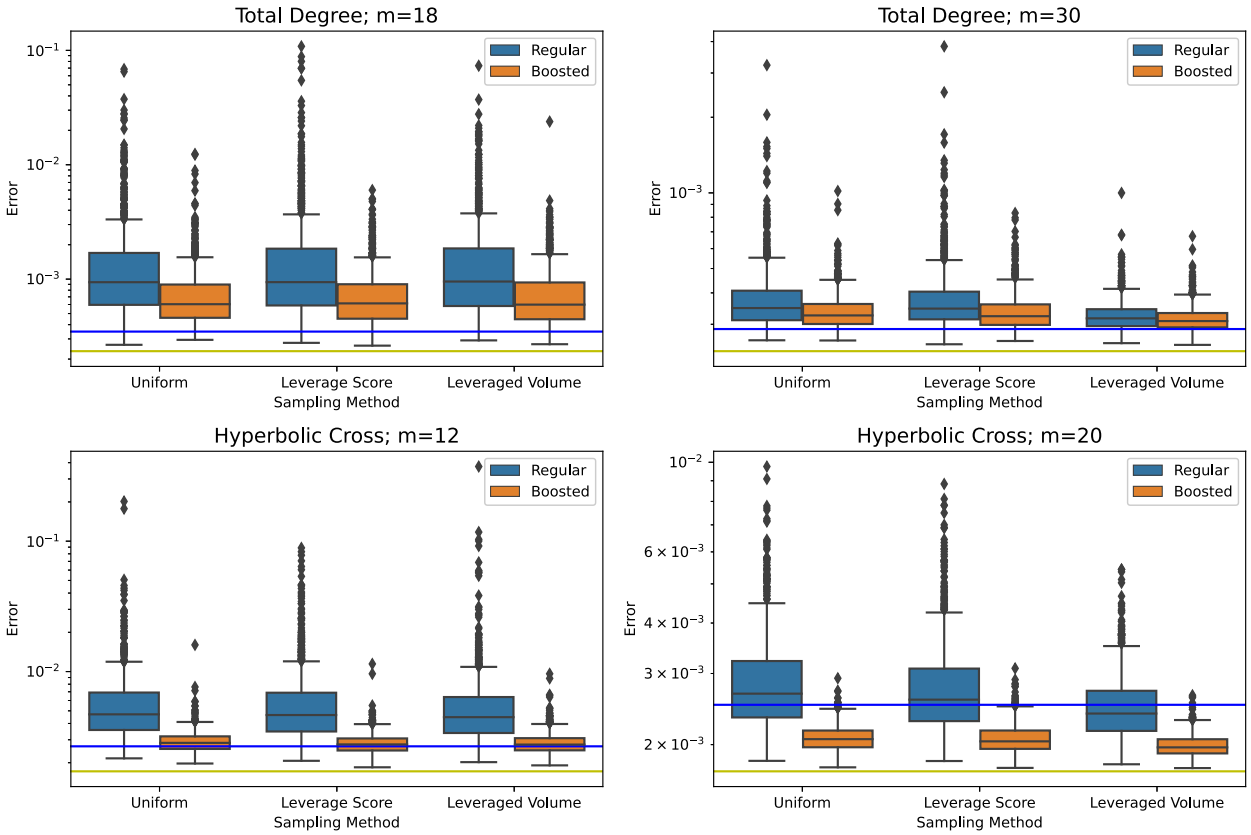


Figure 1.5: Relative error for different sampling methods and polynomial spaces when fitting the surrogate model to the cavity fluid flow data. Yellow lines show the relative error E in (1.61) for the unsketched solution in (1.2). Blue lines show E when the coefficients $\boldsymbol{x}$ are computed via the QR decomposition-based method in Section 1.3.2.1. The blue box plots shows the distribution of E based on 1000 trials when $\boldsymbol{x}$ is computed as in (1.7). The orange box plots shows the same things, but for the solution $\hat{\boldsymbol{x}}_{\mathsf{BFB}}$ computed via Algorithm 2.

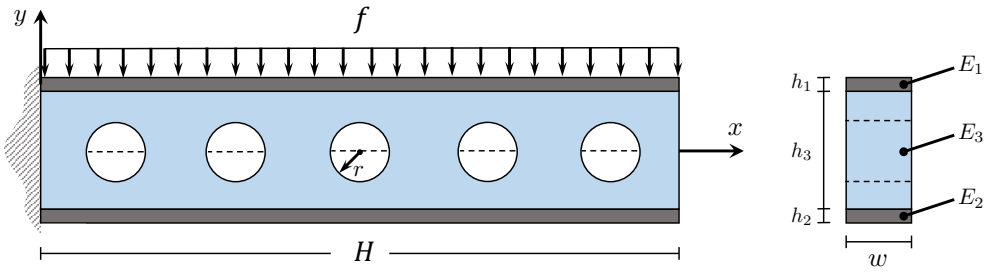


Figure 1.6: Cantilever beam (left) and the composite cross section (right) adapted from [217].

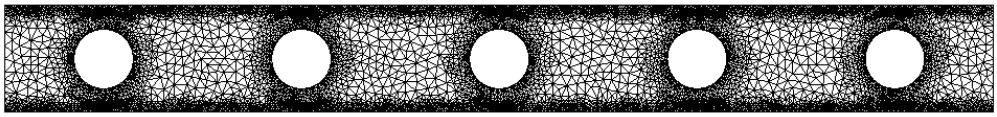


Figure 1.7: Finite element mesh used to generate high-fidelity solutions.

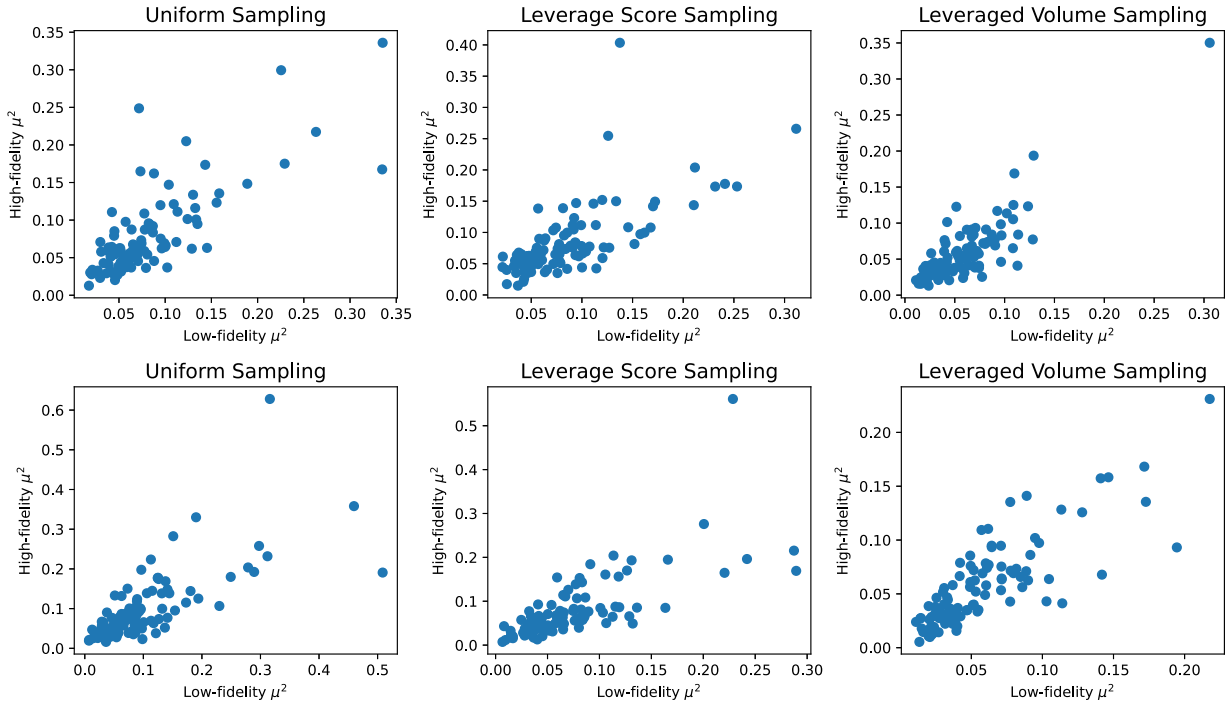


Figure 1.8: Scatter plots of the square of the optimality coefficient for high- and low-fidelity data from the composite beam problem for different polynomial spaces (top: total degree; bottom: hyperbolic cross) and types of sampling. Each point is equal to $(\mu^2(\tilde{\boldsymbol{b}}, \boldsymbol{S}), \mu^2(\boldsymbol{b}, \boldsymbol{S}))$ for one realization of the sketch $\boldsymbol{S}$, and each subplot contains 100 points (i.e., is based on 100 sketch realizations). For the total degree space $m = 30$ samples are used and for the hyperbolic cross space $m = 18$ samples are used. The corresponding correlation coefficients are presented in Table 1.4.

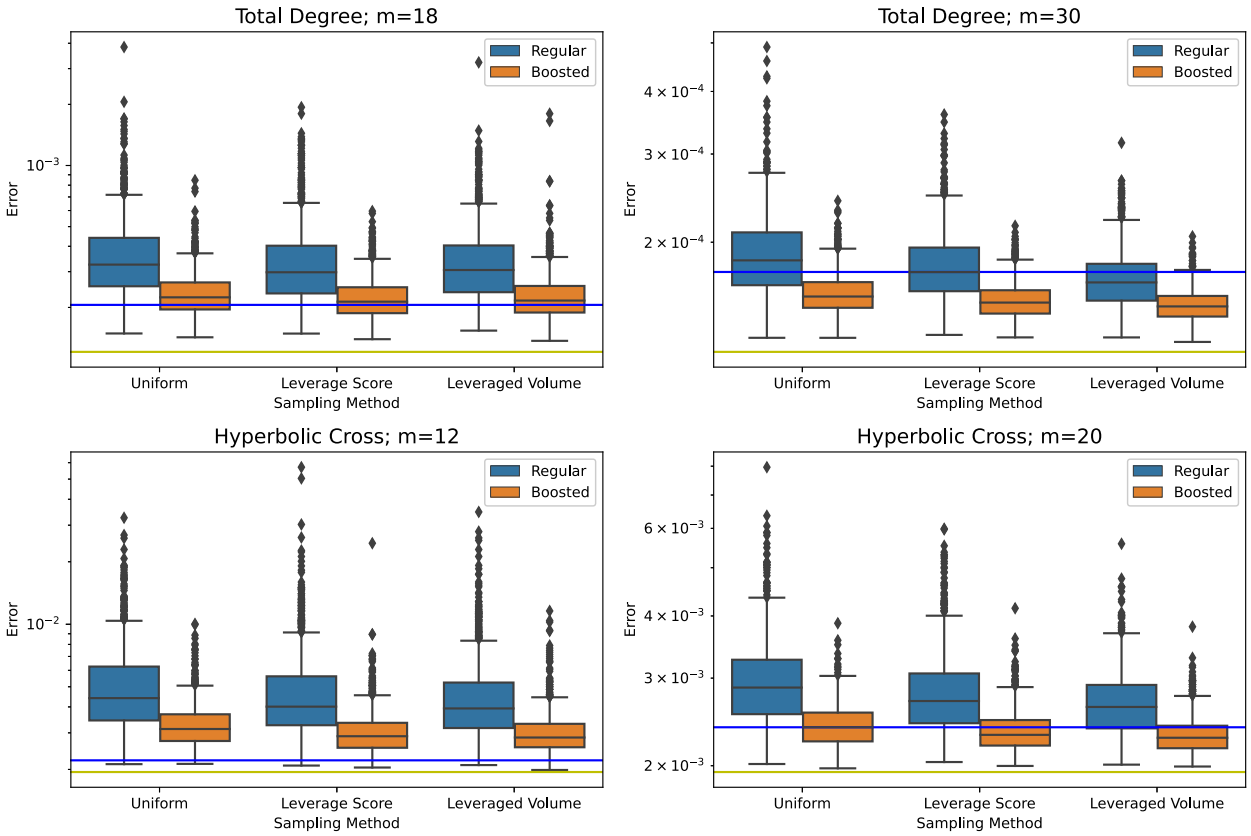


Figure 1.9: Relative error for different sampling methods and polynomial spaces when fitting the surrogate model to the beam problem data. Yellow lines show the relative error E in (1.61) for the unsketched solution in (1.2). Blue lines show E when the coefficients $\boldsymbol{x}$ are computed via the QR decomposition-based method in Section 1.3.2.1. The blue box plots shows the distribution of E based on 1000 trials when $\boldsymbol{x}$ is computed as in (1.7). The orange box plots shows the same things, but for the solution $\hat{\boldsymbol{x}}_{\mathsf{BFB}}$ computed via Algorithm 2.

Chapter 2

Bi-fidelity Variational Auto-encoder for Uncertainty Quantificaiton

2.1 Abstraction

Quantifying the uncertainty of quantities of interest (QoIs) from physical systems is a primary objective in model validation. However, achieving this goal entails balancing the need for computational efficiency with the requirement for numerical accuracy. To address this trade-off, we propose a novel bi-fidelity formulation of variational auto-encoders (BF-VAE) designed to estimate the uncertainty associated with a QoI from low-fidelity (LF) and high-fidelity (HF) samples of the QoI. This model allows for the approximation of the statistics of the HF QoI by leveraging information derived from its LF counterpart. Specifically, we design a bi-fidelity auto-regressive model in the latent space that is integrated within the VAE's probabilistic encoder-decoder structure. An effective algorithm is proposed to maximize the variational lower bound of the HF log-likelihood in the presence of limited HF data, resulting in the synthesis of HF realizations with a reduced computational cost. Additionally, we introduce the concept of the bi-fidelity information bottleneck (BF-IB) to provide an information-theoretic interpretation of the proposed BF-VAE model. Our numerical results demonstrate that the BF-VAE leads to considerably improved accuracy, as compared to a VAE trained using only HF data, when limited HF data is available. [1]

[1] The original version of this work is presented in [94], co-authored with O. Malik, S. De, S. Becker, and A. Doostan.

2.2 Introduction

Uncertainty pervades numerous engineering applications due to various factors, such as material properties, operating environments, and boundary conditions, which impact the prediction of a performance metric or quantity of interest (QoI), denoted as $\boldsymbol{x} \in \mathbb{R}^D$, following an unknown probability density function (pdf) $p(\boldsymbol{x})$. The quantification of uncertainty in $\boldsymbol{x}$ through the estimation of its moments or distribution has been an active area of research within the field of uncertainty quantification (UQ). One approach to accomplish this involves collecting independent and identically distributed (i.i.d.) realizations of $\boldsymbol{x}$ to estimate its empirical properties. However, when the realizations of $\boldsymbol{x}$ are obtained through the solution of computationally intensive models, generating a large enough set of realizations to ensure statistical convergence becomes infeasible. To address this challenge, a surrogate model of the forward map between uncertain inputs $\boldsymbol{\xi} \in \mathbb{R}^M$ and $\boldsymbol{x}$ can be constructed. This approach has been demonstrated through a range of techniques, including polynomial chaos expansion [185, 430, 213, 477], Gaussian process regression [556, 49], and deep neural networks [516, 601, 410, 477]. Once established, the surrogate model can be employed, often at a negligible cost, to generate realizations of the QoI and estimate its statistics. However, it should be noted that the complexity of constructing these surrogate models often increases rapidly with the number of uncertain variables, M, a phenomenon referred to as the curse of dimensionality.

To mitigate the problem caused by high-dimensional uncertainty, one remedy is to build a reduced-order model (ROM), [227], where the solution to the governing equations is approximated in a basis of size possibly independent of M. One widely adopted technique for identifying such a reduced basis is proper orthogonal decomposition (POD), often also referred to as principle component analysis (PCA) or Karhunen–Loéve expansion [86]. POD is commonly employed on a collection of forward problem simulations, known as snapshots, to determine the optimal subspace via the solution of a singular value decomposition problem.

The utility of ROMs has been extensively investigated for problems that exhibit a small Kolmogorov n-width [436], e.g., diffusion-dominated flows. However, for advection-dominated problems

where solutions do not align closely with any linear subspace, conventional ROMs may yield inaccurate approximations. This has resulted in the development of non-linear (manifold-based) ROM formulations, including kernel principal component analysis [597, 445], tangent space alignment [587], and auto-encoders (AEs) [305, 359, 392, 271]. Among these manifold-based ROMs, AEs have gained significant attention due to their expressive neural-network-based encoder-decoder structure, enabling them to capture the underlying patterns of the input data by learning a latent representation. The latent variable, denoted by $\boldsymbol{z}$, is of much lower dimension than the input data and is learned through a non-linear encoder function. The decoder function, which typically has a structure mirroring the encoder, takes $\boldsymbol{z}$ and maps it back to the original data space.

While AE-based UQ models [305, 359, 392] have demonstrated success, they are intrinsically deterministic as they do not automatically produce new samples of $p(\boldsymbol{x})$ and, therefore, are not generative. Furthermore, as shown in [493], the lack of regularization in the fully-connected AE architecture can lead to overfitting, hindering the discovery of meaningful latent representations. To address these limitations, several probabilistic frameworks have been proposed to regularize the problem and, more importantly, enable uncertainty estimation. These include Bayesian convolutional AE [600] designed specifically for flow-based problems and the auto-regressive encoder-decoder model for turbulent flows [182].

In this study, we consider the use of variational autoencoders (VAEs) and present a novel training strategy aimed at reducing the training cost in terms of the number of high-fidelity realizations required. VAEs belong to a class of machine learning models that seek to approximate the unknown underlying distribution $p(\boldsymbol{x})$ from which the QoI is derived and generate new realizations from it. Deep generative models, including VAEs [273, 450], generative adversarial networks (GANs) [191], normalizing flows [449], and diffusion models [233, 486], have achieved significant success in various applications in computer vision and natural language processing [551, 433, 441]. VAEs, in particular, offer a well-suited solution for UQ problems owing to their ability to encode a low-dimensional representation of the QoI, regularized via the probabilistic formulation, and generate new realizations of the QoI. Further details on the VAE methodology can be found in

Section 2.3.1.

Despite their benefits, training deep generative models, such as VAEs, typically requires access to a substantial amount of high-fidelity (HF) data, which may be difficult to obtain in large-scale scientific applications. One way to address this issue is to apply the bi-fidelity approach, in which a larger set of cheaper, possibly less accurate, low-fidelity (LF) realizations of the QoI, $\boldsymbol{x}^L \in \mathbb{R}^D$ along with a relatively small set of HF realizations of the QoI, $\boldsymbol{x}^H \in \mathbb{R}^D$, are leveraged to jointly build the model. There have been rich studies of bi-fidelity modeling for UQ, including Monte-Carlo-based [194, 171], graph-based [437], and ROM-based [218, 445]. In this work, we follow the ROM-based approaches and propose a bi-fidelity VAE (BF-VAE) training method by constructing a VAE model that captures the underlying distribution of the HF QoI. These low-dimensional mappings are key in reducing the number of HF realizations required for training. In more detail, we train a VAE, with the same architecture and activation functions as in the intended HF model, but using LF data. Let $\boldsymbol{z}^L$ and $q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)$ denote the latent variable and encoder of this model. The BF-VAE adapts this VAE using HF data in two ways. Firstly, we assume an auto-regressive model with pdf $p_{\boldsymbol{\psi}}(\boldsymbol{z}^H|\boldsymbol{z}^L)$, parameterized by $\boldsymbol{\psi}$, to set the latent variable of the HF model, $\boldsymbol{z}^H$. As depicted in Figure 2.1, such a regression is performed in the d-dimensional latent space, instead of the observation space of dimension $D \gg d$. Secondly, the subset of the decoder parameters $\boldsymbol{\theta}$ corresponding to the last layer of the decoder $p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}^H)$ are updated (with warm start) to adjust the map between the latent and observation space of the HF data. We note that the latter update also involves a relatively small set of parameters.

To summarize, the core contributions of this work are:

- We introduce a novel approach — dubbed the BF-VAE — for training a VAE model, utilizing primarily LF data and a small set of HF data. While trained using both low- and high-fidelity data, the BF-VAE aims at approximating the pdf of the HF QoI $\boldsymbol{x}^H$. This, in turn, enables the generation of approximate samples of $\boldsymbol{x}^H$.

- The BF-VAE model is theoretically motivated as the maximizer of a training objective

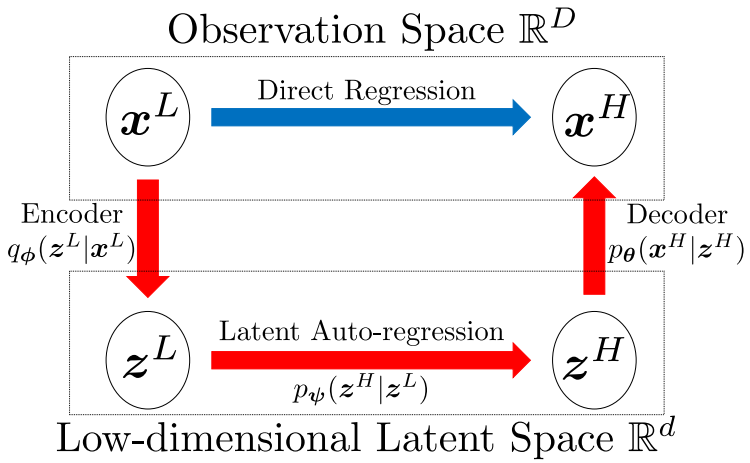


Figure 2.1: Instead of conducting bi-fidelity regression directly in high-dimensional observation space (blue path), we introduce an approach via low-dimensional latent space (red path).

criterion we call BF evidence lower bound (BF-ELBO), an extension of the original ELBO formulation introduced in [273]. We then extend the information bottleneck theory of Tishby et al. [513] to formulate the bi-fidelity information bottleneck (BF-IB) theory and provide an interpretation of BF-ELBO from an information-theoretic perspective.

- We conduct an empirical evaluation of the BF-VAE model through three numerical experiments, comparing its performance with a VAE trained only on HF data. The numerical results indicate that the BF-VAE improves the accuracy of learned HF QoI pdf when the number of HF data is small.

The rest of the paper is structured as follows. In Section 2.3, we provide an overview of the VAE and linear auto-regressive methods for bi-fidelity regression. Section 2.4 elaborates on the proposed BF-VAE model, along with a theoretical interpretation. The implementation details of the BF-VAE model, including prior density selection and hyperparameter tuning, are presented in Section 2.5. Section 2.6 showcases three numerical examples that demonstrate the performance of the BF-VAE. Our conclusions are summarized in Section 2.7. The proof of our main statements and an introduction to an evaluation metric used to assess the quality of data generated from the

VAE models are presented in the appendix.

For consistency with other VAE-related papers, we do not differentiate random vectors and their realizations in this work. Additionally, we simplify density functions by omitting their subscripts. For example, we use $p(\boldsymbol{x}|\boldsymbol{y})$ instead of $p_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}|\boldsymbol{y})$. As such, the densities $p(\boldsymbol{x}^H)$ and $p(\boldsymbol{x}^L)$ may not be the same.

2.3 Motivation and Background

Forward UQ is concerned with quantifying the uncertainty of QoIs from a physical system due to intrinsic variations or limited knowledge about model inputs (or structure). Within a BF setting, an HF model generates $\boldsymbol{x}^H \in \mathbb{R}^D$ with pdf $p(\boldsymbol{x}^H)$ and an LF model generates an approximation to $\boldsymbol{x}^H$ denoted by $\boldsymbol{x}^L \in \mathbb{R}^D$ and following pdf $p(\boldsymbol{x}^L)$. One goal of forward UQ is to estimate $p(\boldsymbol{x}^H)$. With a random input vector $\boldsymbol{\xi} \in \mathbb{R}^M$ and its pdf $p(\boldsymbol{\xi})$, the widely-adopted surrogate modeling approaches seek to approximate $p(\boldsymbol{x}^H|\boldsymbol{\xi})$ and estimate $p(\boldsymbol{x}^H)$ as

$$p(\boldsymbol{x}^H) = \int_{\mathbb{R}^M} p(\boldsymbol{x}^H|\boldsymbol{\xi})p(\boldsymbol{\xi})d\boldsymbol{\xi}. \tag{2.1}$$

This formulation, however, suffers from two major issues: the complexity of building $p(\boldsymbol{x}^H|\boldsymbol{\xi})$ when the dimension of $\boldsymbol{\xi}$ is high and the expensive cost of collecting HF QoI realizations $\boldsymbol{x}^H$ for estimating $p(\boldsymbol{x}^H|\boldsymbol{\xi})$. The Bayesian multi-fidelity approach of [281, 393], summarized next, provides a solution to tackle these aforementioned issues. To alleviate the first issue, bi-fidelity approaches usually introduce an LF pdf $p(\boldsymbol{x}^L)$ with cheaper sampling cost, approximate $p(\boldsymbol{x}^H|\boldsymbol{x}^L)$ instead of $p(\boldsymbol{x}^H|\boldsymbol{\xi})$ due to a closer relation between $\boldsymbol{x}^H$ and $\boldsymbol{x}^L$, and marginalize the random input $\boldsymbol{\xi}$ to mitigate the effect of its high-dimensionality,

$$\begin{aligned} p(\boldsymbol{x}^H) &= \int_{\mathbb{R}^D}\int_{\mathbb{R}^M} p(\boldsymbol{x}^H, \boldsymbol{x}^L, \boldsymbol{\xi})d\boldsymbol{\xi}d\boldsymbol{x}^L && \text{introduce LF model} \\ &= \int_{\mathbb{R}^D}\int_{\mathbb{R}^M} p(\boldsymbol{x}^H, \boldsymbol{\xi}\,|\boldsymbol{x}^L)p(\boldsymbol{x}^L)d\boldsymbol{\xi}d\boldsymbol{x}^L && \text{condition on LF model} \\ &= \int_{\mathbb{R}^D} p(\boldsymbol{x}^H|\boldsymbol{x}^L)p(\boldsymbol{x}^L)d\boldsymbol{x}^L. && \text{marginalization} \end{aligned} \tag{2.2}$$

For the second issue, ROMs introduce a low-dimensional latent variable $\boldsymbol{z} \in \mathbb{R}^d$ with $d \ll D$ to establish a connection between the LF and HF models via $p(\boldsymbol{x}^H|\boldsymbol{x}^L)$,

$$p(\boldsymbol{x}^H) = \int_{\mathbb{R}^D} p(\boldsymbol{x}^H|\boldsymbol{x}^L)p(\boldsymbol{x}^L)d\boldsymbol{x}^L \tag{2.3}$$

$$= \int_{\mathbb{R}^D}\int_{\mathbb{R}^d} p(\boldsymbol{x}^H|\boldsymbol{z},\boldsymbol{x}^L)p(\boldsymbol{z}|\boldsymbol{x}^L)p(\boldsymbol{x}^L)d\boldsymbol{z}d\boldsymbol{x}^L. \tag{2.4}$$

The latent variable $\boldsymbol{z}$ determines a low-dimensional representation of $\boldsymbol{x}$ that captures the relationship between the LF and HF QoIs, possibly with considerably less HF data for training [281, 393].

In this work, we assume that the condition $p(\boldsymbol{x}^H|\boldsymbol{z},\boldsymbol{x}^L) = p(\boldsymbol{x}^H|\boldsymbol{z})$ holds, which leads to an AE model for $p(\boldsymbol{x}^H)$, given by

$$p(\boldsymbol{x}^H) = \int_{\mathbb{R}^D}\int_{\mathbb{R}^d} \underbrace{p(\boldsymbol{x}^H|\boldsymbol{z})}_{\text{decoder}}\underbrace{p(\boldsymbol{z}|\boldsymbol{x}^L)}_{\text{encoder}}p(\boldsymbol{x}^L)d\boldsymbol{z}d\boldsymbol{x}^L. \tag{2.5}$$

Once the AE model is built, the HF QoI can be estimated following

$$p(\boldsymbol{x}^H) = \int_{\mathbb{R}^d} p(\boldsymbol{x}^H|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}. \tag{2.6}$$

This work aims at generating new (approximate) samples from $p(\boldsymbol{x}^H)$ – rather than deriving an explicit representation $p(\boldsymbol{x}^H)$ — which allows us to estimate statistical properties of $\boldsymbol{x}^H$, e.g., $\mathbb{E}[\boldsymbol{x}^H]$ and $\mathrm{Cov}[\boldsymbol{x}^H]$. This involves three key components, namely, an encoder $p(\boldsymbol{z}|\boldsymbol{x}^L)$, the latent variable $p(\boldsymbol{z})$, and a decoder $p(\boldsymbol{x}^H|\boldsymbol{z})$. In the remaining of this section, we discuss two main ingredients to construct these components. In Section 2.3.1, we introduce a VAE approach to building the encoder and decoder in a Bayesian setting. In Section 2.3.2, we discuss an option to the structure of the latent variable $\boldsymbol{z}$.

2.3.1 Variational Autoencoder (VAE)

This section introduces the VAE [273, 450], a widely-used deep generative model capable of using samples of $\boldsymbol{x}$ to construct an estimate of $p(\boldsymbol{x})$ from which new samples of $\boldsymbol{x}$ can be drawn. As a deep Bayesian model, the VAE compresses and reconstructs data in a non-linear and probabilistic manner, while regularizing the model via a Kullback–Leibler (KL) divergence term (Equation (2.9),

which distinguishes it from regular AE models. The VAE is composed of two distinct probabilistic components, namely the encoder and the decoder as depicted in Figure 2.2. In contrast to AEs, the encoder and decoder of a VAE map data to random vectors, rather than deterministic values. The probabilistic encoder produces two separate vectors, representing the mean and standard deviation of a resulting multivariate Gaussian random vector $\boldsymbol{z}$. In this context, the covariance matrix of $\boldsymbol{z}$ is assumed to be diagonal. The probabilistic decoder maps the latent variable $\boldsymbol{z}$ back to the observation space by sampling from the decoder's output distribution. When the expected output is a continuous random variable, which is the primary focus of this work, the decoder result is traditionally assumed to be deterministic and returns the mean value of the decoder distribution [273]. In other words, the decoder $p(\boldsymbol{x}|\boldsymbol{z})$ becomes a Dirac distribution located at $D(\boldsymbol{z})$, where D is the deterministic decoder function. By enforcing a prior on the latent variable $\boldsymbol{z}$, the VAE can synthesize new samples of $\boldsymbol{x}$ by sampling the latent variable and evaluating the decoder.

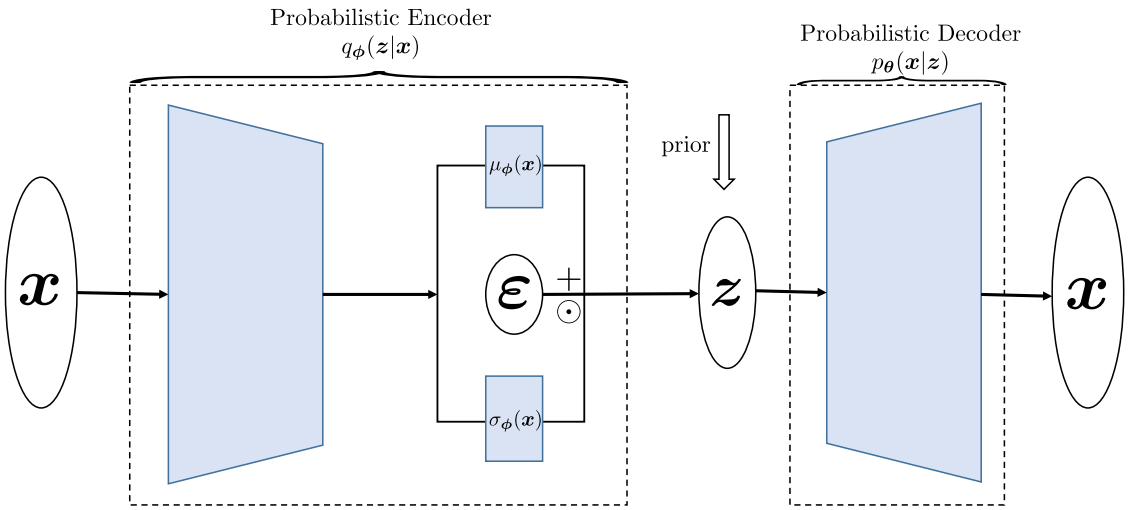


Figure 2.2: The probabilistic encoder $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ of a VAE produces two separate vectors, $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x})$ and $\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{x})$, which respectively represent the mean and standard deviation of resulting latent variable $\boldsymbol{z}$ following a multivariate Gaussian distribution. The random vector $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ provides randomness for the encoder output $\boldsymbol{z}$ and is used for the reparameterization trick in Equation (2.12).

In detail, the VAE introduces a latent variable $\boldsymbol{z}$ with its prior $p(\boldsymbol{z})$ in a low-dimensional

latent space and parameterizes a probabilistic decoder $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ with parameters $\boldsymbol{\theta}$ to establish a joint pdf $p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})$. According to the Bayes' rule, the posterior density is given by

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{z})p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})}{\int p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})d\boldsymbol{z}}. \tag{2.7}$$

In practice, computing $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ is intractable due to the unknown marginal density $\int p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})d\boldsymbol{z}$. To address this issue, the VAE employs a variational inference approach [59] and approximates the posterior density with a pdf $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ parameterized by $\boldsymbol{\phi}$. By introducing the variational replacement $q_{\boldsymbol{\phi}}$, the log-likelihood of $\boldsymbol{x}$ can be decomposed as

$$\log(p_{\boldsymbol{\theta}}(\boldsymbol{x})) = \mathrm{KL}\big(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}))\big) + \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}} \log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right)}_{\text{ELBO}}, \tag{2.8}$$

where $\mathrm{KL}(\cdot\|\cdot)$ is the Kullback-Leibler (KL) divergence and $\mathbb{E}_{q_{\boldsymbol{\phi}}}$ is the expectation over $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$. The KL divergence term in Equation (2.8) measures the discrepancy between the true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ and the variational posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ and is unknown in practice. The second term in Equation (2.8) is known as the evidence lower bound (ELBO), which is a lower bound of the log-likelihood due to the non-negativity of the KL divergence. In variational inference, the ELBO is maximized instead of the log-likelihood due to its tractable form. By maximizing the ELBO, the VAE model enhances a lower bound value of the log-likelihood and mitigates the discrepancy between the variational and true posteriors.

The VAE objective function, ELBO, can be further decomposed into two parts

$$\mathrm{ELBO}(\boldsymbol{\phi}, \boldsymbol{\theta}) = \mathbb{E}_{q_{\boldsymbol{\phi}}} \log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right) \tag{2.9}$$

$$= -\underbrace{\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z}))}_{\text{regularization term}} + \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}} \log(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}))}_{\text{reconstruction term}}. \tag{2.10}$$

The first part is the KL divergence between the prior $p(\boldsymbol{z})$ and the variational posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ measuring the distance between the two densities. The second term, $\mathbb{E}_{q_{\boldsymbol{\phi}}} \log(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}))$, is the log-conditional-probability of $\boldsymbol{x}$ that is averaged over the variational posterior $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}$. This component is often perceived as a negative reconstruction error. For example, when the conditional density

$p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ is Gaussian centered at the decoder output $D_{\boldsymbol{\theta}}(\boldsymbol{z})$, where $D_{\boldsymbol{\theta}}(\boldsymbol{z})$ is a neural-network-based decoder function, $\log(p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}))$ becomes the negative 2-norm reconstruction error $-\|\boldsymbol{x} - D_{\boldsymbol{\theta}}(\boldsymbol{z})\|^2$.

In order to estimate $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, gradient ascent is applied to maximize ELBO with gradients $\nabla_{\boldsymbol{\phi}}$ELBO and $\nabla_{\boldsymbol{\theta}}$ELBO. However, the gradient of ELBO with respect to $\boldsymbol{\phi}$, i.e.,

$$\nabla_{\boldsymbol{\phi}}\text{ELBO} = \nabla_{\boldsymbol{\phi}}\mathbb{E}_{q_{\boldsymbol{\phi}}} \log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right) \tag{2.11}$$

cannot be computed directly since the expectation $\mathbb{E}_{q_{\boldsymbol{\phi}}}$ depends on $\boldsymbol{\phi}$. Instead,the VAE uses a new random vector $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \boldsymbol{I})$ and represents latent samples as $\boldsymbol{z}_{\boldsymbol{\varepsilon}} = \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{x}) \odot \boldsymbol{\varepsilon} + \boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x})$, where $\odot$ is the Hadamard (element-wise) product. Stochastic gradient ascent (or its variants) is performed for each mini-batch of samples $\{\boldsymbol{x}_i\}_{i=1}^B$ by passing them through the encoder and obtaining $\sigma_{\boldsymbol{\phi}}(\boldsymbol{x}_i)$ and $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}_i)$, and generating new $\boldsymbol{z}_{\boldsymbol{\varepsilon}_i}$ by sampling $\boldsymbol{\varepsilon}_i$. An unbiased estimate of the gradient is generated via

$$\begin{aligned}
\nabla_{\boldsymbol{\phi}}\text{ELBO} &= \nabla_{\boldsymbol{\phi}}\mathbb{E}_{\boldsymbol{\varepsilon}} \log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}_{\boldsymbol{\varepsilon}})p(\boldsymbol{z}_{\boldsymbol{\varepsilon}})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\varepsilon}}|\boldsymbol{x})}\right) \\
&= \mathbb{E}_{\boldsymbol{\varepsilon}}\nabla_{\boldsymbol{\phi}} \log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}_{\boldsymbol{\varepsilon}})p(\boldsymbol{z}_{\boldsymbol{\varepsilon}})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\varepsilon}}|\boldsymbol{x})}\right) \\
&\approx \frac{1}{B}\sum_{i=1}^{B} \nabla_{\boldsymbol{\phi}} \log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}=\boldsymbol{x}_i|\boldsymbol{z}=\boldsymbol{z}_{\boldsymbol{\varepsilon}_i})p(\boldsymbol{z}=\boldsymbol{z}_{\boldsymbol{\varepsilon}_i})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}=\boldsymbol{z}_{\boldsymbol{\varepsilon}_i}|\boldsymbol{x}=\boldsymbol{x}_i)}\right).
\end{aligned} \tag{2.12}$$

The method for estimating the gradient in Equation (2.12), known as the reparametrization trick [273], can be applied to any form of $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, provided that it is associated with an easy-to-sample distribution. It further allows for decoupling of the expectation from $\boldsymbol{\phi}$ in Equation (2.12), thereby enabling the optimization of the objective function.

2.3.2 Auto-regressive Method

The central challenge of bi-fidelity modeling is to establish a connection between LF and HF model outputs. The VAE in Section 2.3.1 presents a methodology for building the encoder and decoder, which involves the exploration of an appropriate latent space. When using bi-fidelity data, we additionally require a suitable architecture for the latent variable $\boldsymbol{z}$ to model the relation between the LF and HF solutions. This architecture must be relatively simple as we assume only

limited HF data is available. For example, in [90], the authors use an encoder-decoder structure in conjunction with a latent bi-fidelity modeling approach that minimizes the distance between the reduced basis coefficients. We extend this method to a more general form.

For the case of the probabilistic encoder and decoder, we split the latent random vector $\boldsymbol{z}$ into two parts, $\boldsymbol{z}^L$ and $\boldsymbol{z}^H$, and apply a linear auto-regression from $\boldsymbol{z}^L$ to $\boldsymbol{z}^H$, inspired by the well-known Gaussian process (GP) based linear auto-regressive method [261, 262, 300]. This approach incorporates multivariate Gaussian priors for both fidelity models and postulates a linear, element-wise relationship between the models. The HF latent random vector $\boldsymbol{z}^H$ can be represented as a transformation of the LF latent random vector $\boldsymbol{z}^L$ through

$$z_i^H = a_i z_i^L + b_i, \forall i = 1, 2, \ldots, d \tag{2.13}$$

where a_i serves as a scaling factor and b_i is a Gaussian random variable. In some works, e.g., [262, 261], z_i^H and z_i^L are indexed with a spatial variable, which has been omitted here for clarity. The model assumes that no knowledge of z_i^H can be extracted from z_j^L if z_i^L is known and $i \neq j$, which implies $\mathrm{Cov}(z_i^H, z_j^L | z_i^L) = 0, \forall i \neq j$.

2.4 Bi-fidelity Variational Auto-encoder (BF-VAE)

In this section, we present the BF-VAE model. Section 2.4.1 outlines the architecture of the BF-VAE, a bi-fidelity extension of the ELBO objective function, and an algorithm designed to train the BF-VAE. The bi-fidelity information bottleneck (BF-IB) theory is introduced in Section 2.4.2, providing an interpretation of the BF-VAE from the perspective of information theory. Section 2.4.3 delves into the analysis of an error stemming from the probabilistic encoder trained by LF data.

2.4.1 Architecture, Objective Functions, and Algorithm

The principle behind the BF-VAE involves maximizing a lower bound of the HF log-likelihood, as the VAE does, but primarily utilizing LF data. To achieve this, a VAE-based structure is devised to leverage a latent space to model the relationship between the LF and HF data. The BF-VAE

model is comprised of three probabilistic components: an encoder, a latent auto-regression, and a decoder. The probabilistic encoder $q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)$, parameterized with $\boldsymbol{\phi}$ and trained using LF data, maps LF observations into LF latent representations. The latent auto-regression $p_{\boldsymbol{\psi}}(\boldsymbol{z}^H|\boldsymbol{z}^L)$, parameterized with $\boldsymbol{\psi}$ and specifically designed for a bi-fidelity regression in the latent space, as shown in Equation (2.14), significantly reduces the amount of HF data required for training due to its low-dimensionality. The probabilistic decoder $p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}^H)$, parameterized with $\boldsymbol{\theta}$, is first pre-trained with LF data and then refined with HF data, mapping the HF latent representations back into the observation space by returning the mean of the resulting HF distribution. A schematic illustration of the proposed BF-VAE model is depicted in Figure 2.3.

A crucial part of the BF-VAE is building a connection from the LF latent variable $\boldsymbol{z}^L$ to the HF latent variable $\boldsymbol{z}^H$. As presented in Section 2.3.2, we specify the latent conditional density $p_{\boldsymbol{\psi}}(\boldsymbol{z}^H|\boldsymbol{z}^L)$ to be a linear auto-regressive model, assumed to follow the Gaussian distribution $\mathcal{N}(\boldsymbol{K}_{\boldsymbol{\psi}}(\boldsymbol{z}^L), \gamma^2\boldsymbol{I})$ with parameters $\boldsymbol{\psi}$. The dimensions of $\boldsymbol{z}^L$ and $\boldsymbol{z}^H$ are assumed to be the same in order to enforce the symmetric structure between the encoder $q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)$ and the decoder $p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{x}^H)$. Note that in practical scenarios, HF and LF QoIs may inherently possess different latent dimensions. For the sake of simplicity and computational convenience, their dimensions are constrained to be equal within this study. The mapping $\boldsymbol{K}_{\boldsymbol{\psi}}$ consists of two parameterized vector components, $\boldsymbol{a}_{\boldsymbol{\psi}}$ and $\boldsymbol{b}_{\boldsymbol{\psi}}$, defined by the affine transformation

$$\boldsymbol{K}_{\boldsymbol{\psi}}(\boldsymbol{z}^L) = \boldsymbol{a}_{\boldsymbol{\psi}} \odot \boldsymbol{z}^L + \boldsymbol{b}_{\boldsymbol{\psi}}. \tag{2.14}$$

In this work, $\boldsymbol{K}_{\boldsymbol{\psi}}$ is implemented as a simplified single-layer neural network with a diagonal weight matrix and a bias vector. The hyperparameter γ is fixed for all entries for simplicity. When $\gamma \to 0$, $p_{\boldsymbol{\psi}}(\boldsymbol{z}^H|\boldsymbol{z}^L)$ converges in distribution to the Dirac distribution $\delta_{\boldsymbol{K}_{\boldsymbol{\psi}}(\boldsymbol{z}^L)}$, which makes the latent auto-regression a deterministic map. The hyperparameter γ represents our confidence on how accurately $\boldsymbol{K}_{\boldsymbol{\psi}}$ captures the relation between the LF and HF latent variables; see more discussion about γ in Section 2.5.2.

The objective function of the BF-VAE is a variational lower bound of the HF log-likelihood

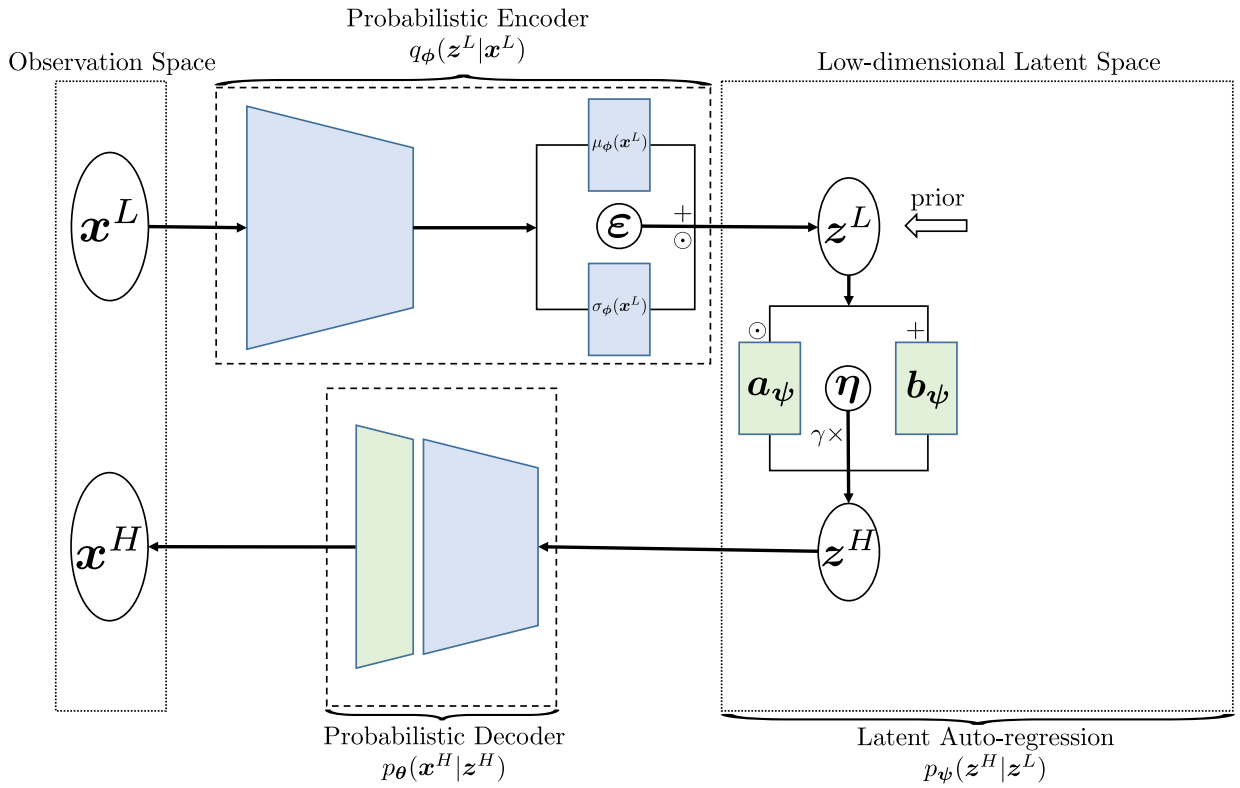


Figure 2.3: Structure of the proposed BF-VAE model. The probabilistic encoder $q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)$ produces two independent vectors, $\mu_{\boldsymbol{\phi}}(\boldsymbol{x}^L)$ and $\sigma_{\boldsymbol{\phi}}(\boldsymbol{x}^L)$, which represent the mean and standard deviation of a resulting multivariate Gaussian. The latent auto-regression $p_{\boldsymbol{\psi}}(\boldsymbol{z}^H|\boldsymbol{z}^L)$ is a simplified single-layer neural network $\boldsymbol{K}_{\boldsymbol{\psi}}$ defined in Equation (2.14) added with a noise $\gamma\boldsymbol{\eta}$. The probabilistic decoder $p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}^H)$ is pre-trained by LF data via the transfer learning technique, with its last layer tuned by LF and HF data pairs. White circles are random vectors and colored blocks are parameterized components for training. Blue blocks are solely trained by LF data and green blocks are trained by both LF and HF data.

as follows

$$\begin{aligned}\log p_{\boldsymbol{\theta},\boldsymbol{\psi}}(\boldsymbol{x}^H) &= \mathrm{KL}\big(q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)\|p_{\boldsymbol{\theta}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^H)\big) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\left[\log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}^H,\boldsymbol{z}_{\boldsymbol{\psi}})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\right)\right] \\ &\geq \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\left[\log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}^H,\boldsymbol{z}_{\boldsymbol{\psi}})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\right)\right] = \mathrm{ELBO}^{\mathrm{BF}}(\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\theta}),\end{aligned} \tag{2.15}$$

where the pdf of $\boldsymbol{z}_{\boldsymbol{\psi}} \coloneqq (\boldsymbol{z}^L, \boldsymbol{z}^H)$ is determined by the latent conditional density $p_{\boldsymbol{\psi}}(\boldsymbol{z}^H|\boldsymbol{z}^L)$ and the prior $p(\boldsymbol{z}^L)$. The above inequality follows from the non-negativity property of KL divergence. The lower bound of the HF log-likelihood in Equation (2.15) is called the bi-fidelity ELBO (BF-ELBO),

and denoted as $\text{ELBO}^{\text{BF}}(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\theta})$. The BF-ELBO consists of two terms, namely

$$\text{ELBO}^{\text{BF}}(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\theta}) = \underbrace{-\text{KL}\big(q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)\|p(\boldsymbol{z}^L)\big)}_{\text{regularization term}} + \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}_{\boldsymbol{\psi}})\big]}_{\text{HF reconstruction term}}. \tag{2.16}$$

The first term regularizes the encoder training by enforcing the encoder output to be close to the prior $p(\boldsymbol{z}^L)$. The second term is the HF log-likelihood conditioned on the latent variable $\boldsymbol{z}_{\boldsymbol{\psi}}$ and perceived as the HF reconstruction term. For example, when $p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}_{\boldsymbol{\psi}})$ is a Gaussian centered at the decoder output $D_{\boldsymbol{\theta}}(\boldsymbol{z}^H)$ with covariance $\beta\boldsymbol{I}$, the HF reconstruction term is a negative 2-norm $-\beta^{-1}\|\boldsymbol{x}^H - D_{\boldsymbol{\theta}}(\boldsymbol{z}^H)\|^2$ with $\boldsymbol{z}^H$ drawn from the encoder and the latent auto-regression with input $\boldsymbol{x}^L$. Note that by the condition, $p(\boldsymbol{x}^H|\boldsymbol{z}_{\boldsymbol{\psi}})$ is equivalent to $p(\boldsymbol{x}^H|\boldsymbol{z}^H)$. We use $\boldsymbol{z}_{\boldsymbol{\psi}}$ as the conditional variable for $p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}_{\boldsymbol{\psi}})$ so that it is consistent with the expectation $\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}$. A detailed derivation of Equations (2.15) and (2.16) are presented in B.1.

Optimizing BF-ELBO requires a large amount of both LF and HF data from their joint distribution $p(\boldsymbol{x}^L, \boldsymbol{x}^H)$ for convergence. However, the scarcity of HF data presents a challenge under the bi-fidelity setting. To address this issue, we apply a transfer learning technique, in which we opt to train the encoder and decoder using a large set of LF data, considering that the parameter spaces of $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ are significantly larger than that of $\boldsymbol{\psi}$. The small parameter space of $\boldsymbol{\psi}$ as a single layer in the low-dimensional latent space allows it to be trained solely with pairs of LF and HF data. As a result, we optimize the BF-ELBO in two steps, with two separate objectives,

$$\text{ELBO}^{\text{LF}}(\boldsymbol{\phi}, \boldsymbol{\theta}) = -\text{KL}\big(q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)\|p(\boldsymbol{z}^L)\big) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)}\big[\log\big(p_{\boldsymbol{\theta}}(\boldsymbol{x}^L|\boldsymbol{z}^L)\big)\big], \tag{2.17}$$

$$\text{ELBO}^{\text{HF}}(\boldsymbol{\psi}, \boldsymbol{\theta}) = \mathbb{E}_{q_{\boldsymbol{\phi}^{L*}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}_{\boldsymbol{\psi}})\big], \tag{2.18}$$

where $\boldsymbol{\phi}^{L*}$ in ELBO^{HF} is the optimal $\boldsymbol{\phi}$ for maximizing ELBO^{LF}. The first objective function $\text{ELBO}^{\text{LF}}(\boldsymbol{\phi}, \boldsymbol{\theta})$ is equivalent to a regular ELBO function discussed in Equation (2.9), as it trains a low-fidelity VAE (LF-VAE) solely using LF data. The trained LF-VAE returns the optimal LF encoder parameters $\boldsymbol{\phi}^{L*}$ and LF decoder parameters $\boldsymbol{\theta}^{L*}$. We assume the optimal HF decoder parameters $\boldsymbol{\theta}^{H*}$ is close to $\boldsymbol{\theta}^{L*}$ in the parameter space. Furthermore, we fix the decoder's parameters except for the last layer and set $\boldsymbol{\theta}^{L*}$ as the initial value for further optimizing ELBO^{HF} using both

LF and HF data to obtain optimal HF parameters $\boldsymbol{\psi}^{H*}, \boldsymbol{\theta}^{H*}$. Note that $\boldsymbol{\theta}^{H*}$ and $\boldsymbol{\theta}^{L*}$ are the same except for entries corresponding to the decoder's last layer, due to this transfer learning technique.

The presence of the parameter $\boldsymbol{\psi}$ in the expectation term in Equation (2.18) poses a challenge for the estimation of the gradients with respect to $\boldsymbol{\psi}$. To address this, we leverage the reparameterization trick outlined in Equation (2.12). Specifically, we introduce an auxiliary vector $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and set

$$\boldsymbol{z}_{\boldsymbol{\eta}}^{H} = \gamma\boldsymbol{\eta} + \boldsymbol{K}_{\boldsymbol{\psi}}(\boldsymbol{z}^{L}). \tag{2.19}$$

With mini-batch bi-fidelity samples $\{\boldsymbol{x}_i^L, \boldsymbol{x}_i^H\}_{i=1}^B$, the gradient w.r.t. $\boldsymbol{\psi}$ is estimated as

$$\begin{aligned}
\nabla_{\boldsymbol{\psi}}\mathrm{ELBO}^{\mathrm{HF}}(\boldsymbol{\psi}, \boldsymbol{\theta}) &= \mathbb{E}_{p_{\boldsymbol{\phi}^{L*}}(\boldsymbol{z}^L|\boldsymbol{x}^L)}[\nabla_{\boldsymbol{\psi}}\mathbb{E}_{p_{\boldsymbol{\psi}}(\boldsymbol{z}^H|\boldsymbol{z}^L)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}^H)]] \\
&= \mathbb{E}_{p_{\boldsymbol{\phi}^{L*}}(\boldsymbol{z}^L|\boldsymbol{x}^L)}[\mathbb{E}_{\boldsymbol{\eta}}[\nabla_{\boldsymbol{\psi}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}_{\boldsymbol{\eta}}^H)]] \\
&\approx \frac{1}{B}\sum_{i=1}^{B} \nabla_{\boldsymbol{\psi}} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^H = \boldsymbol{x}_i^H|\boldsymbol{z}^H = \boldsymbol{z}_{\boldsymbol{\eta}_i}^H),
\end{aligned} \tag{2.20}$$

where B is batch size and $\boldsymbol{z}_{\boldsymbol{\eta}_i}^H$ is the i-th sample from $\boldsymbol{x}_i^L$ and $\boldsymbol{\eta}_i$ as shown in Equation (2.19). Using the estimated gradients, we maximize $\mathrm{ELBO}^{\mathrm{LF}}(\boldsymbol{\phi}, \boldsymbol{\theta})$ and $\mathrm{ELBO}^{\mathrm{HF}}(\boldsymbol{\psi}, \boldsymbol{\theta})$ via stochastic gradient ascent (or its variants). To synthesize HF QoI samples, we sample from $p(\boldsymbol{z}^L)$ and subsequently propagate the samples through the trained latent auto-regressor with parameters $\boldsymbol{\psi}^{H*}$ and subsequently the decoder with parameters $\boldsymbol{\theta}^{H*}$. A summary of the steps in the BF-VAE is provided in Algorithm 3.

2.4.2 Bi-fidelity Information Bottleneck

One of the core ideas of bi-fidelity modeling is to fully exploit information from LF data for building HF results with limited HF data. However, to the best of the authors' knowledge, there is no previous work that explicitly models the bi-fidelity information transfer process incorporating information theory. In this work, we apply the information bottleneck (IB) principle [513, 478] to the BF-VAE model. The IB principle aims to define the essence of a "good" latent representation of data by finding a balance between information preservation and compression. According to

Algorithm 3: Bi-Fidelity Variational Auto-Encoder (BF-VAE)

Input: LF training set $\{\tilde{\boldsymbol{x}}_i^L\}_{i=1}^N$, LF-HF joint training set $\{(\boldsymbol{x}_i^L, \boldsymbol{x}_i^H)\}_{i=1}^n$
Output: Parameters $\boldsymbol{\psi}^{H*}, \boldsymbol{\theta}^{H*}$ for a HF pdf $p_{\boldsymbol{\theta},\boldsymbol{\psi}}(\boldsymbol{x}^H)$

1: Train a LF-VAE by maximizing $\text{ELBO}^{\text{LF}}(\boldsymbol{\phi}, \boldsymbol{\theta})$ in Equation (2.17) with LF realizations $\{\tilde{\boldsymbol{x}}_i^L\}_{i=1}^N$ to attain maximizers $\boldsymbol{\phi}^{L*}, \boldsymbol{\theta}^{L*} = \arg\max_{\boldsymbol{\phi},\boldsymbol{\theta}} \text{ELBO}^{\text{LF}}(\boldsymbol{\phi}, \boldsymbol{\theta})$.
2: Build a BF-VAE as shown in Figure 2.3 with parameters of the encoder and the decoder assigned to be $\boldsymbol{\phi}^{L*}, \boldsymbol{\theta}^{L*}$, and the latent auto-regression map $\boldsymbol{K}_{\boldsymbol{\psi}}(\cdot)$ in Equation (2.14) being initialized as an identity map.
3: Fix all the parameters of the BF-VAE except the decoder's last layer and the latent auto-regression's parameters.
4: Train the BF-VAE by maximizing $\text{ELBO}^{\text{HF}}(\boldsymbol{\psi}, \boldsymbol{\theta})$ in Equation (2.18) with sample pairs $\{(\boldsymbol{x}_i^L, \boldsymbol{x}_i^H)\}_{i=1}^n$ and find maximizers $\boldsymbol{\psi}^{H*}, \boldsymbol{\theta}^{H*} = \arg\max_{\boldsymbol{\psi},\boldsymbol{\theta}} \text{ELBO}^{\text{HF}}(\boldsymbol{\psi}, \boldsymbol{\theta})$.

IB, an optimal latent representation of data is maximally informative about the output while simultaneously compressive with respect to a given input.

In this section, we propose an interpretation of the BF-VAE model through the lens of the bi-fidelity IB (BF-IB) theory. We show that maximizing ELBO^{BF} in Equation (2.16) is equivalent to maximizing the BF-IB objective function in Equation (2.22) with $\beta = 1$ using $(\boldsymbol{x}^L, \boldsymbol{x}^H)$ data. Our analysis in this section builds a bridge between information theory and log-likelihood maximization in the bi-fidelity setting and presents a novel information-theoretic perspective on the BF-VAE model.

The mutual information, which is a non-negative, symmetric function, reflects the information that can be obtained about one random vector by observing another random vector. The definition of mutual information is as follows.

Definition 2.4.1. The mutual information [113] between random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is

$$\mathbb{I}(\boldsymbol{x}, \boldsymbol{y}) := \text{KL}\big(p(\boldsymbol{x}, \boldsymbol{y}) \| p(\boldsymbol{x})p(\boldsymbol{y})\big) = \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{y})} \log\left(\frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}\right), \tag{2.21}$$

where $p(\boldsymbol{x}, \boldsymbol{y})$ is the joint distribution of $\boldsymbol{x}$ and $\boldsymbol{y}$.

In the BF-VAE, our goal is to find a latent representative random vector $\boldsymbol{z}_{\boldsymbol{\psi}}$ corresponding to $\boldsymbol{x}^L$ for re-building HF QoI $\boldsymbol{x}^H$. According to the formula (15) in [513] or formula (5.164) in

[378], the bi-fidelity information bottleneck (BF-IB) objective function that we will maximize is

$$\mathrm{IB}_\beta^{\mathrm{BF}}(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\theta}) \coloneqq \mathbb{I}(\boldsymbol{z}_{\boldsymbol{\psi}}, \boldsymbol{x}^H) - \beta \mathbb{I}(\boldsymbol{x}^L, \boldsymbol{z}_{\boldsymbol{\psi}}), \tag{2.22}$$

where β is a non-negative hyperparameter, and $\boldsymbol{\phi}, \boldsymbol{\theta}$ are parameters of the encoder and decoder, respectively. The first term $\mathbb{I}(\boldsymbol{z}_{\boldsymbol{\psi}}, \boldsymbol{x}^H)$ represents the preserved information from $\boldsymbol{z}_{\boldsymbol{\psi}}$ to $\boldsymbol{x}^H$ by the decoder, while the second term $\mathbb{I}(\boldsymbol{x}^L, \boldsymbol{z}_{\psi})$ represents the information compressed by the encoder. The hyperparameter β is adjusted to balance the tradeoff between the information compression and the preservation. By maximizing the BF-IB objective function, we aim to find an optimal latent random vector $\boldsymbol{z}_{\boldsymbol{\psi}}$ as well as its relation with $\boldsymbol{x}^L, \boldsymbol{x}^H$, which are parameterized by $\boldsymbol{\psi}, \boldsymbol{\phi}$, and $\boldsymbol{\theta}$. Note that when the mutual information between $\boldsymbol{x}^L$ and $\boldsymbol{x}^H$ is zero, which means LF and HF data are independent, the searching for latent variable $\boldsymbol{z}_{\boldsymbol{\psi}}$ is vacuous. The schematic in Figure 2.4 describes the concept of BF-IB.

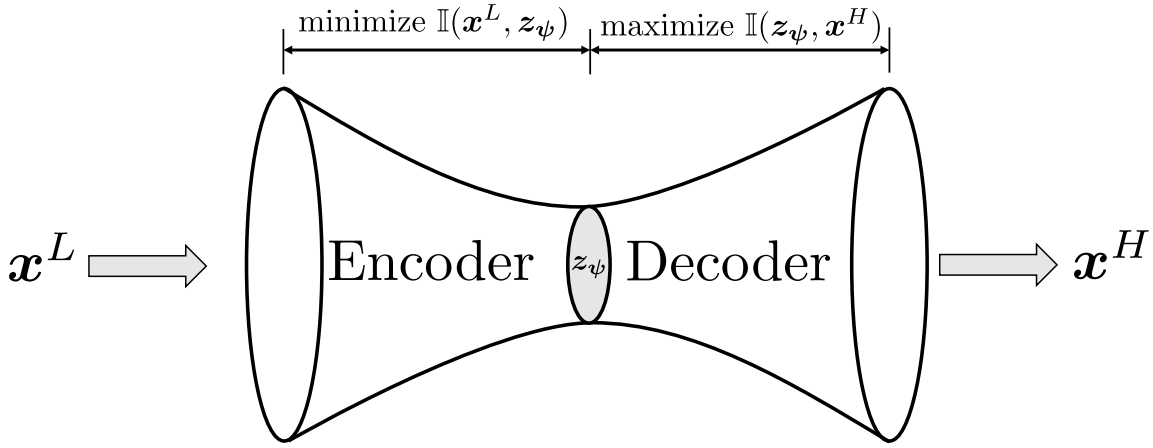


Figure 2.4: The bi-fidelity information bottleneck architecture has an encoder and a decoder, impacted by the information compression function $\mathbb{I}(\boldsymbol{x}^L, \boldsymbol{z}_{\boldsymbol{\psi}})$ and information preservation function $\mathbb{I}(\boldsymbol{z}_{\boldsymbol{\psi}}, \boldsymbol{x}^H)$, respectively. The random vector $\boldsymbol{z}_{\boldsymbol{\psi}}$ is designed to disclose the relation between LF and HF data in the latent space. The bottleneck part is necessary since only a limited number of HF realizations are available for learning the relationship between LF and HF data.

The BF-IB objective function can be decomposed as follows,

$$\mathrm{IB}_\beta^{\mathrm{BF}}(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\theta}) \equiv \mathbb{E}_{p(\boldsymbol{x}^L, \boldsymbol{x}^H)}\Big[\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}_{\boldsymbol{\psi}})\big] - \beta \mathrm{KL}\big(q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L) \| p(\boldsymbol{z}^L)\big)\Big]. \tag{2.23}$$

When $\beta = 1$, the BF-IB objective function becomes

$$\begin{aligned} \mathrm{IB}^{\mathrm{BF}}_{\beta=1}(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\theta}) &\equiv \mathbb{E}_{p(\boldsymbol{x}^L, \boldsymbol{x}^H)} \Big[\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)} \big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}_{\boldsymbol{\psi}}) \big] - \mathrm{KL}\big(q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L) \| p(\boldsymbol{z}^L)\big) \Big] \\ &= \mathbb{E}_{p(\boldsymbol{x}^L, \boldsymbol{x}^H)}[\mathrm{ELBO}^{\mathrm{BF}}(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\theta})]. \end{aligned} \tag{2.24}$$

This proves that the BF-IB function with $\beta = 1$ is equivalent to BF-ELBO in Equation (2.16) averaged with respect to the true joint distribution $p(\boldsymbol{x}^L, \boldsymbol{x}^H)$. The proof of (2.23) is presented in B.2. In Section 2.5.2, we incorporate the hyperparameter β into a prior of the decoder pdf $p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}^H)$, yielding an equivalent objective function containing β. Because the BF-VAE Algorithm 3 approximately maximizes BF-ELBO using joint realizations from $p(\boldsymbol{x}^L, \boldsymbol{x}^H)$, it produces an output that not only maximizes a variational lower bound of the HF log-likelihood but also the IB-BF objective function.

2.4.3 Bi-fidelity Approximation Error

Similar tothe VAE in Section 2.3.1, the BF-VAE model introduces an encoder to approximate the posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^H)$, which produces an approximation error stemming from its variational form. Moreover, since we employ LF data as the input of the encoder, the error also depends on the similarity between LF and HF data. In this section, we give the form of this error and provide insight into a measurement of similarity between LF and HF data under the current Bayesian framework.

Specifically, this error, denoted by $\mathcal{E}$, is the gap between HF log-likelihood and BF-ELBO averaged with respect to the true data distribution $p(\boldsymbol{x}^L, \boldsymbol{x}^H)$. The BF-VAE model assigns a multivariate Gaussian distribution $q_{\boldsymbol{\phi}}$ to the encoder without any guarantee that the given family includes the true HF posterior. The error $\mathcal{E}$,

$$\mathcal{E}(\boldsymbol{\psi}, \boldsymbol{\theta}) \coloneqq \min_{\boldsymbol{\phi}} \mathbb{E}_{p(\boldsymbol{x}^L, \boldsymbol{x}^H)} \big[\log p_{\boldsymbol{\theta}, \boldsymbol{\psi}}(\boldsymbol{x}^H) - \mathrm{ELBO}^{BF}(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\theta}) \big] \tag{2.25}$$

$$= \min_{\boldsymbol{\phi}} \mathbb{E}_{p(\boldsymbol{x}^L, \boldsymbol{x}^H)} \big[\mathrm{KL}\big(q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L) \| p_{\boldsymbol{\theta}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^H)\big) \big], \tag{2.26}$$

is directly derived from Equation (2.15). Since the error is a function of $\boldsymbol{\psi}$ and $\boldsymbol{\theta}$, the final performance of the trained BF-VAE model is determined by $\mathcal{E}(\boldsymbol{\psi}^{H*}, \boldsymbol{\theta}^{H*})$, where $\boldsymbol{\psi}^{H*}$ and $\boldsymbol{\theta}^{H*}$ are the

trained parameters. As a KL divergence averaged on the bi-fidelity data $p(\boldsymbol{x}^L, \boldsymbol{x}^H)$, the error $\mathcal{E}$ can be interpreted as the average difference between the latent representations from LF and HF, which depends on the similarity between the LF and HF data.

To improve the BF-ELBO's proximity to the HF log-likelihood, it is helpful to identify a form of $q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)$ that is potentially close to $p_{\boldsymbol{\theta}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^H)$. However, in practice, determining such a form is often infeasible [470]. Alternatively, bringing LF data closer to HF data can also reduce the error by making their latent representations more similar.

2.5 Priors and Hyperparameters

In the previous section, we introduced the principle concept of the BF-VAE model. In this section, we show two practical components of the BF-VAE model. We discuss the prior distribution selection in Section 2.5.1. An introduction to the hyperparameters and their effects on the BF-VAE performance is given in Section 2.5.2.

2.5.1 Choices of Prior Distributions

Prior distribution, a crucial aspect of Bayesian modeling, is chosen to reflect our prior belief of the parameter or facilitate computation. All the prior distributions utilized in the BF-VAE model are outlined in Table 2.1.

Table 2.1: Selected distributions for different components are presented. Here, $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}^L)$ and $\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{x}^L)$ are the outputs of the variational encoder. $\boldsymbol{K}_{\boldsymbol{\psi}}$ is the parameterized latent mapping in Equation (2.14). $\gamma \in \mathbb{R}$ and $\beta > 0$ are hyperparameters.

Component	Notation	VAE Model(s)	Prior Distribution
LF Latent Variable	$p(\boldsymbol{z}^L)$	LF-VAE, BF-VAE	$\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
Variational Encoder	$q_{\boldsymbol{\phi}}(\boldsymbol{z}^L\|\boldsymbol{x}^L)$	LF-VAE, BF-VAE	$\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}}(\boldsymbol{x}^L), \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\boldsymbol{x}^L))$
Latent Auto-regression	$p_{\boldsymbol{\psi}}(\boldsymbol{z}^H\|\boldsymbol{z}^L)$	BF-VAE	$\mathcal{N}(\boldsymbol{K}_{\boldsymbol{\psi}}(\boldsymbol{z}^L), \gamma^2\boldsymbol{I})$
LF Decoder	$p_{\boldsymbol{\theta}}(\boldsymbol{x}^L\|\boldsymbol{z}^L)$	LF-VAE	$\mathcal{N}(D_{\boldsymbol{\theta}}(\boldsymbol{z}^L), \beta\boldsymbol{I})$
HF Decoder	$p_{\boldsymbol{\theta}}(\boldsymbol{x}^H\|\boldsymbol{z}^H)$	BF-VAE	$\mathcal{N}(D_{\boldsymbol{\theta}}(\boldsymbol{z}^H), \beta\boldsymbol{I})$

Let $\{\tilde{\boldsymbol{x}}_i^L\}_{i=1}^N \sim p(\boldsymbol{x}^L)$ and $\{\boldsymbol{x}_j^L, \boldsymbol{x}_j^H\}_{j=1}^n \sim p(\boldsymbol{x}^L, \boldsymbol{x}^H)$ denote the LF and BF training datasets,

respectively. When using the priors in Table 2.1, the LF-ELBO in Equation (2.17) becomes

$$\text{ELBO}_{\beta}^{\text{LF}}(\boldsymbol{\phi},\boldsymbol{\theta}) = \underbrace{-\frac{1}{2}\big(\|\boldsymbol{\mu}_{\boldsymbol{\phi}}(\tilde{\boldsymbol{x}}_i^L)\|_2^2 + \|\boldsymbol{\sigma}_{\boldsymbol{\phi}}(\tilde{\boldsymbol{x}}_i^L)\|_2^2 - \mathbf{1}^T \log \boldsymbol{\sigma}_{\boldsymbol{\phi}}^2(\tilde{\boldsymbol{x}}_i^L)\big)}_{\text{regularization}} \tag{2.27}$$

$$+ \frac{1}{N}\sum_{i=1}^{N}\Bigg[\underbrace{\beta^{-1}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L=\tilde{\boldsymbol{x}}_i^L)}\|D_{\boldsymbol{\theta}}(\boldsymbol{z}^L) - \boldsymbol{x}_i^L\|_2^2}_{\text{LF reconstruction}}\Bigg]. \tag{2.28}$$

Similarly, the HF-ELBO in Equation (2.18) becomes

$$\text{ELBO}_{\beta}^{\text{HF}}(\boldsymbol{\psi},\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\Bigg[\underbrace{\beta^{-1}\mathbb{E}_{p_{\boldsymbol{\phi}^{L*}}(\boldsymbol{z}^L|\boldsymbol{x}^L=\boldsymbol{x}_i^L)}\big[\|D_{\boldsymbol{\theta}}(\boldsymbol{z}_{\boldsymbol{\eta}_i}^H) - \boldsymbol{x}_i^H\|_2^2\big]}_{\text{BF reconstruction}}\Bigg], \tag{2.29}$$

where $\boldsymbol{z}_{\boldsymbol{\eta}_i}^H$ is computed as in Equation (2.19) with $\boldsymbol{\eta}_i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

2.5.2 Hyperparameter Setting

The BF-VAE consists of two primary hyperparameters, namely β in Equation (2.23) and Table 2.1 and γ in Equation (2.19), which must be specified prior to training. Note that $\text{ELBO}^{\text{BF}}(\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\theta})$ in Equation (2.16) with priors outlined in Table 2.1 is

$$\text{ELBO}^{\text{BF}}(\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\theta}) = -\text{KL}\big(q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)\|p(\boldsymbol{z}^L)\big) + \beta^{-1}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\big[\|\boldsymbol{x}^H - D_{\boldsymbol{\theta}}(\boldsymbol{z}^H)\|^2\big] \tag{2.30}$$

$$\equiv -\beta\text{KL}\big(q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)\|p(\boldsymbol{z}^L)\big) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\big[\|\boldsymbol{x}^H - D_{\boldsymbol{\theta}}(\boldsymbol{z}^H)\|^2\big], \tag{2.31}$$

where β is a hyperparameter adjusting the contribution of the KL regularization term and also aligns with the β in Equation (2.23). Thus, the parameter β in the decoder prior of Table 2.1 is analogous to the one in the BF-IB objective function in Equation (2.22), which also plays a similar role to the β parameter in β-VAE [229]. As discussed in Section 2.4.2, the value of β balances the tradeoff between the information compression and preservation from the perspective of IB and may be derived from prior knowledge or may be tuned using the validation error of the LF-VAE model.

Following the discussion of Section 2.4.1, the hyperparameter γ serves as the variance of the latent auto-regressive model. It indicates the degree of confidence in the accuracy of $\boldsymbol{K}_{\boldsymbol{\psi}}$ (defined in Equation (2.14)) when modeling the latent variables of the LF and HF models. A larger value of γ allows the auto-regression output to deviate further from $\boldsymbol{K}_{\boldsymbol{\psi}}$ but also increases the variance

of the ELBO gradients due to a more noisy reparameterized $\boldsymbol{z}_{\boldsymbol{\epsilon}}^H$, as shown in Equation (2.19). In our numerical experiments, we observe that linear auto-regression in the latent space is capable of accurately capturing the relationship between the LF and HF latent representations, which means that we are able to choose γ to be small. Since a smaller γ ensures faster convergence when optimizing the HF-ELBO, we therefore choose $\gamma = 0$.

2.6 Empirical Results

In this section, we present empirical results obtained by applying the BF-VAE to three PDE-based forward UQ problems. In more details, we first simulate a composite beam in Section 2.6.1, then we discuss studying a thermally-driven cavity fluid flow with a high-dimensional uncertain input in Section 2.6.2. Finally, we consider a 1D viscous Burgers' equation in Section 2.6.3. For each problem, we present the computational cost ratio between the HF and LF models. The outcomes of the BF-VAE model are then compared with the HF-VAE, a standard VAE model trained exclusively with high-fidelity data. These two models have the same architecture and activation functions.

Our primary objective is to showcase the efficacy of the BF-VAE in improving the accuracy of VAE models trained using high-fidelity training data only, particularly when limited high-fidelity data is available.[2]

To examine the quality of data produced by a generative model, it is crucial to use an appropriate evaluation metric. While human evaluation may be adequate for determining the quality of outputs from models generating images and text, such an approach is not generally appropriate for evaluating the quality generated data corresponding to PDE solutions. Therefore, we seek to identify a statistical distance that allows us to compare the difference between the true $p(\boldsymbol{x}^H)$ and the VAE surrogates $p_{\boldsymbol{\psi},\boldsymbol{\theta}}(\boldsymbol{x}^H)$ without incurring excessive computational cost. For deep generative models, there are two major evaluation options: Frechet inception distance (FID) [228] and kernel inception distance (KID) [54]. FID is most appropriate for evaluating image-based

[2] The Python code implementation is available at https://github.com/CU-UQ/Bi-fidelity-VAE.

generative models as it uses a pre-trained convolutional neural network [508]. In this study, we employ KID, which stems from a statistical distance named maximum mean discrepancy (MMD) [196]. KID represents the deviation of the distribution of the generated realizations from the distribution of the true test data, and can be conveniently computed when data is high-dimensional. A smaller KID value indicates a closer distance between two empirical distributions. Given a non-negative and symmetric kernel function $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ and data $\{\boldsymbol{x}_i\}_{i=1}^T$ and $\{\boldsymbol{y}_i\}_{i=1}^T$, the KID is defined as

$$\begin{aligned}\mathrm{KID}(\{\boldsymbol{x}_i\}_{i=1}^T, \{\boldsymbol{y}_j\}_{j=1}^T) = \frac{1}{T(T-1)} \sum_{\substack{i,j=1 \\ i\neq j}}^{T} k(\boldsymbol{x}_i, \boldsymbol{x}_j) - \frac{2}{T^2} \sum_{i=1}^{T}\sum_{j=1}^{T} k(\boldsymbol{x}_i, \boldsymbol{y}_j) \\ + \frac{1}{T(T-1)} \sum_{\substack{i,j=1 \\ i\neq j}}^{T} k(\boldsymbol{y}_i, \boldsymbol{y}_j).\end{aligned} \tag{2.32}$$

Following [54], the kernel function we choose is the rational quadratic kernel

$$k_{\mathrm{rq}}(\boldsymbol{x}_i, \boldsymbol{y}_j) \coloneqq \sum_{\ell \in \mathcal{I}} \left(1 + \frac{\|\boldsymbol{x}_i - \boldsymbol{y}_j\|^2}{2\ell}\right)^{-\ell}, \tag{2.33}$$

where $\mathcal{I} = \{0.2, 0.5, 1.0, 2.0, 5.0\}$ is a mixture of length scales to balance the bias effects from the different values. In order to evaluate the efficacy of the BF-VAE and HF-VAE models, we generated new realizations from the trained BF-VAE and HF-VAE models, denoted by $\{\boldsymbol{x}_i^{\mathrm{BF}}\}_{i=1}^T, \{\boldsymbol{x}_j^{\mathrm{HF}}\}_{j=1}^T$, respectively, where T is the test data size. The KIDs between the generated realizations and the actual data for testing $\{\boldsymbol{x}_l^H\}_{l=1}^T$ are computed as

$$\mathrm{KID}^{\mathrm{BF}} \coloneqq \mathrm{KID}(\{\boldsymbol{x}_l^H\}_{l=1}^T, \{\boldsymbol{x}_i^{\mathrm{BF}}\}_{i=1}^T), \tag{2.34}$$

$$\mathrm{KID}^{\mathrm{HF}} \coloneqq \mathrm{KID}(\{\boldsymbol{x}_l^H\}_{l=1}^T, \{\boldsymbol{x}_j^{\mathrm{HF}}\}_{j=1}^T). \tag{2.35}$$

We also compute the KID value between LF data $\{\boldsymbol{x}_m^L\}_{m=1}^T$ and true test data as a baseline

$$\mathrm{KID}^{\mathrm{LF}} \coloneqq \mathrm{KID}(\{\boldsymbol{x}_l^H\}_{l=1}^T, \{\boldsymbol{x}_m^L\}_{m=1}^T). \tag{2.36}$$

Further discussion and technical details regarding KID are presented in B.3. In addition to KID, we provide 1,000 synthesized QoI realizations generated by both the trained HF-VAE and BF-VAE models, along with their corresponding true HF counterparts. Their statistics including the

relative errors of the first and the second moments generated from both models are also reported. These additional results serve to further validate the KID outcomes. The hyperparameter β in Equation (2.23) is tuned to reduce the KID value of the LF-VAE and, as discussed in Section 2.5.2, γ in Equation (2.19) is assumed to be zero. The neural network architecture outlined in this section is determined based on the performance of VAEs trained using LF data. The evaluation of VAE performance is measured using KID and ELBO values. Our experiments indicate that VAE performance is not highly sensitive to architectural details, such as layer width and latent space size. The results presented with the chosen architecture serve demonstration purposes. In applications, the specific architecture should be determined by practical constraints, such as the budget for high-fidelity evaluations, the size of LF data for training, available computational resources, and other relevant considerations.

2.6.1 Composite Beam

Following [218, 123, 124, 95], we consider a plane stress, cantilever beam with composite cross section and hollow web, as shown in Figure 2.5. The quantities of interest, in this case, are the displacements of the top cord at 128 equi-spaced points and represented as a vector with 128 entries. The uncertain inputs of the model are denoted as $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3, \xi_4)$, where ξ_1, ξ_2 and ξ_3 are the Young's moduli of the three components of the cross section and ξ_4 is the intensity of the applied distributed force on the beam; see Figure 2.5. These are assumed to be statistically independent and uniformly distributed. The range of the input parameters, as well as the other deterministic parameters, are provided in Table 2.2.

Table 2.2: The values of the parameters in the composite cantilever beam model. The centers of the holes are at $x = \{5, 15, 25, 35, 45\}$. The entries of $\boldsymbol{\xi}$ are drawn independently and uniformly at random from the specified intervals.

L	h_1	h_2	h_3	w	r	ξ_1	ξ_2	ξ_3	ξ_4
50	0.1	0.1	5	1	1.5	[0.9e6, 1.1e6]	[0.9e6, 1.1e6]	[0.9e4, 1.1e4]	[9, 11]

The HF QoI $\boldsymbol{x}^H$ is based on a finite element discretization of the beam using a triangular

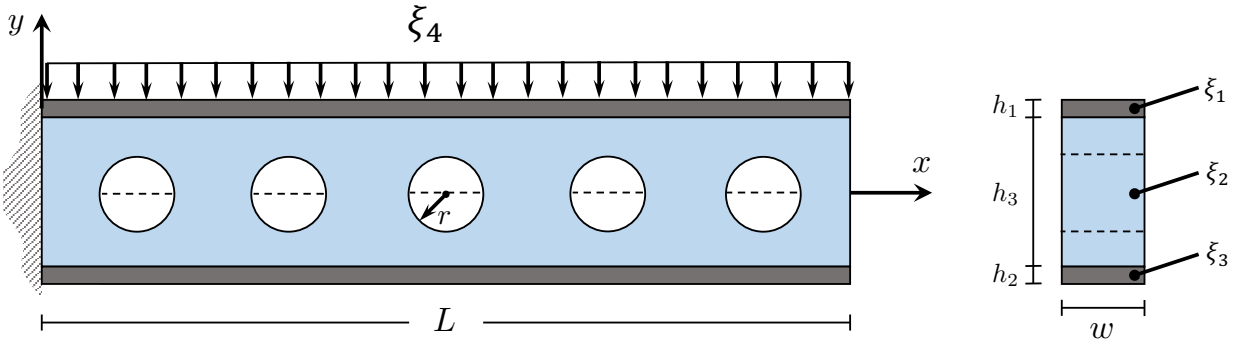


Figure 2.5: Cantilever beam (left) and the composite cross section (right) adapted from [218].

mesh, as Figure 2.7 shows. The LF QoI $\boldsymbol{x}^L$ is derived from the Euler–Bernoulli beam theory in which the vertical cross sections are assumed to remain planes throughout the deformation. The LF model ignores the shear deformation of the web and does not take the circular holes into account, which makes the LF results smoother than their HF counterparts, as displayed in Figure 2.6. Considering the Euler–Bernoulli theorem, the vertical displacement u is

$$EI_n \frac{d^4u(x)}{dx^4} = -\xi_4, \tag{2.37}$$

where E and I_n are, respectively, the Young's modulus and the moment of inertia of an equivalent cross section consisting of a single material. We let $E = \xi_3$, and the width of the top and bottom sections are $w_1 = (\xi_1/\xi_3)w$ and $w_2 = (\xi_2/\xi_3)w$, while all other dimensions are the same, as Figure 2.5 shows. The solution of (2.37) is

$$u(x) = -\frac{qL^4}{24EI_n}\left(\left(\frac{x}{L}\right)^4 - 4\left(\frac{x}{L}\right)^3 + 6\left(\frac{x}{L}\right)^2\right). \tag{2.38}$$

Since the LF data are directly obtained through an explicit formula in Equation (2.38), its computational cost is negligible.

The VAE models are implemented using fully-connected neural networks for both the encoder and decoder, each with two hidden layers and widths of 64 and 16 units, and GeLU activation functions. The latent space dimension is fixed at 4. The optimization of the VAE models is performed using the Adam optimizer with a learning rate of 1×10^{-3} and Adam-betas of 0.9 and

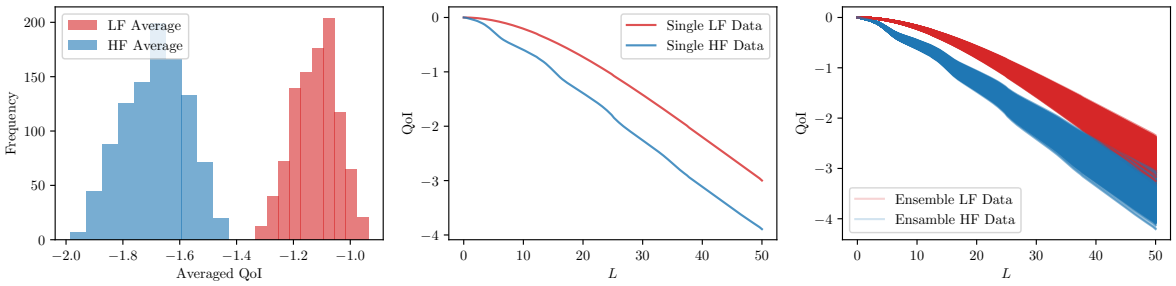


Figure 2.6: A histogram of the averaged QoI solutions along 128 spatial points from the LF and HF composite beam models (left), one single realization of LF and HF data from the same random input (middle), and 1,000 realizations of LF and HF QoIs (right).

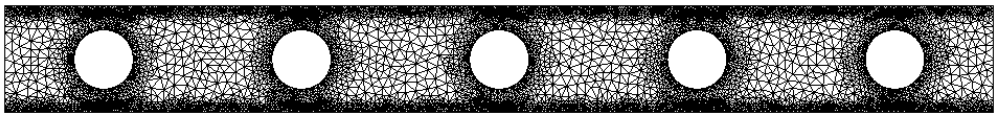


Figure 2.7: Finite element mesh used to generate HF solutions.

0.99. The batch size is set to 64, and the number of epochs for the initial training of the LF-VAE (line 1 in Algorithm 3) is 2,000, followed by 1,000 for the BF-VAEs (line 4 in Algorithm 3). The hyperparameter β is 0.04.

A LF-VAE model is first trained with $N = 4{,}000$ samples drawn from $p(\boldsymbol{x}^L)$. Since LF data are directly generated from Equation (2.38), the cost of LF data is trivial and can be ignored. A BF-VAE is built with parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ initialized from the trained LF-VAE following Algorithm 3. We examine the performance of the BF-VAE as a function of the number of HF training samples, with HF-VAE trained solely on the same HF data as a baseline. The KID performance is evaluated using 1,000 test data and 1,000 samples from each of the trained VAEs across 10 trials, with the results averaged over the trials.

Figure 2.8 illustrates the KID performance of both the BF-VAE and HF-VAE, with the x-axis representing the number of HF data used for training and y-axis being KID values evaluated following Equation (2.32). The results show that KID^{BF} begins to converge with a small number of HF data, while KID^{HF} only starts to converge when the number of HF data exceeds 100. Both model outputs are better than simply using LF data for inferring uncertainty statistics. Given the practical limitations on the acquisition of HF data, the superiority of the BF-VAE model is thus evident. We also observe that when the size of HF data is large, e.g., more than 1,000, KID^{HF} surpasses KID^{BF} and achieves a better accuracy level. This is typical of multi-fidelity strategies and explanations are available in [123]. Figure 2.9 presents 1,000 realizations drawn from the trained HF-VAE and BF-VAE. We expect the displacement as a function of horizontal distance to be smooth, but the HF-VAE samples fail to present these properties when $n < 1{,}000$. It shows that the BF-VAE is able to provide a reliable result with only a small number of HF training samples, while the HF-VAE requires more HF data to converge. In Table 2.3, we show the relative errors of first (mean value) and element-wise second moments estimated by the corresponding generated results shown in Figure 2.9. We observe that the BF-VAE generated smaller statistical errors compared to the HF-VAE. Both figures and the table demonstrate the effectiveness of the BF-VAE algorithm in utilizing the information from LF data to estimate the distribution of the HF QoI.

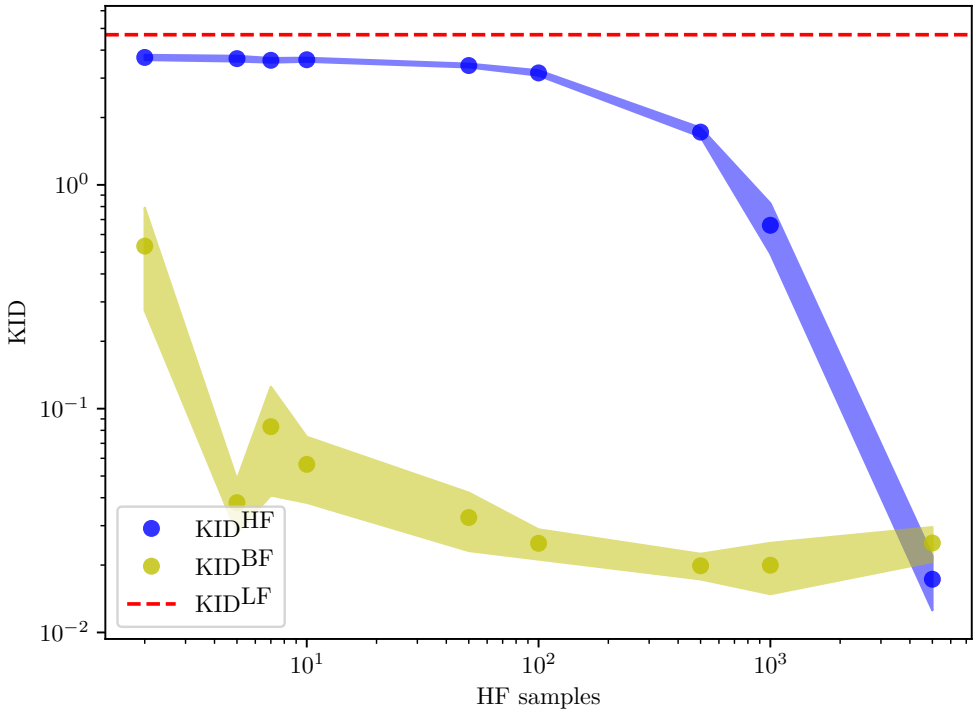


Figure 2.8: The KID results for the composite beam example given different sizes of HF data. Each circle represents the average KID between test data and the VAEs' realizations over 10 separate trials. The shaded area is half the empirical standard deviation of these 10 trials. The red dashed line represents the KID between HF and LF data.

Table 2.3: The relative errors of the first and second moments of HF-VAE/BF-VAE generated QoI shown in Figure 2.9.

	$n = 10$	$n = 100$	$n = 1,000$
First moment (HF-VAE)	7.41e-1	6.50e-1	8.18e-2
First moment (BF-VAE)	1.13e-2	6.64e-3	4.88e-3
Second moment (HF-VAE)	8.99e-1	8.05e-1	1.01e-1
Second moment (BF-VAE)	1.77e-2	9.33e-3	1.40e-2

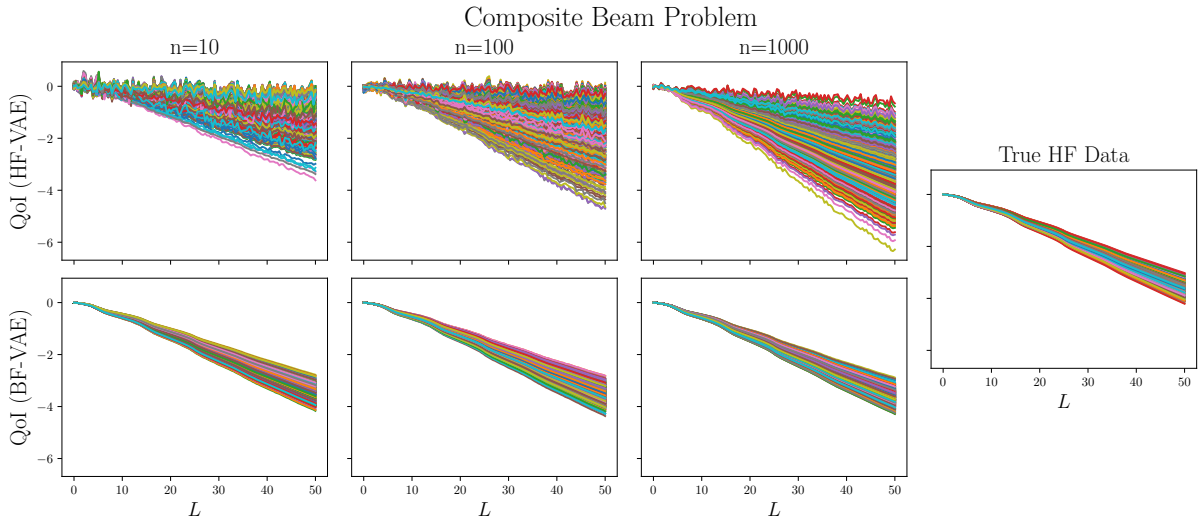


Figure 2.9: Comparison of 1,000 samples generated from the trained HF-VAE (top row), BF-VAE (bottom row) and the true HF model (right). A different number of HF realizations are used in each of the first three columns: $n = 10$ (left column), $n = 100$ (middle left column), and $n = 1{,}000$ (middle right column).

2.6.2 Cavity Flow

Here we consider the case of the temperature-driven fluid flow in a 2D cavity, with the quantity of interest being the heat flux along the hot wall as Figure 2.10 shows. The left-hand wall is considered as the hot wall with a random temperature T_h, while the right-hand wall, referred to as the cold wall, has a smaller random temperature T_c with a constant mean of $\bar{T}_c$. The horizontal walls are treated as adiabatic. The reference temperature and the temperature difference are given by $T_{\text{ref}} = (T_h + \bar{T}_c)/2$ and $\Delta T_{\text{ref}} = T_h - \bar{T}_c$, respectively. The normalized governing equations are given by

$$\begin{aligned} &\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} = -\nabla p + \frac{\text{Pr}}{\sqrt{\text{Ra}}} \nabla^2 \boldsymbol{u} + \text{Pr}\Theta \mathbf{e}_y, \\ &\nabla \cdot \boldsymbol{u} = 0, \\ &\frac{\partial \Theta}{\partial t} + \nabla \cdot (\boldsymbol{u}\Theta) = \frac{1}{\sqrt{\text{Ra}}} \nabla^2 \Theta, \end{aligned} \tag{2.39}$$

where $\mathbf{e}_y$ is the unit vector $(0, 1)$, $\boldsymbol{u} = (u, v)$ is the velocity vector field, $\Theta = (T - T_{\text{ref}})/\Delta T_{\text{ref}}$ is normalized temperature, p is pressure, and t is time. The hot wall at $x = 0$, the cold wall at $x = 1$, and two other walls at $y = 0$ and $y = 1$ are subject to no-slip boundary conditions. The dimensionless Prandtl and Rayleigh numbers are defined as $\text{Pr} = \nu_{\text{visc}}/\alpha$ and $\text{Ra} = g\tau\Delta T_{\text{ref}} W^3/(\nu_{\text{visc}}\alpha)$, respectively, where W is the width of the cavity, g is gravitational acceleration, ν_{visc} is kinematic viscosity, α is thermal diffusivity, and τ is the coefficient of thermal expansion. We set $g = 10$, $W = 1$, $\tau = 0.5$, $\Delta T_{\text{ref}} = 100$, $\text{Ra} = 10^6$, and $\text{Pr} = 0.71$. On the cold wall, we apply a temperature distribution with stochastic fluctuations as

$$T(x = 1, y) = \bar{T}_c + \sigma_T \sum_{i=1}^{M} \sqrt{\lambda_i} \varphi_i(y) \xi_i, \tag{2.40}$$

where $\bar{T}_c = 100$ is a constant, $\{\lambda_i\}_{i\in[M]}$ and $\{\varphi_i(y)\}_{i\in[M]}$ are the M largest eigenvalues and corresponding eigenfunctions of the kernel $k(y_1, y_2) = \exp(-|y_1 - y_2|/0.15)$, and each $\xi_i \overset{\text{i.i.d.}}{\sim} U[-1, 1]$. We let the input dimension $M = 52$ and $\sigma_T = 2$. The vector $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_{52})$ is the uncertain input of the model. These considerations align with previous works in [29, 430, 212, 210, 214, 95].

Unlike the composite beam problem, the low-fidelity model is based on a coarser spatial discretization of the governing equation. Specifically, we employ the finite volume method with a

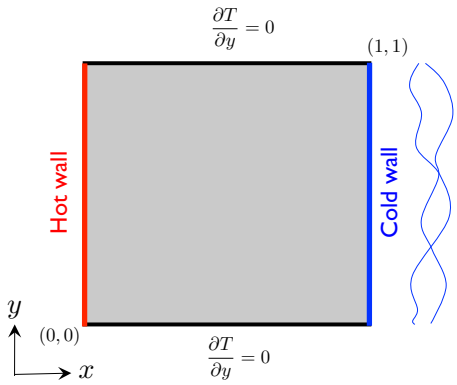


Figure 2.10: A figure of the temperature-driven cavity flow problem, reproduced from Figure 5 of [170].

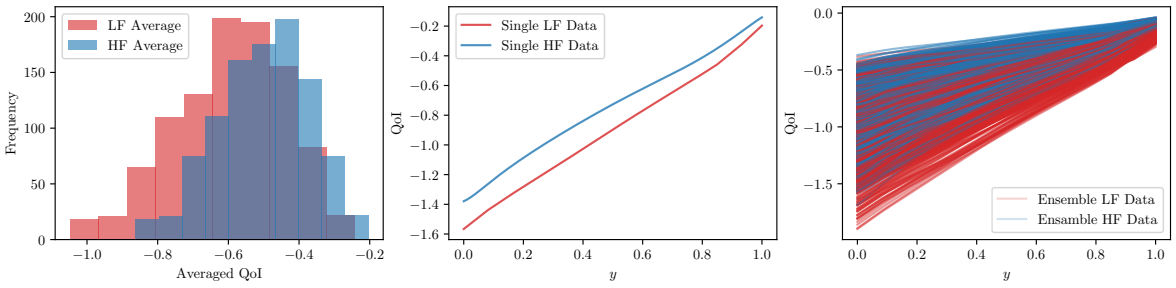


Figure 2.11: A histogram of the QoI solutions averaged across all spatial points from the LF and HF cavity flow models is shown in the left figure, two single realizations separately from LF and HF with the same input are demonstrated in the middle figure, and 1,000 LF and HF QoIs are presented in the right figure.

grid of size 256×256 to produce the HF QoI $\boldsymbol{x}^H$ and a coarser grid of size 16×16 to produce the LF QoI $\boldsymbol{x}^L$. A comparison of LF and HF estimates of the QoI is presented in Figure 2.11. Based on the analysis from [170], the HF/LF ratio of the computational cost for this problem is 9410.14, which means the time for computing one HF realization is equivalent to the time for computing approximately 9410 LF realizations. Since the auto-encoder structure requires both LF and HF input data to have the same dimension, we interpolate the LF data linearly on the fine grid and let the QoI be the (interpolated) steady-state heat flux along the hot wall at 221 equispaced points over $[0.067, 0.933]$, including the endpoints. For the VAE models, we use fully connected neural networks to model the encoder and decoder with ReLU activation functions, three hidden layers, and internal widths 221–128–64–16 determined by some preliminary tests. The dimension of the latent space is 4. The number of LF samples used for training the LF-VAE is $N = 4{,}000$. The cost of generating these 4,000 samples is equal to 42% of the cost of generating a single HF realization. As this equivalent cost is sufficiently small compared to the number of HF realizations we used for testing, we ignore the cost of generating the LF data. The optimizer is Adam with a learning rate 1×10^{-3} and Adam-betas $0.9, 0.99$. The batch size for the optimization is set to 64. The epoch number is 2,000 for the initial LF-VAE training (line 1 in Algorithm 3) followed by 1,000 epochs for the BF-VAE training (line 4 in Algorithm 3). The value of the hyperparameter β is set to 4.5.

The average KID between HF data and data generated by the HF-VAE and BF-VAE, for different numbers of HF training samples sizes, are shown in Figure 2.12. The averages are computed over 10 trials between 1,000 real samples and 1,000 VAE-simulated realizations. Newly generated realizations of HF-VAEs and BF-VAEs based on different HF training sample sizes are shown in Figure 2.13, with their first moments' (mean value) and element-wise second moments' relative errors collected in Table 2.4. The result of Figure 2.12 indicates that KID^{BF} is consistently lower than KID^{HF} but gets closer when more HF data is available, which is further validated by the results of moments' relative errors. Figure 2.13 suggests that the BF-VAE returns smoother and more reliable predictions compared to those of HF-VAEs, especially with limited HF training data, which means the BF-VAE produces more realistic results. Both figures and the table reveal that

the BF-VAE has better performance than the HF-VAE when the two models are given the same amount of HF training data.

Table 2.4: The relative errors of the first and second moments of HF-VAE/BF-VAE generated QoI shown in Figure 2.13.

	$n = 10$	$n = 100$	$n = 1,000$
First moment (HF-VAE)	2.57e-2	5.60e-3	3.57e-3
First moment (BF-VAE)	1.49e-2	4.60e-3	1.10e-3
Second moment (HF-VAE)	9.32e-2	6.02e-2	4.61e-2
Second moment (BF-VAE)	8.67e-2	5.83e-2	7.13e-2

2.6.3 Burgers' Equation

The last example is a one-dimensional unsteady viscous Burgers' equation with uncertain initial conditions and viscosity. The random velocity field $u(x, t, \boldsymbol{\xi})$ with parameters $\boldsymbol{\xi}$ is governed by

$$\begin{aligned} &\frac{\partial u(x,t,\boldsymbol{\xi})}{\partial t} + u(x,t,\boldsymbol{\xi})\frac{\partial u(x,t,\boldsymbol{\xi})}{\partial x} = \frac{\partial}{\partial x}\left(\nu \frac{\partial u(x,t,\boldsymbol{\xi})}{\partial x}\right), \quad (x,t) \in [0,1] \times [0,2], \\ &u(0,t,\boldsymbol{\xi}) = u(1,t,\boldsymbol{\xi}) = 0, \quad t \in [0,2] \\ &u(x,0,\boldsymbol{\xi}) = g(x,\boldsymbol{\xi}), \quad x \in [0,1], \end{aligned} \tag{2.41}$$

where the viscosity ν is modeled by a shifted beta random variable Beta$(0.5, 5)$ over $[0.01, 0.05]$. The initial condition $g(x, \boldsymbol{\xi})$ is a stochastic field given by

$$g(x,\boldsymbol{\xi}) = \sin(\pi x) + \sigma_g \sum_{k=2}^{M} \frac{1}{k} \sin(\pi k x)\xi_{k-1}, \tag{2.42}$$

where $\sigma_g = 1.2840 \times 10^{-1}$ and $M = 6$. The random inputs $\xi_1, \xi_2, \ldots, \xi_{M-1}$ are i.i.d. uniformly distributed between -1 and 1, resulting in a random input vector $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_{M-1}, \nu)$. The QoIs are the values of $u(x, t = 2, \boldsymbol{\xi})$ at 254 equi-spaced x nodes between 0 and 1, excluding the boundary points. To generate bi-fidelity data, the discretization of the Equation (2.41) is carried out using two space/time grid sizes. The LF data is obtained using the semi-implicit, two-step Adam-Bashforth solver with a spatial grid of size $\Delta x = 1.176 \times 10^{-2}$ and time step size of $\Delta t = 2 \times 10^{-2}$.

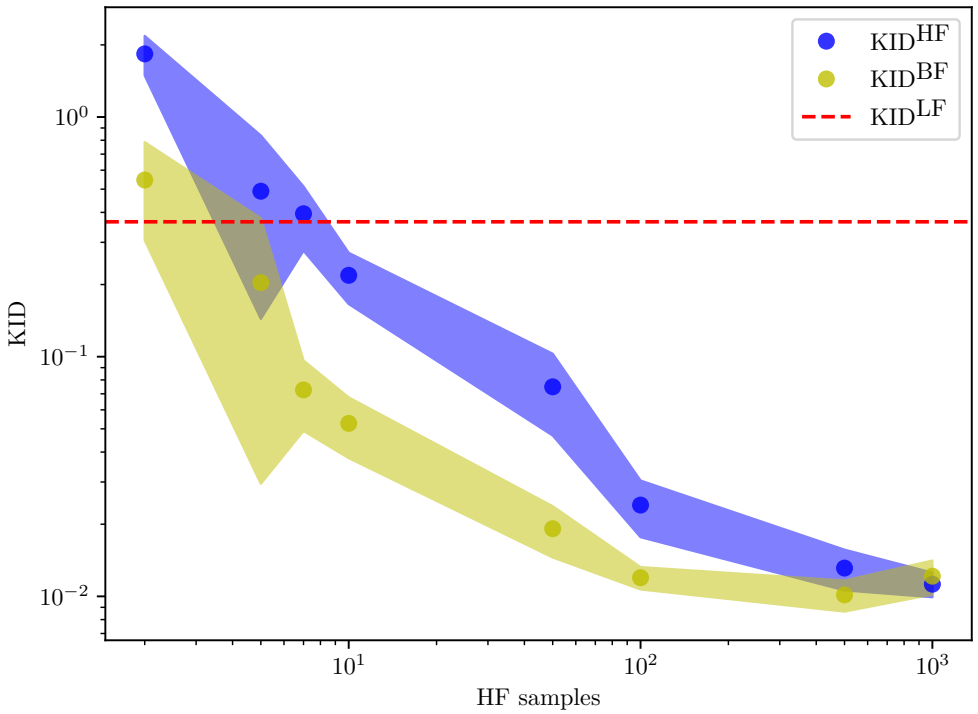


Figure 2.12: The KID result for the cavity flow problem given different sizes of HF data. Each point represents the average KID between test data and the VAEs' realizations over 10 separate trials. The shaded area corresponds to half the empirical standard deviation of these 10 trials. The red dashed line is the KID value between LF and HF data.

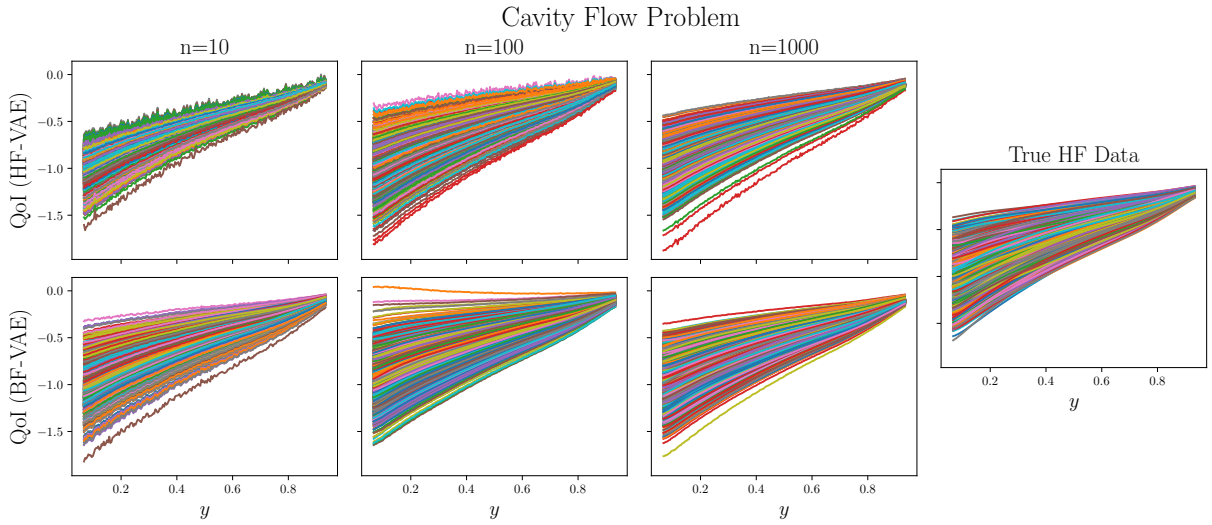


Figure 2.13: Comparison of 1,000 samples generated from the trained HF-VAE (top row), BF-VAE (bottom row) and the true HF model (right). A different number of HF realizations are used in each of the first three columns: $n = 10$ (left column), $n = 100$ (middle left column), and $n = 1{,}000$ (middle right column).

The same solver is applied for HF data, but with smaller space/time grid sizes, $\Delta x = 3.922 \times 10^{-3}$ and $\Delta t = 2 \times 10^{-4}$. The LF data is interpolated linearly on the finer grid so the dimensions of the LF and HF data are the same. The ratio of HF/LF computational cost is 98.07. A comparison between LF and HF data is presented in Figure 2.14.

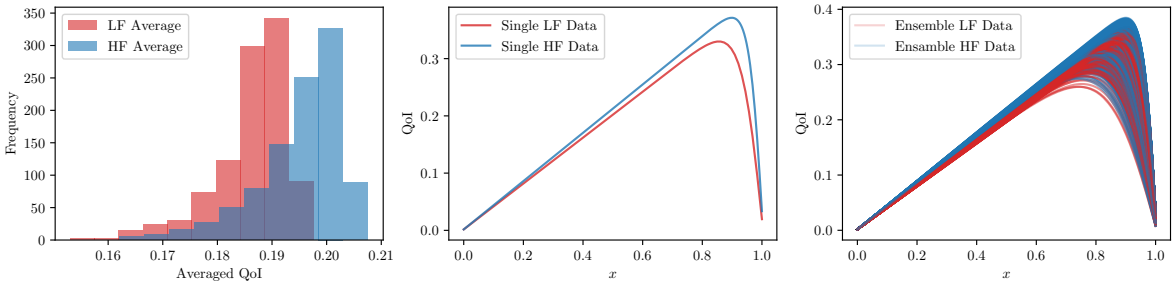


Figure 2.14: Histogram of the QoI values averaged across all spatial points from the LF and HF viscous Burgers' models is shown in the left figure, two single realizations separately from LF and HF models with the same input are presented in the middle figure, and 1,000 LF and HF QoIs are plotted in the right figure.

For the VAE implementations, we use fully connected neural networks to model the encoder and decoder, with four hidden layers as 254–256–128–64–16–4 with GeLU as activation functions. The dimension of the latent space is 4. As before, the Adam optimizer with a learning rate 1×10^{-3} and Adam-betas $0.9, 0.99$ is applied. The batch size for the optimization is 64. The epoch number is 2,000 for the initial LF-VAE training (line 1 in Algorithm 3) with an additional 1,000 epochs for the BF-VAE training (line 4 in Algorithm 3). We use $N = 400$ LF samples to train the LF-VAE, whose cost is equivalent to 4.08 HF realizations and sufficiently small to be ignored in the following presented results. The value of the hyperparameter β is set to 5×10^{-4}.

To validate the performance of the BF-VAE model, we compare its results with those of the HF-VAE using KID. The BF and HF KID results in Figure 2.15 are computed as the average over ten trials consisting of 1,000 test samples and 1,000 VAE-generated samples, with the KID^{LF} presented as the baseline. Additionally, we demonstrate the validity of the BF-VAE model by generating realizations and comparing them with the HF-VAE counterparts, as shown in Figure 2.16. The relative errors of the first moments (mean values) and element-wise second moments from HF-

VAE/BF-VAE generated QoI are presented in Table 2.5. Based on our evaluation, we observe that the BF-VAE model achieves better accuracy in estimating the HF QoI when n is small (< 100). We also observe that when the size of HF data is large, e.g., more than 600, KID^{HF} surpasses KID^{BF} and achieves a better accuracy. Similar with the case in Section 2.6.1, this is typical of multi-fidelity strategies and explanations are available in [123].

Table 2.5: The relative errors of the first and second moments of HF-VAE/BF-VAE generated QoI shown in Figure 2.16.

	$n = 10$	$n = 100$	$n = 1,000$
First moment (HF-VAE)	6.83e-2	3.49e-2	3.63e-3
First moment (BF-VAE)	7.72e-3	3.82e-3	7.20e-3
Second moment (HF-VAE)	1.07e-1	5.60e-2	1.45e-2
Second moment (BF-VAE)	1.34e-2	8.58e-3	1.39e-2

2.7 Conclusion

This paper presents a novel deep generative model, the bi-fidelity variational auto-encoder (BF-VAE), for generating synthetic realizations of spatio and/or temporal QoIs from parametric/stochastic PDEs through bi-fidelity data. With an autoencoder architecture, the BF-VAE exploits a low-dimensional latent space for bi-fidelity auto-regression, which significantly reduces the number of high-fidelity (HF) samples required for training. As such, the construction of the BF-VAE model is largely independent of the dimension of the stochastic input and applicable to QoIs that do not admit low-rank representations. A training criterion for the BF-VAE is proposed and analyzed using information bottleneck theory [513]. The empirical experiments demonstrate the efficacy of the proposed algorithm in scenarios when the amount of HF data is limited.

VAE-based approaches, including the BF-VAE, typically impose a multivariate Gaussian distribution on the encoder. As discussed in Section 2.4.3, this results in approximation errors. An interesting future research direction is to try using other deep generative models that do not suffer from this shortcoming in bi-fidelity UQ applications. Examples of promising models that

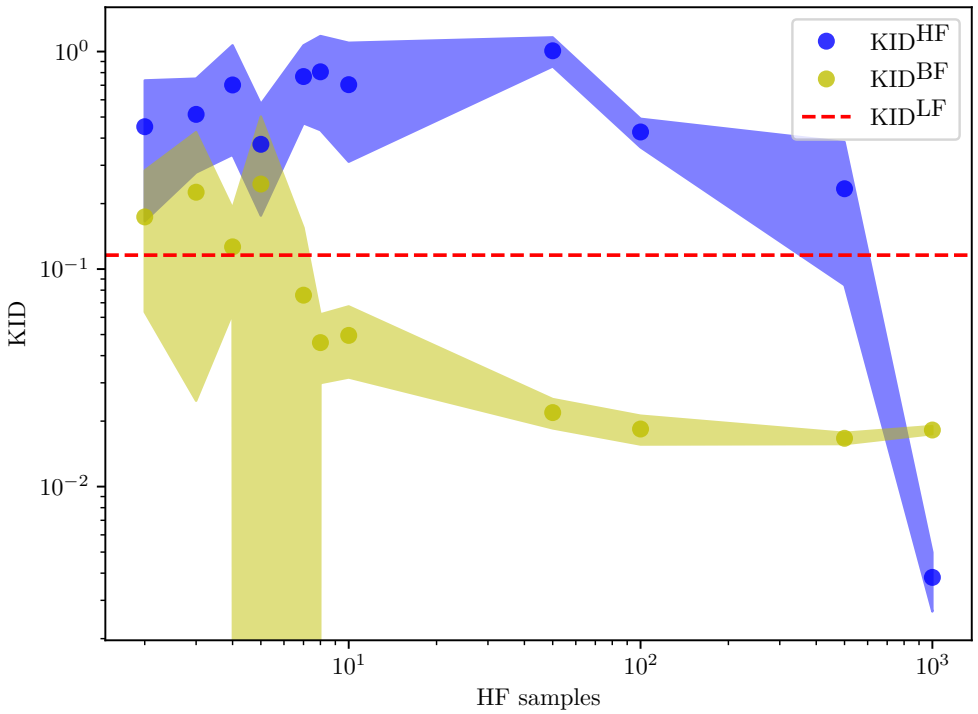


Figure 2.15: The KID result for the viscous Burgers' equation given different numbers of HF realizations. Each point represents the average KID between the test data and the VAEs' realizations over 10 separate trials. The shaded area corresponds to half the empirical standard deviation of these 10 trials. The red dash line is the KID between LF and HF data.

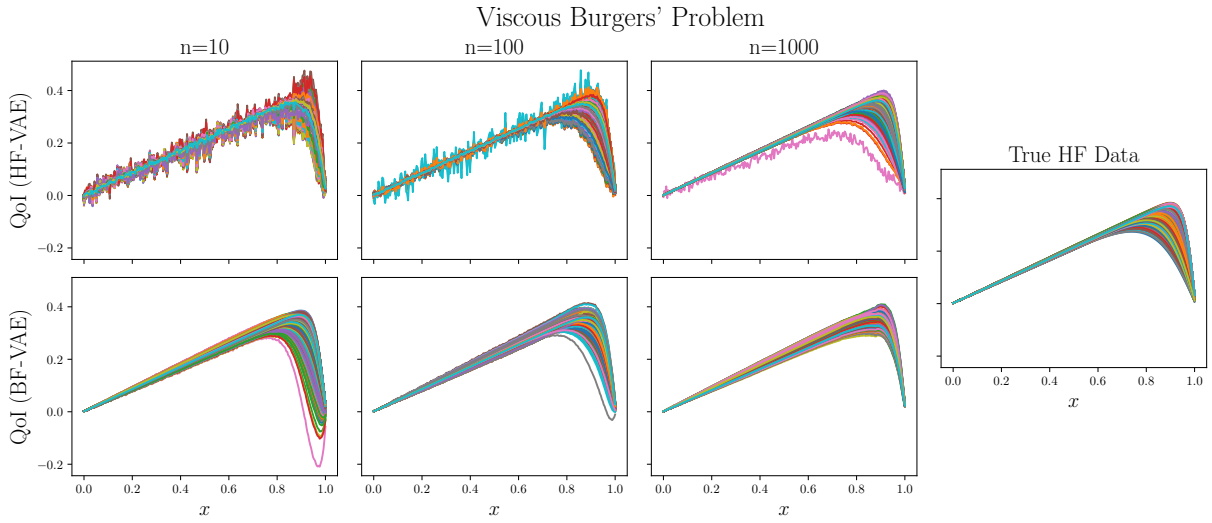


Figure 2.16: Comparison of 1,000 samples generated from the trained HF-VAE (top row), BF-VAE (bottom row) and the true HF model (right). A different number of HF realizations are used in each of the first three columns: $n = 10$ (left column), $n = 100$ (middle left column), and $n = 1000$ (middle right column).

have achieved state-of-the-art results in other domains include normalizing flows and diffusion models.

Chapter 3

Langevin Bi-fidelity Importance Sampling

3.1 Abstract

Estimating failure probability is a key task in the field of uncertainty quantification. In this domain, importance sampling has proven to be an effective estimation strategy; however, its efficiency heavily depends on the choice of the biasing distribution. An improperly selected biasing distribution can significantly increase estimation error. One approach to address this challenge is to leverage a less expensive, lower-fidelity surrogate. Building on the accessibility to such a model and its derivative on the random uncertain inputs, we introduce an importance sampling-based estimator, termed the Langevin bi-fidelity importance sampling (L-BF-IS), which uses score-function-based sampling algorithms to generate new samples and substantially reduces the mean square error (MSE) of failure probability estimation. The proposed method demonstrates lower estimation error, especially in high-dimensional input spaces and when limited high-fidelity evaluations are available. The L-BF-IS estimator's effectiveness is validated through experiments with two synthetic functions and two real-world applications governed by partial differential equations. These real-world applications involve a composite beam, which is represented using a simplified Euler-Bernoulli equation as a low-fidelity surrogate, and a steady-state stochastic heat equation, for which a pre-trained neural operator serves as the low-fidelity surrogate. [1]

[1] The original version of this work is presented in [93], co-authored with A. Doostan.

3.2 Introduction

Uncertainty ubiquitously appears in many real-world applications, such as weather forecasting, financial modeling, healthcare decision-making, and engineering design. In computational modeling, uncertainty is often represented by a random vector, defined within a specific probability distribution based on prior knowledge or observation data. One of the key goals of uncertainty quantification (UQ) is to estimate the probability of a device or system failure based on model outputs, also known as the quantity of interest (QoI). There are many methods to solve this problem, including the first-order reliability method (FORM) [221, 133] and its extension to the second order [175]. Other works involve the Monte Carlo sampling method [24]. However, the standard Monte Carlo method faces the challenge of slow convergence relative to sample size, especially when the probability of the failure event is small. In practical scenarios, model evaluation demands substantial computational resources, limiting the Monte Carlo method's feasibility. Consequently, there is significant interest in enhancing the convergence of the Monte Carlo method by reducing the number of model evaluations, primarily achieved by reducing the variance of estimators.

Variance reduction in Monte Carlo estimators can be primarily approached in two ways. The first method involves control variates [21, 171, 194], which uses correlated random variables to adjust the original estimator based on the covariance between the control and target variables. This adjustment yields a new estimator with reduced variance, provided the appropriately chosen control variable is well-correlated with the target variable. Despite its widespread application in UQ, the efficient control variate method is contingent on the availability of highly correlated control variables with known (or cheap to evaluate) mean and accurate covariance estimation, limiting its applicability. The second method is importance sampling (IS), which is the focus of this work. IS samples the input random variables following a different probability distribution (referred to as biasing distribution) to emphasize the regions that significantly impact the estimation. This approach effectively reduces the estimator variance and focuses on crucial input space areas, proving particularly useful in scenarios involving rare events or tail probability estimations. The critical

challenge in IS is constructing a suitable biasing distribution, a task complicated by limited access to model outputs under the UQ setting.

Several studies have examined the construction of biasing distributions specifically tailored for failure probability estimation. Adaptive importance sampling techniques, such as those in [72, 293, 184, 426], tune the biasing density within a parameterized family by adaptively finding the optimal density under the cross entropy criteria. Papaioannou et al. [415] discuss the application of sequential importance sampling (SIS) for estimating the probability of failure in structural reliability analysis. Initially developed for exploring posterior distributions and estimating normalizing constants in Bayesian inference, SIS involves a sequential reweighting operation that progressively shifts samples from the prior to the posterior distribution. This work was later adapted using the ensemble Kalman filter [538] and consensus sampling [17]. However, these methods may require extensive forward model computations, limiting their practical applicability.

To mitigate the computational burdens associated with high-fidelity (HF) models, adopting a "low-fidelity" (LF) model proves advantageous. This model, for instance, derived from the same solver but employing a coarser grid or an approximate surrogate function—either based on fixed basis functions or data-driven—offers reduced accuracy in exchange for significantly lower computational cost. This approach, often named bi-fidelity or multi-fidelity, has been integrated into many of the aforementioned methods. For instance, Li et al. [311] utilized surrogates of the limit state function as low-fidelity models to enhance adaptive importance sampling [310]. Similarly, Wagner et al. [537] extended sequential importance sampling to multi-level cases where coarse grid solutions serve as low-fidelity models. Peherstorfer et al. [424] proposed the multi-fidelity importance sampling method (MF-IS), which constructs the biasing distribution by applying a Gaussian mixture model to inputs whose LF evaluations indicate potential failures, suggesting that inputs failing under LF conditions are likely to fail under HF conditions as well. This strategy preserves the unbiased nature of the importance sampling estimator and does not confine the format of the LF model. Subsequent extensions of this framework [425, 287, 16] have integrated a collection of different estimators and explored the balance between computation and accuracy.

However, the aforementioned multi-fidelity methods using polynomial chaos surrogates or based on Gaussian mixture models are known for their rapidly growing complexity with the increase in the dimension of the inputs, denoted as D. Moreover, identifying the number of failure clusters for the Gaussian mixture model poses challenges without prior knowledge. In response to the identified challenges, a recent work by Cui et al. [115] introduced a deep importance sampling method. This method is notable for its biasing distribution construction with linear complexity $\mathcal{O}(D)$. This was achieved through the push-forward of a reference distribution under a series of order-preserving transformations, each shaped by a squared tensor-train decomposition. While this method offers theoretical and numerical advancements over [424], challenges related to the training of neural networks and its associated optimization error persist.

In practical applications, low-fidelity models often possess additional properties and information that can be leveraged. For instance, when a low-fidelity model is a simplified model, such as the Euler-Bernoulli equation for beam deflections [96, 94], its explicit formulation facilitates simple forward evaluation at minimal cost and provides derivative information. Similarly, when the low-fidelity model is a data-driven surrogate model, the recent development of auto-differentiation-enabled libraries [4, 421] produces derivatives of the forward surrogate map. These examples highlight the potential of utilizing additional knowledge from low-fidelity models to construct more effective biasing distributions for importance sampling estimators.

In this work, we introduce a new importance sampling estimator, named Langevin Bi-fidelity Importance Sampling (L-BF-IS). By leveraging a new parameterization of the biasing density function and the Metropolis-adjusted Langevin algorithm (MALA) [457, 455], this estimator scales favorably in high-dimensioned scenarios ($D \geq 100$). Specifically, the required number of iterations for this algorithm depends on $\mathcal{O}(D^{1/3})$ [177]. The contributions of this work are threefold:

(1) We introduce a new parameterization of the biasing density function leveraging a low-fidelity model; see Equation (3.5). Two approaches are proposed to tune the only hyper-parameter ℓ;

(2) We analyze the L-BF-IS estimator's statistical properties and estimation performance based on the relation between low-fidelity and high-fidelity models;

(3) We empirically demonstrate the effectiveness of the MALA on a multimodal biasing density function and the L-BF-IS performance through synthetic and real-world problems governed by differential equations with high-dimensional random inputs.

The structure of this work is as follows. Section 3.3 details the construction and implementation of L-BF-IS, presents a discussion on its error analysis. Section 3.4 demonstrates the performance of L-BF-IS using three numerical examples. [2] Finally, Section 1.6 concludes the paper and discusses avenues for future research.

3.3 Langevin Bi-fidelity Importance Sampling Estimator and its Properties

In this section, a detailed motivation, construction, and theoretical analysis of the proposed L-BF-IS estimator is presented. Section 3.3.1 introduces the concepts of Monte Carlo method and importance sampling. Section 3.3.2 presents the groundwork of L-BF-IS: the designed biasing distribution $q(\boldsymbol{z})$ and the formulation of L-BF-IS estimator. In Section 3.3.3, the statistical properties of the proposed L-BF-IS estimator, including its unbiasedness, variance, and consistency are discussed. Section 3.3.4 includes two approaches to estimate the most important parameter in our estimator, ℓ. Section 3.3.5 presents the MALA-based technique employed to sample the biasing distribution. A discussion on the influence of the relation between low-fidelity and high-fidelity models on the performance of L-BF-IS estimation is presented in Section 3.3.6. Section 3.3.7 provides insights on the potential sources of errors in L-BF-IS.

3.3.1 Background

We consider an input-output system encompassing an input random vector of dimension $D \in \mathbb{N}$ and an output random variable, named the quantity of interest (QoI), of dimension $d \in \mathbb{N}$.

[2] The codes are available at https://github.com/CU-UQ/L-BF-IS.

A probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is embedded in the input space so that $\Omega \subset \mathbb{R}^D$. The system is represented as a $\mathcal{F}$-measurable function that is equipped with two distinct levels of fidelity: a high-fidelity (HF) QoI function $f^{\mathrm{HF}} : \Omega \to \mathbb{R}^d$ and a low-fidelity (LF) QoI function $f^{\mathrm{LF}} : \Omega \to \mathbb{R}^d$, with $\Omega \subset \mathbb{R}^D$. The inputs are random variables $\boldsymbol{z}$ that are assumed to obey an absolutely continuous (with respect to Lebesgue measure) probability distribution, yielding a density function $p(\boldsymbol{z})$ with associated law $\mathbb{P}_p$. Additionally, for the failure probability, we define Borel-measurable performance functions $g^{\mathrm{HF}} : \mathbb{R}^d \to \mathbb{R}$ and $g^{\mathrm{LF}} : \mathbb{R}^d \to \mathbb{R}$. These two functions evaluate the failure result given a QoI and provide a value reflecting the outputs. For simplicity, we define $h^{\mathrm{HF}} \coloneqq g^{\mathrm{HF}} \circ f^{\mathrm{HF}}$ and $h^{\mathrm{LF}} \coloneqq g^{\mathrm{LF}} \circ f^{\mathrm{LF}}$. If $h^{\mathrm{HF}}, h^{\mathrm{LF}}(\boldsymbol{z}) < 0$, the result represents failures. In the following contexts, we call h^{HF} and h^{LF} as HF and LF functions, respectively. In the literature [311, 310], limit state function that describes $\{\boldsymbol{z} \mid h^{\mathrm{HF}}(\boldsymbol{z}) = 0\}$ is also discussed. The existence of such a limit state function is based on certain continuity of the function h^{HF}, which is not assumed in this work. We also define failure regions $\mathcal{A}_L \coloneqq (h^{\mathrm{LF}})^{-1}((-\infty, 0))$ and $\mathcal{A}_H \coloneqq (h^{\mathrm{HF}})^{-1}((-\infty, 0))$. Both $\mathcal{A}_H$ and $\mathcal{A}_L$ belong to $\mathcal{F}$ due to the measurable-function assumption and can be multi-modal. We let $\mathbb{E}_p$ and $\mathbb{V}_p$ denote the expectation and variance associated with the density $p(\boldsymbol{z})$, respectively.

Under the multi-fidelity scheme, we aim to evaluate the expected HF failure probability,

$$P_f \coloneqq \mathbb{P}_p[h^{\mathrm{HF}}(\boldsymbol{z}) < 0] = \mathbb{E}_p[\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}] = \int_\Omega \mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0} p(\boldsymbol{z}) d\boldsymbol{z} = \int_{\mathcal{A}_H} p(\boldsymbol{z}) d\boldsymbol{z} = \mathbb{P}_p[\mathcal{A}_H], \tag{3.1}$$

where $\mathbb{1}$ is the indicator function. The Monte Carlo estimator, with N samples, is

$$\widehat{P}_N^{\mathrm{MC}} \coloneqq \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z}_i)<0}, \quad \{\boldsymbol{z}_i\}_{i=1}^N \overset{\mathrm{iid}}{\sim} p(\boldsymbol{z}). \tag{3.2}$$

In this work, we use the hat notation to denote estimators. The mean square error (MSE) of Monte Carlo estimation $\mathbb{E}_p[(\widehat{P}_N^{\mathrm{MC}} - P_f)^2]$ is $\mathbb{V}_p[\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}]/N$. In applications like failure probability estimation, when the failure probability is small, the aforementioned variance becomes large, which requires more HF evaluations to reduce the MSE. Importance sampling (IS) [428] is one of the methods that effectively reduces the estimator variance by re-weighting the samples with a carefully chosen alternative density function $q(\boldsymbol{z})$, named biasing density function. By building $q(\boldsymbol{z})$ to

replace $p(\boldsymbol{z})$, the IS estimator is then defined as

$$\widehat{P}_N^{\mathrm{IS}} \coloneqq \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{h^{\mathrm{HF}}(\tilde{\boldsymbol{z}}_i)<0} \frac{p(\tilde{\boldsymbol{z}}_i)}{q(\tilde{\boldsymbol{z}}_i)}, \quad \{\tilde{\boldsymbol{z}}_i\}_{i=1}^{N} \overset{\mathrm{iid}}{\sim} q(\boldsymbol{z}). \tag{3.3}$$

Note that the IS estimator in Equation (3.3) is unbiased, i.e., $\mathbb{E}_q[\widehat{P}_N^{\mathrm{IS}}] = P_f$.

3.3.2 Biasing Distribution and L-BF-IS Estimator

It is known that the optimal biasing density for failure probability estimation is (see [428])

$$q^*(\boldsymbol{z}) \coloneqq \frac{1}{P_f} \mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0} p(\boldsymbol{z}). \tag{3.4}$$

However, we cannot simply use the LF indicator function $\mathbb{1}_{h^{\mathrm{LF}}(\cdot)<0}$ to replace its counterpart $\mathbb{1}_{h^{\mathrm{HF}}(\cdot)<0}$ due to singularity issue on the IS weight $p(\boldsymbol{z})/q(\boldsymbol{z})$. Instead, we aim to design a "soft version" for the conceptually optimal biasing density while providing it with flexibility to adjust for unmatching support between $\mathbb{1}_{h^{\mathrm{HF}}(\cdot)<0}$ and $\mathbb{1}_{h^{\mathrm{LF}}(\cdot)<0}$.

Similar to the smoothing strategy in [414, 522], we propose the biasing distribution

$$q(\boldsymbol{z}) \coloneqq \frac{1}{\mathcal{Z}(\ell)} \exp\left(-\ell \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z})\right) p(\boldsymbol{z}), \tag{3.5}$$

where ℓ is a length scale and $\mathcal{Z}(\ell)$, a function of ℓ, is the normalisation constant. The value of $\mathcal{Z}(\ell)$ is given by

$$\mathcal{Z}(\ell) = \int_{\Omega} \exp\left(-\ell \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z})\right) p(\boldsymbol{z}) d\boldsymbol{z} = \mathbb{E}_p\left[\exp\left(-\ell \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z})\right)\right]. \tag{3.6}$$

Note that $q(\boldsymbol{z})$ is strictly positive when the input is in the support of $p(\boldsymbol{z})$, which guarantees that $p(\boldsymbol{z})$ is absolutely continuous with respect to $q(\boldsymbol{z})$ and the weight $p(\boldsymbol{z})/q(\boldsymbol{z})$ is well-defined. Based on the initial density $p(\boldsymbol{z})$, the formulation of $q(\boldsymbol{z})$ in Equation (3.5) prioritizes higher probability weights for samples $\boldsymbol{z}$ whose LF counterparts indicate a failure outcome. This approach is based on an assumed connection between the HF function h^{HF} and its LF counterpart h^{LF}, which will be discussed in more details in Section 3.3.6. Figure 3.1 illustrates this concept with a two-dimensional ($D = 2$) example, demonstrating the application of our proposed method. Similar to the strategy employed by [310], the tanh function facilitates a "buffer" region within the importance sampling

framework. However, unlike the method above, our approach does not aim to directly approximate limit state functions due to its complexities in high-dimensional space.

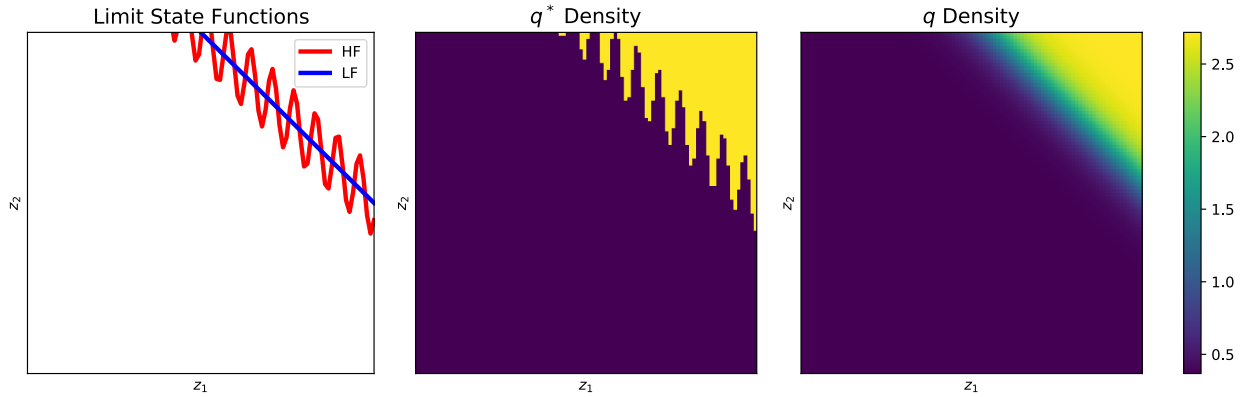


Figure 3.1: Illustration of the concept of limit state functions and biasing densities in the inputs $\boldsymbol{z}$. The left figure displays the limit state functions that separate the failure region from the safe region, highlighting the HF limit function in red and the LF surrogate in blue. The middle figure shows the optimal biasing density as derived from Equation (3.4). The right figure displays the proposed biasing density, as defined in Equation (3.5), which utilizing the LF function.

Given $q(\boldsymbol{z})$ in Equation (3.5) and $p(\boldsymbol{z})$, the importance sampling weight function is

$$\frac{p(\boldsymbol{z})}{q(\boldsymbol{z})} = \mathcal{Z}(\ell) \exp\left(\ell \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z})\right). \tag{3.7}$$

We approximate $\mathcal{Z}(\ell)$ using Monte Carlo estimation

$$\widehat{\mathcal{Z}}_M(\ell) = \frac{1}{M} \sum_{m=1}^{M} \exp\left(-\ell \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z}_m)\right), \tag{3.8}$$

where $\{\boldsymbol{z}_m\}_{m=1}^M \overset{\text{iid}}{\sim} p(\boldsymbol{z})$. Since the estimation of $\mathcal{Z}(\ell)$ only involves evaluating the inexpensive LF function, M can be sufficiently large so that $\mathcal{Z}(\ell)$ can be estimated with high accuracy.

According to the definition of importance sampling estimator in Equation (3.3), and given our $q(\boldsymbol{z})$, we define the L-BF-IS estimator as follows

$$\widehat{P}_{M,N}^{\mathrm{BF}} = \left(\frac{1}{M} \sum_{m=1}^{M} \exp\left(-\ell \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z}_m)\right)\right) \left(\frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{h^{\mathrm{HF}}(\tilde{\boldsymbol{z}}_i)<0} \exp\left(\ell \tanh \circ h^{\mathrm{LF}}(\tilde{\boldsymbol{z}}_i)\right)\right), \tag{3.9}$$

where $\{\boldsymbol{z}_m\}_{m=1}^M \overset{\text{iid}}{\sim} p(\boldsymbol{z})$ and $\{\tilde{\boldsymbol{z}}_i\}_{i=1}^N \overset{\text{iid}}{\sim} q(\boldsymbol{z})$.

3.3.3 Statistical Properties of L-BF-IS Estimator

Analyzing biased, variance, and consistency, is crucial for evaluating the performance of an estimator. Firstly, due to the independence between samples from $p(\boldsymbol{z})$ and $q(\boldsymbol{z})$, the L-BF-IS estimator in Equation (3.9) is unbiased, i.e.,

$$\begin{aligned}\mathbb{E}_{p\otimes q}\left[\widehat{P}^{\mathrm{BF}}_{M,N}\right] &= \mathbb{E}_p\left[\exp\left(-\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)\right]\mathbb{E}_q\left[\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}\exp\left(\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)\right]\\ &= \mathcal{Z}(\ell)\mathbb{E}_q\left[\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}\exp\left(\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)\right] = P_f.\end{aligned} \tag{3.10}$$

Here, $p\otimes q$ represents the Cartesian product of the two densities, indicating their independence and the last equality is from the unbiasedness of the important sampling estimator. Secondly, following the relation

$$\mathbb{V}[XY] = \mathbb{V}[X]\mathbb{V}[Y] + \mathbb{E}^2[X]\mathbb{V}[Y] + \mathbb{V}[X]\mathbb{E}^2[Y], \tag{3.11}$$

for two independent variables X and Y the variance of L-BF-IS estimator is given by

$$\begin{aligned}\mathbb{V}_{p\otimes q}[\widehat{P}^{\mathrm{BF}}_{M,N}] = &\frac{1}{MN}\mathbb{V}_p\left[\exp\left(-\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)\right]\mathbb{V}_q\left[\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}\exp\left(\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)\right]\\ &+\frac{\mathcal{Z}^2(\ell)}{N}\mathbb{V}_q\left[\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}\exp\left(\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)\right]\\ &+\frac{1}{M}\mathbb{V}_p\left[\exp\left(-\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)\right](P_f)^2.\end{aligned} \tag{3.12}$$

With the results from Equation (3.10) and Equation (3.12), the consistency of the L-BF-IS can be shown by applying the Chebyshev's inequality,

$$\mathbb{P}_{p\otimes q}\left(|\widehat{P}^{\mathrm{BF}}_{M,N} - P_f| \geq \epsilon\right) \leq \frac{\mathbb{V}_{p\otimes q}\left[\widehat{P}^{\mathrm{BF}}_{M,N}\right]}{\epsilon^2}, \quad \forall\epsilon > 0. \tag{3.13}$$

Notice that the variance $\mathbb{V}_{p\otimes q}\left[\widehat{P}^{\mathrm{BF}}_{M,N}\right]$ decays when both M and N increase. Additionally, if we assume the value of M is sufficiently large so that $1/M$ is small enough to be ignored, the variance in Equation (3.12) can be approximated as

$$\mathbb{V}_{p\otimes q}[\widehat{P}^{\mathrm{BF}}_{M,N}] \approx \frac{\mathcal{Z}^2(\ell)}{N}\mathbb{V}_q\left[\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}\exp\left(\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)\right]. \tag{3.14}$$

3.3.4 Selection of Lengthscale ℓ

The value of the parameter ℓ in Equation (3.5) plays a key role in determining the performance of the L-BF-IS estimator. Given that the estimator is unbiased as shown in Equation (3.10)

and the values of M and N are held fixed, the goal is to find an optimal value of ℓ so that the variance of the L-BF-IS estimator is minimized. Leveraging the variance approximation presented in Equation (3.14) and acknowledging the dependency of $q(\boldsymbol{z})$ on ℓ, we re-formulate the approximated variance as

$$\mathbb{V}_{p\otimes q}\left[\widehat{P}^{\mathrm{BF}}_{M,N}\right] \approx \frac{\mathcal{Z}(\ell)}{N}\mathbb{E}_p\left[\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}\exp\left(\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)\right] - \frac{(P_f)^2}{N}. \tag{3.15}$$

For the interest of brevity, more detils on the derivation of Equation (3.15) are presented in C.1. Focusing solely on the relationship between the variance in Equation (3.15) and ℓ,

$$\mathbb{V}_{p\otimes q}\left[\widehat{P}^{\mathrm{BF}}_{M,N}\right] \approx \underbrace{\frac{\mathcal{Z}(\ell)}{N}}_{\ell\downarrow}\underbrace{\mathbb{E}_p\left[\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}\exp\left(\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)\right]}_{\ell\uparrow} + \mathcal{O}(1). \tag{3.16}$$

Upon examining Equation (3.16) closely, it is clear that the value of $\mathcal{Z}(\ell)$, as defined in Equation (3.6), decreases monotonically with ℓ while the expectation component exhibits a monotonic increase with the value of ℓ. This dichotomy highlights a trade-off between larger and smaller ℓ values, underscoring the importance of designing an algorithm to optimally determine ℓ.

Since estimating the expectation term requires evaluating the HF function h^{HF}, two practical approaches are next introduced to choose an optimal value for ℓ.

3.3.4.1 Approach One: Using Pilot HF Evaluations

In the first approach, one consider a small sample approximation of the variance in (3.16)

$$\widehat{V}_L(\ell) = \frac{\widehat{\mathcal{Z}}_M(\ell)}{NL}\sum_{j=1}^{L}\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z}_j)<0}\exp\left(\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z}_j)\right), \quad \{\boldsymbol{z}_j\}_{j=1}^{L}\sim p(\boldsymbol{z}) \tag{3.17}$$

using $L \ll M$ HF function $h^{\mathrm{HF}}(\cdot)$ evaluations. We then choose the optimal ℓ^* such that

$$\ell^* = \arg\min_{\ell}\widehat{V}_L(\ell), \tag{3.18}$$

which, as a 1D optimization problem, can be solved using a simple grid search or a first/second order method. However, when the failure probability is small, e.g. $P_f \leq \mathcal{O}(1/L)$, a risk of this approach is that $\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z}_j)<0}$ can be 0 for all $\boldsymbol{z}_j$, thus making it invalid. Indeed, since the HF function is evaluated only L times, the probability that no failure case is sampled is $(1-P_f)^L$ and can be non-trivial.

3.3.4.2 Approach Two: Only Using LF Evaluations

An alternative approach is to replace $\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z}_j)<0}$ with $\mathbb{1}_{h^{\mathrm{LF}}(\boldsymbol{z}_j)<0}$, which produces the variance estimator

$$\widehat{V}'_M(\ell) = \frac{\widehat{\mathcal{Z}}_M(\ell)}{NM} \sum_{m=1}^{M} \mathbb{1}_{h^{\mathrm{LF}}(\boldsymbol{z}_m)<0} \exp\left(\ell \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z}_j)\right), \tag{3.19}$$

with samples $\{\boldsymbol{z}_m\}_{m=1}^{M} \overset{\text{iid}}{\sim} p(\boldsymbol{z})$. We choose the optimal ℓ^* as

$$\ell^* = \arg\min_{\ell} \widehat{V}'_M(\ell). \tag{3.20}$$

This approach provides a less accurate estimation for the variance in exchange for avoiding directly evaluating the HF function. We suggest applying this approach when the value of $(1 - P_f)^L$ is large, where P_f can be replaced by some prior knowledge of the failure probability and, $L \ll M$, is an affordable number of HF function evaluations.

3.3.5 Sampling the Biasing Distributions

The formulation of the biasing density $q(\boldsymbol{z})$ in Equation (3.5), as well as the availability of the LF function derivative $\nabla_{\boldsymbol{z}} h^{\mathrm{LF}}(\boldsymbol{z})$ facilitates the evaluation of the score function $\nabla_{\boldsymbol{z}} \log q(\boldsymbol{z})$. This capability significantly enhances the selection of sampling methods that utilize the score function, which includes, but are not limited to, Langevin Monte Carlo, Hamiltonian Monte Carlo, and Stein Variational Gradient Descent [325].

Among the various options, we opt for the Metropolis-adjusted Langevin algorithm (MALA), a variant Langevin Monte Carlo, to generate samples from the biasing distribution. This choice is made because of its simplicity and widely-used implementation. However, it is important to note that any score-based sampling method is compatible with the importance sampling framework proposed in this work. MALA effectively integrates the discretization of Langevin dynamics with the Metropolis-Hastings algorithm [455], offering a robust framework for sampling.

Assuming the score function $\nabla_{\boldsymbol{z}} \log p(\boldsymbol{z})$ exists and is bounded, and the LF function h^{LF} is

differentiable and Lipschitz, the biasing density $q(\boldsymbol{z})$ can be written as

$$q(\boldsymbol{z}) = \frac{1}{\mathcal{Z}(\ell)} \exp(-U(\boldsymbol{z})), \tag{3.21}$$

where the potential function $U(\boldsymbol{z})$ is given by

$$U(\boldsymbol{z}) \coloneqq \ell \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z}) - \log p(\boldsymbol{z}). \tag{3.22}$$

According to [455], the density $q(\boldsymbol{z})$ is the unique invariant distribution of the Langevin stochastic differential equation (SDE)

$$d\boldsymbol{z} = -\nabla U(\boldsymbol{z}) + \sqrt{2} d\boldsymbol{W}_t, \tag{3.23}$$

where $\boldsymbol{W}_t$ is the Brownian motion. Therefore, by simulating the SDE in Equation (3.23) via Euler-Maruyama method,

$$\boldsymbol{z}^{(t+\tau)} = \boldsymbol{z}^{(t)} - \tau \nabla U(\boldsymbol{z}^{(t)}) + \sqrt{2}(\boldsymbol{W}_{t+\tau} - \boldsymbol{W}_t), \tag{3.24}$$

where $\boldsymbol{z}^{(t)}$ represents the discretized $\boldsymbol{z}$ and τ is the step size. The property of Brownian motion, $\boldsymbol{W}_{t+\tau} - \boldsymbol{W}_t \sim \mathcal{N}(\boldsymbol{0}, \tau \boldsymbol{I}_D)$ with the identity matrix $\boldsymbol{I}_D \in \mathbb{R}^{D \times D}$, allows to re-write Equation (3.24) as

$$\boldsymbol{z}^{(t+1)} = \boldsymbol{z}^{(t)} - \tau \nabla U(\boldsymbol{z}^{(t)}) + \sqrt{2\tau}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_D). \tag{3.25}$$

Following Equations (3.21) and (3.22), $\nabla_{\boldsymbol{z}} U(\boldsymbol{z})$ is given by

$$\nabla_{\boldsymbol{z}} U(\boldsymbol{z}) = -\nabla_{\boldsymbol{z}} \log q(\boldsymbol{z}) = \ell \nabla_{\boldsymbol{z}} \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z}) - \nabla_{\boldsymbol{z}} \log p(\boldsymbol{z}). \tag{3.26}$$

Besides sampling $\boldsymbol{z}^{(t)}$ iteratively, MALA implements a Metropolis-Hastings accept-reject mechanism to reject proposals in low-density regions [456]. The rejection of the new proposed sample $\boldsymbol{z}^{(t+1)}$ is triggered if

$$u \geq \exp\left(U(\boldsymbol{z}^{(t)}) + \pi(\boldsymbol{z}^{(t)}, \boldsymbol{z}^{(t+1)}) - U(\boldsymbol{z}^{(t+1)}) - \pi(\boldsymbol{z}^{(t+1)}, \boldsymbol{z}^{(t)})\right), \tag{3.27}$$

where u is a random variable sampled from uniform distribution $U[0,1]$ and π is a function defined as

$$\pi(\boldsymbol{z}_1, \boldsymbol{z}_2) \coloneqq -\frac{1}{4\tau}\|\boldsymbol{z}_1 - \boldsymbol{z}_2 - \tau\nabla U(\boldsymbol{z}_2)\|_2^2. \tag{3.28}$$

Numerically, we discard the first B samples of the Markov chain, referred to as burn-in samples, with B varying depending on the problem scale. The sampling algorithm is detailed in Algorithm 4. Notice that, once ℓ is set, Algorithm 4 requires only $\mathcal{O}(T+B)$ LF evaluations. The construction of the L-BF-IS estimator is concluded in Algorithm 5.

Algorithm 4: Langevin Algorithm for Sampling from Biasing Distribution
$\mathcal{O}(T+B)$ LF evaluations

Input: Length scale ℓ, burn-in number B, LF function h^{LF}, its gradient ∇h^{LF}, step size τ, iteration number T, and initial state $\boldsymbol{z}^{(0)}$ (Optional)
Output: A collection of samples $\{\tilde{\boldsymbol{z}}_i\}_{i=1}^N \sim q(\boldsymbol{z})$

1: Sample initial state $\boldsymbol{z}^{(0)} \overset{\text{iid}}{\sim} p(\boldsymbol{x})$ if $\boldsymbol{z}^{(0)}$ is not given
2: **for** $t = 1 : T+B$ **do**
3: update $\boldsymbol{z}^{(t)}$ following Equation (3.25) and Equation (3.26)
4: reject the step if Equation (3.27) satisfied.
5: **end for**
6: $\{\tilde{\boldsymbol{z}}_t\}_{t=1}^T \leftarrow \{\boldsymbol{z}^{(t)}\}_{t=B+1}^{T+B}$

Remark 3.3.1. When implementing Algorithm 4 on a bounded domain Ω, we introduce a penalty value $q(\boldsymbol{z}) \gg 0$ for all $\boldsymbol{z} \notin \Omega$ to discourage the chain from moving outside the domain.

3.3.6 Further Discussion on Bi-fidelity Modeling

A crucial aspect of any bi-fidelity modeling is understanding how the similarity between LF and HF models affects the performance of the proposed bi-fidelity algorithm, while we investigate from two perspectives: the variance of the L-BF-IS estimator and the Kullback-Leibler (KL) divergence between the optimal and the proposed biasing distributions.

Recall that in Section 3.3.1 we define subsets $\mathcal{A}_H \subset \Omega$ and $\mathcal{A}_L \subset \Omega$ such that $\boldsymbol{z} \in \mathcal{A}_H$ if and only if $h^{\mathrm{HF}}(\boldsymbol{z}) < 0$, and $\boldsymbol{z} \in \mathcal{A}_L$ if and only if $h^{\mathrm{LF}}(\boldsymbol{z}) < 0$. Note that under this definition,

Algorithm 5: L-BF-IS Method
$\mathcal{O}(M+T+B)$ LF evaluations $\mathcal{O}(N+L)$ HF evaluations

Input: LF sample size M, HF sample size N, LF function h^{LF}, HF function h^{HF}, and additional HF sample size L (optional)
Output: A value of L-BF-IS estimator $\widehat{P}^{\mathrm{BF}}_{M,N}$

1: Determine Langevin dynamics step size τ, burn-in number B, and iteration number T based on available computational resource ($T > N$)
2: **if** L is provided **then**
3: Determine ℓ that minimizes the variance estimator in Equation (3.17);
4: **else**
5: Determine ℓ that minimizes the variance estimator in Equation (3.19);
6: **end if**
7: Build estimator $\widehat{\mathcal{Z}}_M(\ell)$ using $\{\boldsymbol{z}_m\}_{m=1}^M \overset{\text{iid}}{\sim} p(\boldsymbol{z})$ following Equation (3.8);
8: $\{\tilde{\boldsymbol{z}}_t\}_{t=1}^T \leftarrow$ Langevin algorithm$(\ell, B, h^{\mathrm{LF}}, \nabla h^{\mathrm{LF}}, \tau, T)$ in Algorithm 4;
9: Uniformly select subset $\{\tilde{\boldsymbol{z}}_i\}_{i=1}^N \subseteq \{\tilde{\boldsymbol{z}}_t\}_{t=1}^T$
10: Evaluate $\widehat{P}^{\mathrm{BF}}_{M,N}$ as in Equation (3.9) using $\{\tilde{\boldsymbol{z}}_i\}_{i=1}^N$ and $\widehat{\mathcal{Z}}_M(\ell)$.

$P_f = \mathbb{P}_p[\mathcal{A}_H]$. The analysis in this section assumes ℓ is already fixed. Based on the approximated variance in Equation (3.15), we decompose the expectation term into two parts:

$$\begin{aligned}\mathbb{E}_p\left[\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}\exp\left(\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)\right] &= \int_{\mathcal{A}_H}\exp\left(\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)p(\boldsymbol{z})d\boldsymbol{z}\\ &= \int_{\mathcal{A}_H\cap\mathcal{A}_L}\exp\left(\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)p(\boldsymbol{z})d\boldsymbol{z} + \int_{\mathcal{A}_H\cap\mathcal{A}_L^C}\exp\left(\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)p(\boldsymbol{z})d\boldsymbol{z},\end{aligned} \tag{3.29}$$

where $\mathcal{A}_L^C \coloneqq \Omega \setminus \mathcal{A}_L$ is the complement. For the first term in Equation (3.29), since $\boldsymbol{z} \in \mathcal{A}_L$, we have $h^{\mathrm{LF}}(\boldsymbol{z}) < 0$ and thus $\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z}) < 0$, making this term upper bounded by $\mathbb{P}_p[\mathcal{A}_H \cap \mathcal{A}_L]$, which is equivalent to $P_f - \mathbb{P}_p[\mathcal{A}_H \cap \mathcal{A}_L^C]$. The second term, since $\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z}) < 1$ for all $\boldsymbol{z}$, is thereby bounded above by $e^\ell \mathbb{P}_p[\mathcal{A}_H \cap \mathcal{A}_L^C]$.

Applying a similar methodology, we also bound $\mathcal{Z}(\ell) < 1 + (e^\ell - 1)\mathbb{P}_p[\mathcal{A}_L]$; see C.2.1 for detailed proofs. Thus, assuming M is sufficiently large, the variance of the L-BF-IS estimator in Equation (3.14) is upper bounded as:

$$\mathbb{V}_{p\otimes q}[\widehat{P}^{\mathrm{BF}}_{M,N}] \lesssim \frac{1+(e^\ell-1)\mathbb{P}_p[\mathcal{A}_L]}{N}(P_f + (e^\ell-1)\mathbb{P}_p[\mathcal{A}_H\cap\mathcal{A}_L^C]) - \frac{(P_f)^2}{N}. \tag{3.30}$$

The terms $\mathbb{P}_p[\mathcal{A}_L]$ and $e^\ell\mathbb{P}_p[\mathcal{A}_H \cap \mathcal{A}_L^C]$ represent penalties arising from mismatches between the HF and LF models. Should the LF model perfectly align with the HF model, these terms vanish; see

Figure 3.2. This bound elucidates that the performance of L-BF-IS is contingent on $\mathbb{P}_p[\mathcal{A}_H \cap \mathcal{A}_L^C]$, and further analysis of the KL divergence will verify this observation.

In addition to the variance analysis, we examine the KL divergence between the proposed biasing distribution in Equation (3.5) and the optimal distribution in Equation (3.4), given by

$$\mathrm{KL}(q^*\|q) = \mathbb{E}_{q^*}\left[\log \frac{\mathcal{Z}(\ell)\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}}{P_f \exp\left(-\ell \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z})\right)}\right], \tag{3.31}$$

or its simplification

$$\mathrm{KL}(q^*\|q) = \log \frac{\mathcal{Z}(\ell)}{P_f} + \ell \int_{\mathcal{A}_H \cap \mathcal{A}_L} \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z} + \ell \int_{\mathcal{A}_H \cap \mathcal{A}_L^C} \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z}. \tag{3.32}$$

Here, the integrals represent contributions from the regions where high-fidelity and low-fidelity models coincide and where they do not, respectively. Consequently, the KL divergence can be bounded by

$$\mathrm{KL}(q^*\|q) < \log \frac{1 + (e^{\ell} - 1)\mathbb{P}_p[\mathcal{A}_L]}{P_f} + \ell \mathbb{P}_p[\mathcal{A}_H \cap \mathcal{A}_L^C]. \tag{3.33}$$

The expression in Equation (3.33) indicates that the optimality of the proposed biasing distribution depends significantly on $\mathbb{P}_p[\mathcal{A}_H \cap \mathcal{A}_L^C]$. The proofs supporting these claims are provided in C.3. Note that since the optimal biasing distribution q^* is fixed, the KL divergence is equivalent to the cross entropy criteria presented in [293, 184].

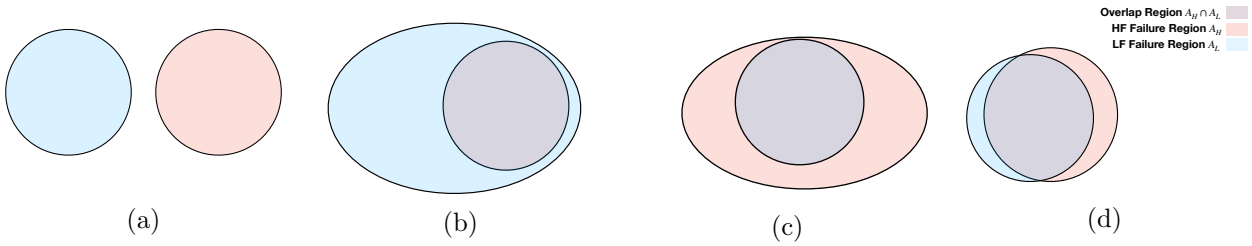


Figure 3.2: Illustration of the trade-off between $\mathbb{P}_p[\mathcal{A}_L]$ and $\mathbb{P}_p[\mathcal{A}_H \cap \mathcal{A}_L^C]$ when $D = 2$. Case 1 (a) represents the worst scenario, where there is no overlap between $\mathcal{A}_H$ and $\mathcal{A}_L$. In Case 2 (b), we observe an extreme case where $\mathbb{P}_p[\mathcal{A}_H \cap \mathcal{A}_L^C]$ is zero, but $\mathbb{P}_p[\mathcal{A}_L]$ becomes excessively large. Case 3 (c) presents a scenario where $\mathbb{P}_p[\mathcal{A}_L]$ is small, but $\mathbb{P}_p[\mathcal{A}_H \cap \mathcal{A}_L^C]$ is significantly large. Lastly, Case 4 (d) shows a favorable scenario resulting in a small values for both $\mathbb{P}_p[\mathcal{A}_L]$ and $\mathbb{P}_p[\mathcal{A}_H \cap \mathcal{A}_L^C]$.

While $\mathbb{P}_p[\mathcal{A}_H \cap \mathcal{A}_L^C]$ is a key in describing the alignment between the HF and LF models, computing it requires many HF model evaluations, which is inpractical. One possible way to

address this problem is to use a small number of pilot HF samples to evaluate the KL divergence in Equaiton (3.31). A systematic framework of the alignment between the LF and HF models is out of the scope of this work but can be the focus of the future works.

3.3.7 Error Analysis

Two principal types of errors are identified as contributing to an increase in the MSE: bias-inducing error and variance-inducing error. This section delves into both those error types.

The bias-inducing error arises from inaccuracies in MALA, as outlined in Section 3.3.5. A series of studies have investigated the convergence behavior of Langevin Monte Carlo, especially under the convexity assumption of the potential function $U(\boldsymbol{z})$ in Equation (3.22). These studies have shown that Langevin algorithm's output tends to converge to the target distribution $q(\boldsymbol{z})$ across several metrics, including total variation [120, 163], Wasserstein-2 distance [162], and KL divergence [98]. However, the convexity of $U(\boldsymbol{z})$ may not always hold, particularly for target densities $q(\boldsymbol{z})$ with multimodal features. The inaccurate sampling of $q(\boldsymbol{z})$ lead to biases in L-BF-IS estimations. A mitigation strategy involves launching multiple Langevin dynamics chains from different initial states.

The variance-inducing error originates from two sources. The first is the discrepancy between LF and HF functions. According to the analysis in Section 3.3.6, this discrepancy is quantified by the probabilities $\mathbb{P}_p[\mathcal{A}_H \cap \mathcal{A}_L^C]$ and $\mathbb{P}_p[\mathcal{A}_L]$. Lower values of these probabilities suggest a smaller estimation variance, hence smaller MSE. The second source of variance-inducing error relates to the selection of the parameter ℓ, as described in Equations (3.17) and (3.19). Given the limited access to HF function evaluations in one approach (Section 3.3.4.1) or the complete avoidance of HF samples for selecting ℓ in another approach (Section 3.3.4.2), a deviation between the chosen ℓ^* and the true optimal ℓ that minimized Equation (3.15) inevitably arises. This deviation contributes to an increase in the variance of L-BF-IS estimator and, consequently, its MSE. We acknowledge that fully addressing these challenges, particularly in mitigating bias-inducing and variance-inducing errors, remains an open problem that forms the basis of a future work.

3.4 Empirical Results

In this section, empirical results are presented to illustrate the effectiveness of the L-BF-IS estimator. In Section 3.4.1, a simple 1D function demonstrates the applicability of the MALA on sampling a multi-modal biasing distribution. Then, in Section 3.4.2, the L-BF-IS is applied to two different cases: an 8-dimensional Borehole function (detailed in 3.4.2.1) and a 1000-dimensional synthetic function (detailed in 3.4.2.2). The application of the L-BF-IS is shown on two real-world failure probability estimation problems in Section 3.4.3, including a composite beam problem (explained in Section 3.4.3.1) that uses the Euler-Bernoulli equation as an LF model and a steady-state stochastic heat equation (explained in Section 3.4.3.2) with a data-driven LF model based on a pre-trained physics-informed neural operator.

To evaluate the accuracy of our estimations, we use the relative root mean square error (rRMSE),

$$\mathrm{rRMSE}(N) \coloneqq \sqrt{\mathbb{E}\left[\frac{(\widehat{P}_N - P_f)^2}{P_f^2}\right]}, \tag{3.34}$$

where $\widehat{P}_N$ is the estimator using N iid HF samples. The performance of the L-BF-IS estimator $\widehat{P}_N^{\mathrm{BF}}$ (formulated in Equation (3.9)) is compared with the standard Monte Carlo estimator $\widehat{P}_N^{\mathrm{MC}}$ (defined in Equation (3.2)) across all problems. We also produce the LF failure probability, denoted as P_f^{LF}, which is solely generated from 1×10^6 h^{LF} evaluations. For the problems where the input dimension $D \leq 10$, we also consider the results from the Multi-fidelity Importance Sampling (MF-IS) estimator [424], which uses a biasing distribution created by a Gaussian mixture model. The number of clusters for MF-IS is chosen from $k = \{1, 3, 5, 10\}$, so that the chosen k yields the best performance. We assume the computational costs of HF models are substantially higher than those of the LF models so that the costs of LF forward and derivative evaluations can be ignored. The initial point $\boldsymbol{z}^{(0)}$ of the MALA is typically chosen as the center of the input space. The proposed method requires $\mathcal{O}(M+T+B)$ forward LF model evaluations (typically around $\sim 1 \times 10^6$ evaluations) and $\mathcal{O}(N+L)$ forward HF model evaluations, usually between 1 and $\sim 1 \times 10^4$.

The experimental component of this study is primarily concerned with scenarios exhibiting a failure probability between 1% and 5%. For the purpose of identifying an appropriate LF failure threshold, 1000 LF QoIs are generated to establish a tentative threshold, ensuring its inducing failure probability is also between 1% and 5% and potentially closed to P_f. This procedure is adopted because, for certain LF/HF models (such as the 1000-dimensional problem discussed in Section 3.4.2.2), there is a notable discrepancy between the ranges of LF and HF QoIs. Consequently, applying the same threshold to both models may result in inaccurate probability estimates. In practice, while the HF failure probability P_f is the goal of estimation, a prior knowledge on a range of values is available. Such an estimate is instrumental in establishing a valid criterion for the assessment of LF QoIs within L-BF-IS.

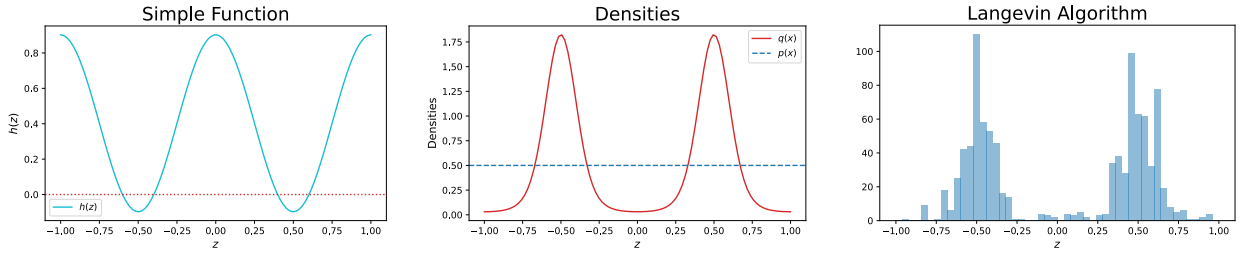
3.4.1 A Simple Bimodal Function for Demonstrating Langevin Algorithm

In failure probability estimation, the multimodal issue occurs when multiple sub-areas in Ω correspond to failure. The goal of this example is to empirically show that the MALA is capable to address this issue through a 1D example, where

$$h(z) = -(\sin(\pi z) + 0.95)(\sin(\pi z) - 0.95). \tag{3.35}$$

The density $p(z)$ is assumed to be uniform between -1 and 1. We choose $\ell = 5.0$. The function $h(z)$ in Figure 3.3a and the densities $p(z), q(z)$ in Figure 3.3b are provided. The biasing density $q^*(z)$ shows the bimodal property and we will show that the Langevin algorithm is possible to generate samples from it.

We implement the Langevin algorithm described in Algorithm 4, where we set the starting point at $z = 0$. The step size $\tau = 0.05$ and burn-in number $B = 200$. We initiate the Langevin algorithm 100 times and for each chain, we collect 10 samples after the burn-in number. The collected samples are shown in Figure 3.3c. As we can see, the bimodal shape of the biasing density $q(z)$ is captured by the Langevin algorithm.



(a) (b) (c)

Figure 3.3: (a) The example function $h(z)$ and the 0 threshold. (b) Densities $p(z)$ and $q(z)$ with $\ell = 5.0$. (c) Histogram of $1,000$ samples of $q(z)$ generated from the Langevin algorithm described in Algorithm 4.

3.4.2 Synthetic Examples with Prescribed Functions

3.4.2.1 Borehole Function

We applied the borehole function described in [374], which is extended to a multi-fidelity setting in [563]. It is an 8-dimensional problem that models water flow through a borehole. Following [563], the HF QoI function is

$$f^{\mathrm{HF}}(\boldsymbol{z}) = \frac{2\pi z_3(z_4 - z_5)}{(z_2 - \log z_1)\left(1 + \frac{2z_7 z_3}{(z_2 - \log z_1)z_1^2 z_8} + \frac{z_3}{z_5}\right)}, \tag{3.36}$$

and the LF QoI function is

$$f^{\mathrm{LF}}(\boldsymbol{z}) = \frac{5z_3(z_4 - z_5)}{(z_2 - \log z_1)\left(1.5 + \frac{2z_7 z_3}{(z_2 - \log z_1)z_1^2 z_8} + \frac{z_3}{z_5}\right)}. \tag{3.37}$$

The random inputs $\boldsymbol{z}$ and their distributions are presented in Table 3.1. We define the HF function $h^{\mathrm{HF}}(\boldsymbol{z})$ as $800 - f^{\mathrm{HF}}(\boldsymbol{z})$. To empirically prevent the Langevin Markov chain from moving outside the domain Ω, we introduce an additional penalty term of $100\|\boldsymbol{z}\|^2$ when $\boldsymbol{z}$ is outside Ω. Similarly, the LF function $h^{\mathrm{LF}}(\boldsymbol{z})$ is defined as $1000 - f^{\mathrm{LF}}(\boldsymbol{z})$ within the specified domain; otherwise, it takes the penalty term $100\|\boldsymbol{z}\|^2$.

In Figure 3.4, the estimated variance of L-BF-IS estimator using two different approaches for tuning ℓ are demonstrated with $L = 1 \times 10^2$ HF trials and $M = 1 \times 10^6$ LF evaluations for length scale selection. The uncertainty of the variance estimate is notably higher in the first approach

Table 3.1: The stochastic input ranges, distributions, and physical meanings of the Borehole function.

Range	Distribution	Physical Meaning
$z_1 \in [0.05, 0.15]$	$\mathcal{N}(0.10, 0.016)$	radius of borehole (m)
$z_2 \in [4.605, 10.820]$	$\mathcal{N}(7.71, 1.0056)$	radius of influence (m)
$z_3 \in [63070, 115600]$	$U[63070, 115600]$	transmissivity of upper aquifer (m^2/yr)
$z_4 \in [990, 1110]$	$U[990, 1110]$	potentiometric head of upper aquifer (m)
$z_5 \in [63.1, 116]$	$U[63.1, 116]$	transmissivity of lower aquifer (m^2/yr)
$z_6 \in [700, 820]$	$U[700, 820]$	potentiometric head of lower aquifer (m)
$z_7 \in [1120, 1680]$	$U[1120, 1680]$	length of borehole (m)
$z_8 \in [9855, 12045]$	$U[9855, 12045]$	hydraulic conductivity of borehole (m/yr)

compared to the second, primarily due to the limited number of HF function evaluations available for choosing ℓ, which significantly raises the likelihood of estimating the variance as zero.

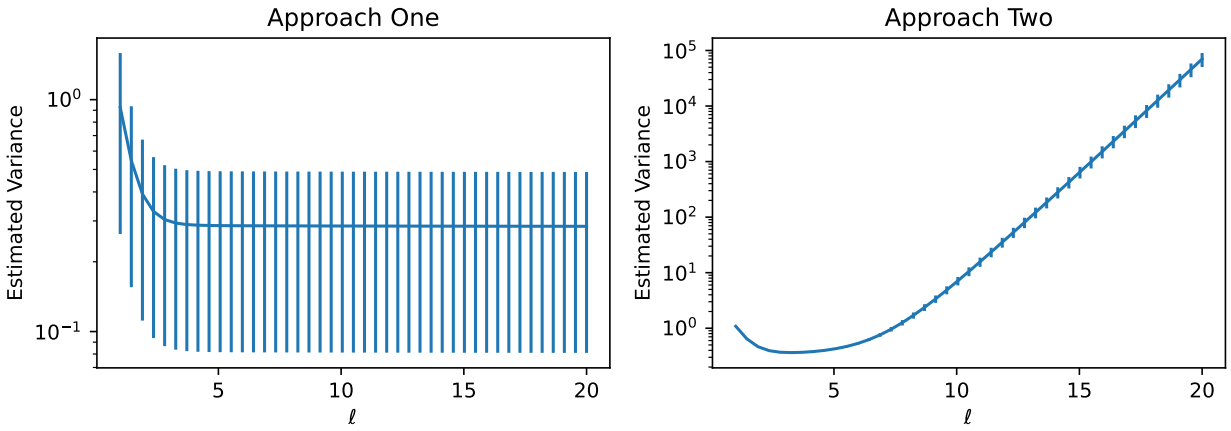


Figure 3.4: Estimated variance of L-BF-IS for different ℓ values with 95% confidence interval using $L = 1 \times 10^2$ HF evaluations (approach one) and $M = 1 \times 10^6$ LF evaluations (approach two) for the borehole function in Section 3.4.2.1. Approach one exhibits higher estimation uncertainty, whereas approach two is more robust.

To demonstrate the robustness of L-BF-IS with respect to the choice of ℓ, we compare the convergence results of L-BF-IS with standard Monte Carlo and MF-IS across three different ℓ values in Figure 3.5. The MALA step size τ is set to 1×10^{-4}, with a burn-in value of $B = 1 \times 10^3$ and an iteration number $T = 1 \times 10^4$. For the convergence analysis, the estimates are computed for HF sample size N as $10, 21, 46, 100, 215, 464, 1000, 2154, 4641$, and 10000 across 1000 trials to determine

the 95% confidence intervals. In this example, the LF model produces similar results to the HF model, with 5% relative error in estimating P_f. When ℓ is set to 3.26 (following approach two), the L-BF-IS successfully reduces the relative RMSE to 0.3%. However, when ℓ is not optimally chosen, as shown in Figure 3.5d with $\ell = 5.80$ or Figure 3.5f with $\ell = 7.34$, the improvements are limited to $5\% \sim 8\%$.

We also investigate the performance of L-BF-IS when the value of P_f is smaller and the LF model is less accurate. We choose the new LF and HF functions as $h^{\mathrm{LF}}(\boldsymbol{z}) = 1100 - f^{\mathrm{LF}}(\boldsymbol{z})$ and $h^{\mathrm{HF}}(\boldsymbol{z}) = 900 - f^{\mathrm{HF}}(\boldsymbol{z})$, respectively. With a smaller value of failure probability, the region that the biasing distribution should place more probabilities becomes smaller. With updated LF and HF functions, the value of ℓ is chosen as 3.71 using approach two, and the corresponding convergence results are presented in Figure 3.6. The relative RMSE of the LF model is 36%, which is significantly larger than the previous case. We notice that the relative RMSE of L-BF-IS maintains its quality and is 1%, which is one order of magnitude better than the standard Monte Carlo method on the HF function.

3.4.2.2 1000 Dimensional Synthetic Function

To evaluate the performance of L-BF-IS on high-dimensional problems, we examine a 1000-dimensional problem following [215]. The HF QoI function is defined as

$$f^{\mathrm{HF}}(\boldsymbol{z}) = \exp\left(2 - \sum_{k=1}^{1000} \frac{\sin(k) z_k}{k}\right) \tag{3.38}$$

and the LF QoI function is established based on the truncated Taylor series expansion of f^{HF},

$$f^{\mathrm{LF}}(\boldsymbol{z}) = \sum_{m=0}^{2} (m!)^{-1} \left(2 - \sum_{k=1}^{1000} \frac{\sin(k) z_k}{k}\right)^m. \tag{3.39}$$

The HF function $h^{\mathrm{HF}}(\boldsymbol{z})$ is set to $20 - f^{\mathrm{HF}}(\boldsymbol{z})$ within the hypercube $[-1, 1]^{1000}$ domain, and we apply a penalty of $100\|\boldsymbol{z}\|^2$ outside this domain. Similarly, the LF function $h^{\mathrm{LF}}(\boldsymbol{z})$ is defined as $8 - f^{\mathrm{LF}}(\boldsymbol{z})$ with the same penalty applied.

For tuning the value of ℓ, we employed two approaches, utilizing $L = 1 \times 10^2$ HF trial evaluations (for approach one) and $M = 1 \times 10^6$ LF evaluations (for both methods) for variance

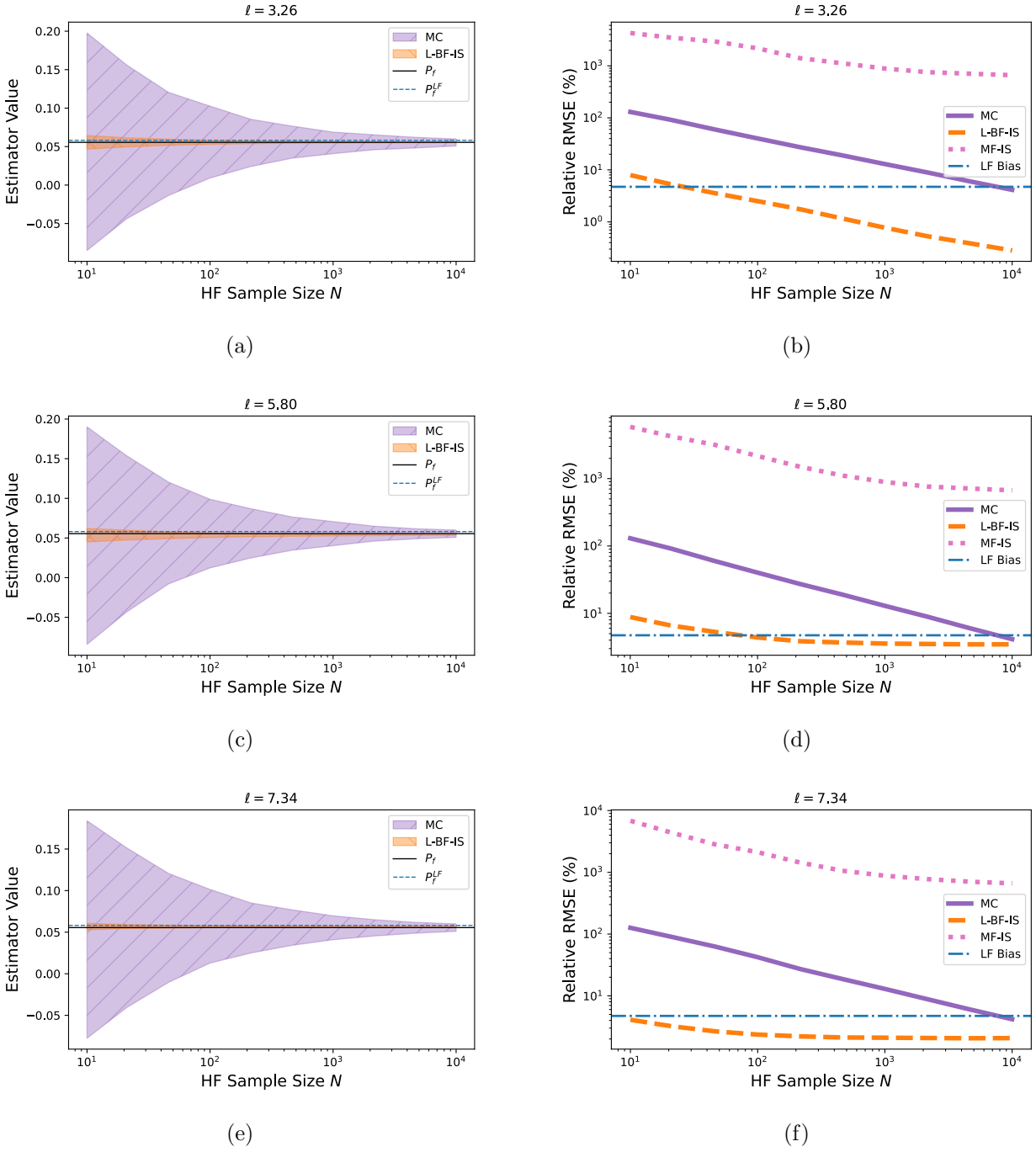
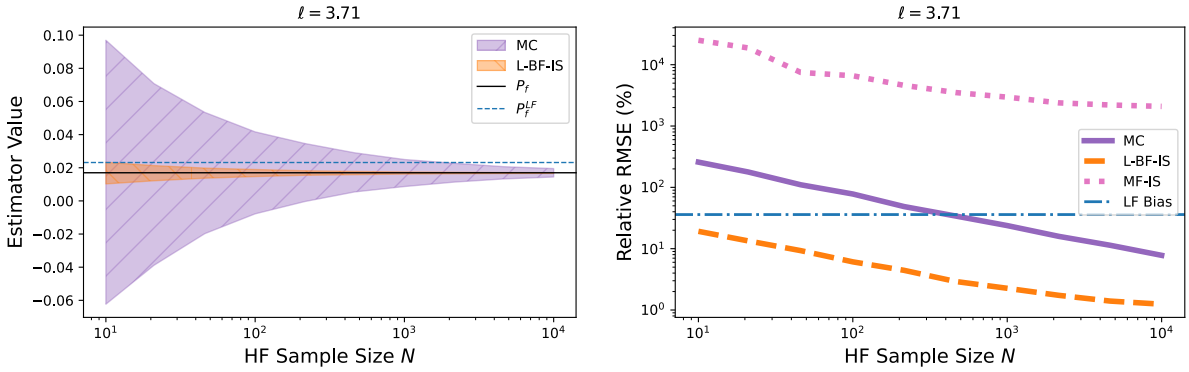


Figure 3.5: Convergence behavior of L-BF-IS (dash) for ℓ values of 3.26 (a-b), 5.80 (c-d), and 7.34 (e-f), compared with standard Monte Carlo (solid), MF-IS (dot), and LF failure probability (dash dot) using 10 Gaussian mixture clusters for the borehole function in Section 3.4.2.1. The blue dash dotted lines are LF failure probabilities. The shaded areas represent the 95% confidence interval from 1,000 trials.



(a) (b)

Figure 3.6: Convergence behavior of L-BF-IS (dash) for $\ell = 3.71$ compared with standard Monte Carlo (solid) and LF failure probability (dash dot) with updated LF and HF functions for the borehole function in Section 3.4.2.1. The shaded areas represent the 95% confidence interval from $1,000$ trials.

estimation. These calculations were repeated ten times to estimate their variability. Unlike the borehole example in Section 3.4.2.1, both approaches yielded similar variance estimates for this high-dimensional problem, though approach one exhibited larger variability, as depicted in Figure 3.7.

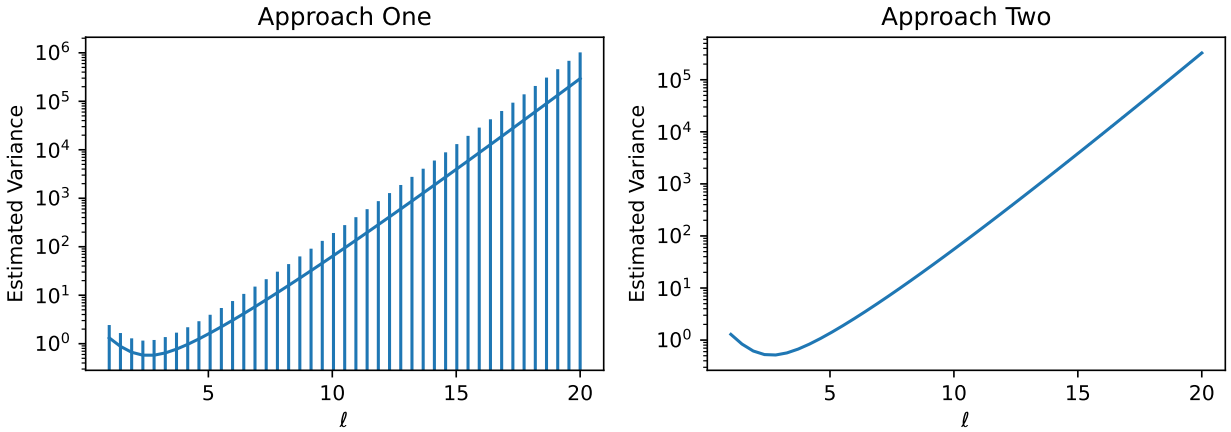
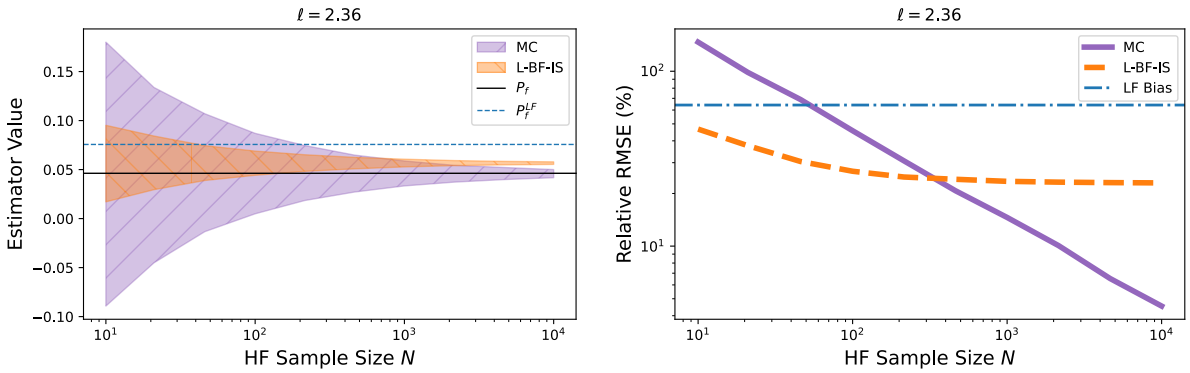


Figure 3.7: Estimated variance of L-BF-IS across different ℓ values, with uncertainty bars indicating a 95% confidence interval. Estimates are based on $L = 1 \times 10^2$ HF evaluations (approach one) and $M = 1 \times 10^6$ LF evaluations (both approaches).

Given the consistent results in Figure 3.7, we selected an ℓ value of 2.36 for this problem. Due to the high-dimensionality of this problem, we compared the convergence of L-BF-IS solely with the

Monte Carlo method. The MALA step size τ is set to 1×10^{-5}, with a burn-in number B of 1×10^4 and $T = 1 \times 10^4$ iterations. The L-BF-IS is compared with MC estimator with HF sample sizes N as $10, 21, 46, 100, 215, 464, 1000, 2154, 4641$, and 10000 across 1000 trials to calculate the 95% confidence intervals. The failure probability produced by the LF model P_f^{LF} has relative RMSE of around 64%, while the L-BF-IS is able to reduce it to around 20%. However, the convergence outcomes in Figure 3.8 reveal a bias of 2% in the L-BF-IS estimate, which we attribute to the Langevin algorithm's inaccuracies discussed in Section 3.3.7. Despite this bias, L-BF-IS still offers a significant improvement of the MSE for smaller HF sample sizes ($N \leq 300$).



(a) (b)

Figure 3.8: Convergence of L-BF-IS (dash) for selected $\ell = 2.36$ value compared with standard Monte Carlo (solid) and LF failure probability (dash dot) for the $1000D$ problem in Section 3.4.2.2.

3.4.3 Physics-based Examples

3.4.3.1 Composite Beam

Building on the work of [218, 123, 124, 96, 94], we examine a plane-stress, cantilever beam featuring a composite cross-section and hollow web, as depicted in Figure 3.9. The focus is on the maximum displacement of the top cord, with uncertain parameters z_1, z_2, z_3, z_4. Here, z_1 represents the intensity of the distributed force applied to the beam, while z_2, z_3, and z_4 denote Young's moduli of the cross-section's three components. These parameters are independent and uniformly

distributed, with the input parameter dimension being $D = 4$. The QoI of this problem is the maximum displacement at the top of the beam. Table 3.2 outlines the range of input parameters alongside other deterministic parameters.

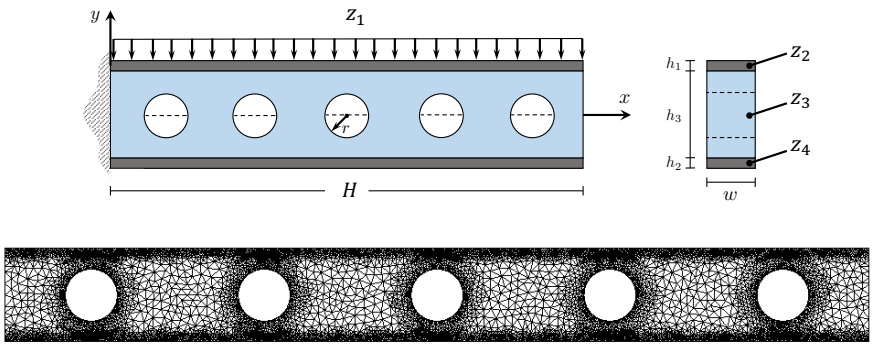


Figure 3.9: Top: Cantilever beam (left) and the composite cross section (right) adapted from [218]. Bottom: Finite element mesh used to generate high-fidelity solutions.

Table 3.2: The parameter values in the composite cantilever beam model. The center of the holes are at $x = \{5, 15, 25, 35, 45\}$. The parameters z_1, z_2, z_3 and z_4 are drawn independently and uniformly at random from the specified intervals.

H	h_1	h_2	h_3	w	r	z_1	z_2	z_3	z_4
50	0.1	0.1	5	1	1.5	$[9, 11]$	[0.9e6, 1.1e6]	[0.9e6, 1.1e6]	[0.9e4, 1.1e4]

This study employs two models to represent the HF and LF QoI functions. The HF QoI function f^{HF} is obtained from a finite element analysis using a triangular mesh, while the LF QoI function f^{LF} is evaluated based on the Euler-Bernoulli beam theory, which simplifies the model by ignoring shear deformation and circular holes. The Euler-Bernoulli theorem provides a differential equation for vertical displacement $u(x)$, which can be explicitly solved as

$$EI\frac{d^4u(x)}{dx^4} = -z_1 \quad \Longrightarrow \quad u(x) = -\frac{z_1 H^4}{24EI}\left(\left(\frac{x}{H}\right)^4 - 4\left(\frac{x}{H}\right)^3 + 6\left(\frac{x}{H}\right)^2\right), \tag{3.40}$$

where E and I represent Young's modulus and the moment of inertia, respectively. We take $E = z_4$, and the width of the top and bottom sections are $w_1 = (z_2/z_4)w$ and $w_2 = (z_3/z_4)w$ respectively, while all other dimensions are the same as Figure 3.9 shows. For simulation convenience, we generate $10,000$ realizations from both HF and LF QoI functions, constructing their surrogates

$\tilde{f}^{\mathrm{HF}}$ and $\tilde{f}^{\mathrm{LF}}$ using polynomial chaos expansion (PCE) with a total degree of 3. The relative MSE of $\tilde{f}^{\mathrm{HF}}$ and $\tilde{f}^{\mathrm{LF}}$ are 1.38×10^{-2} and 1.26×10^{-2}, respectively. We define the HF function $h^{\mathrm{HF}}(\boldsymbol{z}) \coloneqq \tilde{f}^{\mathrm{HF}}(\boldsymbol{z}) + 4.04$ and LF function $h^{\mathrm{LF}}(\boldsymbol{z}) \coloneqq \tilde{f}^{\mathrm{LF}}(\boldsymbol{z}) + 3.18$. These two functions output negative values if the displacement of the composite beam is less than -4.04 (for HF) or -3.18 (for LF). These two functions are defined so that negative values indicate failures.

To determine the optimal ℓ, we replicated the experimental setup used in the previous examples, employing $L = 1 \times 10^2$ HF trials and $M = 1 \times 10^6$ LF trials, with the process repeated ten times to account for uncertainty. The estimated variances for various ℓ values are illustrated in Figure 3.10, showing slight differences between the two approaches. The values of ℓ that minimize the variance are 14.90 and 18.57 for approach one and approach two, respectively. Based on these findings, we proceed with the convergence analysis using the two identified ℓ values.

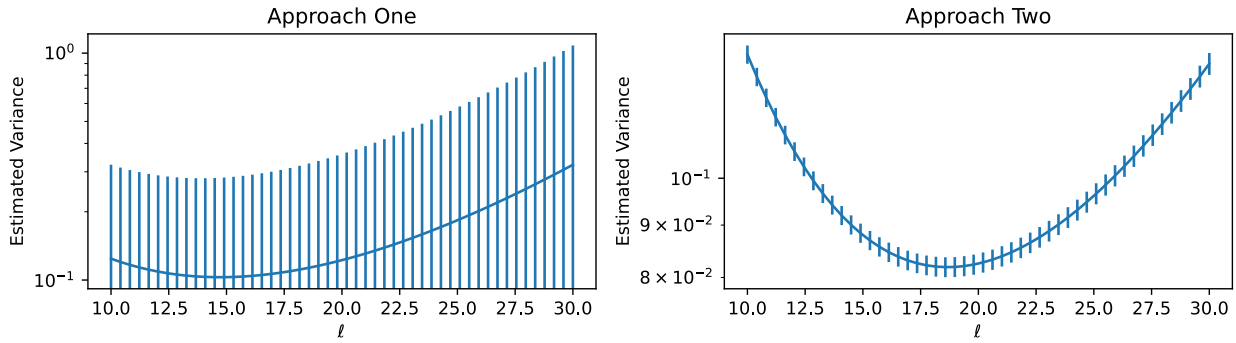


Figure 3.10: Estimated variance of L-BF-IS for different ℓ values, with uncertainty bars representing the 95% confidence interval. Using $L = 100$ HF evaluations (approach one) and $M = 1,000,000$ LF evaluations (both approaches).

In this study, we set the MALA step size $\tau = 1 \times 10^{-3}$ and chose both the burn-in number B and iteration number T to be $10,000$. The starting value $\boldsymbol{z}^{(0)}$ is $[1 \times 10^1, 1 \times 10^6, 1 \times 10^6, 1 \times 10^4]$. This approach was compared with the MF-IS technique [424], which uses a Gaussian mixture model with 10 cluster centers. For the convergence analysis, we used HF sample sizes N as $10, 21, 46, 100, 215, 464, 1000, 2154, 4641$, and 10000 with experiments repeated $1,000$ times to assess the standard deviation of the results. The outcomes, illustrated in Figure 3.11, highlight a noteworthy observation regarding the impact of an inaccurately chosen parameter ℓ on the L-BF-IS

estimates. Specifically, with $\ell = 18.57$, the biasing distribution derived from the MALA significantly reduces the RMSE at the early stage, whereas the convergence with $\ell = 14.90$ demonstrated relatively inferior performance. Given that $\ell = 14.90$ was obtained using approach one and $\ell = 18.57$ using approach two, our findings suggest that the latter provides a more accurate determination of the optimal ℓ value. Note that the L HF evaluations for approach one is not included in this figure. A potential reason for the observed bias of $\ell = 14.90$ is that varying ℓ values alter the smoothness conditions of the resultant biasing densities, which in turn negatively affects the convergence performance of the Langevin algorithm.

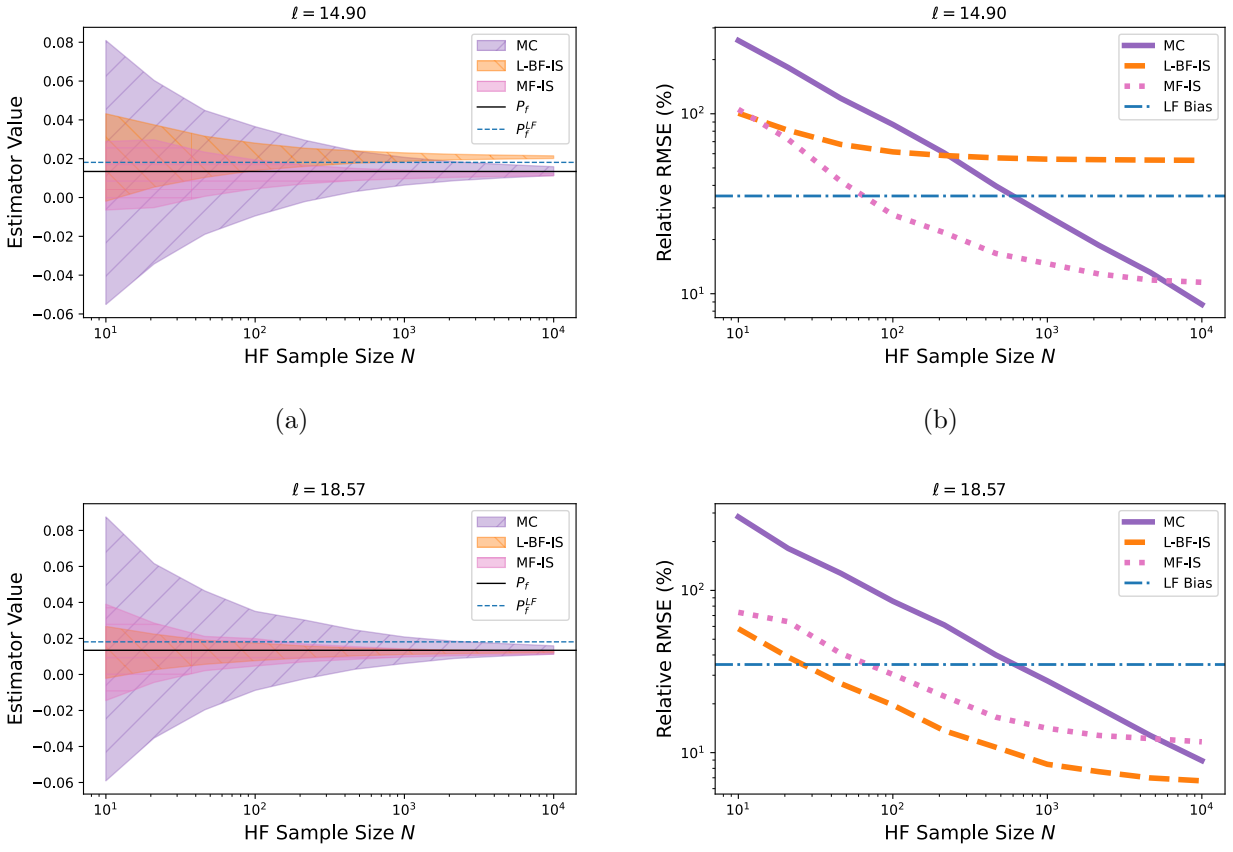


Figure 3.11: Convergence behavior of L-BF-IS (dash) for ℓ values of 14.90 (a-b) and 18.57 (c-d), compared with standard Monte Carlo (solid) and MF-IS (dot) using 10 Gaussian mixture clusters for the beam problem in Section 3.4.3.1. The shaded areas represent the 95% confidence interval from $1,000$ trials.

3.4.3.2 Steady-state Heat Equation with Random Inputs

In this section, we discuss the performance of a 2D steady-state stochastic heat equation, with uncertain thermal coefficient K. The steady-state heat equation can be described as

$$\begin{aligned} -\frac{\partial}{\partial \boldsymbol{x}}\left(K(\boldsymbol{x},\boldsymbol{z})\frac{\partial u(\boldsymbol{x},\boldsymbol{z})}{\partial \boldsymbol{x}}\right) &= 1.0, \quad \boldsymbol{x}\in(0,1)^2 \\ u(\boldsymbol{x},\boldsymbol{z}) &= 0, \quad x_1\in\{0,1\} \text{ or } x_2\in\{0,1\}. \end{aligned} \tag{3.41}$$

The thermal coefficient $K(\boldsymbol{x},\boldsymbol{z})$ is defined as a stochastic process given by [440],

$$K(\boldsymbol{x},\boldsymbol{z}) = \bar{K} + \exp\left(\frac{\sqrt{2}}{\sqrt{D'}}\sum_{i=1}^{D'} z_i^w \cos\left(z_i^{a_1}x_1 + z_i^{a_2}x_2 + z_i^b\right)\right), \tag{3.42}$$

where $\bar{K}=3$, $z_i^w, z_i^{a_1}, z_i^{a_2} \overset{iid}{\sim} \mathcal{N}(0,1)$ and $z_i^b \overset{iid}{\sim} U[0,2\pi]$. The corresponding covariance kernel function of the Gaussian process for the exponent part of Equation (3.42) is $k(x_1,x_2) = \exp\big(-(x_1-x_2)^2\big)$. The dimension of the problem is $4D' = D = 400$. The QoI is the solution on the domain $(0,1)^2$ with a grid size of $\Delta x_1 = \Delta x_2 = 1.67\times10^{-2}$ in each direction, then leading to a solution in $\mathbb{R}^{61\times61}$. Therefore the QoI functions are defined as $f^{\text{LF}}, f^{\text{HF}} : \mathbb{R}^{400} \to \mathbb{R}^{61\times61}$. The finite difference method is used to compute the HF QoI and a pre-trained Physics-informed Neural Operator (PINO) [314] as a surrogate LF QoI function. The core idea of PINO is to construct a deep-learning-based surrogate that learns the operator $\mathcal{G}$, such that $\mathcal{G}(K)(\boldsymbol{x},\boldsymbol{z}) \approx u(\boldsymbol{x},\boldsymbol{z})$. We use a PINO model pre-trained following [314]; however, since the distribution of K in [314] differs from our K defined in Equation (3.42), this setting can be treated as a transfer learning problem. In this case, the training data for PINO are not counted as additional HF evaluations. The deep-learning structure of the PINO provides the Jacobian $\partial\mathcal{G}(K)(\boldsymbol{x},\boldsymbol{z})/\partial\boldsymbol{z}$ given $\boldsymbol{x}$ is defined over a fixed grid. In Figure 3.13a, three samples of $K(\boldsymbol{x},\boldsymbol{z})$ and the corresponding realizations of $u(\boldsymbol{x},\boldsymbol{z})$ are presented.

We assume the system fails if the maximum value of $u(\boldsymbol{x},\boldsymbol{z})$ is larger than some thresholds. The LF and HF functions are defined as $h^{\text{LF}}(\boldsymbol{z}) = 0.019 - \max(f^{\text{LF}})$ and $h^{\text{HF}}(\boldsymbol{z}) = 0.022 - \max(f^{\text{HF}})$, respectively. We choose $M = 1\times10^6$ and $L = 1\times10^2$ for selecting ℓ, with the result presented in Figure 3.12. The optimal value of ℓ, as selected by the approach two, is 1786.0.

The Langevin algorithm is employed with step size $\tau = 1\times10^{-4}$ and burn-in number $B = 1\times10^4$. We provide three examples of the thermal coefficients that are sampled from the biasing density

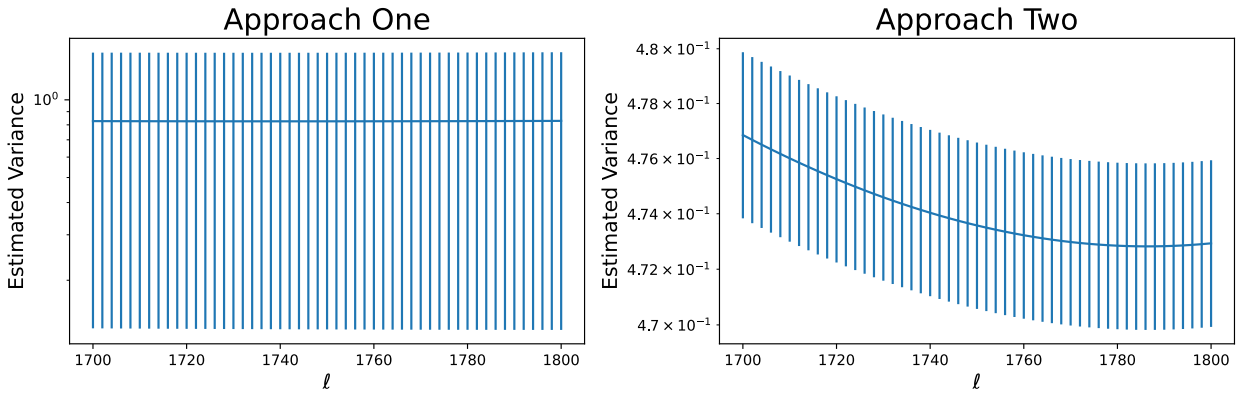


Figure 3.12: The estimated variance of L-BF-IS with 95% confidence intervals across varying values of ℓ is illustrated in the left figure using approach one and in the right figure using approach two. It is worth noting that the left figure exhibits a minimum point; however, the uncertainty is sufficiently large to obscure its depiction.

$q(\boldsymbol{z})$ with the associated LF and HF QoIs presented in Figure 3.13b. Comparing Figure 3.13a and Figure 3.13b, we note that the results generated from the Langevin algorithm are more likely to produce failure results defined by functions $h(\cdot)$ and with smaller variance relative to the original reference density $p(\boldsymbol{z})$. These examples explain why importance sampling using the Langevin algorithm can help reduce the variance of the estimates and eventually the MSE.

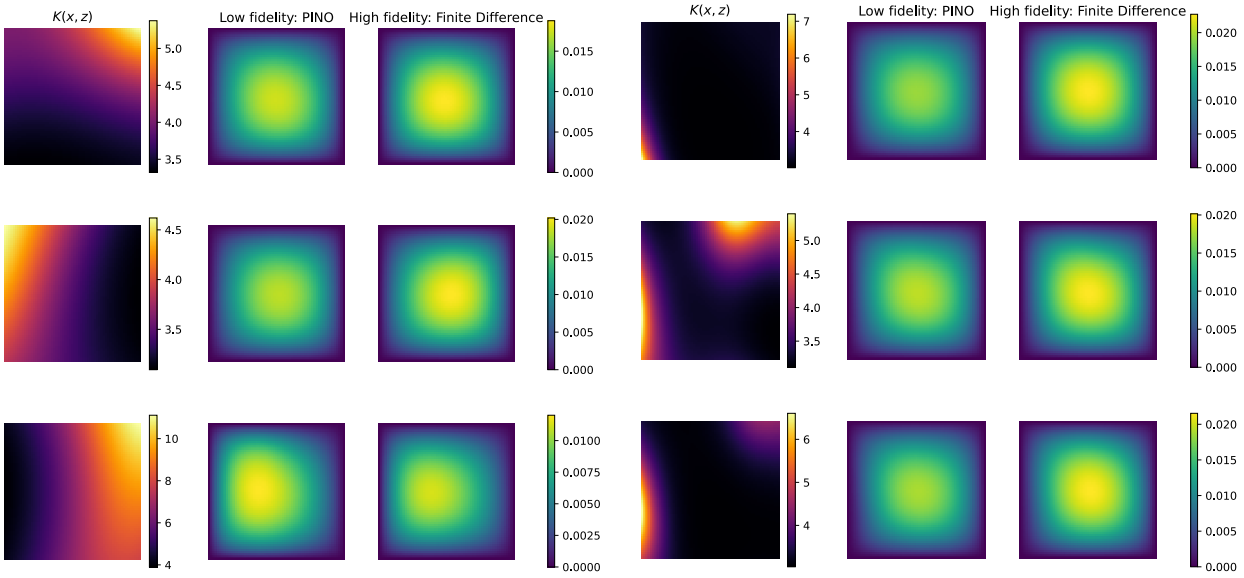
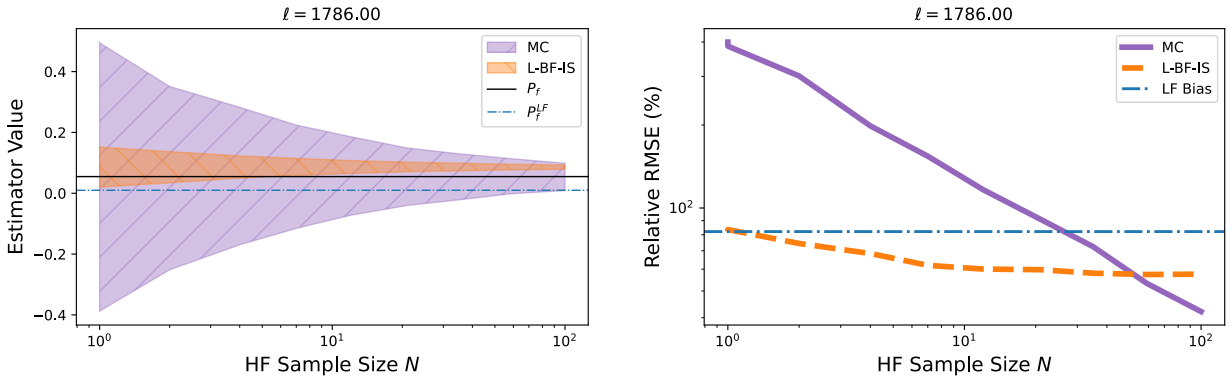


Figure 3.13: The solutions of the steady-state heat equation in Equation (3.41) given three different realizations of the thermal coefficient $K(\boldsymbol{x}, \boldsymbol{z})$ on a 61×61 grid over $(0, 1)^2$ sampled from Equation (3.42) (a) or $q(\boldsymbol{z})$ (b). For both figures, the left column is the visualization of the thermal coefficient, the middle column is the LF QoI solution provided by a pre-trained PINO, and the right column is the HF QoI solution computed using the finite difference method.

The convergence of the estimator is depicted in Figure 3.14. We restrict the performance comparison to cases where the number of HF evaluations is small (≤ 63). Notably, a non-trivial bias of approximately 2% is observed. Despite demonstrating that the L-BF-IS estimator is unbiased in Section 3.3.2, the samples produced by the Langevin algorithm, as described in Section 3.3.5, cannot be guaranteed to exactly represent $q(\boldsymbol{z})$, thereby generating numerical bias in the observations. Given the problem's high dimensionality and suboptimal LF model, the L-BF-IS is able to reduce the relative RMSE from 85% to 65% using less than $N = 100$ HF samples. Further-



(a) (b)

Figure 3.14: Convergence of the L-BF-IS (dashed) against standard Monte Carlo (solid) and LF failure probability (dashed dot) with 95% confidence bound computed from 1,000 trials for the steady-state heat equation problem in Section 3.4.3.2.

more, the variance observed in the results generated by the L-BF-IS is substantially lower than that of standard Monte Carlo methods. This scenario underscores a case for employing pre-trained deep-learning-based operator learning strategies as a LF model.

3.5 Conclusion

In this study, we present an importance sampling estimator, referred to as the Langevin bi-fidelity importance sampling (L-BF-IS). This estimator operates under the premise that in many practical applications, a considerably cheaper, differentiable lower-fidelity model is available. L-BF-IS employs the Metropolis-adjusted Langevin algorithm for sampling from a biasing distribution informed from the low-fidelity model, aiming to estimate failure probabilities with limited high-fidelity evaluations. The algorithm demonstrates superior performance in scenarios characterized by high input dimensions and multimodal failure regions. Two methodologies are introduced to tune a key parameter of the biasing distribution. Our empirical tests include a 1D manufactured bimodal function and two experimental setups using synthetic functions, with one involving 1000 random inputs. Additional experiments are conducted estimating failure probabilities of physics-based problems with failure probabilities of magnitude 1-5%. These experiments illustrate the efficiency of the L-BF-IS estimator relative to standard Monte Carlo simulation and a different importance sampling approach.

L-BF-IS demonstrates significant advantages, and our findings reveal opportunities to enhance the proposed estimator further. One promising direction is incorporating prior knowledge when selecting the Langevin algorithm's starting point and step size. Additionally, future research could explore treating the low-fidelity surrogate as dynamic, updating it at each iteration. This approach would extend the methodology into adaptive importance sampling, making it applicable when a fixed low-fidelity model is unavailable. These directions not only have the potential to refine the effectiveness of existing models but also pave the way for advancing state-of-the-art importance sampling techniques.

Chapter 4

Bi-fidelity Stochastic Subspace Descent: A Surrogated Line Search Approach

4.1 Abstract

Efficient optimization is a fundamental problem in many scientific and engineering applications, particularly when dealing with expensive-to-evaluate objective functions. In this work, we propose the Bi-fidelity Stochastic Subspace Descent (BF-SSD) algorithm, a novel zeroth-order optimization approach that leverages a bi-fidelity framework to significantly reduce computational costs. The method constructs a surrogate model by combining high-fidelity (HF) and low-fidelity (LF) evaluations, enabling efficient step size selection via backtracking line search, for which we can prove convergence guarantees for some sets of assumptions. We evaluate BF-SSD on four distinct problems: a synthetic optimization benchmark and three machine learning tasks, including dual-form kernel ridge regression, black-box adversarial attacks, and transformer-based black-box language model fine-tuning. The results demonstrate that BF-SSD consistently outperforms competing methods and achieves superior performance with fewer HF function evaluations. This study underscores the potential of bi-fidelity frameworks for addressing large-scale, high-dimensional optimization problems in a computationally efficient manner, making BF-SSD a promising tool for real-world applications.

4.2 Introduction

In this work, we are interested in the unconstrained optimization problem

$$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}), \tag{4.1}$$

where the objective function $f : \mathbb{R}^D \to \mathbb{R}$ is L-smooth but ∇f is difficult to obtain and the dimension is large enough (i.e., $D \gtrsim 100$) that traditional derivative free methods struggle. The focus of this work is on selecting an appropriate step size (learning rate) α_k for the iterative descent scheme

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \boldsymbol{v}_k, \tag{4.2}$$

where $\boldsymbol{v}_k \in \mathbb{R}^D$ is an estimate of $\nabla f(\boldsymbol{x}_k)$.

Selecting an appropriate step size can significantly improve the convergence performance of the optimization process. This is illustrated in Figure 4.1, where an example function is optimized using different methods with and without a step size tuning scheme. However, most machine learning problems either use a fixed step size throughout the entire optimization process or employ an adaptive step size scheduling strategy [159]. Both of these methods, although convenient to implement, ignore the intrinsic characteristics of the objective function. In contrast, line search methods, including exact line search and backtracking, produce better step sizes but at the cost of additional function evaluations. This makes them impractical when the function evaluation budget is limited, as is often the case in black-box machine learning problems. To address this, we propose a novel bi-fidelity approach to tune the step size by considering the objective function from a multi-fidelity perspective. The concept of multi-fidelity refers to two (or more) levels: the high-fidelity (HF) objective, which provides accurate but expensive function evaluations, and one or more low-fidelity (LF) objectives which are cheap but inaccurate approximations to the true HF objective. We emphasize that this multi-fidelity structure is more prevalent than often recognized in machine learning applications, making our proposed method broadly applicable, as demonstrated in the experimental section.

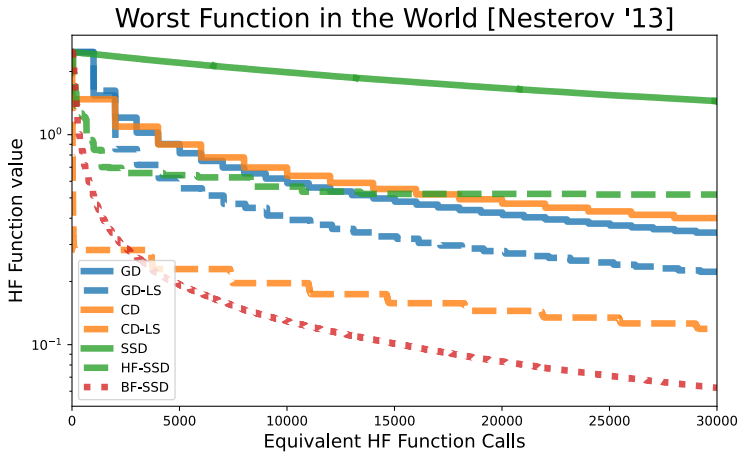


Figure 4.1: Gradient Descent (GD), Coordinate Descent (CD), and Stochastic Subspace Descent (SSD), along with their respective backtracking line search (LS) variants for step size tuning, as well as the proposed Bi-fidelity SSD (BF-SSD), are evaluated on the "worst function in the world" example, detailed in Section 4.5.1.

For simplicity, we focus on the bi-fidelity case, where only two fidelity levels are considered. The high-fidelity objective, f^{HF}, is treated as the ground-truth objective function, so we treat f^{HF} and the f from Eq. (4.1) synonymously. We construct simple bi-fidelity surrogates **after** obtaining the gradient estimation $\boldsymbol{v}_k$. Specifically, given the LF objective $f^{\mathrm{LF}} : \mathbb{R}^D \to \mathbb{R}$, the current position $\boldsymbol{x}_k$, $\boldsymbol{v}_k$, and a budget n_k for HF evaluations at this step, the **local** 1D surrogate is constructed as

$$\tilde{\varphi}_k(\alpha; n_k) = \rho f^{\mathrm{LF}}(\boldsymbol{x}_k - \alpha \boldsymbol{v}_k) + \tilde{\psi}_k(\alpha; n_k), \quad \alpha \in [0, \alpha_{\max}], \tag{4.3}$$

where $\alpha_{\max}$ is the initial step size, in order to approximate the HF counterpart $\varphi_k(\alpha) \coloneqq f^{\mathrm{HF}}(\boldsymbol{x}_k - \alpha \boldsymbol{v}_k)$. Here, ρ is a scalar, and $\tilde{\psi}_k(\cdot; n_k) : \mathbb{R} \to \mathbb{R}$ is a piecewise linear function constructed using n_k HF evaluations. Once the surrogate $\tilde{\varphi}_k : \mathbb{R} \to \mathbb{R}$ is constructed, the step size is selected using backtracking line search by (approximately) solving

$$\alpha_k = \operatorname*{arg\,min}_{\alpha \in [0, \alpha_{\max}]} \tilde{\varphi}_k(\alpha; n_k), \tag{4.4}$$

thereby providing a reasonable estimation for the step size α_k.

Assuming the scalar ρ is properly chosen so that the difference

$$d(\boldsymbol{x}) \coloneqq f^{\mathrm{HF}}(\boldsymbol{x}) - \rho f^{\mathrm{LF}}(\boldsymbol{x}) \tag{4.5}$$

is Lipschitz continuous, we show that the convergence of this descent method is guaranteed, and $K_\epsilon = \mathcal{O}(L/\epsilon)$ iterations are needed to ensure that $\min_k \|\nabla f^{\mathrm{HF}}(\boldsymbol{x}_k)\|^2$ is ϵ-small. Moreover, when the HF and LF functions are well-aligned, i.e., the Lipschitz constant W of $d(\boldsymbol{x})$ is small, the required number of HF function evaluations $N_\epsilon = \mathcal{O}(WL^2/\epsilon + DL/\epsilon)$ is not large.

For implementation, we focus on high-dimensional zeroth-order optimization problems, using the Stochastic Subspace Descent (SSD) method [285] combined with the proposed step size tuning strategy, and call the resulting method Bi-fidelity Stochastic Subspace Descent (BF-SSD). BF-SSD demonstrates strong empirical performance across various tasks and holds great potential for future applications.

4.2.1 Related Work

Line Search for Optimization Line search is a widely used method for determining step sizes in optimization algorithms. Line searches can be either exact, meaning that α is chosen to exactly or almost exactly minimize $f^{\mathrm{HF}}(\boldsymbol{x}_k - \alpha \boldsymbol{v}_k)$, or inexact. Exact line searches are computationally expensive, so other than in special cases, they are rarely used in practice. Common inexact line search methods include backtracking line search [395], the Polyak step size [438], spectral methods such as [35], and learning rate scheduling [159]. Among these, backtracking line search is particularly popular due to its simplicity and explainable design, often employing stopping criteria like the Armijo and Wolfe conditions [395]. However, backtracking line search increases the high computational costs due to the numerous function evaluations required at each iteration to determine the step size. One way to mitigate this issue is by constructing surrogate models to guide step size selection. For example, Yue et al. [579] and Grundvig et al. [200] used reduced-order models to approximate the objective function during line search, while Mahsereci and Hennig [340] employed a probabilistic Gaussian model for step size selection. Paquette and Scheinberg [418] provided a theoretical analysis of line search in stochastic optimization. However, these approaches do not account for the multi-fidelity structure of objective functions, which is the focus of this work.

Derivative-Free and Zeroth-Order Optimization Derivative-free optimization refers

to a family of optimization techniques that rely solely on function evaluations, without requiring gradient information, to find the optimum of an objective function. This category includes methods such as Bayesian optimization [467], direct search [277], trust region methods [112], genetic algorithms [491], and zeroth-order optimization [326]. Among these, zeroth-order methods stand out for their scalability to high-dimensional problems and reliable convergence properties. Following [326], we refer to zeroth-order algorithms as the type of algorithms that approximate gradients using finite difference techniques and subsequently apply strategies similar to first-order methods. These methods have shown great promise in various machine learning applications where objective functions are smooth but lack accessible derivatives. Recent advances include their use in solving black-box adversarial attacks [89, 88] and fine-tuning large models with minimal memory overhead in models such as MeZO, S-MeZO and SubZO among others [501, 502, 350, 329, 577, 584].

Randomized Zeroth-Order Optimization for High-Dimensional Problems In high-dimensional zeroth-order optimization problems, estimating gradients via finite differences can be computationally prohibitive. To address this, randomized algorithms have been proposed to reduce the cost of gradient estimation. The Simultaneous Perturbation Stochastic Approximation (SPSA) [489, 490] uses Rademacher random vectors for gradient estimation, while Gaussian smoothing methods [386] employ Gaussian random vectors. These algorithms typically provide gradient estimators projected onto one-dimensional subspaces. However, for certain problems it is worth the increased functional calls to get an improved estimate of the gradient. Stochastic Subspace Descent (SSD) [285] explores this idea by projecting the gradient onto a random subspace of dimension ℓ for any $1 \leq \ell \leq D$, providing a more generalized framework for randomized zeroth-order optimization.

Multi-Fidelity Model and Optimization Multi-fidelity modeling is a well-established approach in engineering and scientific computing for reducing computational costs. It has been widely applied across various domains, including aerodynamic design [583], structural optimization [389, 126], data sampling [96], and uncertainty quantification [427, 94]. As a strategy to tackle expensive problems while minimizing computational burden, multi-fidelity modeling has been employed in hyperparameter tuning [559], accelerating Bayesian optimization [256, 509], and

reinforcement learning [116] within machine learning. However, despite its relevance in settings where function evaluations are costly, its application in zeroth-order optimization remains largely unexplored [603] and has not been applied to any randomized zeroth-order method.

4.2.2 Contributions

In this work, we propose a multi-fidelity approach for constructing surrogate models in line search. Unlike previous works that use static surrogate (reduced-order) models which do not change from iteration-to-iterate [579, 200] or which incorporate inexactness [81, 526, 418, 251], our method constructs a temporary surrogate **after** the gradient is estimated. This allows us to focus on building a **one-dimensional surrogate** which is a much easier task than building an accurate D-dimensional surrogate. By leveraging a low-fidelity (LF) model, we construct a simple linear surrogate using a small number of high-fidelity (HF) evaluations, n_k. This surrogate facilitates the identification of an optimal step size under certain conditions between the LF and HF models.

Specifically, this work makes the following contributions:

(1) We develop the general BF-SSD algorithm which is a stochastic zeroth-order optimization method with a bi-fidelity line search that allows the user to choose the approximation quality of the gradient by tuning ℓ (reducing to deterministic gradient descent when $\ell = D$).

(2) When the error of the gradient estimate is negligible (e.g., ℓ is sufficiently large), we give specific conditions on the relation between the HF and LF functions that will guarantee convergence to a stationary point (or a global minimizer when f is convex).

(3) We highlight that many machine learning problems naturally have a corresponding sub-problem (low-fidelity model) that can be used to construct a surrogate model, significantly improving optimization efficiency. Despite its potential, this strategy has not received sufficient attention in prior work.

(4) We systematically evaluate the proposed optimization method, BF-SSD, against other

zeroth-order optimization methods on one synthetic function and the following three real-world applications:

- Kernel ridge regression with a Nyström-based low-fidelity approximation;
- Black-box image-based adversarial attacks with a low-fidelity model trained via knowledge distillation;
- Soft prompting of language models using a smaller training set to construct the bi-fidelity surrogate.

The rest of the paper is organized as follows. Section 4.3 introduces the proposed bi-fidelity line search method and provides convergence results. Section 4.4 details the implementation of the proposed method with SSD. Section 4.5 presents the experimental results, and Section 4.6 concludes the paper.

4.3 Line Search on Bi-fidelity Surrogate

In this section we discuss the proposed algorithm and the main theoretical results delivered in this work. Unless specified, $\|\cdot\|$ denotes the Euclidean norm. In the purpose of simplicity and focusing on the main contribution, we assume the gradient $\nabla f^{\mathrm{HF}}(\boldsymbol{x}_k)$ can be accurately estimated by $\boldsymbol{v}_k$ in proofs.

4.3.1 Algorithm

First we define the algorithm, which consists of three step for each iteration k:

(1) Given the current position $\boldsymbol{x}_k \in \mathbb{R}^D$, gradient $\boldsymbol{v}_k \in \mathbb{R}^D$, and initial step size $\alpha_{\max} \in \mathbb{R}$, sample n_k equi-spaced HF evaluations in $[0, \alpha_{\max}]$ and build the surrogate $\tilde{\varphi}_k : \mathbb{R} \to \mathbb{R}$ following Equation (4.3) (see Algo. 6 for details);

(2) Given Armijo condition parameters $c \in (0, 1)$, $\beta \leq 1/2$, and initial step size $\alpha_{\max} \geq$

$c/(L+cL)$, conduct bi-fidelity adjusted Armijo backtracking so that

$$\begin{aligned} \alpha_k &= \max_{m\in\mathbb{N}} c^m \alpha_{\max} \\ \text{s.t.} \quad \tilde{\varphi}_k(c^m\alpha_{\max}; n_k) &\leq f^{\mathrm{HF}}(\boldsymbol{x}_k) - \beta c^m \alpha_{\max}\|\boldsymbol{v}_k\|^2. \end{aligned} \tag{4.6}$$

See Algo. 7 for details.

(3) Evaluating f^{HF} at the new point and continue the iterations.

4.3.2 Convergence Results

For convergence, we make the following assumptions:

Assumption 4.3.1. The objective function $f^{\mathrm{HF}} : \mathbb{R}^D \to \mathbb{R}$ attains its minimum f^* and ∇f^{HF} is L-Lipschitz continuous, i.e., there exists $L \in \mathbb{R}$ such that

$$\|\nabla f^{\mathrm{HF}}(\boldsymbol{x}) - \nabla f^{\mathrm{HF}}(\boldsymbol{y})\| \leq L\|\boldsymbol{x}-\boldsymbol{y}\|, \quad \forall \boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^D. \tag{4.7}$$

Note that Assumption 4.3.1 is standard for analysis of zeroth and first-order methods. The constant L must be known to the algorithm since it is used to set $\alpha_{\max}$.

Assumption 4.3.2. The difference between f^{HF} and f^{LF} is assumed to be smooth with Lipschitz constant; specifically, we assume there exists $W, \rho \in \mathbb{R}$ such that

$$\|\big(f^{\mathrm{HF}}(\boldsymbol{x}) - \rho f^{\mathrm{LF}}(\boldsymbol{x})\big) - \big(f^{\mathrm{HF}}(\boldsymbol{y}) - \rho f^{\mathrm{LF}}(\boldsymbol{y})\big)\| \leq W\|\boldsymbol{x}-\boldsymbol{y}\|, \forall \boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^D. \tag{4.8}$$

Clearly there needs to be some assumption made about the relationship of f^{LF} to f^{HF}. Our particular assumption allows for f^{LF} to be **uncalibrated**, meaning we do not even require $f^{\mathrm{LF}}(\boldsymbol{x}) \approx f^{\mathrm{HF}}(\boldsymbol{x})$ since this can be corrected for by the surrogate.

Assumption 4.3.3. For each iteration k, the surrogate must be accurate (i.e., satisfies Eq. (4.13)); in particular, a sufficient condition is that the the number of HF evaluations for building the surrogate, n_k, satisfies

$$n_k \geq \frac{WL(1+c)\alpha_{\max}}{c\beta\|\boldsymbol{v}_k\|^2}, \quad \text{i.e.,} \quad n_k = \Omega\left(\frac{W(\alpha_{\max}L+1)}{\|\boldsymbol{v}_k\|^2}\right). \tag{4.9}$$

Using $\boldsymbol{v}_k = \nabla f(\boldsymbol{x}_k)$ and with the above assumptions satisfied and sufficiently large initial step size $\alpha_{\max} \geq c/(cL + L)$, the designed bi-fidelity line search leads to the following result:

Theorem 4.3.4. *Given an initial point $\boldsymbol{x}_0$, then the algorithm generates a sequence $(\boldsymbol{x}_k)$ such that*

$$\min_{k\in\{0,\ldots,K\}} \|\nabla f^{\mathrm{HF}}(\boldsymbol{x}_k)\|^2 \leq \frac{2L(1+c)(f^{\mathrm{HF}}(\boldsymbol{x}_0) - f^*)}{(K+1)c\beta}. \tag{4.10}$$

That is to say, $K_\epsilon = \mathcal{O}(L/\epsilon)$ iterations are required to obtain $\min_{k\leq K_\epsilon} \|\nabla f^{\mathrm{HF}}(\boldsymbol{x}_k)\|^2 \leq \epsilon$.

Remark 4.3.5. Theorem 4.3.4 holds when $\boldsymbol{v}_k = \nabla f(\boldsymbol{x}_k)$. The error in approximating $\nabla f(\boldsymbol{x}_k)$ using finite difference methods with $\mathcal{O}(D)$ samples is typically negligible in comparison to the optimization error (see [286] for a precise quantitative statement for the case of SSD). Hence assuming we accurately estimate $\boldsymbol{v}_k = \nabla f(\boldsymbol{x}_k)$ with $\mathcal{O}(D)$ samples per step, then a bound for the total number of HF evaluations for ϵ-convergence is

$$N_\epsilon = \sum_{k=1}^{K_\epsilon} (n_k + \mathcal{O}(D)) = \mathcal{O}\left(\frac{WL^2}{\epsilon} + \frac{DL}{\epsilon}\right). \tag{4.11}$$

The result in Equation (4.11) suggests that the number of function evaluation for ϵ-convergence for the proposed algorithm can be significantly reduced by a small value of W.

Remark 4.3.6. When using zeroth-order gradient descent making the same assumption that we accurately estimate $\boldsymbol{v}_k = \nabla f(\boldsymbol{x}_k)$ with $\mathcal{O}(D)$ samples per step, a bound for the total number of HF evaluations for ϵ-convergence is

$$N_\epsilon = \sum_{k=1}^{K_\epsilon} (\log_{c^{-1}}(\alpha_{\max} L) + \mathcal{O}(D)) = \mathcal{O}\left(\frac{L\log(L)}{\epsilon} + \frac{DL}{\epsilon}\right). \tag{4.12}$$

The proof of Remark 4.3.6 follows the convergence proof of gradient descent using backtracking line search. Comparing the results in Equation (4.11) and Equation (4.12), we notice that the advantage of using our bi-fidelity surrogate depends on the value of W. From a theoretical bound perspective, if W is sufficiently small so that $WL^2 \leq L\log L$, then the worst-case bound of our method is better than that of zeroth-order gradient descent. However, we emphasize that the convergence analysis is loose, due to the global nature of the assumptions and difficulty in precisely

describing the quality of the LF function. Hence we mostly view the convergence analysis simply as a reassurance that the method does converge, and rely on numerical experiments to elucidate when the method improves over baseline methods.

4.3.3 Proof of Theorem 4.3.4

Before the proof, we first introduce the following lemma:

Lemma 4.3.7. *With Assumption 4.3.2 and Assumption 4.3.3 satisfied, for any $\alpha \in [0, \alpha_{\max}]$, the 1D surrogate $\tilde{\varphi}_k(\alpha)$ satisfies the following bound,*

$$|\tilde{\varphi}_k(\alpha; n_k) - \varphi(\alpha)| \leq \frac{\|\boldsymbol{v}_k\|^2}{2} \min\left\{\frac{c}{(1+c)^2 L}, \frac{c\beta}{(1+c)L}, \beta\alpha_{\max}\right\} = \frac{c\beta\|\boldsymbol{v}_k\|^2}{2(1+c)L}. \tag{4.13}$$

The proof of Lemma 4.3.7 is in D.1. Following this lemma, the proof of Theorem 4.3.4 is as follows.

Proof. Following Lemma 4.3.7, we have

$$|f^{\mathrm{HF}}(\boldsymbol{x}_{k+1}) - \tilde{\varphi}_k(\alpha_k; n_k)| = |\varphi(\alpha_k) - \tilde{\varphi}_k(\alpha_k; n_k)| \leq \frac{\|\boldsymbol{v}_k\|^2}{2} \min\left\{\frac{c}{(1+c)^2 L}, \frac{c\beta}{(1+c)L}, \beta\alpha_{\max}\right\} \tag{4.14}$$

and, using the standard descent lemma for L-smooth functions (guaranteed by Assumption 4.3.1),

$$f^{\mathrm{HF}}(\boldsymbol{x}_{k+1}) \leq f^{\mathrm{HF}}(\boldsymbol{x}_k) - \alpha_k\|\boldsymbol{v}_k\|^2 + \frac{\alpha_k^2 L}{2}\|\boldsymbol{v}_k\|^2, \tag{4.15}$$

therefore using the triangle inequality, the surrogate $\tilde{\varphi}_k$ is bounded as

$$\begin{aligned}\tilde{\varphi}_k(\alpha_k; n_k) &\leq f^{\mathrm{HF}}(\boldsymbol{x}_{k+1}) + |f^{\mathrm{HF}}(\boldsymbol{x}_{k+1}) - \tilde{\varphi}_k(\alpha_k; n_k)| \\ &\leq f^{\mathrm{HF}}(\boldsymbol{x}_k) + \left(-\alpha_k + \frac{\alpha_k^2 L}{2} + \frac{c}{2(1+c)^2 L}\right)\|\boldsymbol{v}_k\|^2.\end{aligned}$$

When the step size satisfies $\alpha_k \in [c/(L+cL), 1/(L+cL)]$, the quadratic inequality $-\alpha_k + \alpha_k^2 L/2 + c/(2(1+c)^2 L) \leq -\alpha_k/2$ holds, along with the fact that $\beta \leq 1/2$, which implies the following

bi-fidelity-adjusted Armijo condition

$$\begin{aligned}\tilde{\varphi}_k\left(\alpha_k; n_k\right) &\leq f^{\mathrm{HF}}(\boldsymbol{x}_k) + \left(-\alpha_k + \frac{\alpha_k^2 L}{2} + \frac{c}{2(1+c)^2 L}\right)\|\boldsymbol{v}_k\|^2 \\ &\leq f^{\mathrm{HF}}(\boldsymbol{x}_k) - \frac{\alpha_k}{2}\|\boldsymbol{v}_k\|^2 \\ &\leq f^{\mathrm{HF}}(\boldsymbol{x}_k) - \beta\alpha_k\|\boldsymbol{v}_k\|^2.\end{aligned} \tag{4.16}$$

The last line in Equation (4.16) satisfies the bi-fidelity-adjusted Armijo condition in Equation (4.6). Therefore, the bi-fidelity backtracking either terminates immediately with $\alpha_k = \alpha_{\max}$ or else $\alpha_k \geq c/(L + cL)$, and implies

$$\tilde{\varphi}_k\left(\alpha_k; n_k\right) \leq f^{\mathrm{HF}}(\boldsymbol{x}_k) - \beta\|\boldsymbol{v}_k\|^2 \min\left\{\frac{c}{(1+c)L}, \alpha_{\max}\right\} = f^{\mathrm{HF}}(\boldsymbol{x}_k) - \frac{\beta c}{(1+c)L}\|\boldsymbol{v}_k\|^2, \tag{4.17}$$

where the last equality comes from $\alpha_{\max} \geq c/((1+c)L)$. Since $|f^{\mathrm{HF}}(\boldsymbol{x}_{k+1}) - \tilde{\varphi}_k\left(\alpha_k; n_k\right)| \leq \beta c\|\boldsymbol{v}_k\|^2/(2(1+c)L)$ from Lemma 4.3.7, combined with Equation (4.17), we have

$$\begin{aligned}f^{\mathrm{HF}}(\boldsymbol{x}_{k+1}) &\leq \tilde{\varphi}_k\left(\alpha_k; n_k\right) + \left|f^{\mathrm{HF}}(\boldsymbol{x}_{k+1}) - \tilde{\varphi}_k\left(\alpha_k; n_k\right)\right| \\ &\leq \tilde{\varphi}_k\left(\alpha_k; n_k\right) + \frac{\beta c\|\boldsymbol{v}_k\|^2}{2(1+c)L} \\ &\leq f^{\mathrm{HF}}(\boldsymbol{x}_k) - \frac{\beta c\|\boldsymbol{v}_k\|^2}{2(1+c)L}.\end{aligned} \tag{4.18}$$

Equation (4.18) leads to the telescope series

$$\begin{aligned}\frac{\beta c}{2(1+c)L}\sum_{k=0}^{K}\|\boldsymbol{v}_k\|^2 &\leq \sum_{k=0}^{K}\left(f^{\mathrm{HF}}(\boldsymbol{x}_k) - f^{\mathrm{HF}}(\boldsymbol{x}_{k+1})\right) \\ &= f^{\mathrm{HF}}(\boldsymbol{x}_0) - f^{\mathrm{HF}}(\boldsymbol{x}_{K+1}) \leq f^{\mathrm{HF}}(\boldsymbol{x}_0) - f^*.\end{aligned} \tag{4.19}$$

Hence,

$$\begin{aligned}(K+1)\min_{k\in\{0,\ldots,K\}}\|\boldsymbol{v}_k\|^2 &\leq \left(\frac{\beta c}{2(1+c)L}\right)^{-1}\left(f^{\mathrm{HF}}(\boldsymbol{x}_0) - f^*\right) \\ &= \frac{2(1+c)L}{\beta c}\left(f^{\mathrm{HF}}(\boldsymbol{x}_0) - f^*\right).\end{aligned} \tag{4.20}$$

To guarantee $\min_{k\leq K_\epsilon}\|\nabla f^{\mathrm{HF}}(\boldsymbol{x}_k)\|^2 \leq \epsilon$, the value of K_ϵ should be

$$K_\epsilon \geq \frac{2(f^{\mathrm{HF}}(\boldsymbol{x}_0) - f^*)(1+c)L}{\beta c\epsilon} = \mathcal{O}\left(\frac{L}{\epsilon}\right). \tag{4.21}$$

□

Remark 4.3.8. Even if f^{HF} is non-convex, Eq. (4.18) implies that the method is a descent method, meaning $f^{\mathrm{HF}}(\boldsymbol{x}_{k+1}) \leq f^{\mathrm{HF}}(\boldsymbol{x}_k)$, hence after K iterations it is natural to use $\boldsymbol{x}_K$ as the output. This descent property is not enjoyed by other methods like subgradient descent, stochastic gradient descent or Polyak step size gradient descent.

Remark 4.3.9. If f^{HF} is convex, then the theorem implies convergence to a global minimizer. Or, if f^{HF} satisfies the Polyak-Lojasiewicz inequality with parameter μ (which includes some non-convex functions, as well as all strongly convex functions), then the theorem in conjunction with the descent property implies $f^{\mathrm{HF}}(\boldsymbol{x}_K) - f^* \leq \frac{L(1+c)}{(K+1)\mu c\beta}(f^{\mathrm{HF}}(\boldsymbol{x}_0) - f^*)$, cf. [259].

4.3.4 Examples of Possible Low-Fidelity Functions

In practice, the low-fidelity function f^{LF} can be constructed in various ways. The most straightforward approach is when a multi-fidelity structure is intrinsically present in the problem. For example, in [94, Section 5.1], the low-fidelity model is the exact solution to a simplified physical model; in particular, the LF objective ignores the holes in a cantilevered beam and thus can use the closed-form Euler-Bernoulli equation whereas the HF objective relies on an expensive finite-element simulation. However, in most machine learning problems, the low-fidelity model is not explicitly given, making its construction necessary. In this section, we discuss some possible cases for building the LF model and their resulting upper bound of W.

Affine Bi-Fidelity Relationship The most ideal case occurs when the HF model is an affine transformation of the LF model, i.e., $f^{\mathrm{HF}}(\boldsymbol{x}) = \rho f^{\mathrm{LF}}(\boldsymbol{x}) + c$. In this case, the Lipschitz constant W, as defined in Assumption 4.3.2, is zero, and the number of function evaluations required for convergence is proportional to the number of iterations, as $n_k = 1$ is sufficient (besides $\boldsymbol{x}_k$).

Quadratic Objective with Low-Rank LF Approximation Consider the case where the objective is quadratic with a positive semi-definite matrix $\boldsymbol{A} \in \mathbb{R}^{D\times D}$, and denote its rank-$r$ approximation $\widetilde{\boldsymbol{A}} \in \mathbb{R}^{D\times D}$, assuming that $\mathrm{rank}(\boldsymbol{A}) \gg \mathrm{rank}(\widetilde{\boldsymbol{A}})$. The HF objective is $f^{\mathrm{HF}}(\boldsymbol{x}) = \frac{1}{2}\langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{x}\rangle + \langle \boldsymbol{x}, \boldsymbol{a}\rangle$, and the LF objective is $f^{\mathrm{LF}}(\boldsymbol{x}) = \frac{1}{2}\langle \boldsymbol{x}, \tilde{\boldsymbol{A}}\boldsymbol{x}\rangle + \langle \boldsymbol{x}, \boldsymbol{a}\rangle$. Assuming the input space

$\mathcal{X}$ is bounded by a unit ball with radius R, the Lipschitz constant W is upper bounded as

$$W \leq \sup_{\boldsymbol{x}} \|\nabla f^{\mathrm{HF}}(\boldsymbol{x}) - \nabla f^{\mathrm{LF}}(\boldsymbol{x})\| = \sup_{\boldsymbol{x}} \|\boldsymbol{A} - \tilde{\boldsymbol{A}}\| \cdot \|\boldsymbol{x}\| \leq \lambda_{r+1} R, \tag{4.22}$$

where λ_{r+1} is the $(r+1)$-th eigenvalue of $\boldsymbol{A}$. The empirical problems in Section 4.5.1 and Section 4.5.2.1 fall into this category.

Full-Batch HF and Mini-Batch LF Objectives A common scenario in machine learning involves an objective function consisting of a large number of sub-functions evaluating on individual data samples. In this case, a natural choice for the LF objective is the summation over a smaller subset of the data. Specifically, assuming that the HF objective sums over datapoints $i = 1, \ldots, n$ and (without loss of generality, i.e., by relabeling) that the LF objective sums over datapoints $i = 1, \ldots, r$ for $r \ll n$, then the HF objective is $f^{\mathrm{HF}}(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x})$, and the LF objective is $f^{\mathrm{LF}}(\boldsymbol{x}) = \frac{1}{r}\sum_{i=1}^{r} f_i(\boldsymbol{x})$, so the Lipschitz constant W is upper bounded as

$$W \leq \sup_{\boldsymbol{x}} \left\| \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\boldsymbol{x}) - \frac{1}{r}\sum_{i=1}^{r} \nabla f_i(\boldsymbol{x}) \right\| \leq \frac{2(n-r)}{n} \max_{1 \leq i \leq n} \|\nabla f_i(\boldsymbol{x})\| \tag{4.23}$$

using the triangle inequality. The terms $\|\nabla f_i(\boldsymbol{x})\|$ are bounded if each f_i is Lipschitz or equivalently if f_i is continuous and $\boldsymbol{x}$ is constrained to a compact set. An empirical problem with this setting is presented in Section 4.5.2.3. Our analysis is deterministic, so W is a worst-case bound, but if r is large and the LF subsamples are chosen uniformly at random, it would be reasonable to expect that due to the law of large numbers, the average case behavior is significantly better than our worst-case bounds predict.

Generic Case Finally, we consider the most general case, without assuming specific relationships between the high-fidelity and low-fidelity objectives. By assuming the Lipschitz continuity of both the high-fidelity and low-fidelity objectives, W can be bounded as

$$W = \|f^{\mathrm{HF}}(\boldsymbol{x}) - \rho f^{\mathrm{LF}}(\boldsymbol{x})\|_L \leq \|f^{\mathrm{HF}}(\boldsymbol{x})\|_L + |\rho| \cdot \|f^{\mathrm{LF}}(\boldsymbol{x})\|_L, \tag{4.24}$$

for any choice of ρ where $\|\cdot\|_L$ denotes the Lipschitz constant. The proportionality ρ should **not** be chosen to minimize this bound (since that leads to $\rho = 0$) but can instead be chosen by any

heuristic, such as the one used in control variate techniques where $\rho = -\hat{c}/\hat{v}$ where $\hat{c}$ is an estimate of the covariance between f^{HF} and f^{LF}, and $\hat{v}$ is an estimate of the variance of f^{HF}.

4.4 Bi-Fidelity Line Search with Stochastic Subspace Descent

In applications, we focus on zeroth order optimization, utilizing Stochastic Subspace Descent (SSD) as the implementation method. Following the algorithmic steps introduced in Section 4.3.1, combined with SSD, the entire process is divided into three key components: gradient estimation to construct $\boldsymbol{v}_k$, bi-fidelity surrogate construction, and Armijo backtracking on the surrogate.

Gradient Estimation SSD employs a random projection matrix $\boldsymbol{P}_k \in \mathbb{R}^{D\times\ell}$ with $\ell \ll D$. The random matrix $\boldsymbol{P}_k$ satisfies the properties $\mathbb{E}[\boldsymbol{P}_k\boldsymbol{P}_k^\top] = \boldsymbol{I}_D$ and $\boldsymbol{P}_k^\top\boldsymbol{P}_k = (D/\ell)\boldsymbol{I}_\ell$. A common choice for $\boldsymbol{P}_k$ is based on the Haar measure, where $\boldsymbol{P}_k$ is derived from the Gram-Schmidt orthogonalization of a random Gaussian matrix. The gradient estimation is given by $\boldsymbol{v}_k = \boldsymbol{P}_k\boldsymbol{g}_k$, where $\boldsymbol{g}_k$ is the finite difference estimator of the gradient:

$$\boldsymbol{g}_k \coloneqq \left[\frac{f^{\mathrm{HF}}(\boldsymbol{x}_k+\Delta\boldsymbol{p}_1)-f^{\mathrm{HF}}(\boldsymbol{x}_k)}{\Delta}, \frac{f^{\mathrm{HF}}(\boldsymbol{x}_k+\Delta\boldsymbol{p}_2)-f^{\mathrm{HF}}(\boldsymbol{x}_k)}{\Delta}, \ldots, \frac{f^{\mathrm{HF}}(\boldsymbol{x}_k+\Delta\boldsymbol{p}_\ell)-f^{\mathrm{HF}}(\boldsymbol{x}_k)}{\Delta}\right]^\top, \tag{4.25}$$

where $\Delta \in \mathbb{R}$ is a small step size and $\boldsymbol{p}_i$ is the i-th column of $\boldsymbol{P}_k$. Estimating $\boldsymbol{v}_k$ using Equation (4.25) requires ℓ function evaluations; a more accurate $\mathcal{O}(\Delta^2)$ approximation is also possible at the cost of 2ℓ function evaluations if more than 8 digits of precision are needed. Up to the finite-difference error, $\boldsymbol{g}_k \approx \boldsymbol{P}_k^\top\nabla f^{\mathrm{HF}}(\boldsymbol{x}_k)$ so that $\boldsymbol{v}_k \approx \boldsymbol{P}_k\boldsymbol{P}_k^\top\nabla f^{\mathrm{HF}}(\boldsymbol{x}_k)$, hence $\mathbb{E}[\boldsymbol{v}_k] \approx \nabla f^{\mathrm{HF}}(\boldsymbol{x}_k)$. It is also possible to construct the same estimator without reference to the Haar measure by rewriting $\boldsymbol{v}_k$ as $\boldsymbol{v}_k = \mathrm{proj}_{\mathrm{col}(\boldsymbol{Q}_k)}(\nabla f^{\mathrm{HF}}(\boldsymbol{x}_k))$ where $\boldsymbol{Q}_k \in \mathbb{R}^{D\times\ell}$ is any random matrix with independent columns from an isotropic probability distribution (such as the standard normal).

Surrogate Construction Given the estimated gradient $\boldsymbol{v}_k$ and the current position $\boldsymbol{x}_k$, the goal of surrogate construction is to build φ, denoted as:

$$\tilde{\varphi}_k(\alpha) \coloneqq \rho f^{\mathrm{LF}}(\boldsymbol{x}_k+\alpha\boldsymbol{v}_k) + \tilde{\psi}_k(\alpha). \tag{4.26}$$

The analysis in Section 4.3 assumes that ρ is known and fixed as a constant. However, in practice, ρ is tuned for better performance. In our application, we set $\rho_k = f^{\mathrm{HF}}(\boldsymbol{x}_k)/f^{\mathrm{LF}}(\boldsymbol{x}_k)$. We model $\tilde{\psi}_k$as a piecewise linear function using n_k additional HF evaluations at equispaced points $\{0, \tilde{\alpha}_1, \ldots, \tilde{\alpha}_{n_k} = \alpha_{\max}\}$. Specifically,

$$\tilde{\psi}_k(\alpha) = \frac{h-\alpha}{h}\psi(\tilde{\alpha}_{j-1}) + \frac{\alpha}{h}\psi(\tilde{\alpha}_j), \quad \alpha \in [\tilde{\alpha}_{j-1}, \tilde{\alpha}_j], \quad j = 1, \ldots, n_k, \tag{4.27}$$

where $h = \alpha_{\max}/n_k$. This piecewise linear interpolation is a simple yet effective approach for interpolating φ in 1D space and satisfies the bounds of Lemma 4.3.7 given sufficient n_k. The detailed algorithm is presented in Algorithm 6.

Figure 4.2 illustrates the bi-fidelity backtracking line search process using the example problem in Section 4.5.2.1. The blue curve represents the bi-fidelity surrogate model ($\tilde{\varphi}_k$) approximating the HF function φ (red curve). Rather than performing the line search directly on the computationally expensive HF function (indicated by potential evaluation points as red dots), the method utilizes the surrogate $\tilde{\varphi}_k$ to estimate an optimal step size. While this surrogate is an approximation and may require more surrogate function evaluations during the search itself, it substantially reduces computational cost. In this example, the expense is decreased from 4 HF function calls (for a direct search) to only 1 HF call (to build the surrogate) combined with 6 LF function calls, yielding significant overall savings.

Algorithm 6: Surrogate Construction

Input: $f^{\mathrm{LF}}, f^{\mathrm{HF}}, \boldsymbol{x}_k, \boldsymbol{v}_k, n_k \in \mathbb{N}, \alpha_{\max} > 0$
Output: 1D surrogate $\tilde{\varphi}_k$

1: Define $\{(\tilde{\alpha}_j, \varphi(\tilde{\alpha}_j))\}_{j=0}^{n_k}$ as equispaced points between 0 and $\alpha_{\max}$ (including endpoints), and compute HF evaluations $\varphi(\tilde{\alpha}_j) \leftarrow f^{\mathrm{HF}}(\boldsymbol{x}_k + \tilde{\alpha}_j \boldsymbol{v}_k)$;
2: $\rho_k \leftarrow f^{\mathrm{HF}}(\boldsymbol{x}_k)/f^{\mathrm{LF}}(\boldsymbol{x}_k)$;
3: $\psi(\tilde{\alpha}_j) \leftarrow \varphi(\tilde{\alpha}_j) - \rho_k f^{\mathrm{LF}}(\boldsymbol{x}_k + \tilde{\alpha}_j \boldsymbol{v}_k), \quad j = 1, \ldots, n_k$;
4: Construct piecewise linear function $\tilde{\psi}_k$ using Equation (4.27);
5: Return $\tilde{\varphi}_k$ using Equation (4.26).

Armijo Backtracking on the Surrogate Based on the criteria in Equation (4.6), we set the maximum number of iterations for testing the Armijo condition as $M \in \mathbb{N}$. The detailed procedure is presented in Algorithm 7.

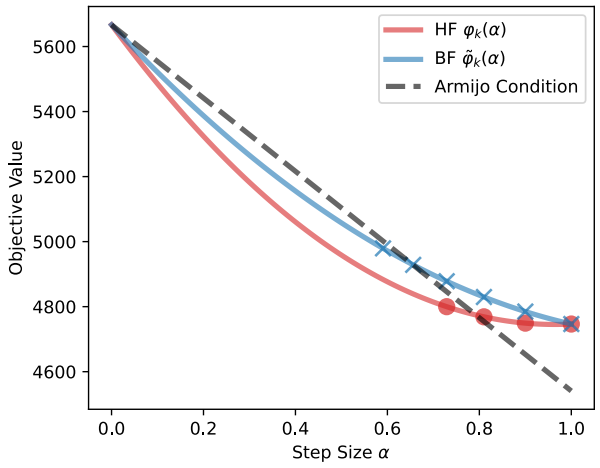


Figure 4.2: Illustration of the bi-fidelity backtracking line search process using the example problem in Section 4.5.2.1. The blue curve represents the bi-fidelity surrogate model ($\tilde{\varphi}_k$) approximating the HF function φ (red curve). It significantly lowers computational cost (e.g., reducing 4 HF calls to 1 HF + 6 LF calls).

Convergence Analysis of SSD with Line Search The convergence results of SSD with line search (on the exact surrogate, $\varphi(\alpha)$) are presented in D.2, under three separate scenarios: strongly convex, convex, and non-convex. The proof shows that in the SSD with line search setting, the value of β can be set as $\ell/2D$.

The proposed bi-fidelity line search algorithm, combined with SSD, will be referred to as Bi-Fidelity SSD (BF-SSD), and is summarized in Algorithm 8. Our theory covers either $\ell = D$ with bi-fidelity linesearch (Thm. 4.3.4) or $1 \leq \ell \leq D$ with HF linesearch (D.2); combining the two analyses is fairly complicated and messy so we do not pursue it.

For practical considerations, as the parameters in Assumption 4.3.3 are often unknown, we set ρ_k as described above, and choose $n_k = 1$ out of simplicity and because it has excellent experimental performance for all of the HF/LF pairs we have examined.

4.5 Empirical Experiments

In this section, we evaluate the proposed BF-SSD Algorithm 8 on four distinct problems: one synthetic optimization problem discussed in Section 4.5.1 and three machine learning-related

Algorithm 7: BF-Backtracking

Input: $\tilde{\varphi}_k, \beta > 0, c \in (0,1), \alpha_{\max} > 0, \boldsymbol{v}_k, M \in \mathbb{N}$ `// typical value of` $c \approx 0.9$
Output: Step size α_k
Initialize $\alpha_k \leftarrow \alpha_{\max}$;
for $m = 0 : M$ **do**
  if $\tilde{\varphi}_k(\alpha_k) \leq f^{\mathrm{HF}}(\boldsymbol{x}_k) - \alpha_k \beta \|\tilde{\boldsymbol{v}}_k\|^2$ **then**
    Break;
  else
    $\alpha_k \leftarrow c\alpha_k$;
  end if
end for
Return α_k;

problems across diverse scenarios presented in Section 4.5.2. These include dual-form kernel ridge regression (Section 4.5.2.1), black-box adversarial attacks (Section 4.5.2.2), and transformer-based black-box language model fine-tuning (soft prompting) in Section 4.5.2.3. We demonstrate that the BF-SSD algorithm consistently outperforms competing methods. To illustrate these advantages, we compare BF-SSD against the following baseline algorithms:

- **Gradient Descent (GD)**: A zeroth-order gradient descent method, where the full-batch gradient is estimated using forward differences, and a fixed step size is used.
- **Coordinate Descent (CD)**: Iteratively optimizes each coordinate individually using finite-difference estimated coordinate gradients.
- **Stochastic Subspace Descent with Fixed Step Size (FS-SSD)**: The standard stochastic subspace descent method, which samples subspaces from the Haar measure and uses a fixed step size.
- **Simultaneous Perturbation Stochastic Approximation (SPSA)**: A randomized optimization method using a Hadamard random variable to estimate the gradient, as proposed by [489] and with step sizes as described in [490]; this is a time-tested, well-respected zeroth-order method.
- **Gaussian Smoothing (GS)**: A method popularized by [386], which is nearly equivalent

Algorithm 8: Bi-Fidelity Line Search SSD Algorithm

Input: $f^{\mathrm{HF}}, f^{\mathrm{LF}}, \ell, c, M, \alpha_{\max}, n$ `// by default,` $n = 1$ `and` $\beta = \ell/2D$

Output: HF minimum value

Initialize $\boldsymbol{x}_0$ and set of HF values $\mathcal{D} = \{f^{\mathrm{HF}}(\boldsymbol{x}_0)\}$
$\beta \leftarrow \ell/2d$;
for $k = 0 : K$ **do**
  Sample random matrix $\boldsymbol{P}_k$;
  Approximate $\tilde{\boldsymbol{v}}_k \approx \boldsymbol{P}_k \boldsymbol{P}_k^T \nabla f(\boldsymbol{x}_k)$ using finite difference (ℓ HF evaluations);
  Normalize $\boldsymbol{v}_k \leftarrow \tilde{\boldsymbol{v}}_k / \|\tilde{\boldsymbol{v}}_k\|$;
  Construct $\tilde{\varphi}_k \leftarrow$ surrogate-construction$(f^{\mathrm{LF}}, f^{\mathrm{HF}}, \boldsymbol{x}_k, \boldsymbol{v}_k, n, \alpha_{\max})$ (n HF evaluations);
  $\alpha_k \leftarrow$ BF-backtracking$(\tilde{\varphi}_k, \beta, c, \alpha_{\max}, \boldsymbol{v}_k, M)$;
  Update $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k - \alpha_k \boldsymbol{v}_k$;
  Evaluate $f^{\mathrm{HF}}(\boldsymbol{x}_{k+1})$ and update $\mathcal{D}$;
end for
Return $\min \mathcal{D}$;

to SSD with $\ell = 1$, and uses a fixed step size.

- **High-Fidelity Stochastic Subspace Descent (HF-SSD)**: A single-fidelity SSD method that utilizes a high-fidelity function for backtracking line search, with its convergence analysis detailed in D.2.

- **Variance-Reduced Stochastic Subspace Descent (VR-SSD)**: A variance-reduced version of the SSD method inspired by SVRG [253], as described in the technical report [284, Section 2.2] of the SSD authors.

- **Bi-fidelity Stochastic Subspace Descent (BF-SSD)**: The proposed method detailed in Section 4.4.

The performance of the optimizers is assessed based on the number of HF objective function evaluations required, accounting for LF calls (in terms of fractional equivalent HF function calls) as appropriate.

4.5.1 Synthetic Problem: Worst Function in the World

In this section we investigate the performance of our proposed BF-SSD algorithm on the "worst function in the world" [385]. With a fixed Lipschitz constant $L > 0$, the function is

$$f(\boldsymbol{x}; r, L) = L\left(\frac{x_1^2 + \sum_{i=1}^{r-1}(x_i - x_{i+1})^2 + x_r^2}{8} - \frac{x_1}{4}\right) - \frac{Lr}{8(r+1)}, \tag{4.28}$$

where x_i denotes the i^{th} entry of the input $\boldsymbol{x}$ and $r < D$ is a constant integer defining the intrinsic dimension of the problem. The function is convex with global minimum value 0. The Lipschitz constant of the gradient of this function is L. Nesterov has shown a wide ranges of iterative first-order method that performs poorly when minimizing $f(\boldsymbol{x}; r, L)$ with initial point $\boldsymbol{x}_0 = \mathbf{0}$.

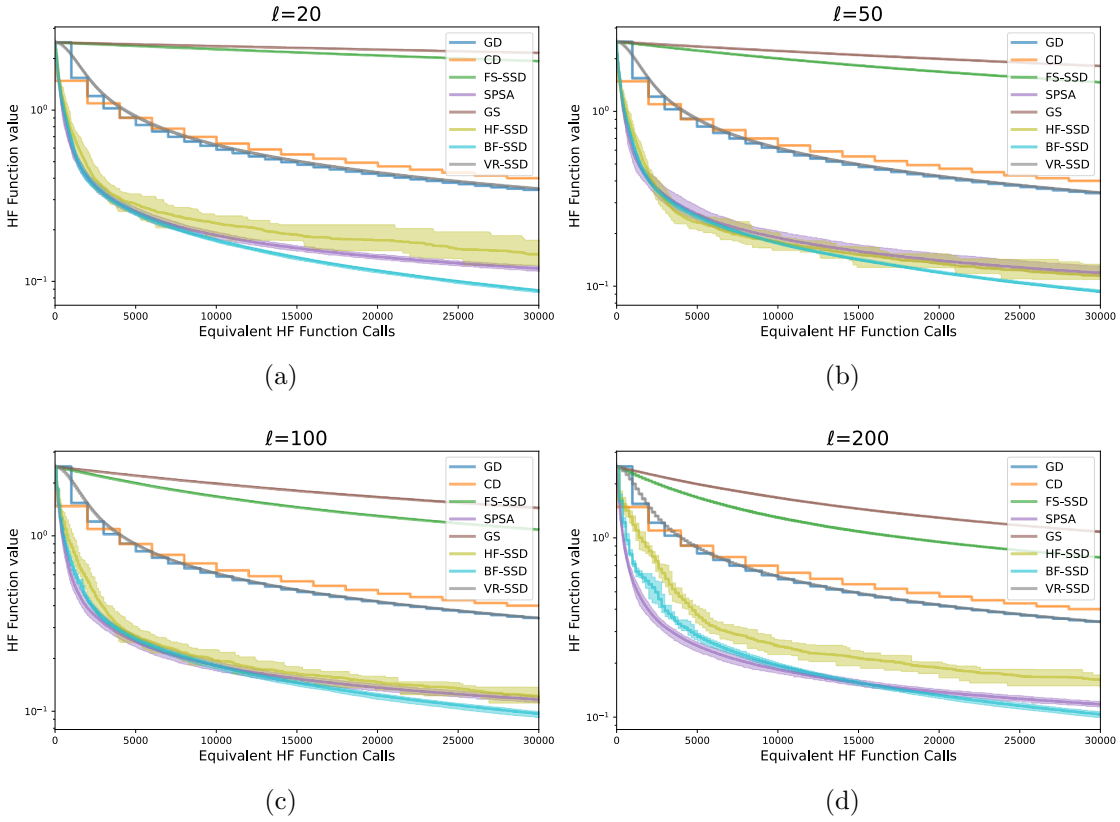


Figure 4.3: The convergence performance for different optimizers. The x-axis is the equivalent number of HF function evaluations, and the y-axis is the HF function evaluation value at the current stage. We investigate the results when $\ell = 20, 50, 100, 200$ with $r_L = 2, r_H = 100$. The corresponding results are presented with their titles indicating the specific choices. The shadow regions are the area between the best and the worst behavior by 10 trials.

We set the dimension $d = 1{,}000$, $\ell = 20$, and $L = 20$. The intrinsic dimension of the HF function is r_H and of the LF function is r_L. We choose $r_L \ll r_H$ and assume the computational cost ratio between HF and LF evaluations is $r_H : r_L$. For Gradient Descent, we choose the standard step size of $1/L = 0.05$ and for the GS and SSD-based methods the step size is $\ell/(LD)$. The backtracking parameter is $\beta = \ell/(2D)$. The hyperparameter study is conducted according to different values of $c \in \{0.8, 0.9, 0.99\}$ and $\ell \in \{5, 10, 20\}$. All the experiments are repeated 10 times with shaded regions denoting the worst and the best performance over 10 trials.

Figure 4.3 illustrates the performance of various optimizers across different values of ℓ. Detailed results for $\ell = 20$ and $c = 0.99$ at N from 500 to 8,000 are presented in Table 4.1, while additional comparisons across different ℓ and c configurations are included in Table D.1. These results show that BF-SSD consistently outperforms the other optimizers in most scenarios. For different SSD methods, the effect of ℓ on the final performance varies. Large values of ℓ improve the optimization results for FS-SSD and VR-SSD, while HF-SSD and BF-SSD prefer relatively smaller ℓ, as highlighted in Table 4.2.

	Equivalent HF function evaluations N				
Method	$N = 100$	$N = 1000$	$N = 10000$	$N = 20000$	$N = 30000$
GD	2.48 ± 0.00	2.48 ± 0.00	0.62 ± 0.00	0.43 ± 0.00	0.34 ± 0.00
CD	$\mathbf{1.48 \pm 0.00}$	1.48 ± 0.00	0.70 ± 0.00	0.49 ± 0.00	0.40 ± 0.00
FS-SSD	2.47 ± 0.00	2.45 ± 0.00	2.26 ± 0.00	2.08 ± 0.00	1.92 ± 0.00
SPSA	1.95 ± 0.05	$\mathbf{0.61 \pm 0.04}$	0.19 ± 0.00	0.14 ± 0.00	0.12 ± 0.00
GS	2.47 ± 0.00	2.46 ± 0.00	2.36 ± 0.00	2.25 ± 0.00	2.15 ± 0.00
HF-SSD	2.07 ± 0.15	0.77 ± 0.06	0.22 ± 0.02	0.17 ± 0.02	0.14 ± 0.02
BF-SSD	2.00 ± 0.05	0.66 ± 0.03	$\mathbf{0.17 \pm 0.00}$	$\mathbf{0.11 \pm 0.00}$	$\mathbf{0.09 \pm 0.00}$
VR-SSD	2.47 ± 0.00	2.10 ± 0.02	0.63 ± 0.01	0.43 ± 0.00	0.35 ± 0.00

Table 4.1: Performance values (mean $\pm$ std over 10 runs) showing the objective function for different optimization methods at various HF function evaluations N with $\ell = 20, c = 0.99$. The minimum values in each **column** are highlighted in bold.

Table 4.2: Comparison of SSD methods for different values of ℓ (Mean $\pm$ Std at $N = 20,000$). Bold values indicate the minimum mean for each SSD method, i.e., across each **row**.

Method	$\ell = 20$	$\ell = 50$	$\ell = 100$	$\ell = 200$
FS-SSD	2.0766 ± 0.0038	1.6726 ± 0.0057	1.2943 ± 0.0047	$\mathbf{0.9447 \pm 0.0041}$
HF-SSD	0.1745 ± 0.0209	$\mathbf{0.1357 \pm 0.0073}$	0.1482 ± 0.0067	0.1893 ± 0.0115
BF-SSD	$\mathbf{0.1149 \pm 0.0016}$	0.1206 ± 0.0015	0.1236 ± 0.0028	0.1329 ± 0.0024
VR-SSD	0.4328 ± 0.0030	0.4268 ± 0.0023	0.4232 ± 0.0018	$\mathbf{0.4227 \pm 0.0016}$

4.5.2 Zero-th Order Optimization for Machine Learning Problems

In this section, we present the BF-SSD optimization results with other completing methods under the machine learning-related zero-th order optimization settings. Besides showing the advantages of the BF-SSD, we also show that it is often convenient to design a cheap LF model in many machine learning problems that can be leveraged to accelerate the convergence.

4.5.2.1 Dual Form of Kernel Ridge Regression

Consider a kernel ridge regression problem as follows. By the representer theorem, given datapoints $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^D$ and a kernel function $\kappa : \mathbb{R}^{\tilde{m}} \times \mathbb{R}^{\tilde{m}} \to \mathbb{R}$, the goal is to find the coefficients $\tilde{\boldsymbol{\alpha}}$ such that

$$f_{\text{predict}}(\boldsymbol{x}) = \sum_{i=1}^{D} \tilde{\alpha}_i k(\boldsymbol{x}, \boldsymbol{x}_i). \tag{4.29}$$

One way to solve the coefficients is to solve the dual form of the kernel ridge regression,

$$\tilde{\boldsymbol{\alpha}}^* = \arg\min_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha} - 2\langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle + \lambda \|\boldsymbol{\alpha}\|^2, \tag{4.30}$$

where $\boldsymbol{K}$ is the kernel matrix with $[\boldsymbol{K}]_{i,j} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $[\boldsymbol{y}]_i = y_i$, and λ is a positive scalar denoting the regularization parameter. The solution of Equation (4.30) can be explicitly represented as

$$\tilde{\boldsymbol{\alpha}}^* = (\boldsymbol{K} + \lambda \boldsymbol{I})^{-1} \boldsymbol{y}. \tag{4.31}$$

However, solving the explicit solution involves inverting the matrix $\boldsymbol{K} + \lambda \boldsymbol{I}$, which takes $\mathcal{O}(D^3)$ and can be extremely expensive when D is large. In fact, when D is sufficiently large, evaluating the function in Equation (4.29) takes $\mathcal{O}(D^2)$ and becomes expensive.

Therefore, an alternative approach to solve this problem is to build a low-rank surrogate for the kernel matrix $\boldsymbol{K}$. One of the mostly used approach is the Nystroem method, which finds a subset $\mathcal{S} \subset [1, \ldots, D]$ with size $l \ll D$ and builds the kernel surrogate $\tilde{\boldsymbol{K}} = \boldsymbol{K}[:, \mathcal{S}](\boldsymbol{K}[\mathcal{S}, \mathcal{S}])^{-1}\boldsymbol{K}[\mathcal{S}, :]$. By implementing Nystroem method, the complexity of evaluating objective function is reduced to $\mathcal{O}(lD)$, which can be much cheaper as $l \ll D$. We let the low-rank surrogate model using Nystroem method as LF function. Therefore, the ratio of computational cost between HF and LF function evaluation is D/l.

We consider the problem is a black-box format: the accesses to the HF function are presented as a form of API, which means some parts of the objective function in Equation (4.30), e.g. $\boldsymbol{y}$ and/or $\boldsymbol{K}$, are concealed, so that the derivative is unavailable. In this specific case, we assume the values of $\boldsymbol{y}$ and $\boldsymbol{K}$ are unavailable due to privacy reason.

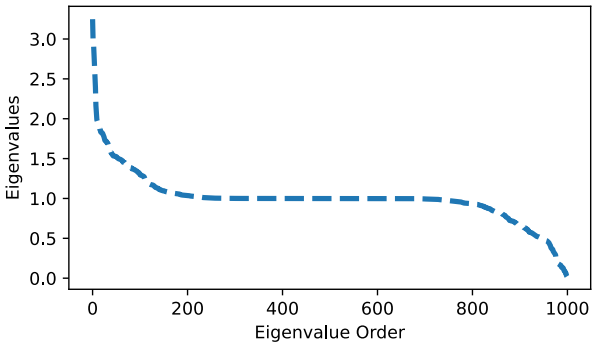
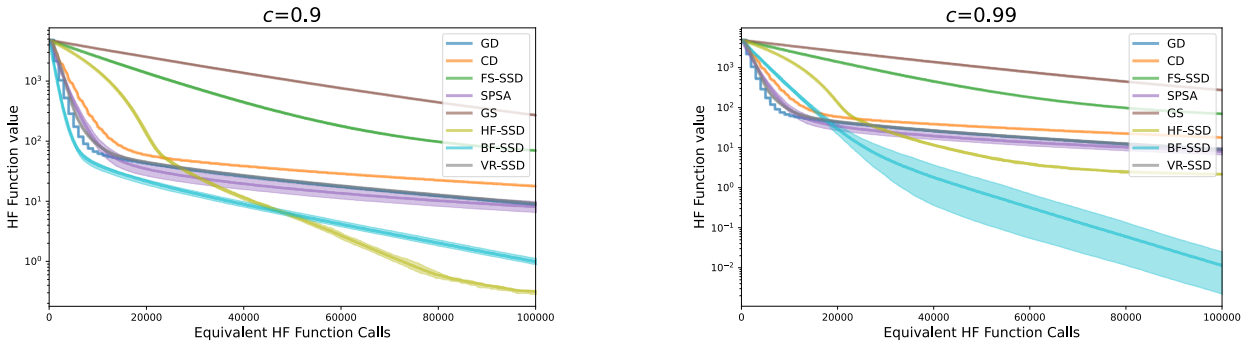


Figure 4.4: The eigenvalues of the kernel matrix implemented in Equation (4.30).

For the regression data, we select the first $D = 1{,}000$ samples from the California housing dataset provided in the scikit-learn library [422]. We use a Gaussian (RBF) kernel with lengthscale 1.0 to generate the corresponding kernel matrix $\boldsymbol{K}$. Figure 4.4 shows the decay of eigenvalues for $\boldsymbol{K}$, with a rapid drop, especially within the first 100 eigenvalues, due to the Gaussian kernel's properties. This fast decay motivates our focus on cases where the values of ℓ are below 100. The starting point $\boldsymbol{x}_0$ is set at the origin $\mathbf{0}$.

The results of kernel ridge regression are shown in Figure 4.5, with values of c chosen from 0.9 and 0.99, and value of ℓ is fixed as 100. According to these results, BF-SSD shows advantages



(a) (b)

Figure 4.5: Similar with Figure 4.3, we compare the optimizer performances with varying parameters $\ell = 10, 50, 100$ and c=0.9, 0.95, 0.99. The corresponding results are presented with their titles indicating the specific choices. The shadow regions are the area between the best and the worst behavior by 10 trials.

over other methods except in Figure 4.5a. When the backtracking factor c decreases, the step sizes determined by the backtracking method become more conservative, leading to suboptimal results, especially for BF-SSD. We also implement different combinations of c and ℓ and collect the SSD performances in Table 4.3. The results suggest that BF-SSD show great advantages over other methods for the most of cases, and larger values of ℓ and c improve the performances of BF-SSD.

c	ℓ	**FS-SSD**	**HF-SSD**	**VR-SSD**	**BF-SSD**
0.9	10	3497.69 ± 3.56	$\mathbf{8.98 \pm 0.43}$	23.65 ± 0.83	$\mathbf{8.77 \pm 0.72}$
	50	1016.43 ± 9.26	$\mathbf{3.96 \pm 0.17}$	22.19 ± 0.66	6.49 ± 0.51
	100	268.71 ± 3.32	$\mathbf{5.70 \pm 0.26}$	21.81 ± 0.71	$\mathbf{6.01 \pm 0.36}$
0.95	10	3499.15 ± 6.41	19.17 ± 0.99	23.39 ± 0.72	$\mathbf{3.21 \pm 0.40}$
	50	1009.36 ± 9.14	5.52 ± 0.13	21.46 ± 0.55	$\mathbf{2.42 \pm 0.38}$
	100	269.68 ± 2.79	6.07 ± 0.23	21.32 ± 1.09	$\mathbf{2.06 \pm 0.27}$
0.99	10	3499.32 ± 7.90	30.18 ± 1.08	23.04 ± 0.91	$\mathbf{1.59 \pm 0.74}$
	50	1010.93 ± 6.68	6.94 ± 0.21	21.94 ± 0.50	$\mathbf{0.88 \pm 0.49}$
	100	270.82 ± 4.04	6.17 ± 0.19	21.53 ± 0.75	$\mathbf{0.75 \pm 0.38}$

Table 4.3: Black-box kernel ridge regression HF function values (mean $\pm$ std) for FS-SSD, HF-SSD, VR-SSD, and BF-SSD at various combinations of ℓ and c at $N = 50,000$. Considering uncertainties, the minimum values in each row are highlighted in bold.

4.5.2.2 Black-box Adversarial Attack on MNIST

In practice, especially in the area of explainable AI (XAI), researchers have found that many deep learning models are not robust towards noisy data. Specifically, if test data is contaminated with a small noisy perturbation that is imperceptible to humans, many previously well-performing deep learning models fail to produce reasonable results. Generating such biased noise to confuse a trained neural network model is an interesting topic, which is usually referred as an "attack" in the domain of adversarially robust training. This need not be a "black hat" activity, as it can be used as part of hardening a system in order to prevent these attacks in the future.

There are primarily two types of attacks: one is white-box attack, in which we have knowledge of the model and generating the corresponding adversarial samples to confuse the given model. The standard approach under this scenario is to generate the shift in pixel space based on the gradient of loss function in order to **maximize** the loss. The other type of attack is called black-box attack, in which one does not have knowledge of the trained model and would like to generate adversarial data from it. The black-box scenario is more difficult due to the missing knowledge and one way to solve it is to treat this problem as a black box optimization. To generate an adversarial sample for the given datapoint $\boldsymbol{x}^\dagger \in \mathbb{R}^D$, with D representing the number of pixels in the given image, a common formulation of the adversarial attack is to find a noise sample $\boldsymbol{x}^*$ solving

$$\boldsymbol{x}^* = \underset{\|\boldsymbol{x}\| \leq \varepsilon}{\arg\max}\, \mathcal{L}\left(g(\boldsymbol{x}+\boldsymbol{x}^\dagger), y^\dagger\right), \tag{4.32}$$

where $y^\dagger$ is the correct label of $\boldsymbol{x}^\dagger$, $\mathcal{L}$ is the attack loss function, and g represents the model for attack. Following the adversarial attack paradigm of [80] and its black-box extension [89], we use a soft version of the given optimization problem as follows:

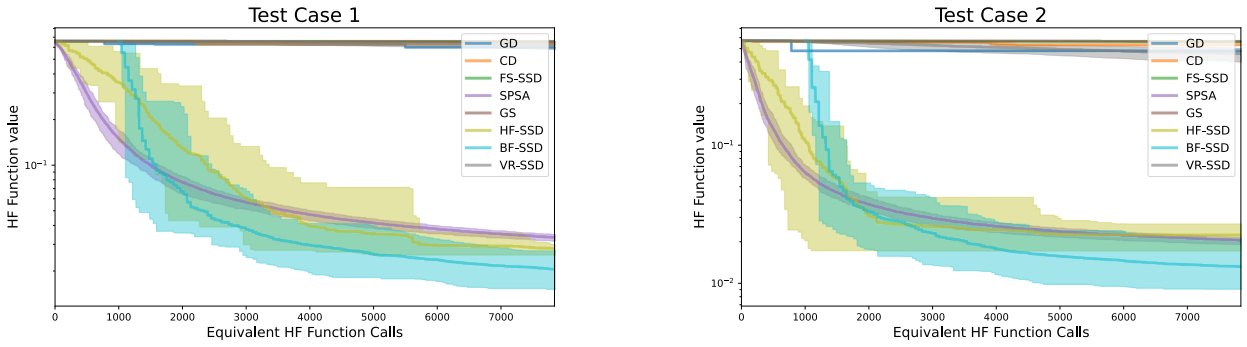
$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\arg\min} -\mathrm{CE}\left(g(\boldsymbol{x}+\boldsymbol{x}^\dagger), y^\dagger\right) + \tilde{\tau}\|\boldsymbol{x}\|^2, \tag{4.33}$$

where cross entropy loss CE is assigned as the attack loss and $g(\cdot)$ outputs the probabilities of different classes, usually using a softmax function for normalization. $\tilde{\tau}$ is a variable balancing the attack loss function CE and the attack norm. The goal of the optimization is to find a small shift

$\boldsymbol{x}$ in pixel space so that the output results are greatly changed.

In this study, we utilized two convolutional neural network (CNN) architectures to model the HF and LF representations for classification tasks on the MNIST dataset with 60,000 training data and 10,000 testing data. The HF model was a deeper CNN consisting of two convolutional layers, the first with 32 filters and the second with 64 filters, both using 5×5 kernels, followed by ReLU activations and 2×2 max-pooling. The flattened output from the convolutional layers ($7 \times 7 \times 64$) was connected to a fully connected layer with 1024 neurons, followed by a 10-class output layer. In contrast, the LF model employed a simplified architecture with a single convolutional layer containing 2 filters and a 3×3 kernel, followed by ReLU activation and 2×2 max-pooling. The output ($13 \times 13 \times 2$) was flattened and passed through a fully connected layer with 16 neurons, leading to a 10-class output layer with log-softmax activation. The HF model was designed to provide high-capacity representations, while the LF model served as a lightweight alternative for computational efficiency. The LF model was trained using knowledge distillation [232], leveraging only 1000 training samples and 1000 evaluations of the HF function. Knowledge distillation is a technique where a smaller, simpler model (the student) learns to replicate the outputs of a larger, more complex model (the teacher), effectively transferring knowledge while reducing computational costs. The classification accuracy for the HF and LF CNN are 99.02% and 82.21%, respectively. There are 27,562 parameters for the LF CNN and 3,274,634 parameters for the HF CNN. We estimate the ratio between HF and LF computational costs as $3274634/27562 \approx 118.8$. The images in MNIST dataset are 28×28 with single channel, hence the dimension is $D = 784$. The starting points are initialized as the origin point for all experiments.

Figure 4.6 illustrates the optimization convergence of various zeroth order methods on two test images. For the SSD methods (including the line search version), the parameters are set to $\ell = 50$ and $\alpha_{\max} = 2.0$. Since BF-SSD uses 1,000 HF evaluations for knowledge distillation training, it begins at $N = 1{,}000$. The convergence results demonstrate that HF-SSD outperforms other methods in this task. Additionally, HF-SSD, BF-SSD, and SPSA exhibit clear advantages over other methods, underscoring the importance of tuning suitable step sizes for the optimization

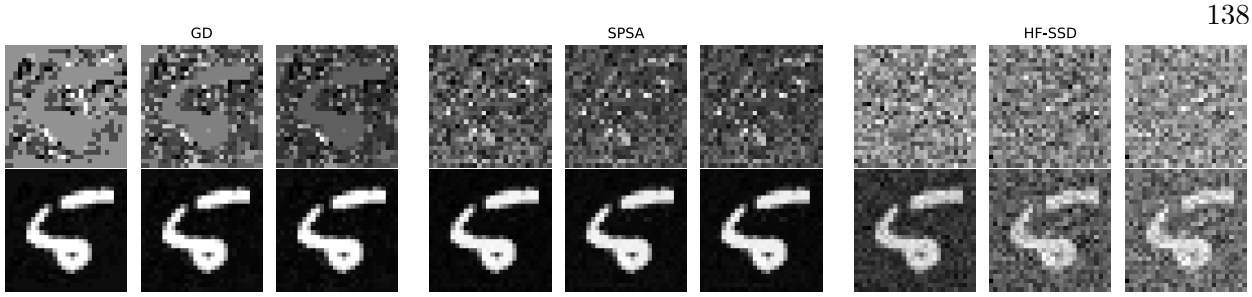


(a) (b)

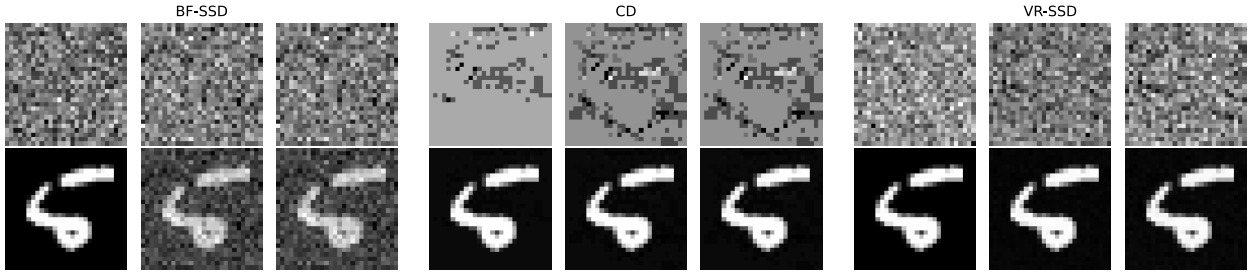
Figure 4.6: Optimization performances according to different attack targets. The images and their attack noises are presented in Figure 4.7.

process.

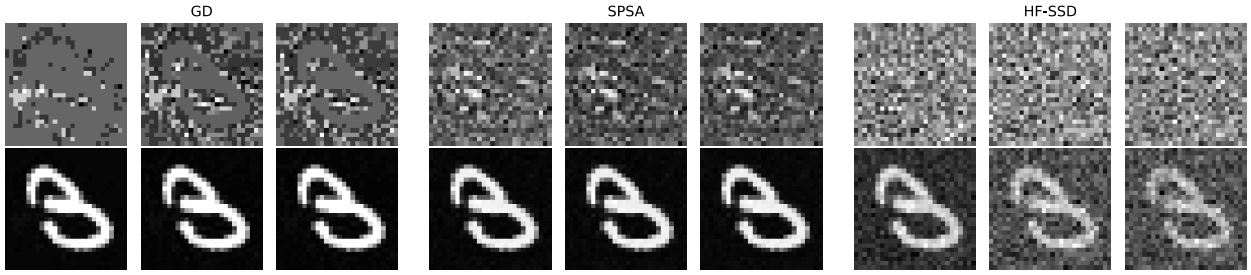
In Figure 4.7, we present the adversarially attacked test images generated by different optimization approaches for $N = 2{,}000$, 5,000, and 7,000. For the first test image (a-f), only HF-SSD, BF-SSD, and SPSA successfully flip the output of the HF model under limited HF evaluations ($N \leq 7{,}000$). Similarly, for the second test image (g-l), HF-SSD, BF-SSD, SPSA, and VR-SSD succeed in flipping the HF model output. However, in both cases, we observe that HF-SSD (Figure 4.7c and Figure 4.7i) and BF-SSD (Figure 4.7d and Figure 4.7j) tend to blur the images more than SPSA (Figure 4.7b and Figure 4.7h). This behavior may result from differences in sampling strategies, such as Haar measure sampling versus Hadamard sampling. From an adversarial attack perspective, a successful attack should flip the model's output without excessively blurring the image. In this regard, SPSA performs better compared to HF-SSD and BF-SSD, despite its loss function remaining higher than the other two methods. Since SPSA performs visually better despite a higher objective function, the problem is not with BF-SSD per se, but instead related to the formulation of the objective function, and will require further investigation in future work.



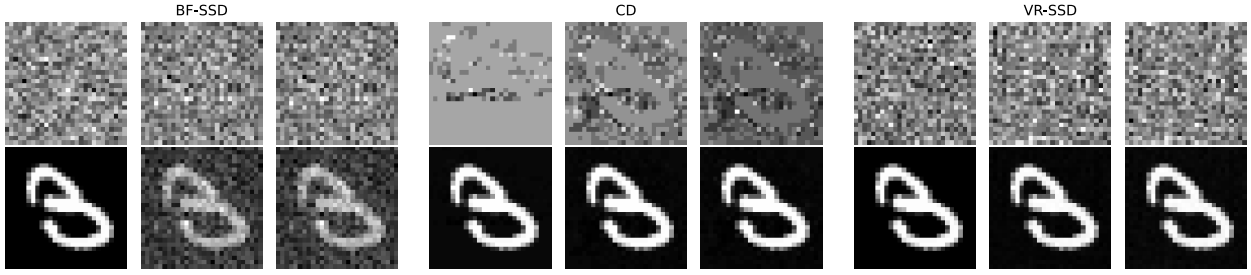
(a) GD (Label=5, Predict=5) (b) SPSA (Label=5, Predict=6) (c) HF-SSD (Label=5, Predict=6)



(d) BF-SSD (Label=5, Predict=6) (e) CD (Label=5, Predict=5) (f) VR-SSD (Label=5, Predict=5)



(g) GD (Label=3, Predict=3) (h) SPSA (Label=3, Predict=8) (i) HF-SSD (Label=3, Predict=8)



(j) BF-SSD (Label=3, Predict=8) (k) CD (Label=3, Predict=3) (l) VR-SSD (Label=3, Predict=8)

Figure 4.7: Adversarial examples for idx = 8 (top two rows) and idx = 18 (bottom two rows) using different methods.

4.5.2.3 Soft Prompting Black-box Language Model

Fine-tuning pre-trained models like BERT or GPT has become a cornerstone of modern natural language processing (NLP). These models, trained on massive corpora, achieve state-of-the-art performance across a wide range of downstream tasks when adapted using task-specific fine-tuning. However, traditional fine-tuning involves updating millions or even billions of parameters, making it computationally expensive and prone to overfitting, especially in low-resource settings. To address these challenges, soft prompting has emerged as a lightweight and efficient alternative. Instead of modifying the model's internal parameters, soft prompting introduces learnable embeddings (soft prompts) that are prepended to the input sequence, enabling task adaptation with minimal computational cost. This approach is particularly appealing for tasks requiring minimal intervention in the model's architecture while leveraging its pre-trained knowledge.

Despite the efficiency of soft prompting, its practical applicability faces challenges when dealing with black-box models where gradients with respect to the model parameters are inaccessible. For instance, many commercial APIs or proprietary models only provide access to predictions or loss values, making gradient-based optimization infeasible. In such scenarios, zeroth order optimization becomes a crucial tool. Specifically, in this section, we consider a black-box pre-trained language classifier $f_c : \mathbb{R}^{L_t \times 768} \to [0, 1]$, a pre-trained tokenizer $f_t : \text{str} \to \mathbb{R}^{L_t \times 768}$, where `str` is any string of arbitrary length, and the sequence length L_t is a positive integer up to 512 representing the length of the embedding. The goal is to find a soft prompt $\boldsymbol{x}^* \in \mathbb{R}^{768}$ such that

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \mathbb{E}_{(\boldsymbol{z},y)} \big[\text{CE}(f_c(\text{cat}[\boldsymbol{x}, f_t(\boldsymbol{z})]), y)\big], \tag{4.34}$$

where $\text{CE}(\cdot, \cdot)$ is the cross-entropy loss function, and the dataset $(\boldsymbol{z}, y) \in \texttt{str} \times \{0, 1\}$. This particular formulation addresses a binary sentiment analysis task where a given string is classified as expressing a positive (1) or negative (0) sentiment. Since the classifier f_c is pre-trained and treated as a black box, gradient information for the loss function is unavailable, necessitating the use of zeroth order optimization to solve the problem.

For the pre-trained classifier and tokenizer, we employed the BERT model, a state-of-the-art

transformer-based architecture trained on large corpora. Specifically, we focused on a simplified version of BERT, named `DistilBERT`. Sentiment analysis on the `aclImdb` dataset was used as a soft prompting task. This dataset comprises movie reviews categorized into positive and negative sentiments, forming a binary classification problem. A $D = 768$-dimensional soft prompt $\boldsymbol{x}$ is considered as the input. The transformer's parameters were kept frozen to focus optimization on the soft prompt, reducing the degrees of freedom and computational overhead. The HF model, as described in Equation (4.34), was evaluated using 1,000 samples from `aclImdb` to approximate the expectation, while the LF model leveraged only 10 samples that randomly selected from them. Consequently, the evaluation cost ratio between HF and LF was 100:1.

We set the initial starting point at the origin. We let $\ell = 50$, $c = 0.99$, and for the non-line search methods we chose a fixed step size of 1×10^{-2}. Additionally, we ran the Adam optimizer (using gradients) to solve the same problem using 500 epochs to generate a reference error. The y-axis in the following figure represents the relative error compared with the Adam optimizer. Figure 4.8 illustrates the performances of different competing methods. The BF-SSD demonstrates significant advantages compared with others, highlighting its efficiency in the optimization.

4.6 Conclusion

In this paper, we introduced Bi-fidelity Stochastic Subspace Descent (BF-SSD), a novel zeroth-order optimization algorithm designed for computationally expensive black-box problems. BF-SSD constructs a bi-fidelity surrogate model using both high-fidelity (HF) and low-fidelity (LF) function evaluations, enabling an efficient backtracking line search on the surrogate to determine step sizes. This approach significantly reduces the required number of expensive HF evaluations while maintaining theoretical convergence guarantees under certain assumptions. We demonstrated the effectiveness of BF-SSD on diverse tasks, including a synthetic benchmark, dual-form kernel ridge regression, black-box adversarial attacks, and transformer-based language model fine-tuning. Our results show consistent outperformance against methods like gradient descent, coordinate descent, SPSA, and HF-SSD, particularly in terms of solution quality achieved per HF evaluation.

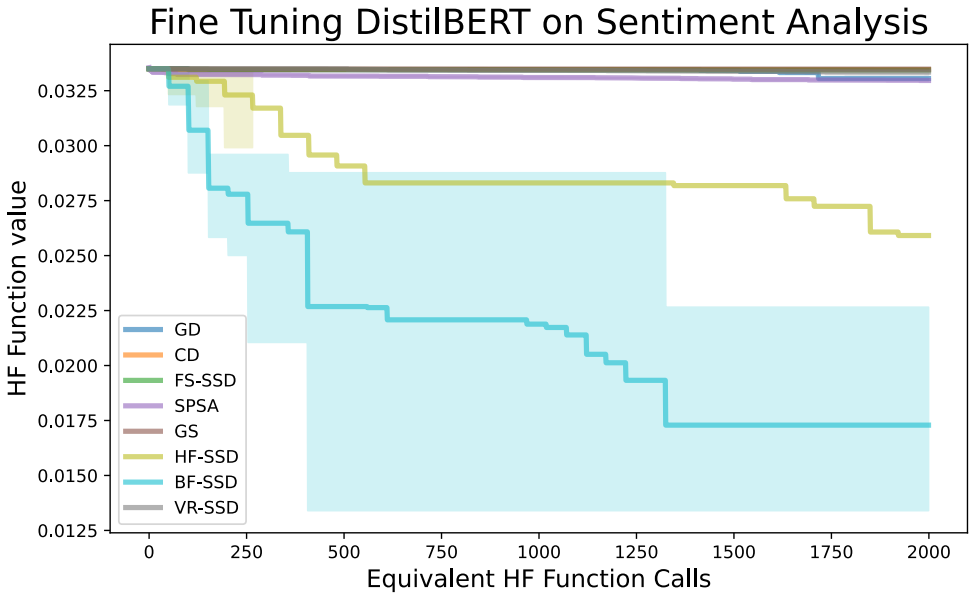


Figure 4.8: Relative errors with respect to Adam optimization using 500 epochs of zero-th order optimizers.

These findings highlight the effectiveness of the bi-fidelity strategy within a stochastic subspace descent framework for tackling large-scale, high-dimensional optimization challenges, positioning BF-SSD as a promising and computationally efficient tool for various real-world applications.

Bibliography

[1] Bittner 2020 dataset. https://schlieplab.org/Static/Supplements/CompCancer/CDNA/bittner-2000/.

[2] Metagenes and molecular pattern discovery using matrix factorization. https://www.pnas.org/content/101/12/4164.

[3] Nimfa Python library. http://nimfa.biolab.si/.

[4] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. {TensorFlow}: a system for {Large-Scale} machine learning. In 12th USENIX symposium on operating systems design and implementation (OSDI 16), pages 265–283, 2016.

[5] Ben Adcock. Infinite-dimensional compressed sensing and function interpolation. Foundations of Computational Mathematics, 18(3):661–701, 2018.

[6] Ben Adcock, Simone Brugiapaglia, and Clayton G Webster. Sparse Polynomial Approximation of High-Dimensional Functions, volume 25. SIAM, 2022.

[7] Sergios Agapiou, Omiros Papaspiliopoulos, Daniel Sanz-Alonso, and Andrew M Stuart. Importance sampling: Intrinsic dimension and computational cost. Statistical Science, pages 405–431, 2017.

[8] Kareem S. Aggour, Alex Gittens, and Bülent Yener. Adaptive Sketching for Fast and Convergent Canonical Polyadic Decomposition. In International Conference on Machine Learning. PMLR, 2020.

[9] Salman Ahmadi-Asl, Andrzej Cichocki, Anh Huy Phan, Maame G. Asante-Mensah, Farid Mousavi, Ivan Oseledets, and Toshihisa Tanaka. Randomized algorithms for fast computation of low-rank tensor ring model. Machine Learning: Science and Technology, 2020.

[10] Nir Ailon and Bernard Chazelle. The fast Johnson–Lindenstrauss transform and approximate nearest neighbors. SIAM Journal on Computing, 39(1):302–322, 2009.

[11] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. arXiv preprint arXiv:1612.00410, 2016.

[12] aleskars. What's the difference between all the brands rtx3080, September 2020.

[13] Forrester Alexander I.J, Sóbester András, and Keane Andy J. Multi-fidelity optimization via surrogate modelling. Proc. R. Soc., 2007.

[14] Bo O Almroth, Perry Stern, and Frank A Brogan. Automatic choice of global shape functions in structural analysis. Aiaa Journal, 16(5):525–528, 1978.

[15] Hannah Alsdurf, Yoshua Bengio, Tristan Deleu, Prateek Gupta, Daphne Ippolito, Richard Janda, Max Jarvie, Tyler Kolody, Sekoul Krastev, and Tegan Maharaj. Covi white paper. arXiv preprint arXiv:2005.08502, 2020.

[16] Terrence Alsup and Benjamin Peherstorfer. Context-aware surrogate modeling for balancing approximation and sampling costs in multifidelity importance sampling and bayesian inverse problems. SIAM/ASA Journal on Uncertainty Quantification, 11(1):285–319, 2023.

[17] Konstantin Althaus, Iason Papaioannou, and Elisabeth Ullmann. Consensus-based rare event estimation. SIAM Journal on Scientific Computing, 46(3):A1487–A1513, 2024.

[18] Jason Altschuler, Aditya Bhaskara, Gang Fu, Vahab Mirrokni, Afshin Rostamizadeh, and Morteza Zadimoghaddam. Greedy Column Subset Selection: New Bounds and Distributed Algorithms. In International Conference on Machine Learning, pages 2539–2548, June 2016.

[19] Maliheh Aramon, Gili Rosenberg, Elisabetta Valiante, Toshiyuki Miyazawa, Hirotaka Tamura, and Helmut G. Katzgraber. Physics-inspired optimization for quadratic unconstrained problems using a digital annealer. Frontiers in Physics, 7:48, 2019.

[20] Maliheh Aramon, Gili Rosenberg, Elisabetta Valiante, Toshiyuki Miyazawa, Hirotaka Tamura, and Helmut G. Katzgraber. Physics-inspired optimization for quadratic unconstrained problems using a digital annealer. Frontiers in Physics, 7:48, 2019.

[21] Søren Asmussen and Peter W Glynn. Stochastic simulation: algorithms and analysis, volume 57. Springer, 2007.

[22] Kendall Atkinson and Weimin Han. Theoretical Numerical Analysis: A Functional Analysis Framework. Number 39 in Texts in Applied Mathematics. Springer-Verlag, New York, 3rd edition, 2009.

[23] Kendall E. Atkinson. An Introduction to Numerical Analysis. John Wiley & Sons, second edition, 1989.

[24] Siu-Kui Au and James L Beck. Estimation of small failure probabilities in high dimensions by subset simulation. Probabilistic engineering mechanics, 16(4):263–277, 2001.

[25] Woody Austin, Grey Ballard, and Tamara G. Kolda. Parallel tensor compression for large-scale scientific data. In 2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pages 912–922. IEEE, 2016.

[26] Haim Avron, Petar Maymounkov, and Sivan Toledo. Blendenpik: Supercharging LAPACK's least-squares solver. SIAM Journal on Scientific Computing, 32(3):1217–1236, 2010.

[27] Haim Avron, Huy L. Nguyen, and David P. Woodruff. Subspace Embeddings for the Polynomial Kernel. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, pages 2258–2266, Cambridge, MA, USA, 2014. MIT Press.

[28] Ivo Babuška, Fabio Nobile, and Raúl Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. SIAM Journal on Numerical Analysis, 45(3):1005–1034, 2007.

[29] Markus Bachmayr and Albert Cohen. Kolmogorov widths and low-rank approximations of parametric elliptic pdes, 2015.

[30] Brett W. Bader and Tamara G. Kolda. Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. ACM Transactions on Mathematical Software (TOMS), 32(4):635–653, 2006.

[31] Brett W. Bader, Tamara G. Kolda, and others. MATLAB Tensor Toolbox, Version 2.6. Available online at https://www.tensortoolbox.org, 2015.

[32] Raj Raghu Bahadur. A Representation of the Joint Distribution of Responses to n Dichotomous Items. In Studies in Item Analysis and Prediction, pages 158–168. Stanford University Press, Stanford, California, 1961.

[33] G. Ballard, A. Benson, A. Druinsky, B. Lipshitz, and O. Schwartz. Improving the Numerical Stability of Fast Matrix Multiplication. SIAM Journal on Matrix Analysis and Applications, 37(4):1382–1418, January 2016.

[34] Eduard Bartl, Radim Belohlavek, Petr Osicka, and Hana Řezanková. Dimensionality reduction in boolean data: Comparison of four bmf methods. In International Workshop on Clustering High-Dimensional Data, pages 118–133. Springer, 2012.

[35] Jonathan Barzilai and Jonathan Borwein. Two-point step size gradient methods. IMA Journal of Numerical Analysis, 8(1):141–148, 01 1988.

[36] Muthu Baskaran, Benoît Meister, Nicolas Vasilache, and Richard Lethin. Efficient and scalable computations with sparse tensors. In 2012 IEEE Conference on High Performance Extreme Computing, pages 1–6. IEEE, 2012.

[37] Casey Battaglino, Grey Ballard, and Tamara G. Kolda. A practical randomized CP tensor decomposition. SIAM Journal on Matrix Analysis and Applications, 39(2):876–901, 2018.

[38] Christian Bauckhage, E. Brito, K. Cvejoski, C. Ojeda, Rafet Sifa, and S. Wrobel. Ising Models for Binary Clustering via Adiabatic Quantum Computing. In Marcello Pelillo and Edwin Hancock, editors, Energy Minimization Methods in Computer Vision and Pattern Recognition, Lecture Notes in Computer Science, pages 3–17, Cham, 2018. Springer International Publishing.

[39] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM journal on imaging sciences, 2(1):183–202, 2009.

[40] Johann A. Bengua, Ho N. Phien, and Hoang D. Tuan. Optimal feature extraction and classification of tensors via matrix product state decomposition. In 2015 IEEE International Congress on Big Data, pages 669–672. IEEE, 2015.

[41] Austin R. Benson and Grey Ballard. A framework for practical parallel fast matrix multiplication. In ACM SIGPLAN Notices, volume 50, pages 42–53. ACM, 2015.

[42] Tanya Y. Berger-Wolf and Jared Saia. A framework for analysis of dynamic social networks. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 523–528. ACM, 2006.

[43] Michael W. Berry, Shakhina A. Pulatova, and G. W. Stewart. Algorithm 844: Computing sparse reduced-rank approximations to sparse matrices. ACM Transactions on Mathematical Software (TOMS), 31(2):252–269, 2005.

[44] Gregory Beylkin and Martin J. Mohlenkamp. Numerical operator calculus in higher dimensions. Proceedings of the National Academy of Sciences, 99(16):10246–10251, 2002.

[45] Gregory Beylkin and Martin J. Mohlenkamp. Algorithms for Numerical Analysis in High Dimensions. SIAM Journal on Scientific Computing, 26(6):2133–2159, July 2006.

[46] Aditya Bhaskara, Moses Charikar, and Aravindan Vijayaraghavan. Uniqueness of tensor decompositions with applications to polynomial identifiability. In Conference on Learning Theory, pages 742–778, 2014.

[47] David J. Biagioni, Daniel Beylkin, and Gregory Beylkin. Randomized interpolative decomposition of separated representations. Journal of Computational Physics, 281(C):116–134, January 2015.

[48] Jacob Bien, Ya Xu, and Michael W. Mahoney. CUR from a sparse optimization viewpoint. In Advances in Neural Information Processing Systems, pages 217–225, 2010.

[49] Ilias Bilionis and Nicholas Zabaras. Multi-output local gaussian process regression: Applications to uncertainty quantification. Journal of Computational Physics, 231(17):5718–5746, 2012.

[50] Dario Bini. Relations between exact and approximate bilinear algorithms. Applications. Calcolo, 17(1):87–97, 1980.

[51] Dario Bini, Milvio Capovani, Francesco Romani, and Grazia Lotti. $O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication. Information Processing Letters, 8(5):234–235, June 1979.

[52] Dario Bini and Grazia Lotti. Stability of Fast Algorithms for Matrix Multiplication. Numerische Mathematik, 36(1):63–72, 1980.

[53] Dario Bini, Grazia Lotti, and Francesco Romani. Approximate solutions for the bilinear form computational problem. SIAM Journal on Computing, 9(4):692–697, 1980.

[54] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. arXiv preprint arXiv:1801.01401, 2018.

[55] Richard L. Bishop and Samuel I. Goldberg. Tensor Analysis on Manifolds. Courier Corporation, 2012.

[56] Meltzer Bittner, Paul Meltzer, Yidong Chen, Youfei Jiang, Elisabeth Seftor, M. Hendrix, M. Radmacher, Richard Simon, Zohar Yakhini, and Amir Ben-Dor. Molecular classification of cutaneous malignant melanoma by gene expression profiling. Nature, 406(6795):536–540, 2000.

[57] Markus Bläser. On the complexity of the multiplication of matrices of small formats. Journal of Complexity, 19(1):43–60, 2003.

[58] Jonah Blasiak, Thomas Church, Henry Cohn, Joshua A. Grochow, and Chris Umans. Which groups are amenable to proving exponent two for matrix multiplication? arXiv preprint arXiv:1712.02302, 2017.

[59] 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.

[60] Ajinkya Borle, Vincent E. Elfving, and Samuel J. Lomonaco. Quantum Approximate Optimization for Hard Problems in Linear Algebra. arXiv preprint arXiv:2006.15438, 2020.

[61] Christos Boutsidis. http://www.boutsidis.org, accessed 25 April 2019.

[62] Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-Optimal Column-Based Matrix Reconstruction. SIAM Journal on Computing, 43(2):687–717, April 2014.

[63] Christos Boutsidis, Michael W. Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 968–977. SIAM, 2009.

[64] Christos Boutsidis and David P. Woodruff. Optimal CUR matrix decompositions. SIAM Journal on Computing, 46(2):543–589, 2017.

[65] Amanda Bower and Laura Balzano. Preference modeling with context-dependent salient features. In International Conference on Machine Learning, pages 1067–1077. PMLR, 2020.

[66] Stephen Boyd and Lieven Vandenberghe. Convex Optimization. Cambridge university press, 2004.

[67] Karen Braman. Third-order tensors as linear operators on a space of matrices. Linear Algebra and its Applications, 433(7):1241–1253, 2010.

[68] Richard P. Brent. Algorithms for matrix multiplication. Technical Report STAN-CS-70-157, Stanford University, 1970.

[69] Richard P. Brent. Error Analysis of Algorithms for Matrix Multiplication and Triangular Decomposition using Winograd's Identity. Numerische Mathematik, 16(2):145–156, 1970.

[70] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203, 2013.

[71] Jean-Philippe Brunet, Pablo Tamayo, Todd R. Golub, and Jill P. Mesirov. Metagenes and molecular pattern discovery using matrix factorization. Proceedings of the national academy of sciences, 101(12):4164–4169, 2004.

[72] Christian G Bucher. Adaptive sampling—an iterative fast monte carlo procedure. Structural safety, 5(2):119–126, 1988.

[73] Zvonimir Bujanović and Daniel Kressner. Norm and trace estimation with random rank-one vectors. arXiv:2004.06433 [cs, math, stat], August 2020.

[74] Aydin Buluc, Tamara G. Kolda, Stefan M. Wild, Mihai Anitescu, Anthony DeGennaro, John Jakeman, Chandrika Kamath, Ramakrishnan, Kannan, Miles E. Lopes, Per-Gunnar Martinsson, Kary Myers, Jelani Nelson, Juan M. Restrepo, C. Seshadhri, Draguna Vrabie, Brendt Wohlberg, Stephen J. Wright, Chao Yang, and Peter Zwart. Randomized algorithms for scientific computing (RASC). 2021.

[75] Peter Bürgisser, Michael Clausen, and Mohammad A. Shokrollahi. Algebraic Complexity Theory, volume 315. Springer Science & Business Media, 2013.

[76] James V Burke, Adrian S Lewis, and Michael L Overton. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. SIAM Journal on Optimization, 15(3):751–779, 2005.

[77] Hongyun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. IEEE Transactions on Knowledge and Data Engineering, 30(9):1616–1637, 2018.

[78] Cesar F. Caiafa and Andrzej Cichocki. Generalizing the column–row matrix decomposition to multi-way arrays. Linear Algebra and its Applications, 433(3):557–573, 2010.

[79] K. I. Caputi, G. B. Caminha, S. Fujimoto, K. Kohno, F. Sun, E. Egami, S. Deshmukh, F. Tang, Y. Ao, L. Bradley, D. Coe, D. Espada, C. Grillo, B. Hatsukade, K. K. Knudsen, M. M. Lee, G. E. Magdis, K. Morokuma-Matsui, P. Oesch, M. Ouchi, P. Rosati, I. Smail, H. Umehata, F. Valentino, E. Vanzella, W.-H. Wang, J. F. Wu, and A. Zitrin. ALMA Lensing Cluster Survey: An ALMA galaxy signposting a MUSE galaxy group at z=4.3 behind 'El Gordo'. arXiv:2009.04838 [astro-ph], September 2020.

[80] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In 2017 ieee symposium on security and privacy (sp), pages 39–57. IEEE, 2017.

[81] Coralia Cartis and Katya Scheinberg. Global convergence rate analysis of unconstrained optimization methods based on probabilistic models. Mathematical Programming, 169:337–375, 2018.

[82] Rick R. Castrapel and John L. Gustafson. Precision improvement method for the Strassen/Winograd matrix multiplication method. U.S. Patent No., 7209939B2:1–11, April 2007.

[83] Thomas Chaigne, Bastien Arnal, Sergey Vilov, Emmanuel Bossy, and Ori Katz. Super-resolution photoacoustic imaging via flow-induced absorption fluctuations. Optica, 4(11):1397–1404, 2017.

[84] Thomas Chaigne, Jérôme Gateau, Marc Allain, Ori Katz, Sylvain Gigan, Anne Sentenac, and Emmanuel Bossy. Super-resolution photoacoustic fluctuation imaging with multiple speckle illumination. Optica, 3(1):54–57, 2016.

[85] Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding Frequent Items in Data Streams. Theoretical Computer Science, 312(1):3–15, January 2004.

[86] Anindya Chatterjee. An introduction to the proper orthogonal decomposition. Current Science, pages 808–817, 2000.

[87] Maolin Che and Yimin Wei. Randomized algorithms for the approximations of Tucker and the tensor train decompositions. Advances in Computational Mathematics, 45(1):395–428, 2019.

[88] Aochuan Chen, Yimeng Zhang, Jinghan Jia, James Diffenderfer, Jiancheng Liu, Konstantinos Parasyris, Yihua Zhang, Zheng Zhang, Bhavya Kailkhura, and Sijia Liu. Deepzero: Scaling up zeroth-order optimization for deep model training. arXiv preprint arXiv:2310.02025, 2023.

[89] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In Proceedings of the 10th ACM workshop on artificial intelligence and security, pages 15–26, 2017.

[90] Wenqian Chen and Panos Stinis. Feature-adjacent multi-fidelity physics-informed machine learning for partial differential equations. arXiv preprint arXiv:2303.11577, 2023.

[91] Dehua Cheng, Richard Peng, Yan Liu, and Ioakeim Perros. SPALS: Fast alternating least squares via implicit leverage scores sampling. In Advances In Neural Information Processing Systems, pages 721–729, 2016.

[92] Hongwei Cheng, Zydrunas Gimbutas, Per-Gunnar Martinsson, and Vladimir Rokhlin. On the Compression of Low Rank Matrices. SIAM Journal on Scientific Computing, 26(4):1389–1404, March 2005.

[93] Nuojin Cheng and Alireza Doostan. Langevin bi-fidelity importance sampling for failure probability estimation. arXiv preprint arXiv:2503.17796, 2025.

[94] Nuojin Cheng, Osman Asif Malik, Subhayan De, Stephen Becker, and Alireza Doostan. Bi-fidelity variational auto-encoder for uncertainty quantification. Computer Methods in Applied Mechanics and Engineering, 421:116793, 2024.

[95] Nuojin Cheng, Osman Asif Malik, Yiming Xu, Stephen Becker, Alireza Doostan, and Akil Narayan. Quadrature sampling of parametric models with bi-fidelity boosting. arXiv preprint arXiv:2209.05705, 2022.

[96] Nuojin Cheng, Osman Asif Malik, Yiming Xu, Stephen Becker, Alireza Doostan, and Akil Narayan. Subsampling of parametric models with bifidelity boosting. SIAM/ASA Journal on Uncertainty Quantification, 12(2):213–241, 2024.

[97] Nuojin Cheng, Leonard Papenmeier, Stephen Becker, and Luigi Nardi. A unified framework for entropy search and expected improvement in bayesian optimization. arXiv preprint arXiv:2501.18756, 2025.

[98] Xiang Cheng and Peter Bartlett. Convergence of langevin mcmc in kl-divergence. In Algorithmic Learning Theory, pages 186–211. PMLR, 2018.

[99] Jocelyn T. Chi and Ilse C. F. Ipsen. A Geometric Analysis of Model- and Algorithm-induced uncertainties for randomized least squares regression. 2019.

[100] Jocelyn T. Chi and Ilse C. F. Ipsen. A Projector-Based Approach to Quantifying Total and Excess Uncertainties for Sketched Linear Regression. arXiv:1808.05924 [cs, math, stat], 2020. arXiv: 1808.05924 version: 3.

[101] Jocelyn T Chi and Ilse CF Ipsen. Multiplicative perturbation bounds for multivariate multiple linear regression in schatten p-norms. Linear Algebra and its Applications, 624:87–102, 2021.

[102] Agniva Chowdhury, Palma London, Haim Avron, and Petros Drineas. Speeding up Linear Programming using Randomized Linear Algebra. arXiv:2003.08072 [cs, math], March 2020.

[103] Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, and Danilo P. Mandic. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. Foundations and Trends® in Machine Learning, 9(4-5):249–429, 2016.

[104] Andrzej Cichocki, Anh-Huy Phan, Qibin Zhao, Namgil Lee, Ivan Oseledets, Masashi Sugiyama, and Danilo P. Mandic. Tensor networks for dimensionality reduction and large-scale optimization: Part 2 applications and future perspectives. Foundations and Trends® in Machine Learning, 9(6):431–673, 2017.

[105] Kenneth L. Clarkson and David P. Woodruff. Numerical linear algebra in the streaming model. In Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, pages 205–214. ACM, 2009.

[106] Kenneth L. Clarkson and David P. Woodruff. Low-Rank Approximation and Regression in Input Sparsity Time. Journal of the ACM, 63(6):54:1–54:45, February 2017.

[107] Albert Cohen, Mark A. Davenport, and Dany Leviatan. On the Stability and Accuracy of Least Squares Approximations. Foundations of Computational Mathematics, 13(5):819–834, 2013.

[108] Albert Cohen and Ronald DeVore. Approximation of high-dimensional parametric PDEs. Acta Numerica, 24:1–159, 2015.

[109] Albert Cohen and Giovanni Migliorati. Optimal weighted least-squares methods. SMAI Journal of Computational Mathematics, 3:181–203, 2017. arxiv:1608.00512 [math.NA].

[110] Eldan Cohen, Avradip Mandal, Hayato Ushijima-Mwesigwa, and Arnab Roy. Ising-based consensus clustering on specialized hardware. In International Symposium on Intelligent Data Analysis, pages 106–118. Springer, 2020.

[111] Nadav Cohen, Or Sharir, and Amnon Shashua. On the expressive power of deep learning: A tensor analysis. In Conference on Learning Theory, pages 698–728, 2016.

[112] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. Trust Region Methods. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[113] Thomas M Cover. Elements of information theory. John Wiley & Sons, 1999.

[114] B. T. Cox and P. C. Beard. Fast calculation of pulsed photoacoustic fields in fluids using k-space methods. The Journal of the Acoustical Society of America, 117(6):3616–3627, 2005.

[115] Tiangang Cui, Sergey Dolgov, and Robert Scheichl. Deep importance sampling using tensor trains with application to a priori and a posteriori rare events. SIAM Journal on Scientific Computing, 46(1):C1–C29, 2024.

[116] Mark Cutler, Thomas J Walsh, and Jonathan P How. Reinforcement learning with multi-fidelity simulators. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 3888–3895. IEEE, 2014.

[117] Michele N. da Costa, Renato R. Lopes, and Joao Marcos T. Romano. Randomized methods for higher-order subspace separation. In 2016 24th European Signal Processing Conference (EUSIPCO), pages 215–219. IEEE, 2016.

[118] Alex P. da Silva and Pierre Comon. A Finite Algorithm to Compute Rank-1 Tensor Approximations. IEEE Signal Processing Letters, 23(7):959–963, July 2016.

[119] J. Christopher Dainty. Laser Speckle and Related Phenomena, volume 9. Springer science & business Media, 2013.

[120] Arnak S Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. Journal of the Royal Statistical Society. Series B (Statistical Methodology), pages 651–676, 2017.

[121] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. Random Structures & Algorithms, 22(1):60–65, 2003.

[122] Timothy A. Davis. Direct Methods for Sparse Linear Systems. Number 2 in Fundamentals of Algorithms. Society for Industrial and Applied Mathematics, 1st edition, 2006.

[123] Subhayan De, Jolene Britton, Matthew Reynolds, Ryan Skinner, Kenneth Jansen, and Alireza Doostan. On transfer learning of neural networks using bi-fidelity data for uncertainty propagation. International Journal for Uncertainty Quantification, 10(6), 2020.

[124] Subhayan De and Alireza Doostan. Neural network training using ℓ_1-regularization and bi-fidelity data. Journal of Computational Physics, 458:111010, 2022.

[125] Subhayan De, Malik Hassanaly, Matthew Reynolds, Ryan N King, and Alireza Doostan. Bi-fidelity modeling of uncertain and partially unknown systems using deeponets. arXiv preprint arXiv:2204.00997, 2022.

[126] Subhayan De, Kurt Maute, and Alireza Doostan. Bi-fidelity stochastic gradient descent for structural optimization under uncertainty. Computational Mechanics, 66:745–771, 2020.

[127] Himeshi De Silva, John L. Gustafson, and Weng-Fai Wong. Making Strassen Matrix Multiplication Safe. In 2018 IEEE 25th International Conference on High Performance Computing (HiPC), pages 173–182. IEEE, 2018.

[128] Fabrizio De Vico Fallani, Jonas Richiardi, Mario Chavez, and Sophie Achard. Graph analysis of functional brain networks: Practical issues in translational neuroscience. Philosophical Transactions of the Royal Society B: Biological Sciences, 369(1653):20130521, 2014.

[129] X. Luís Dean-Ben and Daniel Razansky. Localization optoacoustic tomography. Light: Science & Applications, 7(4):18004–18004, 2018.

[130] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In Advances in Neural Information Processing Systems, pages 3844–3852, 2016.

[131] James Demmel, Ioana Dumitriu, and Olga Holtz. Fast linear algebra is stable. Numerische Mathematik, 108(1):59–91, 2007.

[132] James Demmel, Ioana Dumitriu, Olga Holtz, and Robert Kleinberg. Fast matrix multiplication is stable. Numerische Mathematik, 106(2):199–224, 2007.

[133] Armen Der Kiureghian et al. First-and second-order reliability methods. Engineering design reliability handbook, 14, 2005.

[134] Micha\l Derezinski and Michael W. Mahoney. Determinantal point processes in randomized numerical linear algebra. Notices of the American Mathematical Society, 68(1):34–45, 2021.

[135] Micha\l Dereziński and Manfred K. Warmuth. Subsampling for ridge regression via regularized volume sampling. arXiv preprint arXiv:1710.05110, 2017.

[136] Micha\l Dereziński and Manfred K. Warmuth. Unbiased estimates for linear regression via volume sampling. arXiv preprint arXiv:1705.06908, 2017.

[137] Micha\l Dereziński and Manfred K. Warmuth. Reverse iterative volume sampling for linear regression. The Journal of Machine Learning Research, 19(1):853–891, 2018.

[138] Micha\l Dereziński, Manfred K. Warmuth, and Daniel Hsu. Leveraged volume sampling for linear regression. arXiv preprint arXiv:1802.06749, 2018.

[139] Thomas Dertinger, Ryan Colyer, Gopal Iyer, Shimon Weiss, and Jörg Enderlein. Fast, background-free, 3D super-resolution optical fluctuation imaging (SOFI). Proceedings of the National Academy of Sciences, 106(52):22287–22292, 2009.

[140] Derek DeSantis, Erik Skau, and Boian Alexandrov. Factorizations of Binary Matrices–Rank Relations and the Uniqueness of Boolean Decompositions. arXiv preprint arXiv:2012.10496, 2020.

[141] Amit Deshpande and Luis Rademacher. Efficient volume sampling for row/column subset selection. In 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, pages 329–338. IEEE, 2010.

[142] Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. In Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm, pages 1117–1126. Society for Industrial and Applied Mathematics, 2006.

[143] Amit Deshpande and Santosh Vempala. Adaptive sampling and fast low-rank matrix approximation. In Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, pages 292–303. Springer, 2006.

[144] Huaian Diao, Rajesh Jayaram, Zhao Song, Wen Sun, and David P. Woodruff. Optimal Sketching for Kronecker Product Regression and Low Rank Approximation. arXiv preprint arXiv:1909.13384, 2019.

[145] Huaian Diao, Zhao Song, Wen Sun, and David Woodruff. Sketching for Kronecker Product Regression and P-splines. In Proceedings of the 21st International Conference on Artificial Intelligence and Statistics, pages 1299–1308, 2018.

[146] Paul Diaz, Alireza Doostan, and Jerrad Hampton. Sparse polynomial chaos expansions via compressed sensing and d-optimal design. Computer Methods in Applied Mechanics and Engineering, 336:640–666, 2018.

[147] Mamadou Diop, Anthony Larue, Sebastian Miron, and David Brie. A post-nonlinear mixture model approach to binary matrix factorization. In 2017 25th European Signal Processing Conference (EUSIPCO), pages 321–325. IEEE, 2017.

[148] Alireza Doostan, Roger G. Ghanem, and John Red-Horse. Stochastic model reduction for chaos representations. Computer Methods in Applied Mechanics and Engineering, 196(37):3951–3966, 2007.

[149] Alireza Doostan and Gianluca Iaccarino. A least-squares approximation of partial differential equations with high-dimensional random inputs. Journal of Computational Physics, 228(12):4332–4345, 2009.

[150] Alireza Doostan and Houman Owhadi. A non-adapted sparse approximation of pdes with stochastic inputs. Journal of Computational Physics, 230(8):3015–3034, 2011.

[151] Petros Drineas and Ravi Kannan. Pass efficient algorithms for approximating large matrices. In Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 223–232, Baltimore, Maryland, January 2003. Society for Industrial and Applied Mathematics.

[152] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo algorithms for matrices I: Approximating matrix multiplication. SIAM Journal on Computing, 36(1):132–157, 2006.

[153] Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, and David P. Woodruff. Fast approximation of matrix coherence and statistical leverage. The Journal of Machine Learning Research, 13(1):3475–3506, 2012.

[154] Petros Drineas and Michael W. Mahoney. A randomized algorithm for a tensor-based generalization of the singular value decomposition. Linear algebra and its applications, 420(2-3):553–571, 2007.

[155] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Polynomial time algorithm for column-row based relative-error low-rank matrix approximation. Technical report, Technical report 2006-04, DIMACS, 2006.

[156] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. SIAM Journal on Matrix Analysis and Applications, 30(2):844–881, 2008.

[157] Petros Drineas, Michael W. Mahoney, and Shan Muthukrishnan. Sampling algorithms for ℓ_2 regression and applications. In Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm, pages 1127–1136, 2006.

[158] Petros Drineas, Michael W. Mahoney, Shan Muthukrishnan, and Tamás Sarlós. Faster least squares approximation. Numerische Mathematik, 117(2):219–249, 2011.

[159] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research, 12(7), 2011.

[160] John C Duchi, Michael I Jordan, Martin J Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. IEEE Transactions on Information Theory, 61(5):2788–2806, 2015.

[161] Bogdan Dumitrescu. Improving and estimating the accuracy of Strassen's algorithm. Numerische Mathematik, 79(4):485–499, 1998.

[162] Alain Durmus, Szymon Majewski, and Błażej Miasojedow. Analysis of langevin monte carlo via convex optimization. The Journal of Machine Learning Research, 20(1):2666–2711, 2019.

[163] Alain Durmus and Éric Moulines. Nonasymptotic convergence analysis for the unadjusted langevin algorithm. Annals of Applied Probability, 27(3):1551–1587, 2017.

[164] Alain Durmus and Eric Moulines. High-dimensional bayesian inference via the unadjusted langevin algorithm. 2019.

[165] Stavros Efthymiou, Jack Hidary, and Stefan Leichenauer. TensorNetwork for Machine Learning. arXiv preprint arXiv:1906.06329, 2019.

[166] David M. Egolf, Ryan KW Chee, and Roger J. Zemp. Sparsity-based reconstruction for super-resolved limited-view photoacoustic computed tomography deep in a scattering medium. Optics letters, 43(10):2221–2224, 2018.

[167] Veit Elser. A network that learns strassen multiplication. The Journal of Machine Learning Research, 17(1):3964–3976, 2016.

[168] Lawrence J. Emrich and Marion R. Piedmonte. A method for generating high-dimensional multivariate binary variates. The American Statistician, 45(4):302–304, 1991.

[169] Mike Espig, Kishore Kumar Naraparaju, and Jan Schneider. A note on tensor chain approximation. Computing and Visualization in Science, 15(6):331–344, 2012.

[170] Hillary R. Fairbanks, Alireza Doostan, Christian Ketelsen, and Gianluca Iaccarino. A low-rank control variate for multilevel monte carlo simulation of high-dimensional uncertain systems. Journal of Computational Physics, 341:121–139, 2017.

[171] Hillary R Fairbanks, Alireza Doostan, Christian Ketelsen, and Gianluca Iaccarino. A low-rank control variate for multilevel monte carlo simulation of high-dimensional uncertain systems. Journal of Computational Physics, 341:121–139, 2017.

[172] Hillary R Fairbanks, Lluís Jofre, Gianluca Geraci, Gianluca Iaccarino, and Alireza Doostan. Bi-fidelity approximation for uncertainty quantification and sensitivity analysis of irradiated particle-laden turbulence. Journal of Computational Physics, 402:108996, 2020.

[173] Hadi Fanaee-T and João Gama. Multi-aspect-streaming tensor analysis. Knowledge-Based Systems, 89:332–345, 2015.

[174] Yani Feng, Kejun Tang, Lianxing He, Pingqiang Zhou, and Qifeng Liao. Tensor Train Random Projection. arXiv preprint arXiv:2010.10797, 2020.

[175] Bernd Fiessler, Hans-Joachim Neumann, and Rudiger Rackwitz. Quadratic limit states in structural reliability. Journal of the Engineering Mechanics Division, 105(4):661–676, 1979.

[176] Steven T. Flammia and Yi-Kai Liu. Direct fidelity estimation from few Pauli measurements. Physical review letters, 106(23):230501, 2011.

[177] Yoav Freund, Yi-An Ma, and Tong Zhang. When is the convergence time of langevin algorithms dimension independent? a composite optimization viewpoint. The Journal of Machine Learning Research, 23(1):9604–9635, 2022.

[178] Shmuel Friedland, Volker Mehrmann, Agnieszka Miedlar, and M. Nkengla. Fast low rank approximations of matrices and tensors. Electronic Journal of Linear Algebra, 22:1031–1048, 2011.

[179] Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. Journal of the ACM (JACM), 51(6):1025–1041, 2004.

[180] Timur Garipov, Dmitry Podoprikhin, Alexander Novikov, and Dmitry Vetrov. Ultimate tensorization: Compressing convolutional and fc layers alike. arXiv preprint arXiv:1611.03214, 2016.

[181] Jérôme Gateau, Thomas Chaigne, Ori Katz, Sylvain Gigan, and Emmanuel Bossy. Improving visibility in photoacoustic imaging using dynamic speckle illumination. Optics letters, 38(23):5188–5191, 2013.

[182] Nicholas Geneva and Nicholas Zabaras. Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks. Journal of Computational Physics, 403:109056, 2020.

[183] Nicholas Geneva and Nicholas Zabaras. Multi-fidelity generative deep learning turbulent flows. arXiv preprint arXiv:2006.04731, 2020.

[184] Sebastian Geyer, Iason Papaioannou, and Daniel Straub. Cross entropy-based importance sampling using gaussian densities revisited. Structural Safety, 76:15–27, 2019.

[185] Roger G Ghanem and Pol D Spanos. Stochastic finite elements: a spectral approach. Courier Corporation, 2003.

[186] Kyle Gilman and Laura Balzano. Grassmannian optimization for online tensor completion and tracking in the t-SVD algebra. arXiv preprint arXiv:2001.11419, 2020.

[187] Kyle Gilman and Laura Balzano. Online Tensor Completion and Free Submodule Tracking With The T-SVD. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3282–3286. IEEE, 2020.

[188] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In International Conference on Machine Learning, pages 1263–1272. PMLR, 2017.

[189] Fred Glover, Gary Kochenberger, and Yu Du. A tutorial on formulating and using QUBO models. arXiv preprint arXiv:1811.11538, 2018.

[190] Gene H. Golub and Charles F. Van Loan. Matrix Computations. Johns Hopkins University Press, Baltimore, fourth edition, 2013.

[191] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. Communications of the ACM, 63(11):139–144, 2020.

[192] Sergey A. Goreinov, Eugene E. Tyrtyshnikov, and Nickolai L. Zamarashkin. Pseudo-skeleton approximations by matrices of maximal volume. Mathematical Notes, 62(4):515–519, October 1997.

[193] Sergey A. Goreinov, Eugene E. Tyrtyshnikov, and Nickolai L. Zamarashkin. A theory of pseudoskeleton approximations. Linear Algebra and its Applications, 261(1-3):1–21, August 1997.

[194] Alex A Gorodetsky, Gianluca Geraci, Michael S Eldred, and John D Jakeman. A generalized approximate control variate framework for multifidelity uncertainty quantification. Journal of Computational Physics, 408:109257, 2020.

[195] Palash Goyal, Nitin Kamra, Xinran He, and Yan Liu. Dyngem: Deep embedding method for dynamic graphs. arXiv preprint arXiv:1805.11273, 2018.

[196] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. Journal of Machine Learning Research, 13(25):723–773, 2012.

[197] Thomas Nall Eden Greville. Note on the generalized inverse of a matrix product. Siam Review, 8(4):518–521, 1966.

[198] David Gross and Vincent Nesme. Note on sampling without replacing from a finite collection of matrices. arXiv preprint arXiv:1001.2738, 2010.

[199] Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 855–864, 2016.

[200] Dane Grundvig. Line search based optimization using function approximations with tunable accuracy. Master's thesis, Rice University, 2023.

[201] Ming Gu and Stanley C. Eisenstat. Efficient Algorithms for Computing a Strong Rank-Revealing QR Factorization. SIAM Journal on Scientific Computing, 17(4):848–869, July 1996.

[202] Ekta Gujral, Ravdeep Pasricha, and Evangelos E. Papalexakis. Sambaten: Sampling-based batch incremental tensor decomposition. In Proceedings of the 2018 SIAM International Conference on Data Mining, pages 387–395. SIAM, 2018.

[203] Ling Guo, Akil Narayan, Liang Yan, and Tao Zhou. Weighted Approximate Fekete Points: Sampling for Least-Squares Polynomial Approximation. SIAM Journal on Scientific Computing, 40(1):A366–A387, 2018. arXiv:1708.01296 [math.NA].

[204] Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In International Conference on Machine Learning, pages 1737–1746, 2015.

[205] Venkatesan Guruswami and Ali Kemal Sinop. Optimal column-based low-rank matrix reconstruction. In Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, pages 1207–1214. SIAM, 2012.

[206] Cécile Haberstich, Anthony Nouy, and Guillaume Perrin. Boosted optimal weighted least-squares. Mathematics of Computation, 91(335):1281–1315, 2022.

[207] Mohammad Hadigol and Alireza Doostan. Least squares polynomial chaos expansion: A review of sampling strategies. Computer Methods in Applied Mechanics and Engineering, 332:382–407, 2018.

[208] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. SIAM Review, 53(2):217–288, May 2011.

[209] David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. Applied and Computational Harmonic Analysis, 30(2):129–150, 2011.

[210] Jerrad Hampton and Alireza Doostan. Coherence motivated sampling and convergence analysis of least squares polynomial chaos regression. Computer Methods in Applied Mechanics and Engineering, 290:73–97, 2015.

[211] Jerrad Hampton and Alireza Doostan. Coherence motivated sampling and convergence analysis of least squares polynomial Chaos regression. Computer Methods in Applied Mechanics and Engineering, 290:73–97, 2015.

[212] Jerrad Hampton and Alireza Doostan. Compressive sampling of polynomial chaos expansions: Convergence analysis and sampling strategies. Journal of Computational Physics, 280:363–386, 2015.

[213] Jerrad Hampton and Alireza Doostan. Compressive sampling of polynomial chaos expansions: Convergence analysis and sampling strategies. Journal of Computational Physics, 280:363–386, 2015.

[214] Jerrad Hampton and Alireza Doostan. Basis adaptive sample efficient polynomial chaos (base-pc). Journal of Computational Physics, 371:20–49, 2018.

[215] Jerrad Hampton and Alireza Doostan. Basis adaptive sample efficient polynomial chaos (base-pc). Journal of Computational Physics, 371:20–49, 2018.

[216] Jerrad Hampton, Hillary Fairbanks, Akil Narayan, and Alireza Doostan. Parametric/stochastic model reduction: low-rank representation, non-intrusive bi-fidelity approximation, and convergence analysis. arXiv preprint arXiv:1709.03661, 2017.

[217] Jerrad Hampton, Hillary R. Fairbanks, Akil Narayan, and Alireza Doostan. Practical error bounds for a non-intrusive bi-fidelity approach to parametric/stochastic model reduction. Journal of Computational Physics, 368:315–332, 2018.

[218] Jerrad Hampton, Hillary R. Fairbanks, Akil Narayan, and Alireza Doostan. Practical error bounds for a non-intrusive bi-fidelity approach to parametric/stochastic model reduction. Journal of Computational Physics, 368:315–332, 9 2018.

[219] Jerrad Hampton, Hillary R. Fairbanks, Akil Narayan, and Alireza Doostan. Practical error bounds for a non-intrusive bi-fidelity approach to parametric/stochastic model reduction. Journal of Computational Physics, 368:315–332, 2018.

[220] Ning Hao, Misha E. Kilmer, Karen Braman, and Randy C. Hoover. Facial recognition using tensor-tensor decompositions. SIAM Journal on Imaging Sciences, 6(1):437–463, 2013.

[221] Abraham M Hasofer. An exact and invarient first order reliability format. J. Eng. Mech. Div., Proc. ASCE, 100(1):111–121, 1974.

[222] Johan Håstad. Tensor rank is NP-Complete. In International Colloquium on Automata, Languages, and Programming, pages 451–460. Springer, 1989.

[223] W. Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. Biometrika, 57(1):97–109, April 1970.

[224] W. Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. Biometrika, 57(1):97–109, April 1970.

[225] Tamir Hazan, Simon Polak, and Amnon Shashua. Sparse image coding using a 3D non-negative tensor factorization. In Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, volume 1, pages 50–57. IEEE, 2005.

[226] Sibylle Hess, Katharina Morik, and Nico Piatkowski. The PRIMPING routine—Tiling through proximal alternating linearized minimization. Data Mining and Knowledge Discovery, 31(4):1090–1131, July 2017.

[227] Jan S Hesthaven, Gianluigi Rozza, Benjamin Stamm, et al. Certified reduced basis methods for parametrized partial differential equations, volume 590. Springer, 2016.

[228] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. Advances in Neural Information Processing Systems, 30, 2017.

[229] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: Learning basic visual concepts with a constrained variational framework. International Conference on Learning Representations, 2017.

[230] Nicholas J. Higham. Functions of Matrices: Theory and Computation, volume 104. Siam, 2008.

[231] Christopher J. Hillar and Lek-Heng Lim. Most tensor problems are NP-hard. Journal of the ACM (JACM), 60(6):45, 2013.

[232] Geoffrey Hinton. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531, 2015.

[233] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems, 33:6840–6851, 2020.

[234] Eliel Hojman, Thomas Chaigne, Oren Solomon, Sylvain Gigan, Emmanuel Bossy, Yonina C. Eldar, and Ori Katz. Photoacoustic imaging beyond the acoustic diffraction-limit with dynamic speckle illumination and sparse joint support recovery. Optics express, 25(5):4875–4886, 2017.

[235] David Hong, Jeffrey A. Fessler, and Laura Balzano. Optimally weighted PCA for high-dimensional heteroscedastic data. arXiv preprint arXiv:1810.12862, 2018.

[236] David Hong, Kyle Gilman, Laura Balzano, and Jeffrey A. Fessler. HePPCAT: Probabilistic PCA for Data with Heteroscedastic Noise. arXiv preprint arXiv:2101.03468, 2021.

[237] Michael Hopkins, Mantas Mikaitis, Dave R. Lester, and Steve Furber. Stochastic rounding and reduced-precision fixed-point arithmetic for solving neural ODEs. arXiv preprint arXiv:1904.11263, 2019.

[238] Roger A. Horn and Charles R. Johnson. Topics in Matrix Analysis. Cambridge University Press, 1994.

[239] Roger A. Horn and Charles R. Johnson. Matrix Analysis. Cambridge university press, 2012.

[240] Ming Hou, Jiajia Tang, Jianhai Zhang, Wanzeng Kong, and Qibin Zhao. Deep multimodal multilinear fusion with high-order polynomial pooling. In Advances in Neural Information Processing Systems, pages 12136–12145, 2019.

[241] Jianyu Huang, Chenhan D. Yu, and Robert A. van de Geijn. Implementing Strassen's Algorithm with CUTLASS on NVIDIA Volta GPUs. arXiv:1808.07984 [cs], August 2018.

[242] Benjamin Huber, Reinhold Schneider, and Sebastian Wolf. A randomized tensor train singular value decomposition. In Compressed Sensing and Its Applications, pages 261–290. Springer, 2017.

[243] Steven Huss-Lederman, Elaine M. Jacobson, Jeremy R. Johnson, Anna Tsao, and Thomas Turnbull. Implementation of Strassen's algorithm for matrix multiplication. In Supercomputing'96: Proceedings of the 1996 ACM/IEEE Conference on Supercomputing, pages 32–32. IEEE, 1996.

[244] Jérôme Idier, Simon Labouesse, Marc Allain, Penghuan Liu, Sébastien Bourguignon, and Anne Sentenac. On the superresolution capacity of imagers using unknown speckle illuminations. IEEE Transactions on Computational Imaging, 4(1):87–98, 2018.

[245] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, pages 604–613. ACM, 1998.

[246] Leon Isserlis. On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables. Biometrika, 12(1/2):134–139, 1918.

[247] M. A. Iwen, D. Needell, E. Rebrova, and A. Zare. Lower Memory Oblivious (Tensor) Subspace Embeddings with Fewer Random Bits: Modewise Methods for Least Squares. arXiv preprint arXiv:1912.08294, 2019.

[248] Pavel Izmailov, Alexander Novikov, and Dmitry Kropotov. Scalable gaussian processes with billions of inducing inputs via tensor train decomposition. In International Conference on Artificial Intelligence and Statistics, pages 726–735, 2018.

[249] Jean Jacod and Philip Protter. Probability Essentials. Springer Science & Business Media, 2012.

[250] Inah Jeon, Evangelos E. Papalexakis, Uksong Kang, and Christos Faloutsos. Haten2: Billion-scale tensor decompositions. In 2015 IEEE 31st International Conference on Data Engineering, pages 1047–1058. IEEE, 2015.

[251] Billy Jin, Katya Scheinberg, and Miaolan Xie. High probability complexity bounds for adaptive step search based on stochastic oracles. SIAM Journal on Optimization, 34(3):2411–2439, 2024.

[252] Ruhui Jin, Tamara G Kolda, and Rachel Ward. Faster Johnson-Lindenstrauss transforms via kronecker products. Information and Inference: A Journal of the IMA, October 2020.

[253] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. Advances in Neural Information Processing Systems, 26, 2013.

[254] Rodney W. Johnson and Aileen M. McLoughlin. Noncommutative Bilinear Algorithms for 3*3 Matrix Multiplication. SIAM Journal on Computing, 15(2):595–603, 1986.

[255] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. Contemporary Mathematics, 26(189-206):1, 1984.

[256] Kirthevasan Kandasamy, Gautam Dasarathy, Junier B Oliva, Jeff Schneider, and Barnabás Póczos. Gaussian process bandit optimisation with multi-fidelity evaluations. Advances in neural information processing systems, 29, 2016.

[257] Daniel M. Kane and Jelani Nelson. Sparser Johnson-Lindenstrauss transforms. Journal of the ACM (JACM), 61(1):4, 2014.

[258] Igor Kaporin. The aggregation and cancellation techniques as a practical tool for faster matrix multiplication. Theoretical Computer Science, 315(2-3):469–510, 2004.

[259] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Lojasiewicz condition. In Joint European conference on machine learning and knowledge discovery in databases, pages 795–811. Springer, 2016.

[260] Oguz Kaya and Bora Uçar. High performance parallel algorithms for the tucker decomposition of sparse tensors. In 2016 45th International Conference on Parallel Processing (ICPP), pages 103–112. IEEE, 2016.

[261] Marc C Kennedy and Anthony O'Hagan. Predicting the output from a complex computer code when fast approximations are available. Biometrika, 87(1):1–13, 2000.

[262] Marc C Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63(3):425–464, 2001.

[263] Eric Kernfeld, Misha Kilmer, and Shuchin Aeron. Tensor–tensor products with invertible linear transforms. Linear Algebra and its Applications, 485:545–570, 2015.

[264] Yuehaw Khoo, Jianfeng Lu, and Lexing Ying. Efficient construction of tensor ring representations from sampling. arXiv preprint arXiv:1711.00954, 2019.

[265] Valentin Khrulkov, Alexander Novikov, and Ivan Oseledets. Expressive power of recurrent neural networks. In International Conference on Learning Representations, 2018.

[266] Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. The Annals of Mathematical Statistics, pages 462–466, 1952.

[267] Misha Kilmer, Lior Horesh, Haim Avron, and Elizabeth Newman. Tensor-Tensor Products for Optimal Representation and Compression. arXiv preprint arXiv:2001.00046, 2019.

[268] Misha E. Kilmer, Karen Braman, Ning Hao, and Randy C. Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. SIAM Journal on Matrix Analysis and Applications, 34(1):148–172, 2013.

[269] Misha E. Kilmer and Carla D. Martin. Factorization strategies for third-order tensors. Linear Algebra and its Applications, 435(3):641–658, 2011.

[270] Misha E. Kilmer, Carla D. Martin, and Lisa Perrone. A third-order generalization of the matrix svd as a product of third-order tensors. Tufts University, Department of Computer Science, Tech. Rep. TR-2008-4, 2008.

[271] Youngkyu Kim, Youngsoo Choi, David Widemann, and Tarek Zohdi. A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder. Journal of Computational Physics, 451:110841, 2022.

[272] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[273] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114, 2013.

[274] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.

[275] Tamara G. Kolda. Multilinear operators for higher-order decompositions. Sandia Report SAND2006-2081, April 2006.

[276] Tamara G. Kolda and Brett W. Bader. Tensor Decompositions and Applications. SIAM Review, 51(3):455–500, August 2009.

[277] Tamara G. Kolda, Robert Michael Lewis, and Virginia Torczon. Optimization by direct search: New perspectives on some classical and modern methods. SIAM Review, 45(3):385–482, 2003.

[278] Tamara G. Kolda and Dianne P. O'leary. Algorithm 805: Computation and uses of the semidiscrete matrix decomposition. ACM Transactions on Mathematical Software (TOMS), 26(3):415–435, 2000.

[279] Tamara G. Kolda and Jimeng Sun. Scalable tensor decompositions for multi-aspect data mining. In 2008 Eighth IEEE International Conference on Data Mining, pages 363–372. IEEE, 2008.

[280] Tonu Kollo. Advanced multivariate statistics with matrices. Springer, 2005.

[281] Phaedon-Stelios Koutsourelakis. Accurate uncertainty quantification using inaccurate computational models. SIAM Journal on Scientific Computing, 31(5):3274–3300, 2009.

[282] Reka A. Kovacs, Oktay Gunluk, and Raphael A. Hauser. Binary Matrix Factorisation via Column Generation. arXiv preprint arXiv:2011.04457, 2020.

[283] Mehmet Koyutürk and Ananth Grama. PROXIMUS: A framework for analyzing very high dimensional discrete-attributed datasets. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 147–156, 2003.

[284] David Kozak, Stephen Becker, Alireza Doostan, and Luis Tenorio. Stochastic subspace descent. arXiv preprint arXiv:1904.01145, 2019.

[285] David Kozak, Stephen Becker, Alireza Doostan, and Luis Tenorio. A stochastic subspace approach to gradient-free optimization in high dimensions. Computational Optimization and Applications, 79(2):339–368, 2021.

[286] David Kozak, Cesare Molinari, Lorenzo Rosasco, Luis Tenorio, and Silvia Villa. Zeroth-order optimization with orthogonal random directions. Mathematical Programming, 199(1):1179–1219, 2023.

[287] Boris Kramer, Alexandre Noll Marques, Benjamin Peherstorfer, Umberto Villa, and Karen Willcox. Multifidelity probability estimation via fusion of estimators. Journal of Computational Physics, 392:385–402, 2019.

[288] Alex Kulesza and Ben Taskar. Determinantal point processes for machine learning. arXiv preprint arXiv:1207.6083, 2012.

[289] Ravi Kumar, Rina Panigrahy, Ali Rahimi, and David Woodruff. Faster algorithms for binary matrix factorization. In International Conference on Machine Learning, pages 3551–3559, 2019.

[290] Srijan Kumar, William L. Hamilton, Jure Leskovec, and Dan Jurafsky. Community interaction and conflict on the web. In Proceedings of the 2018 World Wide Web Conference, pages 933–943. International World Wide Web Conferences Steering Committee, 2018.

[291] Srijan Kumar, Francesca Spezzano, V. S. Subrahmanian, and Christos Faloutsos. Edge weight prediction in weighted signed networks. In 2016 IEEE 16th International Conference on Data Mining (ICDM), pages 221–230. IEEE, 2016.

[292] Jérôme Kunegis. Konect: The koblenz network collection. In Proceedings of the 22nd International Conference on World Wide Web, pages 1343–1350. ACM, 2013.

[293] Nolan Kurtz and Junho Song. Cross-entropy-based adaptive importance sampling using gaussian mixture. Structural Safety, 42:35–44, 2013.

[294] Julian D. Laderman. A noncommutative algorithm for multiplying 3x3 matrices using 23 multiplications. Bulletin of the American Mathematical Society, 82(1):126–128, 1976.

[295] J. M. Landsberg. Tensors: Geometry and Applications. American Mathematical Society, Providence, R.I, December 2011.

[296] Joseph M. Landsberg. Tensors: Geometry and applications. Representation theory, 381(402):3, 2012.

[297] Brett W. Larsen and Tamara G. Kolda. Practical Leverage-Based Sampling for Low-Rank Tensor Decomposition. arXiv preprint arXiv:2006.16438v3, 2020.

[298] Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. Annals of Statistics, pages 1302–1338, 2000.

[299] Frédéric Lavancier, Jesper Møller, and Ege Rubak. Determinantal point process models and statistical inference. Journal of the Royal Statistical Society: Series B: Statistical Methodology, pages 853–877, 2015.

[300] Loic Le Gratiet. Multi-fidelity Gaussian process regression for computer experiments. PhD thesis, Université Paris-Diderot-Paris VII, 2013.

[301] Olivier Le Maître and Omar M Knio. Spectral methods for uncertainty quantification: with applications to computational fluid dynamics. Springer Science & Business Media, 2010.

[302] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[303] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755):788–791, 1999.

[304] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In Advances in Neural Information Processing Systems, pages 556–562, 2001.

[305] Kookjin Lee and Kevin T Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. Journal of Computational Physics, 404:108973, 2020.

[306] Tao Lei, Yu Xin, Yuan Zhang, Regina Barzilay, and Tommi Jaakkola. Low-rank tensors for scoring dependency structures. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1381–1391, 2014.

[307] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pages 177–187. ACM, 2005.

[308] Jiajia Li, Casey Battaglino, Ioakeim Perros, Jimeng Sun, and Richard Vuduc. An input-adaptive and in-place approach to dense tensor-times-matrix multiply. In SC'15: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pages 1–12. IEEE, 2015.

[309] Jiajia Li, Yuchen Ma, Chenggang Yan, and Richard Vuduc. Optimizing sparse tensor times matrix on multi-core and many-core architectures. In 2016 6th Workshop on Irregular Applications: Architecture and Algorithms (IA3), pages 26–33. IEEE, 2016.

[310] Jing Li, Jinglai Li, and Dongbin Xiu. An efficient surrogate-based method for computing rare failure probability. Journal of Computational Physics, 230(24):8683–8697, 2011.

[311] Jing Li and Dongbin Xiu. Evaluation of failure probability via surrogate models. Journal of Computational Physics, 229(23):8966–8980, 2010.

[312] Jundong Li, Harsh Dani, Xia Hu, Jiliang Tang, Yi Chang, and Huan Liu. Attributed network embedding for learning in a dynamic environment. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pages 387–396, 2017.

[313] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926, 2017.

[314] Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. ACM/IMS Journal of Data Science, 2021.

[315] Lifan Liang, Kunju Zhu, and Songjian Lu. BEM: Mining Coregulation Patterns in Transcriptomics via Boolean Matrix Factorization. Bioinformatics, 36(13):4030–4037, 2020.

[316] Edo Liberty, Franco Woolfe, Per-Gunnar Martinsson, Vladimir Rokhlin, and Mark Tygert. Randomized algorithms for the low-rank approximation of matrices. Proceedings of the National Academy of Sciences, 104(51):20167–20172, December 2007.

[317] Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 73(4):423–498, 2011.

[318] John Lipor and Laura Balzano. Clustering quality metrics for subspace clustering. Pattern Recognition, 104:107328, 2020.

[319] John Lipor, David Hong, Yan Shuo Tan, and Laura Balzano. Subspace clustering using ensembles of K-subspaces. arXiv preprint arXiv:1709.04744, 2021.

[320] Bangtian Liu, Chengyao Wen, Anand D. Sarwate, and Maryam Mehri Dehnavi. A unified optimization approach for sparse tensor operations on gpus. In 2017 IEEE International Conference on Cluster Computing (CLUSTER), pages 47–57. IEEE, 2017.

[321] Chunchen Liu, Lu Feng, Ryohei Fujimaki, and Yusuke Muraoka. Scalable model selection for large-scale factorial relational models. In International Conference on Machine Learning, pages 1227–1235, 2015.

[322] Ding Liu, Shi-Ju Ran, Peter Wittek, Cheng Peng, Raul Blázquez García, Gang Su, and Maciej Lewenstein. Machine learning by unitary tensor network of hierarchical tree structure. New Journal of Physics, 21(7):073059, 2019.

[323] Feng Liu, Wenkai Xu, Jie Lu, Guangquan Zhang, Arthur Gretton, and Danica J. Sutherland. Learning deep kernels for non-parametric two-sample tests. International Conference on Machine Learning, 2020.

[324] Penghuan Liu. Label-free STORM principle realized by super-Rayleigh speckle in photoacoustic imaging. Optics Letters, 44(19):4642–4645, October 2019.

[325] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. Advances in neural information processing systems, 29, 2016.

[326] Sijia Liu, Pin-Yu Chen, Bhavya Kailkhura, Gaoyuan Zhang, Alfred O Hero III, and Pramod K Varshney. A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications. IEEE Signal Processing Magazine, 37(5):43–54, 2020.

[327] Xien Liu, Xinxin You, Xiao Zhang, Ji Wu, and Ping Lv. Tensor graph convolutional networks for text classification. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 8409–8416, 2020.

[328] Yan Liu, Chi Zhang, and Lihong V. Wang. Effects of light scattering on optical-resolution photoacoustic microscopy. Journal of Biomedical Optics, 17(12):126014, 2012.

[329] Yong Liu, Zirui Zhu, Chaoyu Gong, Minhao Cheng, Cho-Jui Hsieh, and Yang You. Sparse mezo: Less parameters for better performance in zeroth-order llm fine-tuning. arXiv preprint arXiv:2402.15751, 2024.

[330] Benjamin Lochocki, Ksenia Abrashitova, Johannes F. de Boer, and Lyubov V. Amitonova. Ultimate resolution limits of speckle-based compressive imaging. Optics Express, 29(3):3943–3955, 2021.

[331] Canyi Lu, Jiashi Feng, Yudong Chen, Wei Liu, Zhouchen Lin, and Shuicheng Yan. Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5249–5257, 2016.

[332] Claudio Lucchese, Salvatore Orlando, and Raffaele Perego. Mining top-k patterns from binary datasets in presence of noise. In Proceedings of the 2010 SIAM International Conference on Data Mining, pages 165–176. SIAM, 2010.

[333] David J Lucia, Philip S Beran, and Walter A Silva. Reduced-order modeling: New approaches for computational physics. Progress in Aerospace Sciences, 40(1-2):51–117, 2004.

[334] Kathryn Lund. The tensor t-function: A definition for functions of third-order tensors. arXiv preprint arXiv:1806.07261, 2018.

[335] Hanbaek Lyu, Deanna Needell, and Laura Balzano. Online matrix factorization for Markovian data and applications to Network Dictionary Learning. Journal of Machine Learning Research, 21(251):1–49, 2020.

[336] Malik Magdon-Ismail. Row sampling for matrix algorithms via a non-commutative Bernstein bound. arXiv preprint arXiv:1008.0587, 2010.

[337] Michael W. Mahoney. Randomized algorithms for matrices and data. Foundations and Trends in Machine Learning, 3(2):123–224, 2011.

[338] Michael W. Mahoney and Petros Drineas. CUR matrix decompositions for improved data analysis. Proceedings of the National Academy of Sciences, 106(3):697–702, 2009.

[339] Michael W. Mahoney, Mauro Maggioni, and Petros Drineas. Tensor-CUR decompositions for tensor-based data. SIAM Journal on Matrix Analysis and Applications, 30(3):957–987, 2008.

[340] M. Mahsereci and P. Hennig. Probabilistic line searches for stochastic optimization. Journal of Machine Learning Research, 18(119):1–59, 2017.

[341] Konstantin Makarychev, Yury Makarychev, and Ilya Razenshteyn. Performance of Johnson–Lindenstrauss transform for k-means and k-medians clustering. In Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, pages 1027–1038. ACM, 2019.

[342] Osman Asif Malik and Stephen Becker. Low-Rank Tucker Decomposition of Large Tensors Using TensorSketch. In Advances in Neural Information Processing Systems, pages 10096–10106, 2018.

[343] Osman Asif Malik and Stephen Becker. Fast randomized matrix and tensor interpolative decomposition using CountSketch. Advances in Computational Mathematics, 46(6):76, October 2020.

[344] Osman Asif Malik and Stephen Becker. Guarantees for the Kronecker fast Johnson–Lindenstrauss transform using a coherence and sampling argument. Linear Algebra and its Applications, 602:120–137, October 2020.

[345] Osman Asif Malik and Stephen Becker. A Sampling Based Method for Tensor Ring Decomposition. arXiv preprint arXiv:2010.08581, 2020.

[346] Osman Asif Malik and Stephen Becker. Randomization of approximate bilinear computation for matrix multiplication. International Journal of Computer Mathematics: Computer Systems Theory, 6(1):54–93, 2021.

[347] Osman Asif Malik, Shashanka Ubaru, Lior Horesh, Misha E. Kilmer, and Haim Avron. Dynamic graph convolutional networks using the tensor m-product. In Proceedings of the 2021 SIAM International Conference on Data Mining (SDM), pages 729–737, 2021.

[348] Osman Asif Malik, Hayato Ushijima-Mwesigwa, Arnab Roy, Avradip Mandal, and Indradeep Ghosh. Binary Matrix Factorization on Special Purpose Hardware. arXiv preprint arXiv:2010.08693, 2020.

[349] Osman Asif Malik, Yiming Xu, Nuojin Cheng, Stephen Becker, Alireza Doostan, and Akil Narayan. Fast algorithms for monotone lower subsets of kronecker least squares problems, 2022. Preprint.

[350] Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. Advances in Neural Information Processing Systems, 36:53038–53075, 2023.

[351] Franco Manessi, Alessandro Rozza, and Mario Manzo. Dynamic graph convolutional networks. Pattern Recognition, page 107000, 2019.

[352] Carla D. Martin, Richard Shafer, and Betsy LaRue. An order-p tensor factorization with applications in imaging. SIAM Journal on Scientific Computing, 35(1):A474–A490, 2013.

[353] Carla D. Martin, Richard Shafer, and Betsy LaRue. An order-p tensor factorization with applications in imaging. SIAM Journal on Scientific Computing, 35(1):A474–A490, 2013.

[354] Per-Gunnar Martinsson, Vladimir Rokhlin, and Mark Tygert. A Randomized Algorithm for the Approximation of Matrices. Technical Report YALEU/DCS/TR-1361, June 2006.

[355] Per-Gunnar Martinsson, Vladimir Rokhlin, and Mark Tygert. A randomized algorithm for the decomposition of matrices. Applied and Computational Harmonic Analysis, 30(1):47–68, January 2011.

[356] Per-Gunnar Martinsson and Joel Tropp. Randomized numerical linear algebra: Foundations & algorithms. arXiv preprint arXiv:2002.01387, 2020.

[357] Per-Gunnar Martinsson and Joel A. Tropp. Randomized numerical linear algebra: Foundations and algorithms. Acta Numerica, 29:403–572, 2020.

[358] Jiří Matoušek. On variants of the Johnson–Lindenstrauss lemma. Random Structures & Algorithms, 33(2):142–156, 2008.

[359] Romit Maulik, Bethany Lusch, and Prasanna Balaprakash. Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. Physics of Fluids, 33(3):037106, 2021.

[360] Effrosyni Mavroudi, Benjamín Béjar Haro, and René Vidal. Representation Learning on Visual-Symbolic Graphs for Video Understanding. In European Conference on Computer Vision, pages 71–90. Springer, 2020.

[361] Edward Meeds, Zoubin Ghahramani, Radford M. Neal, and Sam T. Roweis. Modeling dyadic data with binary latent factors. In Advances in Neural Information Processing Systems, pages 977–984, 2007.

[362] Raghu Meka, Oanh Nguyen, and Van Vu. Anti-concentration for polynomials of independent random variables. arXiv preprint arXiv:1507.00829, 2015.

[363] Xuhui Meng and George Em Karniadakis. A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse pde problems. Journal of Computational Physics, 401:109020, 2020.

[364] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. The journal of chemical physics, 21(6):1087–1092, 1953.

[365] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. The journal of chemical physics, 21(6):1087–1092, 1953.

[366] J. V. Michalowicz, J. M. Nichols, F. Bucholtz, and C. C. Olson. An Isserlis' theorem for mixed Gaussian variables: Application to the auto-bispectral density. Journal of Statistical Physics, 136(1):89–102, 2009.

[367] Oscar Mickelin and Sertac Karaman. On algorithms for and computing with the tensor ring decomposition. Numerical Linear Algebra with Applications, 27(3):e2289, 2020.

[368] Pauli Miettinen. Boolean tensor factorizations. In 2011 IEEE 11th International Conference on Data Mining, pages 447–456. IEEE, 2011.

[369] Pauli Miettinen, Taneli Mielikäinen, Aristides Gionis, Gautam Das, and Heikki Mannila. The discrete basis problem. IEEE transactions on knowledge and data engineering, 20(10):1348–1362, 2008.

[370] Pauli Miettinen and Jilles Vreeken. Model order selection for boolean matrix factorization. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 51–59, 2011.

[371] Ashley Milsted, Martin Ganahl, Stefan Leichenauer, Jack Hidary, and Guifre Vidal. TensorNetwork on TensorFlow: A Spin Chain Application Using Tree Tensor Networks. arXiv preprint arXiv:1905.01331, 2019.

[372] Junhong Min, Jaeduck Jang, Dongmin Keum, Seung-Wook Ryu, Chulhee Choi, Ki-Hun Jeong, and Jong Chul Ye. Fluorescent microscopy beyond diffraction limits using speckle illumination and joint support recovery. Scientific reports, 3(1):1–6, 2013.

[373] Rachel Minster, Arvind K. Saibaba, and Misha E. Kilmer. Randomized algorithms for low-rank tensor decompositions in the Tucker format. SIAM Journal on Mathematics of Data Science, 2(1):189–215, 2020.

[374] Max D Morris, Toby J Mitchell, and Donald Ylvisaker. Bayesian design and analysis of computer experiments: use of derivatives in surface prediction. Technometrics, 35(3):243–255, 1993.

[375] Rajeev Motwani and Prabhakar Raghavan. Randomized Algorithms. Cambridge university press, 1995.

[376] Emeric Mudry, Kamal Belkebir, J. Girard, Julien Savatier, Emmeran Le Moal, C. Nicoletti, Marc Allain, and Anne Sentenac. Structured illumination microscopy using unknown speckle patterns. Nature Photonics, 6(5):312–315, 2012.

[377] Robb J. Muirhead. Aspects of Multivariate Statistical Theory, volume 197. John Wiley & Sons, 1982.

[378] Kevin P. Murphy. Probabilistic Machine Learning: Advanced Topics. MIT Press, 2023.

[379] Todd W. Murray, Markus Haltmeier, Thomas Berer, Elisabeth Leiss-Holzinger, and Peter Burgholzer. Super-resolution photoacoustic microscopy using blind structured illumination. Optica, 4(1):17–22, 2017.

[380] Akil Narayan, Claude Gittelson, and Dongbin Xiu. A Stochastic Collocation Algorithm with Multifidelity Models. SIAM Journal on Scientific Computing, 36(2):A495–A521, 2014.

[381] Akil Narayan, Claude Gittelson, and Dongbin Xiu. A stochastic collocation algorithm with multifidelity models. SIAM Journal on Scientific Computing, 36(2):A495–A521, 2014.

[382] Akil Narayan, John Jakeman, and Tao Zhou. A Christoffel function weighted least squares algorithm for collocation approximations. Mathematics of Computation, 86(306):1913–1947, 2017. arXiv: 1412.4305 [math.NA].

[383] Christian F. A. Negre, Hayato Ushijima-Mwesigwa, and Susan M. Mniszewski. Detecting multiple communities using quantum annealing on the D-Wave system. Plos one, 15(2):e0227538, 2020.

[384] Sameer A. Nene, Shree K. Nayar, and Hiroshi Murase. Columbia Object Image Library (COIL-100). Technical Report CUCS-006-96, Columbia University, 1996.

[385] Yurii Nesterov. Introductory lectures on convex optimization: A basic course, volume 87. Springer Science & Business Media, 2013.

[386] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. Foundations of Computational Mathematics, 17(2):527–566, 2017.

[387] Felix Newberry, Jerrad Hampton, Kenneth Jansen, and Alireza Doostan. Bi-fidelity reduced polynomial chaos expansion for uncertainty quantification. Computational Mechanics, 69(2):405–424, 2022.

[388] Elizabeth Newman, Lior Horesh, Haim Avron, and Misha Kilmer. Stable Tensor Neural Networks for Rapid Deep Learning. arXiv preprint arXiv:1811.06569, 2018.

[389] Leo WT Ng and Karen E Willcox. Multifidelity approaches for optimization under uncertainty. International Journal for numerical methods in Engineering, 100(10):746–772, 2014.

[390] Giang Hoang Nguyen, John Boaz Lee, Ryan A. Rossi, Nesreen K. Ahmed, Eunyee Koh, and Sungchul Kim. Continuous-time dynamic network embeddings. In Companion Proceedings of the The Web Conference 2018, pages 969–976, 2018.

[391] Lam M Nguyen, Katya Scheinberg, and Trang H Tran. Stochastic ISTA/FISTA adaptive step search algorithms for convex composite optimization. Journal of Optimization Theory and Applications, 205(1):10, 2025.

[392] Stefanos Nikolopoulos, Ioannis Kalogeris, and Vissarion Papadopoulos. Non-intrusive surrogate modeling for parametrized time-dependent partial differential equations using convolutional autoencoders. Engineering Applications of Artificial Intelligence, 109:104652, 2022.

[393] Jonas Nitzler, Jonas Biehler, Niklas Fehn, Phaedon-Stelios Koutsourelakis, and Wolfgang A Wall. A generalized probabilistic learning approach for multi-fidelity uncertainty quantification in complex physical simulations. Computer Methods in Applied Mechanics and Engineering, 400:115600, 2022.

[394] Fabio Nobile, Raúl Tempone, and Clayton G Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. SIAM Journal on Numerical Analysis, 46(5):2309–2345, 2008.

[395] Jorge Nocedal and Stephen J Wright. Numerical optimization. Springer, 1999.

[396] Ahmed K Noor and Jeanne M Peters. Reduced basis technique for nonlinear analysis of structures. Aiaa journal, 18(4):455–462, 1980.

[397] Alexander Novikov, Dmitrii Podoprikhin, Anton Osokin, and Dmitry P. Vetrov. Tensorizing neural networks. In Advances in Neural Information Processing Systems, pages 442–450, 2015.

[398] Alexander Novikov, Mikhail Trofimov, and Ivan Oseledets. Exponential machines. arXiv preprint arXiv:1605.03795, 2016.

[399] Alexander Novikov, Mikhail Trofimov, and Ivan Oseledets. Exponential machines. arXiv preprint arXiv:1605.03795, 2017.

[400] Jinoh Oh, Kijung Shin, Evangelos E. Papalexakis, Christos Faloutsos, and Hwanjo Yu. S-hot: Scalable high-order tucker decomposition. In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, pages 761–770, 2017.

[401] Dianne O'Leary and Shmuel Peleg. Digital image compression by outer product expansion. IEEE Transactions on Communications, 31(3):441–444, 1983.

[402] Daniel O'Malley and Velimir V. Vesselinov. ToQ. jl: A high-level programming language for D-Wave machines based on Julia. In 2016 IEEE High Performance Extreme Computing Conference (HPEC), pages 1–7. IEEE, 2016.

[403] Daniel O'Malley, Velimir V. Vesselinov, Boian S. Alexandrov, and Ludmil B. Alexandrov. Nonnegative/binary matrix factorization with a d-wave quantum annealer. PloS one, 13(12):e0206653, 2018.

[404] Greg Ongie, Daniel Pimentel-Alarcón, Laura Balzano, Rebecca Willett, and Robert D. Nowak. Tensor methods for nonlinear matrix completion. arXiv preprint arXiv:1804.10266, 2020.

[405] Ivan Oseledets and Eugene Tyrtyshnikov. TT-cross approximation for multidimensional arrays. Linear Algebra and its Applications, 432(1):70–88, 2010.

[406] Ivan V. Oseledets. Approximation of $2^d \times 2^d$ matrices using tensor decomposition. SIAM Journal on Matrix Analysis and Applications, 31(4):2130–2145, 2010.

[407] Ivan V. Oseledets. Tensor-train decomposition. SIAM Journal on Scientific Computing, 33(5):2295–2317, 2011.

[408] Ivan V. Oseledets, D. V. Savostianov, and Eugene E. Tyrtyshnikov. Tucker dimensionality reduction of three-dimensional arrays in linear time. SIAM Journal on Matrix Analysis and Applications, 30(3):939–956, 2008.

[409] Daniele Ottaviani and Alfonso Amendola. Low rank non-negative matrix factorization with d-wave 2000q. arXiv preprint arXiv:1808.08721, 2018.

[410] Govinda Anantha Padmanabha and Nicholas Zabaras. Solving inverse problems using conditional invertible neural networks. Journal of Computational Physics, 433:110194, 2021.

[411] Rasmus Pagh. Compressed Matrix Multiplication. ACM Transactions on Computation Theory, 5(3):9:1–9:17, August 2013.

[412] Christopher C. Paige and Michael A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. ACM Transactions on Mathematical Software (TOMS), 8(1):43–71, 1982.

[413] Dimitri J Papageorgiou, Jan Kronqvist, and Krishnan Kumaran. Linewalker: line search for black box derivative-free optimization and surrogate model construction. Optimization and Engineering, pages 1–65, 2024.

[414] Iason Papaioannou, Sebastian Geyer, and Daniel Straub. Improved cross entropy-based importance sampling with a flexible mixture model. Reliability Engineering & System Safety, 191:106564, 2019.

[415] Iason Papaioannou, Costas Papadimitriou, and Daniel Straub. Sequential importance sampling for structural reliability analysis. Structural safety, 62:66–75, 2016.

[416] Evangelos E. Papalexakis, Christos Faloutsos, and Nicholas D. Sidiropoulos. Tensors for data mining and data fusion: Models, applications, and scalable algorithms. ACM Transactions on Intelligent Systems and Technology (TIST), 8(2):1–44, 2016.

[417] Leonard Papenmeier, Nuojin Cheng, Stephen Becker, and Luigi Nardi. Exploring exploration in bayesian optimization. arXiv preprint arXiv:2502.08208, 2025.

[418] Courtney Paquette and Katya Scheinberg. A stochastic line search method with expected complexity analysis. SIAM Journal on Optimization, 30(1):349–376, 2020.

[419] Panos M. Pardalos, Ding-Zhu Du, and Ronald L. Graham. Handbook of Combinatorial Optimization. Springer, 2013.

[420] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, and Charles E. Leisersen. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. arXiv preprint arXiv:1902.10191, 2019.

[421] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32, 2019.

[422] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. Journal of machine learning research, 12(Oct):2825–2830, 2011.

[423] B. Peherstorfer, K. Willcox, and M. Gunzburger. Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization. SIAM Review, 60(3):550–591, 2018.

[424] Benjamin Peherstorfer, Tiangang Cui, Youssef Marzouk, and Karen Willcox. Multifidelity importance sampling. Computer Methods in Applied Mechanics and Engineering, 300:490–509, 2016.

[425] Benjamin Peherstorfer, Boris Kramer, and Karen Willcox. Combining multiple surrogate models to accelerate failure probability estimation with expensive high-fidelity models. Journal of Computational Physics, 341:61–75, 2017.

[426] Benjamin Peherstorfer, Boris Kramer, and Karen Willcox. Multifidelity preconditioning of the cross-entropy method for rare event simulation and failure probability estimation. SIAM/ASA Journal on Uncertainty Quantification, 6(2):737–761, 2018.

[427] Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. Siam Review, 60(3):550–591, 2018.

[428] D. M. Penfold. Monte carlo methods. by j. m. hammersley and d. c. hands-comb. pp. 178. 21s. 1964. (methuen, monographs on applied statistics and probability.). The Mathematical Gazette, 51(378):361–362, 1967.

[429] Ji Peng, Jerrad Hampton, and Alireza Doostan. A weighted ℓ_1-minimization approach for sparse polynomial chaos expansions. Journal of Computational Physics, 267:92–111, 2014.

[430] Ji Peng, Jerrad Hampton, and Alireza Doostan. A weighted ℓ_1-minimization approach for sparse polynomial chaos expansions. Journal of Computational Physics, 267:92–111, 2014.

[431] P. Perdikaris, M. Raissi, A. Damianou, N. D. Lawrence, and G. E. Karniadakis. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 473(2198):20160751, February 2017.

[432] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 701–710, 2014.

[433] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3D human motion synthesis with transformer VAE. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10985–10995, 2021.

[434] Robert NC Pfeifer, Jutho Haegeman, and Frank Verstraete. Faster identification of optimal contraction sequences for tensor networks. Physical Review E, 90(3):033315, 2014.

[435] Ninh Pham and Rasmus Pagh. Fast and Scalable Polynomial Kernels via Explicit Feature Maps. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '13, pages 239–247, New York, NY, USA, 2013. ACM.

[436] Allan Pinkus. N-widths in Approximation Theory, volume 7. Springer Science & Business Media, 2012.

[437] Orazio Pinti and Assad A Oberai. Graph laplacian-based spectral multi-fidelity modeling. Scientific Reports, 13(1):16618, 2023.

[438] Boris T Polyak. Introduction to optimization. 1987.

[439] Amela Prelić, Stefan Bleuler, Philip Zimmermann, Anja Wille, Peter Bühlmann, Wilhelm Gruissem, Lars Hennig, Lothar Thiele, and Eckart Zitzler. A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics, 22(9):1122–1129, 2006.

[440] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. Advances in neural information processing systems, 20, 2007.

[441] Sai Rajeswar, Sandeep Subramanian, Francis Dutil, Christopher Pal, and Aaron Courville. Adversarial generation of natural language. arXiv preprint arXiv:1705.10929, 2017.

[442] Beheshteh T. Rakhshan and Guillaume Rabusseau. Tensorized Random Projections. arXiv preprint arXiv:2003.05101, 2020.

[443] Ignacio Ramírez. Binary matrix factorization via dictionary learning. IEEE journal of selected topics in signal processing, 12(6):1253–1262, 2018.

[444] Siamak Ravanbakhsh, Barnabás Póczos, and Russell Greiner. Boolean Matrix Factorization and Noisy Completion via Message Passing. In ICML, volume 69, pages 945–954, 2016.

[445] Mani Razi, Robert Mike Kirby, and Akil Narayan. Kernel optimization for low-rank multifidelity algorithms. International Journal for Uncertainty Quantification, 11(1), 2021.

[446] Sidney I. Resnick. A Probability Path. Modern Birkhäuser Classics. Birkhäuser Basel, 2014.

[447] Matthew J. Reynolds, Gregory Beylkin, and Alireza Doostan. Optimization via separated representations and the canonical tensor decomposition. Journal of Computational Physics, 348(C):220–230, November 2017.

[448] Matthew J. Reynolds, Alireza Doostan, and Gregory Beylkin. Randomized Alternating Least Squares for Canonical Tensor Decompositions: Application to A PDE With Random Data. SIAM Journal on Scientific Computing, 38(5):A2634–A2664, September 2016.

[449] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In International Conference on Machine Learning, pages 1530–1538. PMLR, 2015.

[450] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In International Conference on Machine Learning, pages 1278–1286. PMLR, 2014.

[451] Roberto Rigamonti, Amos Sironi, Vincent Lepetit, and Pascal Fua. Learning separable filters. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2754–2761, 2013.

[452] Alexander Ritchie, Laura Balzano, and Clayton Scott. Supervised PCA: A Multiobjective Approach. arXiv preprint arXiv:2011.05309, 2020.

[453] Herbert Robbins and Sutton Monro. A stochastic approximation method. The annals of mathematical statistics, pages 400–407, 1951.

[454] Chase Roberts, Ashley Milsted, Martin Ganahl, Adam Zalcman, Bruce Fontaine, Yijian Zou, Jack Hidary, Guifre Vidal, and Stefan Leichenauer. TensorNetwork: A Library for Physics and Machine Learning. arXiv preprint arXiv:1905.01330, 2019.

[455] Gareth O Roberts and Osnat Stramer. Langevin diffusions and metropolis-hastings algorithms. Methodology and computing in applied probability, 4:337–357, 2002.

[456] Gareth O Roberts and Richard L Tweedie. Exponential convergence of langevin distributions and their discrete approximations. Bernoulli, pages 341–363, 1996.

[457] Peter J Rossky, Jimmie D Doll, and Harold L Friedman. Brownian dynamics as smart monte carlo simulation. The Journal of Chemical Physics, 69(10):4628–4633, 1978.

[458] Aravind Sankar, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. Dynamic graph representation learning via self-attention networks. arXiv preprint arXiv:1812.09430, 2018.

[459] Martin Schaefer. Note on the k-dimensional Jensen inequality. The Annals of Probability, pages 502–504, 1976.

[460] Christopher Schinnerl. PyMF - Python Matrix Factorization Module. https://github.com/ChrisSchinnerl/pymf3.

[461] Warren Schudy and Maxim Sviridenko. Concentration and moment inequalities for polynomials of independent random variables. In Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, pages 437–446. SIAM, 2012.

[462] Oylum Şeker, Neda Tanoumand, and Merve Bodur. Digital Annealer for quadratic unconstrained binary optimization: A comparative performance analysis. arXiv preprint arXiv:2012.12264, 2020.

[463] Oguz Semerci, Ning Hao, Misha E. Kilmer, and Eric L. Miller. Tensor-based formulation and nuclear norm regularization for multienergy computed tomography. IEEE Transactions on Image Processing, 23(4):1678–1693, 2014.

[464] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks. In International Conference on Neural Information Processing, pages 362–373. Springer, 2018.

[465] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks. In International Conference on Neural Information Processing, pages 362–373. Springer, 2018.

[466] Pranay Seshadri, Akil Narayan, and Sankaran Mahadevan. Effectively subsampled quadratures for least squares polynomial approximations. SIAM/ASA Journal on Uncertainty Quantification, 5(1):1003–1023, 2017.

[467] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P. Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. Proceedings of the IEEE, 104(1):148–175, January 2015.

[468] Ohad Shamir. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. Journal of Machine Learning Research, 18(52):1–11, 2017.

[469] Ruslan Shaydulin, Hayato Ushijima-Mwesigwa, Ilya Safro, Susan Mniszewski, and Yuri Alexeev. Community detection across emerging quantum architectures. 3rd International Workshop on Post Moore's Era Supercomputing (PMES 2018), 2018.

[470] Alexander Shekhovtsov, Dmitrij Schlesinger, and Boris Flach. VAE approximation error: ELBO and exponential families. arXiv preprint arXiv:2102.09310, 2021.

[471] Bao-Hong Shen, Shuiwang Ji, and Jieping Ye. Mining discrete patterns via binary matrix factorization. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 757–766, 2009.

[472] Navid Shervani-Tabar and Nicholas Zabaras. Physics-constrained predictive molecular latent space discovery with graph scattering variational autoencoder. arXiv preprint arXiv:2009.13878, 2020.

[473] Yang Shi and Animashree Anandkumar. Higher-order Count Sketch: Dimensionality Reduction That Retains Efficient Tensor Operations. arXiv, 2019.

[474] Yang Shi, Uma Naresh Niranjan, Animashree Anandkumar, and Cris Cecka. Tensor contractions with extended blas kernels on cpu and gpu. In 2016 IEEE 23rd International Conference on High Performance Computing (HiPC), pages 193–202. IEEE, 2016.

[475] David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Processing Magazine, 30(3):83–98, May 2013.

[476] Boris Shustin and Haim Avron. Randomized Riemannian Preconditioning for Quadratically Constrained Problems. arXiv:1902.01635 [cs, math], February 2019.

[477] Paz Fink Shustin, Shashanka Ubaru, Vasileios Kalantzis, Lior Horesh, and Haim Avron. PCENet: High dimensional surrogate modeling for learning uncertainty. arXiv preprint arXiv:2202.05063, 2022.

[478] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. arXiv preprint arXiv:1703.00810, 2017.

[479] Martin Slawski, Matthias Hein, and Pavlo Lutsik. Matrix factorization with binary components. In Advances in Neural Information Processing Systems, pages 3210–3218, 2013.

[480] A. V. Smirnov. The bilinear complexity and practical algorithms for matrix multiplication. Computational Mathematics and Mathematical Physics, 53(12):1781–1795, 2013.

[481] Ralph C Smith. Uncertainty quantification: theory, implementation, and applications, volume 12. Siam, 2013.

[482] Ralph C. Smith. Uncertainty Quantification: Theory, Implementation, and Applications. SIAM-Society for Industrial and Applied Mathematics, Philadelphia, December 2013.

[483] Shaden Smith, Kejun Huang, Nicholas D. Sidiropoulos, and George Karypis. Streaming tensor factorization for infinite data sources. In Proceedings of the 2018 SIAM International Conference on Data Mining, pages 81–89. SIAM, 2018.

[484] Shaden Smith, Jongsoo Park, and George Karypis. Sparse tensor factorization on many-core processors with high-bandwidth memory. In 2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pages 1058–1067. IEEE, 2017.

[485] Shaden Smith, Niranjay Ravindran, Nicholas D. Sidiropoulos, and George Karypis. SPLATT: Efficient and parallel sparse tensor-matrix multiplication. In 2015 IEEE International Parallel and Distributed Processing Symposium, pages 61–70. IEEE, 2015.

[486] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456, 2020.

[487] Zhao Song, David P. Woodruff, and Peilin Zhong. Low Rank Approximation with Entrywise L1-norm Error. In Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, pages 688–701, New York, NY, USA, 2017. ACM.

[488] Zhao Song, David P. Woodruff, and Peilin Zhong. Relative Error Tensor Low Rank Approximation. arXiv:1704.08246 [cs], April 2017.

[489] James C Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE transactions on automatic control, 37(3):332–341, 1992.

[490] James C Spall. Implementation of the simultaneous perturbation algorithm for stochastic optimization. IEEE Transactions on aerospace and electronic systems, 34(3):817–823, 1998.

[491] Mandavilli Srinivas and Lalit M. Patnaik. Genetic algorithms: A survey. Computer, 27(6):17–26, June 1994.

[492] Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet. Hilbert space embeddings and metrics on probability measures. The Journal of Machine Learning Research, 11:1517–1561, 2010.

[493] Harald Steck. Autoencoders that don't overfit towards the identity. Advances in Neural Information Processing Systems, 33:19598–19608, 2020.

[494] Gilbert W. Stewart. Matrix Perturbation Theory. Citeseer, 1990.

[495] E. Miles Stoudenmire and David J. Schwab. Supervised learning with quantum-inspired tensor networks. arXiv preprint arXiv:1605.05775.

[496] E. Miles Stoudenmire and David J. Schwab. Supervised Learning with Quantum-Inspired Tensor Networks. arXiv:1605.05775 [cond-mat, stat], May 2017.

[497] E. Miles Stoudenmire and David J. Schwab. Supervised Learning with Quantum-Inspired Tensor Networks. arXiv:1605.05775 [cond-mat, stat], May 2017.

[498] Volker Strassen. Gaussian Elimination is not Optimal. Numerische Mathematik, 13(4):354–356, August 1969.

[499] Felipe Petroski Such, Shagan Sah, Miguel Alexander Dominguez, Suhas Pillai, Chao Zhang, Andrew Michael, Nathan D. Cahill, and Raymond Ptucha. Robust spatial filtering with graph convolutional neural networks. IEEE Journal of Selected Topics in Signal Processing, 11(6):884–896, 2017.

[500] Jimeng Sun, Dacheng Tao, Spiros Papadimitriou, Philip S. Yu, and Christos Faloutsos. Incremental tensor analysis: Theory and applications. ACM Transactions on Knowledge Discovery from Data (TKDD), 2(3):1–37, 2008.

[501] Tianxiang Sun, Zhengfu He, Hong Qian, Yunhua Zhou, Xuan-Jing Huang, and Xipeng Qiu. Bbtv2: Towards a gradient-free future with large language models. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 3916–3930, 2022.

[502] Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In International Conference on Machine Learning, pages 20841–20855. PMLR, 2022.

[503] Yiming Sun, Yang Guo, Charlene Luo, Joel Tropp, and Madeleine Udell. Low-Rank Tucker Approximation of a Tensor From Streaming Data. arXiv preprint arXiv:1904.10951, 2019.

[504] Yiming Sun, Yang Guo, Charlene Luo, Joel Tropp, and Madeleine Udell. Low-rank tucker approximation of a tensor from streaming data. SIAM Journal on Mathematics of Data Science, 2(4):1123–1150, 2020.

[505] Yiming Sun, Yang Guo, Joel A. Tropp, and Madeleine Udell. Tensor random projection for low memory dimension reduction. In NeurIPS Workshop on Relational Representation Learning, 2018.

[506] Robert H. Swendsen and Jian-Sheng Wang. Replica Monte Carlo simulation of spin-glasses. Physical review letters, 57(21):2607, 1986.

[507] Robert H. Swendsen and Jian-Sheng Wang. Replica Monte Carlo simulation of spin-glasses. Physical review letters, 57(21):2607, 1986.

[508] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2818–2826, 2016.

[509] Shion Takeno, Hitoshi Fukuoka, Yuhki Tsukada, Toshiyuki Koyama, Motoki Shiga, Ichiro Takeuchi, and Masayuki Karasuyama. Multi-fidelity bayesian optimization with max-value entropy search and its parallelization. In International Conference on Machine Learning, pages 9334–9345. PMLR, 2020.

[510] Rong Tang and Yun Yang. On the computational complexity of metropolis-adjusted langevin algorithms for bayesian posterior sampling. arXiv preprint arXiv:2206.06491, 2022.

[511] Davoud Ataee Tarzanagh and George Michailidis. Fast Randomized Algorithms for t-Product Based Tensor Operations and Decompositions with Applications to Imaging Data. SIAM Journal on Imaging Sciences, 11(4):2629–2664, 2018.

[512] Jozef L. Teugels. Some representations of the multivariate Bernoulli and binomial distributions. Journal of multivariate analysis, 32(2):256–268, 1990.

[513] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. arXiv preprint physics/0004057, 2000.

[514] Bradley E. Treeby and Benjamin T. Cox. K-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields. Journal of biomedical optics, 15(2):021314, 2010.

[515] Lloyd N. Trefethen and David Bau III. Numerical Linear Algebra, volume 50. Siam, 1997.

[516] Rohit K Tripathy and Ilias Bilionis. Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification. Journal of Computational Physics, 375:565–588, 2018.

[517] Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In International Conference on Machine Learning, pages 3462–3471. PMLR, 2017.

[518] Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In International Conference on Learning Representations, 2019.

[519] Charalampos E. Tsourakakis. Mach: Fast randomized tensor decompositions. In Proceedings of the 2010 SIAM International Conference on Data Mining, pages 689–700. SIAM, 2010.

[520] Eugene E. Tyrtyshnikov. Incomplete Cross Approximation in the Mosaic-Skeleton Method. Computing, 64:367–380, 2000.

[521] Shashanka Ubaru, Lior Horesh, and Guy Cohen. Dynamic graph based epidemiological model for COVID-19 contact tracing data analysis and optimal testing prescription. arXiv preprint arXiv:2009.04971, 2020.

[522] Felipe Uribe, Iason Papaioannou, Youssef M Marzouk, and Daniel Straub. Cross-entropy-based importance sampling with failure-informed dimension reduction for rare event simulation. SIAM/ASA Journal on Uncertainty Quantification, 9(2):818–847, 2021.

[523] Hayato Ushijima-Mwesigwa, Christian F. A. Negre, and Susan M. Mniszewski. Graph partitioning using quantum annealing on the D-Wave system. In Proceedings of the Second International Workshop on Post Moores Era Supercomputing, pages 22–29. ACM, 2017.

[524] Charles F. Van Loan. The ubiquitous Kronecker product. Journal of Computational and Applied Mathematics, 123(1):85–100, November 2000.

[525] M. Alex O. Vasilescu and Demetri Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In European Conference on Computer Vision, pages 447–460. Springer, 2002.

[526] Sharan Vaswani, Aaron Mishkin, Issam Laradji, Mark Schmidt, Gauthier Gidel, and Simon Lacoste-Julien. Painless stochastic gradient: Interpolation, line-search, and convergence rates. Advances in neural information processing systems, 32, 2019.

[527] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. arXiv preprint arXiv:1011.3027, 2010.

[528] Roman Vershynin. High-Dimensional Probability: An Introduction with Applications in Data Science, volume 47. Cambridge University Press, 2018.

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

[530] Cédric Villani. Topics in optimal transportation, volume 58. American Mathematical Soc., 2021.

[531] Sergey Vilov, Bastien Arnal, and Emmanuel Bossy. Overcoming the acoustic diffraction limit in photoacoustic imaging by the localization of flowing absorbers. Optics letters, 42(21):4379–4382, 2017.

[532] Sergey Vilov, Bastien Arnal, Eliel Hojman, Yonina C. Eldar, Ori Katz, and Emmanuel Bossy. Super-resolution photoacoustic and ultrasound imaging with sparse arrays. Scientific reports, 10(1):1–8, 2020.

[533] Sergey Vilov, Guillaume Godefroy, Bastien Arnal, and Emmanuel Bossy. Photoacoustic fluctuation imaging: Theory and application to blood flow imaging. Optica, 7(11):1495–1505, 2020.

[534] Sergey Voronin. LowRankMatrixDecompositionCodes. https://github.com/sergeyvoronin, February 2017.

[535] Sergey Voronin and Per-Gunnar Martinsson. RSVDPACK: An implementation of randomized algorithms for computing the singular value, interpolative, and CUR decompositions of matrices on multi-core and GPU architectures. arXiv:1502.05366 [cs, math], February 2015.

[536] Sergey Voronin and Per-Gunnar Martinsson. Efficient algorithms for CUR and interpolative matrix decompositions. Advances in Computational Mathematics, 43(3):495–516, June 2017.

[537] Fabian Wagner, Jonas Latz, Iason Papaioannou, and Elisabeth Ullmann. Multilevel sequential importance sampling for rare event estimation. SIAM Journal on Scientific Computing, 42(4):A2062–A2087, 2020.

[538] Fabian Wagner, Iason Papaioannou, and Elisabeth Ullmann. The ensemble kalman filter for rare event estimation. SIAM/ASA Journal on Uncertainty Quantification, 10(1):317–349, 2022.

[539] M. Mitchell Waldrop. The chips are down for Moore's law. Nature News, 530(7589):144, 2016.

[540] M. Mitchell Waldrop. The chips are down for Moore's law. Nature News, 530(7589):144, 2016.

[541] Kun Wang, Sergey A. Ermilov, Richard Su, Hans-Peter Brecht, Alexander A. Oraevsky, and Mark A. Anastasio. An imaging model incorporating ultrasonic transducer properties for three-dimensional optoacoustic tomography. IEEE transactions on medical imaging, 30(2):203–214, 2011.

[542] Lihong V. Wang. Tutorial on photoacoustic microscopy and computed tomography. IEEE Journal of Selected Topics in Quantum Electronics, 14(1):171–179, 2008.

[543] Naigang Wang, Jungwook Choi, Daniel Brand, Chia-Yu Chen, and Kailash Gopalakrishnan. Training deep neural networks with 8-bit floating point numbers. In Advances in Neural Information Processing Systems, pages 7675–7684, 2018.

[544] Shusen Wang and Zhihua Zhang. Improving CUR matrix decomposition and the Nyström approximation via adaptive sampling. The Journal of Machine Learning Research, 14(1):2729–2769, 2013.

[545] Weiran Wang, Xinchen Yan, Honglak Lee, and Karen Livescu. Deep variational canonical correlation analysis. arXiv preprint arXiv:1610.03454, 2016.

[546] Wenqi Wang, Vaneet Aggarwal, and Shuchin Aeron. Efficient low rank tensor ring completion. In Proceedings of the IEEE International Conference on Computer Vision, pages 5697–5705, 2017.

[547] Yang Wang, Zhipeng Lü, Fred Glover, and Jin-Kao Hao. Backbone guided tabu search for solving the UBQP problem. Journal of Heuristics, 19(4):679–695, 2013.

[548] Yining Wang and Aarti Singh. Provably Correct Algorithms for Matrix Column Subset Selection with Selectively Sampled Data. Journal of Machine Learning Research, 18(156):1–42, 2018.

[549] Yining Wang, Hsiao-Yu Tung, Alexander J. Smola, and Anima Anandkumar. Fast and guaranteed tensor decomposition via sketching. In Advances in Neural Information Processing Systems, pages 991–999, 2015.

[550] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. arXiv preprint arXiv:1801.07829, 2018.

[551] Zhengwei Wang, Qi She, and Tomas E Ward. Generative adversarial networks in computer vision: A survey and taxonomy. ACM Computing Surveys (CSUR), 54(2):1–38, 2021.

[552] Mark Weber, Jie Chen, Toyotaro Suzumura, Aldo Pareja, Tengfei Ma, Hiroki Kanezashi, Tim Kaler, Charles E. Leiserson, and Tao B. Schardl. Scalable Graph Learning for Anti-Money Laundering: A First Look. arXiv preprint arXiv:1812.00076, 2018.

[553] Gian-Carlo Wick. The evaluation of the collision matrix. Physical review, 80(2):268, 1950.

[554] Wikipedia. Tensor contraction. Wikipedia, October 2020.

[555] Wikipedia. Tensor product. Wikipedia, October 2020.

[556] Christopher KI Williams and Carl Edward Rasmussen. Gaussian processes for machine learning, volume 2. MIT press Cambridge, MA, 2006.

[557] David P. Woodruff. Sketching as a tool for numerical linear algebra. Foundations and Trends in Theoretical Computer Science, 10(1-2):1–157, 2014.

[558] Franco Woolfe, Edo Liberty, Vladimir Rokhlin, and Mark Tygert. A fast randomized algorithm for the approximation of matrices. Applied and Computational Harmonic Analysis, 25(3):335–366, November 2008.

[559] Jian Wu, Saul Toscano-Palmerin, Peter I Frazier, and Andrew Gordon Wilson. Practical multi-fidelity bayesian optimization for hyperparameter tuning. In Uncertainty in Artificial Intelligence, pages 788–798. PMLR, 2020.

[560] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A comprehensive survey on graph neural networks. arXiv preprint arXiv:1901.00596, 2019.

[561] Matt Wytock and Zico Kolter. Sparse gaussian conditional random fields: Algorithms, theory, and application to energy forecasting. In International conference on machine learning, pages 1265–1273. PMLR, 2013.

[562] Jun Xia, Junjie Yao, and Lihong V. Wang. Photoacoustic tomography: Principles and advances. Electromagnetic waves (Cambridge, Mass.), 147:1–22, 2014.

[563] Shifeng Xiong, Peter ZG Qian, and CF Jeff Wu. Sequential design and analysis of high-accuracy and low-accuracy computer codes. Technometrics, 55(1):37–46, 2013.

[564] Dongbin Xiu and Jan S. Hesthaven. High-Order Collocation Methods for Differential Equations with Random Inputs. SIAM Journal on Scientific Computing, 27(3):1118–1139, 2005.

[565] Dongbin Xiu and George Em Karniadakis. The Wiener–Askey Polynomial Chaos for Stochastic Differential Equations. SIAM Journal on Scientific Computing, 24(2):619–644, 2002.

[566] Dongbin Xiu and George Em Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. SIAM journal on scientific computing, 24(2):619–644, 2002.

[567] Miao Xu, Rong Jin, and Zhi-Hua Zhou. CUR algorithm for partially observed matrices. In International Conference on Machine Learning, pages 1412–1421, 2015.

[568] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[569] Bo Yang, Ahmed Zamzam, and Nicholas D. Sidiropoulos. ParaSketch: Parallel Tensor Factorization via Sketching. In Proceedings of the 2018 SIAM International Conference on Data Mining, pages 396–404. SIAM, 2018.

[570] Yinchong Yang, Denis Krompass, and Volker Tresp. Tensor-train recurrent neural networks for video classification. arXiv preprint arXiv:1707.01786, 2017.

[571] Junjie Yao and Lihong V. Wang. Photoacoustic microscopy. Laser & photonics reviews, 7(5):758–778, 2013.

[572] Jinmian Ye, Linnan Wang, Guangxi Li, Di Chen, Shandian Zhe, Xinqi Chu, and Zenglin Xu. Learning compact recurrent neural networks with block-term tensor decomposition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9378–9387, 2018.

[573] Jinmian Ye, Linnan Wang, Guangxi Li, Di Chen, Shandian Zhe, Xinqi Chu, and Zenglin Xu. Learning compact recurrent neural networks with block-term tensor decomposition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9378–9387, 2018.

[574] Ke Ye and Lek-Heng Lim. Tensor network ranks. arXiv preprint arXiv:1801.02662, 2018.

[575] Li-Hao Yeh, Lei Tian, and Laura Waller. Structured illumination microscopy with unknown patterns and a statistical prior. Biomedical optics express, 8(2):695–711, 2017.

[576] Rose Yu, Stephan Zheng, Anima Anandkumar, and Yisong Yue. Long-term forecasting using higher order tensor RNNs. arXiv preprint arXiv:1711.00073, 2017.

[577] Ziming Yu, Pan Zhou, Sike Wang, Jia Li, and Hua Huang. Subzero: Random subspace zeroth-order optimization for memory-efficient llm fine-tuning. arXiv preprint arXiv:2410.08989, 2024.

[578] Longhao Yuan, Chao Li, Jianting Cao, and Qibin Zhao. Randomized tensor ring decomposition and its application to large-scale data reconstruction. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2127–2131. IEEE, 2019.

[579] Yao Yue and Karl Meerbergen. Accelerating optimization of parametric linear systems by model order reduction. SIAM Journal on Optimization, 23(2):1344–1370, 2013.

[580] Dejiao Zhang and Laura Balzano. Convergence of a Grassmannian gradient descent algorithm for subspace estimation from undersampled data. arXiv preprint arXiv:1610.00199, 2018.

[581] Jiani Zhang, Arvind K. Saibaba, Misha E. Kilmer, and Shuchin Aeron. A randomized tensor singular value decomposition based on the t-product. Numerical Linear Algebra with Applications, 25:e2179, 2018.

[582] Tong Zhang, Wenming Zheng, Zhen Cui, and Yang Li. Tensor graph convolutional neural network. arXiv preprint arXiv:1803.10071, 2018.

[583] Xinshuai Zhang, Fangfang Xie, Tingwei Ji, Zaoxu Zhu, and Yao Zheng. Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization. Computer Methods in Applied Mechanics and Engineering, 373:113485, 2021.

[584] Yihua Zhang, Pingzhi Li, Junyuan Hong, Jiaxiang Li, Yimeng Zhang, Wenqing Zheng, Pin-Yu Chen, Jason D. Lee, Wotao Yin, Mingyi Hong, Zhangyang Wang, Sijia Liu, and Tianlong Chen. Revisiting zeroth-order optimization for memory-efficient llm fine-tuning: A benchmark, 2024.

[585] Zemin Zhang and Shuchin Aeron. Exact tensor completion using t-SVD. IEEE Transactions on Signal Processing, 65(6):1511–1526, 2016.

[586] Zemin Zhang, Gregory Ely, Shuchin Aeron, Ning Hao, and Misha Kilmer. Novel methods for multilinear data completion and de-noising based on tensor-SVD. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3842–3849, 2014.

[587] Zhenyue Zhang and Hongyuan Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. SIAM Journal on Scientific Computing, 26(1):313–338, 2004.

[588] Zhong-Yuan Zhang, Tao Li, Chris Ding, Xian-Wen Ren, and Xiang-Sun Zhang. Binary matrix factorization for analyzing gene expression data. Data Mining and Knowledge Discovery, 20(1):28, 2010.

[589] Zhongyuan Zhang, Tao Li, Chris Ding, and Xiangsun Zhang. Binary matrix factorization with applications. In Seventh IEEE International Conference on Data Mining (ICDM 2007), pages 391–400. IEEE, 2007.

[590] Ling Zhao, Yujiao Song, Min Deng, and Haifeng Li. Temporal graph convolutional network for urban traffic flow prediction method. arXiv preprint arXiv:1811.05320, 2018.

[591] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. IEEE Transactions on Intelligent Transportation Systems, 2019.

[592] Qibin Zhao, Guoxu Zhou, Shengli Xie, Liqing Zhang, and Andrzej Cichocki. Tensor ring decomposition. arXiv preprint arXiv:1606.05535, 2016.

[593] Guoxu Zhou, Andrzej Cichocki, and Shengli Xie. Decomposition of big tensors with low multilinear rank. arXiv preprint arXiv:1412.1885, 2014.

[594] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Graph neural networks: A review of methods and applications. arXiv preprint arXiv:1812.08434, 2018.

[595] Lekui Zhou, Yang Yang, Xiang Ren, Fei Wu, and Yueting Zhuang. Dynamic network embedding by modeling triadic closure process. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 32, 2018.

[596] Tao Zhou, Akil Narayan, and Dongbin Xiu. Weighted discrete least-squares polynomial approximation using randomized quadratures. Journal of Computational Physics, 298:787–800, 2015.

[597] Tong Zhou and Yongbo Peng. Kernel principal component analysis-based Gaussian process regression modelling for high-dimensional reliability analysis. Computers & Structures, 241:106358, 2020.

[598] Xueyu Zhu, Akil Narayan, and Dongbin Xiu. Computational Aspects of Stochastic Collocation with Multifidelity Models. SIAM/ASA Journal on Uncertainty Quantification, 2(1):444–463, 2014.

[599] Xueyu Zhu, Akil Narayan, and Dongbin Xiu. Computational aspects of stochastic collocation with multifidelity models. SIAM/ASA Journal on Uncertainty Quantification, 2(1):444–463, 2014.

[600] Yinhao Zhu and Nicholas Zabaras. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. Journal of Computational Physics, 366:415–447, 2018.

[601] Yinhao Zhu, Nicholas Zabaras, Phaedon-Stelios Koutsourelakis, and Paris Perdikaris. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. Journal of Computational Physics, 394:56–81, 2019.

[602] Marinka Zitnik and Blaz Zupan. Nimfa: A python library for nonnegative matrix factorization. Journal of Machine Learning Research, 13:849–853, 2012.

[603] Étienne de Montbrun and Sébastien Gerchinovitz. Certified multifidelity zeroth-order optimization. SIAM/ASA Journal on Uncertainty Quantification, 12(4):1135–1164, December 2024.

Appendix A

Bi-fidelity Sampling

A.1 Efficient leverage score sampling of certain design matrices

In this section, we describe the key elements of the sampling approach developed in [349] as it applies to the problems we consider in this paper. The discussion here will consider the design matrices discussed in Section **??**. Using the same notation as in that section, define the matrices $\boldsymbol{A}_k$ for $k \in [q]$ elementwise via

$$\boldsymbol{A}_k(n_k, j_k) = \sqrt{w_{k,n_k}}\psi_{j_k}(p_{k,n_k}), \quad n_k \in [N_k],\ j_k \in [\zeta]. \tag{A.1}$$

Next, define

$$\boldsymbol{A}_{\mathrm{TP}} \coloneqq \boldsymbol{A}_1 \otimes \cdots \otimes \boldsymbol{A}_q, \tag{A.2}$$

where $\otimes$ denotes the Kronecker product; see Section 12.3 of [190] for a definition. The design matrices corresponding to total degree and hyperbolic cross polynomial spaces discussed in Section **??** are made up of a subset of the columns of $\boldsymbol{A}_{\mathrm{TP}}$. In particular, using Matlab indexing notation, they can be written as

$$\boldsymbol{A} = \boldsymbol{A}_{\mathrm{TP}}(:, \boldsymbol{v}), \tag{A.3}$$

where $\boldsymbol{v}$ is a vector containing distinct column indices of $\boldsymbol{A}_{\mathrm{TP}}$. The sampling scheme we discuss requires the additional assumption that the entries in $\boldsymbol{v}$ are arranged in increasing order. The columns of $\boldsymbol{A}$ can always be permuted to ensure that this is possible when $\boldsymbol{A}$ corresponds to a total degree or hyperbolic cross space. Such a permutation will not change the least squares problem

since it will only permute the order of the entries in the solution vector, and is therefore something that can always be done.

Note that a column index c of $\boldsymbol{A}_{\mathrm{TP}}$ corresponds to a multi-index $(c_1, \ldots, c_q)$ such that

$$\boldsymbol{A}_{\mathrm{TP}}(:, c) = \boldsymbol{A}_1(:, c_1) \otimes \cdots \otimes \boldsymbol{A}_q(:, c_q). \tag{A.4}$$

Each row index r of $\boldsymbol{A}_{\mathrm{TP}}$ corresponds to a multi-index $(r_1, \ldots, r_q)$ in a similar fashion.

Algorithm 9 outlines the sampling algorithm. We provide some intuition for why the algorithm works and refer the reader to [349] for a rigorous treatment. Note that $\boldsymbol{A}$ is full rank and therefore $\mathrm{rank}(\boldsymbol{A}) = d$. Let $\boldsymbol{Q}\boldsymbol{R} = \boldsymbol{A}$ be a compact QR decomposition (i.e., such that $\boldsymbol{Q}$ has d columns and $\boldsymbol{R}$ has d rows). Recall that the leverage score sampling distribution satisfies

$$\boldsymbol{p}(i) = \frac{\|\boldsymbol{Q}(i, :)\|_2^2}{d}. \tag{A.5}$$

Instead of drawing a sample according to the distribution above, we may instead draw a single column $\boldsymbol{Q}(:, j)$ of $\boldsymbol{Q}$ uniformly at random and instead draw a sample according to the probability distribution defined by $\tilde{\boldsymbol{p}}_j(i) = (\boldsymbol{Q}(i, j))^2$. To see this, let $\tilde{I}$ be a random row index drawn according to this alternate strategy. Moreover, let $J \sim \mathrm{Uniform}([d])$ be the random column index, and let $\tilde{I}_j$ be a random row index drawn according to $\tilde{\boldsymbol{p}}_j$. Then we have

$$\mathcal{P}(\tilde{I} = i) = \sum_{j=1}^{d} \mathcal{P}(\tilde{I} = i \mid J = j)\, \mathcal{P}(J = j) = \sum_{j=1}^{d} \mathcal{P}(\tilde{I}_j = i)\, \mathcal{P}(J = j) = \sum_{j=1}^{d} (\boldsymbol{Q}(i, j))^2 \, \frac{1}{d} = \frac{\|\boldsymbol{Q}(i, :)\|_2^2}{d} = \boldsymbol{p}(i). \tag{A.6}$$

This shows that the alternate sampling strategy indeed draws samples according to the leverage score sampling distribution. This is the sampling strategy that our algorithm uses. Moreover, it uses two additional fact:

(1) When $\boldsymbol{A}$ has the particular structure assumed in this section, then the cth column of $\boldsymbol{Q}$ satisfies

$$\boldsymbol{Q}(:, c) = \boldsymbol{Q}_1(:, c_1) \otimes \cdots \otimes \boldsymbol{Q}_q(:, c_q), \tag{A.7}$$

where $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_q$ are defined in line 2 in Algorithm 9.

(2) Due to (A.7), drawing a row index r according to $\tilde{\boldsymbol{p}}_j$ is equivalent to drawing a multi-index $(r_1, \ldots, r_q)$ according to a product distribution with each r_k drawn independently according to the distribution $((\boldsymbol{Q}_k(r_k, j_k))^2)_{r_k}$ where $(j_1, \ldots, j_q)$ is the column multi-index corresponding to j.

Fact (i) makes it possible to sample according to the alternate sampling strategy without every needing to compute the QR decomposition of the large matrix $\boldsymbol{A}$. A more general version of this fact appears in Proposition 4.4 of [349]. Fact (ii) further makes it possible to sample according to $\tilde{\boldsymbol{p}}_j$ without needing to form that probability vector which is of length $\prod_k N_k$.

Algorithm 9: Efficient leverage score sampling of total degree and hyperbolic cross design matrices

Input: Matrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_q$, index vector $\boldsymbol{v}$, number of samples m
Output: Vector $\boldsymbol{s} \in [\prod_k N_k]^m$ of m samples drawn from row indices of $\boldsymbol{A}$

for $k \in [q]$ **do**
  Compute compact QR decomposition $\boldsymbol{Q}_k \boldsymbol{R}_k = \boldsymbol{A}_k$
end for
for $i \in [m]$ **do**
  Draw an entry j from $\boldsymbol{v}$ uniformly at random
  Compute the multi-index $(j_1, \ldots, j_q)$ corresponding to j
  for $k \in [q]$ **do**
    Construct the probability distribution $\boldsymbol{p} = ((\boldsymbol{Q}_k(r_k, j_k))^2)_{r_k} \in \mathbb{R}^{N_k}$
    Draw an index $r_k \in [N_k]$ according to the distribution $\boldsymbol{p}$
  end for
  Set the ith sample $\boldsymbol{s}(i)$ equal the row index corresponding to the row multi-index $(r_1, \ldots, r_q)$
end for
return Vector of samples $\boldsymbol{s}$

A.2 Proof of Theorem 1.4.5

The proof of Theorem 1.4.5 relies on the following lemmas:

Lemma A.2.1. *Let X and Y be two (nonconstant) random variables defined on the same probability space. The correlation coefficient between X and Y, $\mathrm{corr}(X, Y)$, is bounded from below as*

$$\mathrm{corr}(X, Y) \geq \sqrt{\frac{\mathbb{V}[X]}{\mathbb{V}[Y]}} - \sqrt{\frac{\mathbb{V}[Y - X]}{\mathbb{V}[Y]}}. \tag{A.8}$$

Proof. It follows from direct computation that

$$\begin{aligned}\operatorname{corr}(X,Y) &= \frac{\mathbb{E}[XY]-\mathbb{E}[X]\mathbb{E}[Y]}{\sqrt{\mathbb{V}[X]\mathbb{V}[Y]}} \\ &= \frac{\mathbb{E}[X^2]-\mathbb{E}[X]^2}{\sqrt{\mathbb{V}[X]\mathbb{V}[Y]}} + \frac{\mathbb{E}[X(Y-X)]-\mathbb{E}[X]\mathbb{E}[Y-X]}{\sqrt{\mathbb{V}[X]\mathbb{V}[Y]}} \\ &= \sqrt{\frac{\mathbb{V}[X]}{\mathbb{V}[Y]}} + \operatorname{corr}(X,Y-X)\sqrt{\frac{\mathbb{V}[Y-X]}{\mathbb{V}[Y]}} \\ &\geq \sqrt{\frac{\mathbb{V}[X]}{\mathbb{V}[Y]}} - \sqrt{\frac{\mathbb{V}[Y-X]}{\mathbb{V}[Y]}},\end{aligned} \tag{A.9}$$

where the last inequality uses $\operatorname{corr}(X, Y-X) \geq -1$. □

Lemma A.2.2. *Let $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_n)$ be a standard Gaussian vector in $\mathbb{R}^n$. For any $\boldsymbol{w}, \boldsymbol{z} \in \mathbb{R}^n$,*

$$\mathbb{E}[\langle \boldsymbol{w}, \boldsymbol{\xi}\rangle^2 \langle \boldsymbol{z}, \boldsymbol{\xi}\rangle^2] = 2\langle \boldsymbol{w}, \boldsymbol{z}\rangle^2 + \|\boldsymbol{w}\|_2^2 \|\boldsymbol{z}\|_2^2. \tag{A.10}$$

Proof. The proof follows from a direct application of Wick's formula [553]. Denote $X_1 = \langle \boldsymbol{w}, \boldsymbol{\xi}\rangle$ and $X_2 = \langle \boldsymbol{z}, \boldsymbol{\xi}\rangle$. It is easy to verify that

$$\begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{K}), \quad \boldsymbol{K} = \begin{pmatrix} \|\boldsymbol{w}\|_2^2 & \langle \boldsymbol{w}, \boldsymbol{z}\rangle \\ \langle \boldsymbol{w}, \boldsymbol{z}\rangle & \|\boldsymbol{z}\|_2^2 \end{pmatrix}. \tag{A.11}$$

By Wick's formula,

$$\mathbb{E}[\langle \boldsymbol{w}, \boldsymbol{\xi}\rangle^2 \langle \boldsymbol{z}, \boldsymbol{\xi}\rangle^2] = \mathbb{E}[X_1^2 X_2^2] = 2\mathbb{E}[X_1 X_2]^2 + \mathbb{E}[X_1^2]\mathbb{E}[X_2^2] = 2\langle \boldsymbol{w}, \boldsymbol{z}\rangle^2 + \|\boldsymbol{w}\|_2^2 \|\boldsymbol{z}\|_2^2. \tag{A.12}$$

□

Proof of Theorem 1.4.5. Since correlation coefficients are scale-invariant, and both $\boldsymbol{b}$ and $\tilde{\boldsymbol{b}}$ are fixed,

$$\operatorname{corr}(\mu^2(\boldsymbol{b}, \boldsymbol{S}), \mu^2(\tilde{\boldsymbol{b}}, \boldsymbol{S})) = \operatorname{corr}\left(\frac{r_{\boldsymbol{S}}^2(\boldsymbol{A}, \boldsymbol{b}) - r^2(\boldsymbol{A}, \boldsymbol{b})}{\|\boldsymbol{b}\|_2^2}, \frac{r_{\boldsymbol{S}}^2(\boldsymbol{A}, \tilde{\boldsymbol{b}}) - r^2(\boldsymbol{A}, \tilde{\boldsymbol{b}})}{\|\tilde{\boldsymbol{b}}\|_2^2} \right). \tag{A.13}$$

Without loss of generality, we assume $\|\boldsymbol{b}\|_2 = \|\tilde{\boldsymbol{b}}\|_2 = 1$, so that $\boldsymbol{b}_{\mathcal{P}} = \boldsymbol{b}$, $\tilde{\boldsymbol{b}}_{\mathcal{P}} = \tilde{\boldsymbol{b}}$.

Let

$$\begin{aligned} X &= r_{\boldsymbol{S}}^2(\boldsymbol{A}, \boldsymbol{b}) - r^2(\boldsymbol{A}, \boldsymbol{b}) = \|(\boldsymbol{S}\boldsymbol{Q})^\dagger \boldsymbol{S}\boldsymbol{Q}_\perp \boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2^2 \\ Y &= r_{\boldsymbol{S}}^2(\boldsymbol{A}, \tilde{\boldsymbol{b}}) - r^2(\boldsymbol{A}, \tilde{\boldsymbol{b}}) = \|(\boldsymbol{S}\boldsymbol{Q})^\dagger \boldsymbol{S}\boldsymbol{Q}_\perp \boldsymbol{Q}_\perp^T \tilde{\boldsymbol{b}}\|_2^2. \end{aligned} \tag{A.14}$$

To apply Lemma A.2.1, it suffices to estimate $\mathbb{V}[X]/\mathbb{V}[Y]$ and $\mathbb{V}[Y-X]/\mathbb{V}[Y]$.

First of all, due to the rotation invariance of joint Gaussians,

$$\begin{aligned} \boldsymbol{G}_1 &\coloneqq \sqrt{m}\boldsymbol{S}\boldsymbol{Q} \in \mathbb{R}^{m\times d}, \\ \boldsymbol{G}_2 &\coloneqq \sqrt{m}\boldsymbol{S}\boldsymbol{Q}_\perp \in \mathbb{R}^{m\times(N-d)} \end{aligned} \tag{A.15}$$

are independent Gaussian random matrices, i.e., $(\boldsymbol{S}\boldsymbol{Q})^\dagger \boldsymbol{S}\boldsymbol{Q}_\perp \boldsymbol{Q}_\perp^T = \boldsymbol{G}_1^\dagger \boldsymbol{G}_2 \boldsymbol{Q}_\perp^T$, and

$$\begin{aligned} \mathbb{E}[X] &= \mathbb{E}\left[\mathrm{tr}\Big(\boldsymbol{G}_1^\dagger \boldsymbol{G}_2 \boldsymbol{Q}_\perp^T \boldsymbol{b}\boldsymbol{b}^T \boldsymbol{Q}_\perp \boldsymbol{G}_2^T {\boldsymbol{G}_1^\dagger}^T\Big)\right] \\ &= \mathbb{E}\left[\mathrm{tr}\Big(\boldsymbol{G}_1^\dagger \mathbb{E}[\boldsymbol{G}_2 \boldsymbol{Q}_\perp^T \boldsymbol{b}\boldsymbol{b}^T \boldsymbol{Q}_\perp \boldsymbol{G}_2^T] {\boldsymbol{G}_1^\dagger}^T\Big)\right] \\ &= \|\boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2^2 \mathbb{E}\left[\mathrm{tr}\Big(\boldsymbol{G}_1^\dagger {\boldsymbol{G}_1^\dagger}^T\Big)\right] \\ &= \|\boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2^2 \mathbb{E}\left[\mathrm{tr}\big((\boldsymbol{G}_1^T \boldsymbol{G}_1)^{-1}\big)\right], \end{aligned} \tag{A.16}$$

where we have used that $\mathbb{E}[\boldsymbol{G}_2 \boldsymbol{Q}_\perp^T \boldsymbol{b}\boldsymbol{b}^T \boldsymbol{Q}_\perp \boldsymbol{G}_2^T] = \|\boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2^2 \boldsymbol{I}_m$.

Note $\boldsymbol{G}_1^T \boldsymbol{G}_1$ is a Wishart matrix with dimension d and degrees of freedom m, i.e. $\boldsymbol{W} = \boldsymbol{G}_1^T \boldsymbol{G}_1 \sim W_d(\boldsymbol{I}_d, m)$. Consequently, $\mathbb{E}[\boldsymbol{W}^{-1}] = \frac{1}{m-d-1}\boldsymbol{I}_d$ if $m > d+1$, and

$$\mathbb{E}[X] = \|\boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2^2 \frac{d}{m-d-1}. \tag{A.17}$$

Similarly,

$$\mathbb{E}[Y] = \|\boldsymbol{Q}_\perp^T \tilde{\boldsymbol{b}}\|_2^2 \frac{d}{m-d-1}. \tag{A.18}$$

Note $\boldsymbol{G}_1^\dagger \boldsymbol{G}_2 \boldsymbol{Q}_\perp^T \boldsymbol{a} \overset{\mathcal{D}}{=} \|\boldsymbol{Q}_\perp^T \boldsymbol{a}\|_2 (\boldsymbol{G}_1^T \boldsymbol{G}_1)^{-1} \boldsymbol{G}_1^T \boldsymbol{\xi}$ for every $\boldsymbol{a} \in \mathbb{R}^N$, where $\boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_m)$ is independent of $\boldsymbol{G}_1$. If we denote $\boldsymbol{G} = (\boldsymbol{G}_1^T \boldsymbol{G}_1)^{-1} \boldsymbol{G}_1^T$, with rows denoted by $\boldsymbol{g}_i, i \in [d]$, then

$$\begin{aligned} \mathbb{E}[X^2] &= \mathbb{E}[\|\|\boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2 \boldsymbol{G}\boldsymbol{\xi}\|_2^4] \\ &= \|\boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2^4 \mathbb{E}\left[\left(\sum_{i=1}^d \langle \boldsymbol{g}_i, \boldsymbol{\xi}\rangle^2\right)^2\right] \\ &= \|\boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2^4 \left(\sum_{i=1}^d \mathbb{E}[\langle \boldsymbol{g}_i, \boldsymbol{\xi}\rangle^4] + \sum_{i\neq j} \mathbb{E}[\langle \boldsymbol{g}_i, \boldsymbol{\xi}\rangle^2 \langle \boldsymbol{g}_j, \boldsymbol{\xi}\rangle^2]\right) \\ &\overset{\text{(A.10)}}{=} \|\boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2^4 \left(3\sum_{i=1}^d \mathbb{E}[\|\boldsymbol{g}_i\|_2^4] + \sum_{i\neq j} (2\mathbb{E}[\langle \boldsymbol{g}_i, \boldsymbol{g}_j\rangle^2] + \mathbb{E}[\|\boldsymbol{g}_i\|_2^2 \|\boldsymbol{g}_j\|_2^2])\right) \\ &= \|\boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2^4 \left(2\mathbb{E}[\|\boldsymbol{G}\boldsymbol{G}^T\|_F^2] + \mathbb{E}[\mathrm{tr}\big(\boldsymbol{G}\boldsymbol{G}^T\big)^2]\right) \\ &= \|\boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2^4 \left(2\mathbb{E}[\|(\boldsymbol{G}_1^T \boldsymbol{G}_1)^{-1}\|_F^2] + \mathbb{E}[\mathrm{tr}\big((\boldsymbol{G}_1^T \boldsymbol{G}_1)^{-1}\big)^2]\right). \end{aligned} \tag{A.19}$$

To explicitly compute (A.19), we use the following moments formulas of inverse Wishart distributions [280, Theorem 2.4.14]:

$$\begin{aligned}\mathbb{E}[\boldsymbol{W}^{-1}\boldsymbol{W}^{-1}] &= \left(\frac{d}{(m-d)(m-d-3)} + \frac{d}{(m-d)(m-d-1)(m-d-3)}\right)\boldsymbol{I}_d \\ \mathrm{Cov}(\boldsymbol{W}_{ii}^{-1}, \boldsymbol{W}_{jj}^{-1}) &= \frac{2+2(m-d-1)\delta_{ij}}{(m-d)(m-d-1)^2(m-d-3)}.\end{aligned} \tag{A.20}$$

Therefore,

$$\mathbb{E}[\|(\boldsymbol{G}_1^T\boldsymbol{G}_1)^{-1}\|_F^2] = \mathrm{tr}\big(\mathbb{E}[\boldsymbol{W}^{-1}\boldsymbol{W}^{-1}]\big) = \frac{d^2}{(m-d-1)(m-d-3)} \simeq \frac{d^2}{(m-d-1)^2} \tag{A.21}$$

and

$$\begin{aligned}\mathbb{E}[\mathrm{tr}\big((\boldsymbol{G}_1^T\boldsymbol{G}_1)^{-1}\big)^2] &= \sum_{i,j\in[d]} \mathbb{E}[\boldsymbol{W}_{ii}^{-1}\boldsymbol{W}_{jj}^{-1}] \\ &= \sum_{i,j\in[d]} (\mathrm{Cov}(\boldsymbol{W}_{ii}^{-1}, \boldsymbol{W}_{jj}^{-1}) + \mathbb{E}[\boldsymbol{W}_{ii}^{-1}]\mathbb{E}[\boldsymbol{W}_{jj}^{-1}]) \\ &= \frac{d^2}{(m-d-1)^2} + \frac{2d}{(m-d-1)^2(m-d-3)} + \frac{2(d^2-d)}{(m-d)(m-d-1)^2(m-d-3)} \\ &\simeq \frac{d^2}{(m-d-1)^2},\end{aligned} \tag{A.22}$$

where $a_m \simeq b_m$ if $\lim_{m\to\infty} a_m/b_m = 1$. Substituting these back into (A.19) yields

$$\mathbb{E}[X^2] \simeq \|\boldsymbol{Q}_\perp^T\boldsymbol{b}\|_2^4 \frac{3d^2}{(m-d-1)^2}. \tag{A.23}$$

Replacing $\boldsymbol{b}$ by $\tilde{\boldsymbol{b}}$ in the above computation gives a similar estimate for $\mathbb{E}[Y^2]$:

$$\mathbb{E}[Y^2] \simeq \|\boldsymbol{Q}_\perp^T\tilde{\boldsymbol{b}}\|_2^4 \frac{3d^2}{(m-d-1)^2}. \tag{A.24}$$

Combining (A.23), (A.24) with (A.17) and (A.18) produces

$$\begin{aligned}\mathbb{V}[X] &\simeq \|\boldsymbol{Q}_\perp^T\boldsymbol{b}\|_2^4 \frac{2d^2}{(m-d-1)^2}, \\ \mathbb{V}[Y] &\simeq \|\boldsymbol{Q}_\perp^T\tilde{\boldsymbol{b}}\|_2^4 \frac{2d^2}{(m-d-1)^2},\end{aligned} \tag{A.25}$$

which implies

$$\frac{\mathbb{V}[X]}{\mathbb{V}[Y]} \simeq \frac{\|\boldsymbol{Q}_\perp^T\boldsymbol{b}\|_2^4}{\|\boldsymbol{Q}_\perp^T\tilde{\boldsymbol{b}}\|_2^4}. \tag{A.26}$$

On the other hand, using Cauchy–Schwarz inequality, Moreover, we have

$$\begin{aligned}\mathbb{V}[Y-X] &\le \mathbb{E}[(Y-X)^2] \\ &= \mathbb{E}\big[(X^{\frac{1}{2}}+Y^{\frac{1}{2}})^2\cdot(X^{\frac{1}{2}}-Y^{\frac{1}{2}})^2\big] \\ &= \mathbb{E}\big[(X^{\frac{1}{2}}+Y^{\frac{1}{2}})^2\cdot(\|\boldsymbol{G}_1^{\dagger}\boldsymbol{G}_2\boldsymbol{Q}_{\perp}^T\boldsymbol{b}\|_2-\|\boldsymbol{G}_1^{\dagger}\boldsymbol{G}_2\boldsymbol{Q}_{\perp}^T\tilde{\boldsymbol{b}}\|_2)^2\big] \\ &\le \mathbb{E}[(X^{\frac{1}{2}}+Y^{\frac{1}{2}})^2\cdot\|\boldsymbol{G}_1^{\dagger}\boldsymbol{G}_2\boldsymbol{Q}_{\perp}^T(\boldsymbol{b}\pm\tilde{\boldsymbol{b}})\|_2^2],\end{aligned} \tag{A.27}$$

where the last inequality follows from the reverse triangle inequality. Furthermore, using the inequality of arithmetic and geometric means followed by the Cauchy–Schwarz inequality, we have

$$\begin{aligned}&\mathbb{E}[(X^{\frac{1}{2}}+Y^{\frac{1}{2}})^2\cdot\|\boldsymbol{G}_1^{\dagger}\boldsymbol{G}_2\boldsymbol{Q}_{\perp}^T(\boldsymbol{b}\pm\tilde{\boldsymbol{b}})\|_2^2 \\ &\le 2\mathbb{E}[(X+Y)\cdot\|\boldsymbol{G}_1^{\dagger}\boldsymbol{G}_2\boldsymbol{Q}_{\perp}^T(\boldsymbol{b}\pm\tilde{\boldsymbol{b}})\|_2^2] \\ &= 2\mathbb{E}[X\cdot\|\boldsymbol{G}_1^{\dagger}\boldsymbol{G}_2\boldsymbol{Q}_{\perp}^T(\boldsymbol{b}\pm\tilde{\boldsymbol{b}})\|_2^2]+2\mathbb{E}[Y\cdot\|\boldsymbol{G}_1^{\dagger}\boldsymbol{G}_2\boldsymbol{Q}_{\perp}^T(\boldsymbol{b}\pm\tilde{\boldsymbol{b}})\|_2^2] \\ &\le 2\sqrt{\mathbb{E}[X^2]\cdot\mathbb{E}[\|\boldsymbol{G}_1^{\dagger}\boldsymbol{G}_2\boldsymbol{Q}_{\perp}^T(\boldsymbol{b}\pm\tilde{\boldsymbol{b}})\|_2^4]}+2\sqrt{\mathbb{E}[Y^2]\cdot\mathbb{E}[\|\boldsymbol{G}_1^{\dagger}\boldsymbol{G}_2\boldsymbol{Q}_{\perp}^T(\boldsymbol{b}\pm\tilde{\boldsymbol{b}})\|_2^4]}.\end{aligned} \tag{A.28}$$

Combining (A.27) and (A.28) yields

$$\mathbb{V}[Y-X]\le 2\sqrt{\mathbb{E}[X^2]\cdot\mathbb{E}[\|\boldsymbol{G}_1^{\dagger}\boldsymbol{G}_2\boldsymbol{Q}_{\perp}^T(\boldsymbol{b}\pm\tilde{\boldsymbol{b}})\|_2^4]}+2\sqrt{\mathbb{E}[Y^2]\cdot\mathbb{E}[\|\boldsymbol{G}_1^{\dagger}\boldsymbol{G}_2\boldsymbol{Q}_{\perp}^T(\boldsymbol{b}\pm\tilde{\boldsymbol{b}})\|_2^4]}. \tag{A.29}$$

A similar argument as (A.23) shows that

$$\mathbb{E}[\|\boldsymbol{G}_1^{\dagger}\boldsymbol{G}_2\boldsymbol{Q}_{\perp}^T(\boldsymbol{b}\pm\tilde{\boldsymbol{b}})\|_2^4]\simeq\|\boldsymbol{Q}_{\perp}^T(\boldsymbol{b}\pm\tilde{\boldsymbol{b}})\|_2^4\frac{3d^2}{(m-d-1)^2}. \tag{A.30}$$

Plugging (A.30) into (A.29) together with the previous estimates yields that, asymptotically,

$$\frac{\mathbb{V}[Y-X]}{\mathbb{V}[Y]}\le\frac{2\left(\|\boldsymbol{Q}_{\perp}^T\boldsymbol{b}\|_2^2+\|\boldsymbol{Q}_{\perp}^T\tilde{\boldsymbol{b}}\|_2^2\right)\|\boldsymbol{Q}_{\perp}^T(\boldsymbol{b}\pm\tilde{\boldsymbol{b}})\|_2^2}{\|\boldsymbol{Q}_{\perp}^T\tilde{\boldsymbol{b}}\|_2^4}\cdot\frac{3d^2}{2d^2}\le\frac{6\|\boldsymbol{Q}_{\perp}^T(\boldsymbol{b}\pm\tilde{\boldsymbol{b}})\|_2^2}{\|\boldsymbol{Q}_{\perp}^T\tilde{\boldsymbol{b}}\|_2^4}, \tag{A.31}$$

where the last inequality follows from $\|\boldsymbol{b}\|_2=\|\tilde{\boldsymbol{b}}\|_2=1$. Appealing to Lemma A.2.1,

$$\liminf_{m\to\infty}\operatorname{corr}(X,Y)\ge\frac{\|\boldsymbol{Q}_{\perp}^T\boldsymbol{b}\|_2^2-\sqrt{6}\min\{\|\boldsymbol{Q}_{\perp}^T(\boldsymbol{b}\pm\tilde{\boldsymbol{b}})\|_2\}}{\|\boldsymbol{Q}_{\perp}^T\tilde{\boldsymbol{b}}\|_2^2}. \tag{A.32}$$

(1.33) follows by noting $\|\boldsymbol{Q}_{\perp}^T\boldsymbol{a}\|_2=\|\boldsymbol{P}_{\boldsymbol{Q}_{\perp}}\boldsymbol{a}\|_2$ for $\boldsymbol{a}\in\mathbb{R}^N$.

To prove (1.35), we use Proposition 1.4.8 (i.e. (1.44)) to lower bound $\|\boldsymbol{Q}_{\perp}^T\tilde{\boldsymbol{b}}\|_2^2$:

$$\varphi\le\|\boldsymbol{Q}_{\perp}^T\tilde{\boldsymbol{b}}\|_2+\kappa\Longrightarrow(\varphi-\kappa)^2\le\|\boldsymbol{Q}_{\perp}^T\tilde{\boldsymbol{b}}\|_2^2\le\|\tilde{\boldsymbol{b}}\|_2^2=1. \tag{A.33}$$

Also, $\|\boldsymbol{Q}_\perp^T \boldsymbol{b}\|_2^2 = 1 - \kappa^2$ and

$$\min\{\|\boldsymbol{Q}_\perp^T(\boldsymbol{b} \pm \tilde{\boldsymbol{b}})\|_2\} \leq \min\{\|\boldsymbol{b} \pm \tilde{\boldsymbol{b}}\|_2\} = \sqrt{2 - 2\varphi}. \tag{A.34}$$

Hence,

$$\liminf_{m\to\infty} \operatorname{corr}(X, Y) \geq (1 - \kappa^2) - \frac{\sqrt{12(1-\varphi)}}{(\varphi - \kappa)^2}, \tag{A.35}$$

completing the proof. □

A.3 Proof of Theorem 1.4.11

We first prove the case of the sub-Gaussian sketches. According to Lemma 1.4.10, it suffices to verify the conditions (1.52).

Note $\sqrt{m}\boldsymbol{S}\boldsymbol{Q} \in \mathbb{R}^{m\times d}$ is a random matrix whose rows are i.i.d. isotropic random vectors in $\mathbb{R}^d$, with the sub-Gaussian norm $\lesssim K$ (this follows from Definition 3.4.1 and Proposition 2.6.1 in [528]). Applying [528, Theorem 4.6.1] to the matrix $\sqrt{m}\boldsymbol{S}\boldsymbol{Q}$ and using the fact that $\sigma_{\min}(\sqrt{m}\boldsymbol{S}\boldsymbol{Q}) = \sqrt{m}\sigma_{\min}(\boldsymbol{S}\boldsymbol{Q})$, we find that if $m \gtrsim K^4 d \log(4L/\delta)$, then with probability at least $1 - \delta/(2L)$, the first condition in (1.52) is satisfied.

For the second condition in (1.52), we write the i-th component of $\boldsymbol{Q}^T\boldsymbol{S}^T\boldsymbol{S}\boldsymbol{h}$ as

$$\boldsymbol{q}_i^T\boldsymbol{S}^T\boldsymbol{S}\boldsymbol{h} = \frac{1}{m}\sum_{j\in[m]} \langle\sqrt{m}\boldsymbol{s}_j, \boldsymbol{q}_i\rangle\langle\sqrt{m}\boldsymbol{s}_j, \boldsymbol{h}\rangle, \quad i \in [d]. \tag{A.36}$$

Both $\langle\sqrt{m}\boldsymbol{s}_j, \boldsymbol{q}_i\rangle$ and $\langle\sqrt{m}\boldsymbol{s}_j, \boldsymbol{h}\rangle$ are sub-Gaussian random variables [528, Proposition 2.6.1]. Therefore,

$$\|\langle\sqrt{m}\boldsymbol{s}_j, \boldsymbol{q}_i\rangle\langle\sqrt{m}\boldsymbol{s}_j, \boldsymbol{h}\rangle\|_{\psi_1} \leq \|\langle\sqrt{m}\boldsymbol{s}_j, \boldsymbol{q}_i\rangle\|_{\psi_2}\|\langle\sqrt{m}\boldsymbol{s}_j, \boldsymbol{h}\rangle\|_{\psi_2} \leq \|\sqrt{m}\boldsymbol{s}_j\|_{\psi_2}^2 \lesssim K^2, \tag{A.37}$$

where the first inequality follows from [528, Lemma 2.7.7], the second inequality follows from [528, Definition 3.4.1], and the final inequality follows from an application of [528, Proposition 2.6.1]. Moreover, since $\boldsymbol{h} \perp \operatorname{range}(\boldsymbol{Q})$ it is easy to verify that the summands in (A.36) are all zero-mean. By Bernstein's inequality [528, Corollary 2.8.3], if $m \gtrsim K^4 d \log(4dL/\delta)/\varepsilon$, with probability at least

$1-\delta/(2dL)$, $|\boldsymbol{q}_i^T\boldsymbol{S}^T\boldsymbol{S}\boldsymbol{h}| \le \sqrt{\varepsilon/(2d)}$. Taking a union bound over $i \in [d]$ yields that, with probability at least $1-\delta/(2L)$,

$$\max_{i\in[d]} |\boldsymbol{q}_i^T\boldsymbol{S}^T\boldsymbol{S}\boldsymbol{h}| \le \sqrt{\frac{\varepsilon}{2d}}. \tag{A.38}$$

Note that (A.38) implies $\|\boldsymbol{Q}^T\boldsymbol{S}^T\boldsymbol{S}\boldsymbol{h}\|_2^2 \le \frac{\varepsilon}{2}$. Consequently, combining the results we have that there exists an absolute constant C, such that if $m \ge CK^4 d\log(4dL/\delta)/\varepsilon$, then with probability at least $1-\delta/L$,

$$\sigma_{\min}^2(\boldsymbol{S}\boldsymbol{Q}) \ge \frac{\sqrt{2}}{2} \quad \text{and} \quad \|\boldsymbol{Q}^T\boldsymbol{S}^T\boldsymbol{S}\boldsymbol{h}\|_2^2 \le \frac{\varepsilon}{2}, \tag{A.39}$$

which are the conditions in (1.52). This completes the proof for the sub-Gaussian sketch.

We next prove the case for the leverage score sampling matrices, and the proof is again based on Lemma 1.4.10. Note that leverage score sampling can be viewed as a special case of induced measure sampling. The first condition in (1.52) is implied by $\|\boldsymbol{Q}^T\boldsymbol{S}^T\boldsymbol{S}\boldsymbol{Q}-\boldsymbol{I}\|_2 \le 1-\frac{\sqrt{2}}{2}$, which, according to [349, Lemma A.1], is satisfied with probability at least $1-\delta/2L$ if $m \ge 35d\log(4dL/\delta)$. For the second condition in (1.52), the only difference is that one uses Markov's inequality in place of Bernstein's inequality due to the lack of information on the tail of $\boldsymbol{q}_i^T\boldsymbol{S}^T\boldsymbol{S}\boldsymbol{h}$, and the details are omitted. Under additional assumptions in (1.56), Markov's inequality can be replaced by Hoeffding's inequality to yield an improved bound (1.57):

$$\boldsymbol{q}_i^T\boldsymbol{S}^T\boldsymbol{S}\boldsymbol{h} = \frac{1}{m}\sum_{j\in[m]} \langle\sqrt{m}\boldsymbol{s}_j, \boldsymbol{q}_i\rangle\langle\sqrt{m}\boldsymbol{s}_j, \boldsymbol{h}\rangle, \quad i \in [d], \tag{A.40}$$

with each summand $\langle\sqrt{m}\boldsymbol{s}_j, \boldsymbol{q}_i\rangle\langle\sqrt{m}\boldsymbol{s}_j, \boldsymbol{h}\rangle$ centered and bounded as

$$|\langle\sqrt{m}\boldsymbol{s}_j, \boldsymbol{q}_i\rangle\langle\sqrt{m}\boldsymbol{s}_j, \boldsymbol{h}\rangle| \le \max_{i\in[d]}\max_{j\in[N]:\ell_j>0} \frac{r|q_{ij}h_j|}{\ell_j} \le \max_{i\in[d]}\max_{j\in[N]:\ell_j>0} \frac{d|q_{ij}h_j|}{\ell_j} \le C, \tag{A.41}$$

where r is the rank of $\boldsymbol{A}$. By Hoeffding's inequality, for $t > 0$,

$$\mathcal{P}\left(|\boldsymbol{q}_i^T\boldsymbol{S}^T\boldsymbol{S}\boldsymbol{h}| \le t\right) \ge 1 - 2\exp\left(-\frac{mt^2}{2C^2}\right). \tag{A.42}$$

Setting $t = \sqrt{\varepsilon/2d}$ and taking a union bound over i yields that, for $m \ge 4C^2 d\log(4dL/\delta)/\varepsilon$, with probability at least $1-\delta/2L$, $\|\boldsymbol{Q}^T\boldsymbol{S}^T\boldsymbol{S}\boldsymbol{h}\|_2^2 \le \varepsilon/2$.

Appendix B

Bi-fidelity VAE

B.1 Proof of Bi-fidelity ELBO

In this section, we present the detailed proof of Equation (2.15). We assume the conditions $p(\boldsymbol{x}^H|\boldsymbol{z}^L, \boldsymbol{z}^H) = p(\boldsymbol{x}^H|\boldsymbol{z}^H)$ and $p(\boldsymbol{z}^H|\boldsymbol{z}^L, \boldsymbol{x}^L) = p(\boldsymbol{z}^H|\boldsymbol{z}^L)$ hold.

Proof. HF log-likelihood $\log p_{\boldsymbol{\theta},\boldsymbol{\psi}}(\boldsymbol{x}^H)$ can be decomposed and lower bounded as follows

$$\log p_{\boldsymbol{\theta},\boldsymbol{\psi}}(\boldsymbol{x}^H) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}[\log p_{\boldsymbol{\theta},\boldsymbol{\psi}}(\boldsymbol{x}^H)] \tag{B.1}$$

$$= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\left[\log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}^H, \boldsymbol{z}_{\boldsymbol{\psi}})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^H)}\right)\right] \tag{B.2}$$

$$= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\left[\log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}^H, \boldsymbol{z}_{\boldsymbol{\psi}})q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}{p_{\boldsymbol{\theta}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^H)q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\right)\right] \tag{B.3}$$

$$= \mathrm{KL}\big(q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)\|p_{\boldsymbol{\theta}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^H)\big) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\left[\log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}^H, \boldsymbol{z}_{\boldsymbol{\psi}})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\right)\right] \tag{B.4}$$

$$\geq \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\left[\log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}^H, \boldsymbol{z}_{\boldsymbol{\psi}})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\right)\right] \tag{B.5}$$

$$= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\left[\log\left(\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}_{\boldsymbol{\psi}})p_{\boldsymbol{\psi}}(\boldsymbol{z}^H|\boldsymbol{z}^L)p(\boldsymbol{z}^L)}{p_{\boldsymbol{\psi}}(\boldsymbol{z}^H|\boldsymbol{z}^L)q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)}\right)\right] \tag{B.6}$$

$$= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\left[\log\left(\frac{p(\boldsymbol{z}^L)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)}\right) + \log\big(p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}_{\boldsymbol{\psi}})\big)\right] \tag{B.7}$$

$$= -\mathrm{KL}\big(q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)\|p(\boldsymbol{z}^L)\big) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}\big[\log\big(p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}_{\boldsymbol{\psi}})\big)\big] \tag{B.8}$$

$$= \mathrm{ELBO}^{\mathrm{BF}}(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\theta}). \tag{B.9}$$

The only inequality above follows from the non-negativity of KL divergence. The above derivation shows that the HF log-likelihood can be lower bounded by the proposed BF-ELBO in Equa-

tion (2.16), where the tightness of the bound is controlled by the approximation error mentioned in Section 2.4.3 as $\mathrm{KL}\big(q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)\|p_{\boldsymbol{\theta}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^H)\big)$. □

B.2 Proof of Bi-fidelity Information Bottleneck

In this section, we prove that optimizing $\mathrm{ELBO}^{BF}(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\theta})$ in Equation (2.16) is equivalent with optimizing BF-IB objective function $\mathrm{IB}_\beta^{\mathrm{BF}}$ in Equation (2.22) with $\beta = 1$. With the BF-IB graphical model $\boldsymbol{z}_{\boldsymbol{\psi}} \leftarrow \boldsymbol{x}^L \leftrightarrow \boldsymbol{x}^H$ assumed (similar with IB in [378]), we have $q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L) = q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L, \boldsymbol{x}^H)$.

Proof.

$$\mathrm{IB}^{\mathrm{BF}}(\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\theta}) \tag{B.10}$$

$$= -\mathbb{I}(\boldsymbol{x}^L,\boldsymbol{z}_{\boldsymbol{\psi}}) + \mathbb{I}(\boldsymbol{z}_{\boldsymbol{\psi}},\boldsymbol{x}^H) \tag{B.11}$$

$$= -\mathbb{E}_{p(\boldsymbol{x}^L,\boldsymbol{z}_{\boldsymbol{\psi}})}\left[\log\frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}},\boldsymbol{x}^L)}{p(\boldsymbol{z}_{\boldsymbol{\psi}})p(\boldsymbol{x}^L)}\right] + \mathbb{E}_{p(\boldsymbol{x}^H,\boldsymbol{z}_{\boldsymbol{\psi}})}\left[\log\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}^H,\boldsymbol{z}_{\boldsymbol{\psi}})}{p(\boldsymbol{x}^H)p(\boldsymbol{z}_{\boldsymbol{\psi}})}\right] \tag{B.12}$$

$$= -\mathbb{E}_{p_{\boldsymbol{\phi}}(\boldsymbol{x}^L,\boldsymbol{z}_{\boldsymbol{\psi}})}\left[\log\frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\boldsymbol{\psi}}|\boldsymbol{x}^L)}{p_{\boldsymbol{\psi}}(\boldsymbol{z}_{\boldsymbol{\psi}})}\right] + \mathbb{E}_{p(\boldsymbol{x}^H,\boldsymbol{z}_{\boldsymbol{\psi}})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}_{\boldsymbol{\psi}})] + \mathbb{H}[\boldsymbol{x}^H] \tag{B.13}$$

$$= -\mathbb{E}_{p(\boldsymbol{x}^L,\boldsymbol{z}^H,\boldsymbol{z}^L)}\left[\log\frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)p_{\boldsymbol{\psi}}(\boldsymbol{z}^H|\boldsymbol{z}^L)}{p(\boldsymbol{z}^L)p_{\boldsymbol{\psi}}(\boldsymbol{z}^H|\boldsymbol{z}^L)}\right] + \mathbb{E}_{p(\boldsymbol{x}^H,\boldsymbol{z}^H,\boldsymbol{z}^L)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}^H)] + \mathbb{H}[\boldsymbol{x}^H] \tag{B.14}$$

$$= -\mathbb{E}_{p(\boldsymbol{x}^L)}\big[\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)||p(\boldsymbol{z}^L)\big] + \mathbb{E}_{p(\boldsymbol{x}^H,\boldsymbol{z}^H,\boldsymbol{z}^L)}\big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}^H)\big] + \mathbb{H}[\boldsymbol{x}^H] \tag{B.15}$$

$$= -\mathbb{E}_{p(\boldsymbol{x}^L)}\big[\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)||p(\boldsymbol{z}^L)\big] + \int p(\boldsymbol{x}^H,\boldsymbol{z}^H,\boldsymbol{z}^L)\big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}^H)\big]d\boldsymbol{z}^L d\boldsymbol{z}^H d\boldsymbol{x}^H + \mathbb{H}[\boldsymbol{x}^H] \tag{B.16}$$

$$= -\mathbb{E}_{p(\boldsymbol{x}^L)}\big[\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)||p(\boldsymbol{z}^L)\big] \tag{B.17}$$

$$+ \int q_{\boldsymbol{\phi}}(\boldsymbol{z}^L,\boldsymbol{z}^H|\boldsymbol{x}^L,\boldsymbol{x}^H)p(\boldsymbol{x}^H,\boldsymbol{x}^L)\big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}^H)\big]d\boldsymbol{z}^H d\boldsymbol{x}^H d\boldsymbol{x}^L d\boldsymbol{z}^L + \mathbb{H}[\boldsymbol{x}^H] \tag{B.18}$$

$$= -\mathbb{E}_{p(\boldsymbol{x}^L)}\big[\mathrm{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)||p(\boldsymbol{z}^L)\big] \tag{B.19}$$

$$+ \int q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)p(\boldsymbol{x}^H,\boldsymbol{x}^L)p_{\boldsymbol{\psi}}(\boldsymbol{z}^H|\boldsymbol{z}^L)\big[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}^H)\big]d\boldsymbol{z}^H d\boldsymbol{x}^H d\boldsymbol{x}^L d\boldsymbol{z}^L + \mathbb{H}[\boldsymbol{x}^H] \tag{B.20}$$

$$= -\mathbb{E}_{p(\boldsymbol{x}^L)}\big[\mathrm{KL}(q_{\phi}(\boldsymbol{z}^L|\boldsymbol{x}^L)||p(\boldsymbol{z}^L)\big] + \mathbb{E}_{p(\boldsymbol{x}^L,\boldsymbol{x}^H)}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}^L|\boldsymbol{x}^L)}\big[\mathbb{E}_{p_{\boldsymbol{\psi}}(\boldsymbol{z}^H|\boldsymbol{z}^L)}[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}^H|\boldsymbol{z}^H)]\big] + \mathbb{H}[\boldsymbol{x}^H] \tag{B.21}$$

$$= \mathbb{E}_{p(\boldsymbol{x}^L,\boldsymbol{x}^H)}[\mathrm{ELBO}^{BF}(\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\theta})] + \text{constant}, \tag{B.22}$$

where $\mathbb{H}[\cdot]$ is the differential entropy of the input random vector. Since $p(\boldsymbol{x}^H)$ is fixed, its entropy is a constant. □

B.3 A Brief Introduction to KID

In this section, we briefly introduce the kernel inception distance (KID). KID is a commonly-used metric for evaluating the performance of generative models [54], which stems from maximum mean discrepancy (MMD). MMD is a type of statistical distance that falls under the umbrella of

the integral probability metric (IPM). Given two probability distributions $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$, the IPM is defined as

$$\mathrm{IPM}_{\mathcal{F}}(p, q) \coloneqq \sup_{f \in \mathcal{F}} \mathbb{E}_p[f(\boldsymbol{x})] - \mathbb{E}_q[f(\boldsymbol{x})], \tag{B.23}$$

where the function class $\mathcal{F}$ controls the value range of the IPM and $\mathbb{E}_p$ is the expectation with respect to $p(\boldsymbol{x})$. Larger $\mathcal{F}$ brings higher accuracy to the IPM value but also increases the computing complexity. By the Kantorovich-Rubinstein duality theorem [530], the Wasserstein-1 distance is a type of IPM with $\mathcal{F}$ being all Lipschitz continuous functions having Lipschitz constant bounded by 1. However, estimating the Wasserstein distance accurately in high dimensions is difficult. MMD assigns $\mathcal{F}$ to be all functions in a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ with norm bounded by 1, where $\mathcal{H}$ is generated from a given kernel function $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$. The motivation for using RKHS is for its computational convenience, as the following propositions show.

Proposition B.3.1. *MMD can be expressed in the following alternative form.*

$$\mathrm{MMD}(p, q) \coloneqq \sup_{\|f\|_{\mathcal{H}}=1} \mathbb{E}_p[f(\boldsymbol{x})] - \mathbb{E}_q[f(\boldsymbol{x})] \tag{B.24}$$

$$= \|\mathcal{G}_{\mathcal{H}}(p) - \mathcal{G}_{\mathcal{H}}(q)\|_{\mathcal{H}}, \tag{B.25}$$

where $\mathcal{G}_{\mathcal{H}}$ is a Bochner integral defined as $\mathcal{G}_{\mathcal{H}}(p) \coloneqq \int_{\mathbb{R}^D} k(\boldsymbol{x}, \cdot) p(\boldsymbol{x}) d\boldsymbol{x}$.

The proof of Proposition B.3.1 is available from the Lemma 4 in [196]. Suppose we have samples $\{\boldsymbol{x}_i\}_{i=1}^m \sim p(\boldsymbol{x})$ and $\{\tilde{\boldsymbol{x}}_i\}_{i=1}^n \sim q(\boldsymbol{x})$, then it is straightforward to see that the following KID statistic is an unbiased estimate of the MMD:

$$\mathrm{KID}\left(\{\boldsymbol{x}_i\}_{i=1}^m, \{\tilde{\boldsymbol{x}}_i\}_{i=1}^n\right) \tag{B.26}$$

$$= \frac{1}{m(m-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{m} k(\boldsymbol{x}_i, \boldsymbol{x}_j) - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_j) + \frac{1}{n(n-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{n} k(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j). \tag{B.27}$$

Equation (B.26) discloses the connection between KID and MMD. In Section 3.4, we use KID to present the distributional similarity between two given samples.

The authors of [54] have shown that a rational quadratic kernel with a mixture of length scales is a good choice of the kernel function to be used with MMD due to its low rate of tail decay. The rational quadratic kernel has the form

$$k_{\mathrm{rq}}(\boldsymbol{x}_i, \boldsymbol{y}_j) \coloneqq \sum_{\ell \in \mathcal{I}} \left(1 + \frac{\|\boldsymbol{x}_i - \boldsymbol{y}_j\|^2}{2\ell}\right)^{-\ell}, \tag{B.28}$$

where $\mathcal{I} = [0.2, 0.5, 1.0, 2.0, 5.0]$ is a mixture of different length scales.

The following results, which appear as Theorem 10 and Corollary 16 in [196], show the consistency of KID.

Proposition B.3.2. *Assuming both input sample sets have the same size m, and that the kernel function satisfies $0 \leq k(\boldsymbol{x}, \boldsymbol{y}) \leq K$, KID in Equation (B.26) satisfies*

$$\mathcal{P}[|\mathrm{KID} - \mathrm{MMD}^2| > \epsilon] \leq 2 \exp\left(-\frac{\epsilon^2 m}{16K^2}\right), \tag{B.29}$$

$$m^{1/2}(\mathrm{KID} - \mathrm{MMD}^2) \xrightarrow{d} \mathcal{N}(0, \sigma_u^2), \tag{B.30}$$

where σ_u^2 is a value independent of m and D.

It should be noted that the asymptotic mean square error of KID is $m^{-1}\sigma_u^2$, which is independent of the dimension D. It is the key reason why we choose KID over other statistical distances to test our models, considering the problems in this work have large D ($D \geq 100$).

Appendix C

Langevin Bi-fidelity Importance Sampling

C.1 Variance Deviation

The simplification of the $\widehat{P}_N^{\mathrm{BF}}$ variance is as follow:

$$\begin{aligned}
\mathbb{V}_{p\otimes q}\left[\widehat{P}_{M,N}^{\mathrm{BF}}\right] &\approx \frac{\mathcal{Z}^2(\ell)}{N}\mathbb{V}_q\left[\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}\exp\left(\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)\right]\\
&= \frac{\mathcal{Z}^2(\ell)}{N}\left(\mathbb{E}_q\left[\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}\exp\left(2\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)\right]-\left(\mathbb{E}_q\left[\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}\exp\left(\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)\right]\right)^2\right)\\
&= = \frac{\mathcal{Z}^2(\ell)}{N}\left(\mathbb{E}_q\left[\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}\exp\left(2\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)\right]-\left(\frac{1}{\mathcal{Z}(\ell)}\mathbb{E}_p\left[\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}\right]\right)^2\right)\\
&= \frac{\mathcal{Z}(\ell)}{N}\mathbb{E}_p\left[\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}\exp\left(\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)\right]-\frac{1}{N}\left(\mathbb{E}_p\left[\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}\right]\right)^2\\
&= \frac{\mathcal{Z}(\ell)}{N}\mathbb{E}_p\left[\mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}\exp\left(\ell\tanh\circ h^{\mathrm{LF}}(\boldsymbol{z})\right)\right]-\frac{(P_f)^2}{N}.
\end{aligned} \tag{C.1}$$

C.2 An Upper Bound for the Normalization Constant

Lemma C.2.1. *An upper bound for $\mathcal{Z}(\ell)$ is as*

$$\mathcal{Z}(\ell) < (e^{\ell}-1)\mathcal{P}_p[\mathcal{A}_L]+1. \tag{C.2}$$

Proof.

$$
\begin{aligned}
\mathcal{Z}(\ell) &= \mathbb{E}_p[\exp\left(-\ell \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z})\right)] \\
&= \int_{\Omega} \exp\left(-\ell \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z})\right) p(\boldsymbol{z}) d\boldsymbol{z} \\
&= \int_{\mathcal{A}_L} \exp\left(-\ell \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z})\right) p(\boldsymbol{z}) d\boldsymbol{z} + \int_{\mathcal{A}_L^C} \exp\left(-\ell \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z})\right) p(\boldsymbol{z}) d\boldsymbol{z} \\
&< e^{\ell} \mathcal{P}_p[\mathcal{A}_L] + \mathcal{P}_p[\mathcal{A}_L^C] \\
&= (e^{\ell} - 1)\mathcal{P}_p[\mathcal{A}_L] + 1.
\end{aligned} \tag{C.3}
$$

□

C.3 Simplification for KL Divergence

The detailed process to simplify the KL divergence is,

$$
\begin{aligned}
\mathrm{KL}(q^* \| q) &= \mathbb{E}_{q^*}\left[\log \frac{\mathcal{Z}(\ell) \mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0}}{P_f \exp\left(-\ell \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z})\right)}\right] \\
&= \log \frac{\mathcal{Z}(\ell)}{P_f} + \mathbb{E}_{q^*}[\log \mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0} + \ell \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z})] \\
&= \log \frac{\mathcal{Z}(\ell)}{P_f} + \int_{\mathcal{A}_H} \log \mathbb{1}_{h^{\mathrm{HF}}(\boldsymbol{z})<0} + \ell \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z}) d\boldsymbol{z} \\
&= \log \frac{\mathcal{Z}(\ell)}{P_f} + \ell \int_{\mathcal{A}_H \cap \mathcal{A}_L} \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z}) d\boldsymbol{z} + \ell \int_{\mathcal{A}_H \cap \mathcal{A}_L^C} \tanh \circ h^{\mathrm{LF}}(\boldsymbol{z}) d\boldsymbol{z} \\
&< \log \frac{(e^{\ell} - 1)\mathcal{P}_p[\mathcal{A}_L] + 1}{P_f} + \ell \mathcal{P}_p[\mathcal{A}_H \cap \mathcal{A}_L^C].
\end{aligned} \tag{C.4}
$$

Appendix D

Bi-fidelity Stochastic Subspace Descent

D.1 Proof of Lemma 4.3.7

Proof. We let n_k evaluations positioned at equispaced points between $\boldsymbol{x}_k$ and $\boldsymbol{x}_k+\alpha_{\max}\boldsymbol{v}_k$, each sub-interval has length $\alpha_{\max}/n_k$. We also define $\psi_k(\alpha) \coloneqq \varphi(\alpha) - f^{\mathrm{LF}}(\boldsymbol{x}+\alpha\boldsymbol{v}_k)$. For each sub-interval, we define the surrogate $\tilde{\psi}_k(\alpha; n_k)$ as a linear function connecting these values.

WLOG, we prove the bound in Equation (4.13) holds in the interval $\alpha \in [0, h]$ with $h = \alpha_{\max}/n_k$ and this result can be extended to other sub-intervals. The surrogate $\psi(\alpha)$ is defined as

$$\tilde{\psi}_k(\alpha; n_k) \coloneqq \frac{h-\alpha}{h}\psi(0) + \frac{\alpha}{h}\psi(h), \quad \alpha \in [0, h], \tag{D.1}$$

and similar definitions of $\tilde{\psi}_k(\alpha; n_k)$ hold when α in other sub-intervals. Such linear approximation $\tilde{\psi}_k(\alpha)$ satisfies

$$|\varphi(\alpha) - \tilde{\varphi}_k(\alpha; n_k)| = |\psi(\alpha) - \tilde{\psi}_k(\alpha; n_k)| = \left| \frac{h-\alpha}{h}\left(\psi(0) - \psi(\alpha)\right) + \frac{\alpha}{h}\left(\psi(h) - \psi(\alpha)\right) \right|, \tag{D.2}$$

for any $\alpha \in [0, h]$. Since the Lipschitz constant of $\psi(\alpha)$ is strictly controlled by W and the fact that $\boldsymbol{v}_k$ is a unit vector, Equation (D.2) satisfies

$$|\varphi(\alpha) - \tilde{\varphi}_k(\alpha; n_k)| \leq W\frac{\alpha_{\max}/n_k - \alpha}{\alpha_{\max}/n_k}(\alpha - 0) + W\frac{\alpha}{\alpha_{\max}/n_k}(\alpha_{\max}/n_k - \alpha) \leq \frac{W\alpha_{\max}}{2n_k}, \quad \forall \alpha \in [0, h]. \tag{D.3}$$

Since we have

$$n_k \geq \frac{WL(1+c)\alpha_{\max}}{c\beta\|\boldsymbol{v}_k\|^2}, \tag{D.4}$$

the sup-norm is bounded as

$$|\varphi(\alpha) - \tilde{\varphi}_k(\alpha; n_k)| \leq \frac{c\beta\|\boldsymbol{v}_k\|^2}{2(1+c)L} = \frac{\|\boldsymbol{v}_k\|^2}{2} \min\left\{\frac{c}{(1+c)^2 L}, \frac{c\beta}{(1+c)L}, \beta\alpha_{\max}\right\} = \frac{\beta c\|\boldsymbol{v}_k\|^2}{2(1+c)L}, \quad \text{(D.5)}$$

where the last equality stems from the fact that $\beta \leq 1/2$ and $\alpha_{\max} \geq c/(cL+L)$. □

D.2 Single-fidelity SSD with Line Search

D.2.1 Assuming Strong-convexity

Assumption D.2.1. Assume the objective function f^{HF} and algorithm satisfies the following conditions

(1) $\boldsymbol{P}_k \in \mathbb{R}^{d\times\ell}$ are independent random matrices such that $\mathbb{E}[\boldsymbol{P}_k\boldsymbol{P}_k^T] = \boldsymbol{I}_d$ and $\boldsymbol{P}_k^T\boldsymbol{P}_k = (d/\ell)\boldsymbol{I}_\ell$ with $d > \ell$;

(2) Objective function $f^{\text{HF}} : \mathbb{R}^d \to \mathbb{R}$ attains its minimum f^* and ∇f^{HF} is L-Lipschitz continuous;

(3) Objective function $f^{\text{HF}} : \mathbb{R}^d \to \mathbb{R}$ is γ-strongly convex; note $\gamma \leq L$.

Theorem D.2.2. *(Single fidelity) With the assumptions of D.2.1, SSD with line search (either exact line search or backtracking) converges in the sense that* $f(\boldsymbol{x}_k) \overset{a.s.}{\to} f^*$ *and* $f(\boldsymbol{x}_k) \overset{L^1}{\to} f^*$.

Proof. Define the filtration $\mathcal{F}_k \coloneqq \sigma(\boldsymbol{P}_1, \ldots, \boldsymbol{P}_{k-1})$ and $\mathcal{F}_1 = \{\emptyset, \Omega\}$. By Lipschitz continuity, we have

$$f(\boldsymbol{x}_{k+1}) \leq f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^T(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) + \frac{L}{2}\|\boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|^2. \quad \text{(D.6)}$$

By defining $f_e(\boldsymbol{x}) \coloneqq f(\boldsymbol{x}) - f^*$ and plugging $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k\boldsymbol{P}_k\boldsymbol{P}_k^T\nabla f(\boldsymbol{x}_k)$, Equation (D.6) yields

$$\begin{aligned} f_e(\boldsymbol{x}_{k+1}) - f_e(\boldsymbol{x}_k) &\leq -\alpha_k\langle\nabla f(\boldsymbol{x}_k), \boldsymbol{P}_k\boldsymbol{P}_k^T\nabla f(\boldsymbol{x}_k)\rangle + \frac{\alpha_k^2 L}{2}\langle\boldsymbol{P}_k\boldsymbol{P}_k^T\nabla f(\boldsymbol{x}_k), \boldsymbol{P}_k\boldsymbol{P}_k^T\nabla f(\boldsymbol{x}_k)\rangle \\ &= -\alpha_k\langle\nabla f(\boldsymbol{x}_k), \boldsymbol{P}_k\boldsymbol{P}_k^T\nabla f(\boldsymbol{x}_k)\rangle + \frac{d\alpha_k^2 L}{2\ell}\langle\nabla f(\boldsymbol{x}_k), \boldsymbol{P}_k\boldsymbol{P}_k^T\nabla f(\boldsymbol{x}_k)\rangle \quad \text{(D.7)} \\ &= \left(-\alpha_k + \frac{d\alpha_k^2 L}{2\ell}\right)\langle\nabla f(\boldsymbol{x}_k), \boldsymbol{P}_k\boldsymbol{P}_k^T\nabla f(\boldsymbol{x}_k)\rangle, \end{aligned}$$

where the fact $\boldsymbol{P}_k\boldsymbol{P}_k^T\boldsymbol{P}_k\boldsymbol{P}_k^T = (d/\ell)\boldsymbol{P}_k\boldsymbol{P}_k^T$ is applied. We have two line search approaches to determine the step size α_k,

(1) Exact line search:

$$\alpha_k = \arg\min_{\alpha} f(\boldsymbol{x}_k - \alpha \boldsymbol{P}_k \boldsymbol{P}_k^T \nabla f(\boldsymbol{x}_k)); \tag{D.8}$$

(2) Backtracking: for some fixed $\alpha_{\max} > 0$, $\beta \in (0, \ell/2d)$, and $c \in (0,1)$,

$$\begin{aligned} \alpha_k &= \max_{m \in \mathbb{N}} c^m \alpha_{\max} \\ \text{s.t.} \quad f(\boldsymbol{x}_k - c^m \alpha_{\max} \boldsymbol{P}_k \boldsymbol{P}_k^T \nabla f(\boldsymbol{x}_k)) &\leq f(\boldsymbol{x}^k) - \beta c^m \alpha_{\max} \|\boldsymbol{P}_k \boldsymbol{P}_k^T \nabla f(\boldsymbol{x}^k)\|^2. \end{aligned} \tag{D.9}$$

Note that α_k is a scalar random variable with randomness from the Haar measure $\boldsymbol{P}_k$. We will prove the convergence for two line search methods separately. All the following analyses hold for any $\boldsymbol{P}_k$.

Exact line search According to Equation (D.8), the exact line search method can find the optimal α_k such that the quadratic term in Equation (D.7) yields $-\alpha_k + d\alpha_k^2 L/2\ell \leq -\ell/2dL$ for any $\boldsymbol{P}_k$, thereby

$$f_e(\boldsymbol{x}_{k+1}) - f_e(\boldsymbol{x}_k) \leq -\frac{\ell}{2dL} \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{P}_k \boldsymbol{P}_k^T \nabla f(\boldsymbol{x}_k) \rangle \quad \forall \boldsymbol{P}_k. \tag{D.10}$$

With condition on the current filtration $\mathcal{F}_k$, the conditional expectation on both sides turn to

$$\begin{aligned} \mathbb{E}[f_e(\boldsymbol{x}_{k+1})|\mathcal{F}_k] &\leq -\frac{\ell}{2dL} \mathbb{E}\left[\langle \nabla f(\boldsymbol{x}_k), \boldsymbol{P}_k \boldsymbol{P}_k^T \nabla f(\boldsymbol{x}_k) \rangle | \mathcal{F}_k\right] + f_e(\boldsymbol{x}_k) \\ &= -\frac{\ell}{2dL} \|\nabla f(\boldsymbol{x}_k)\|^2 + f_e(\boldsymbol{x}_k), \end{aligned} \tag{D.11}$$

where the equality is from the fact $\mathbb{E}[\boldsymbol{P}_k \boldsymbol{P}_k^T | \mathcal{F}_k] = \boldsymbol{I}_d$. By invoking the Polyak-Lojasiewicz inequality,

$$\mathbb{E}[f_e(\boldsymbol{x}_{k+1})|\mathcal{F}_k] \leq -\frac{\gamma\ell}{dL} f_e(\boldsymbol{x}_k) + f_e(\boldsymbol{x}_k) = \left(1 - \frac{\gamma\ell}{dL}\right) f_e(\boldsymbol{x}_k) \tag{D.12}$$

Recursive application yields

$$\mathbb{E}[f_e(\boldsymbol{x}_{k+1})|\mathcal{F}_k] \leq \left(1 - \frac{\gamma\ell}{dL}\right) f_e(\boldsymbol{x}_k) = \left(1 - \frac{\gamma\ell}{dL}\right) \mathbb{E}[f_e(\boldsymbol{x}_k)|\mathcal{F}_{k-1}] \leq \left(1 - \frac{\gamma\ell}{dL}\right)^{k+1} \mathbb{E}[f_e(\boldsymbol{x}_0)] \tag{D.13}$$

Since $\ell \leq d$ and $\gamma \leq L$, the term $1 - \gamma\ell/dL$ is less than 1. Equation (D.13) implies

$$\mathbb{E}\left[f(\boldsymbol{x}_{k+1})\right] - f^* \leq \left(1 - \frac{\gamma\ell}{dL}\right)^{k+1} \mathbb{E}[(f(\boldsymbol{x}_0) - f^*)] = \left(1 - \frac{\gamma\ell}{dL}\right)^{k+1} (f(\boldsymbol{x}_0) - f^*), \tag{D.14}$$

which proves $f(\boldsymbol{x}_k) \xrightarrow{a.s.} f^*$ and $f(\boldsymbol{x}_k) \xrightarrow{L^1} f^*$.

Backtracking (Show the existence of a feasible set such that Armijo condition satisfies.) Following Equation (D.9), the backtracking method selects the maximal possible step size value that satisfies the Armijo condition with specified parameter $\beta \leq \ell/2d$ and shrinking parameter $c < 1$. When $0 \leq \alpha_k \leq \ell/dL$, $-\alpha_k + d\alpha_k^2 L/2\ell \leq -\alpha_k/2$ holds with Haar measure probability one, which implies the following Armijo stopping condition

$$\begin{aligned} f(\boldsymbol{x}_{k+1}) &\leq f(\boldsymbol{x}_k) - \alpha_k \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{P}_k \boldsymbol{P}_k^T \nabla f(\boldsymbol{x}_k) \rangle + \frac{d\alpha_k^2 L}{2\ell} \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{P}_k \boldsymbol{P}_k^T \nabla f(\boldsymbol{x}_k) \rangle \\ &\leq f(\boldsymbol{x}_k) - \frac{\alpha_k}{2} \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{P}_k \boldsymbol{P}_k^T \nabla f(\boldsymbol{x}_k) \rangle \\ &= f(\boldsymbol{x}_k) - \frac{\alpha_k \ell}{2d} \| \boldsymbol{P}_k \boldsymbol{P}_k^T \nabla f(\boldsymbol{x}_k) \|^2 \\ &\leq f(\boldsymbol{x}_k) - \beta \alpha_k \| \boldsymbol{P}_k \boldsymbol{P}_k^T \nabla f(\boldsymbol{x}_k) \|^2. \end{aligned} \tag{D.15}$$

Therefore, the backtracking terminates when $\alpha_k = \alpha_{\max}$ or $\alpha_k \geq \ell c/dL$, which implies

$$f_e(\boldsymbol{x}_{k+1}) \leq f_e(\boldsymbol{x}_k) - \beta \alpha_{\max} \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{P}_k \boldsymbol{P}_k^T \nabla f(\boldsymbol{x}_k) \rangle, \tag{D.16}$$

or

$$f_e(\boldsymbol{x}_{k+1}) \leq f_e(\boldsymbol{x}_k) - \frac{\ell c \beta}{dL} \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{P}_k \boldsymbol{P}_k^T \nabla f(\boldsymbol{x}_k) \rangle. \tag{D.17}$$

Similar with Equation (D.11), by combining Equation (D.16) and Equation (D.17) and taking expectations conditioned on the filtration $\mathcal{F}_k$, we have

$$\begin{aligned} \mathbb{E}[f_e(\boldsymbol{x}_{k+1}) | \mathcal{F}_k] &\leq -\min \left\{ \beta \alpha_{\max}, \frac{\ell c \beta}{dL} \right\} \mathbb{E} \left[\langle \nabla f(\boldsymbol{x}_k), \boldsymbol{P}_k \boldsymbol{P}_k^T \nabla f(\boldsymbol{x}_k) \rangle | \mathcal{F}_k \right] + f_e(\boldsymbol{x}_k) \\ &= -\min \left\{ \beta \alpha_{\max}, \frac{\ell c \beta}{dL} \right\} \| \nabla f(\boldsymbol{x}_k) \|^2 + f_e(\boldsymbol{x}_k). \end{aligned} \tag{D.18}$$

Similar with Equation (D.12), by invoking the Polyak-Lojasiewicz inequality,

$$\mathbb{E}[f_e(\boldsymbol{x}_{k+1}) | \mathcal{F}_k] \leq -\min \left\{ 2\gamma\beta\alpha_{\max}, \frac{2\ell c \gamma \beta}{dL} \right\} f_e(\boldsymbol{x}_k) + f_e(\boldsymbol{x}) = \left(1 - \min \left\{ 2\gamma\beta\alpha_{\max}, \frac{2\ell c \gamma \beta}{dL} \right\} \right) f_e(\boldsymbol{x}_k). \tag{D.19}$$

By recursively implementing Equation (D.19), we have

$$\mathbb{E}[f_e(\boldsymbol{x}_{k+1})] \leq \left(1 - \min \left\{ 2\gamma\beta\alpha_{\max}, \frac{2\ell c \gamma \beta}{dL} \right\} \right)^{k+1} f_e(\boldsymbol{x}_0). \tag{D.20}$$

Since $0 < c < 0$, $0 < \gamma \leq L$, $0 < \ell \leq d$, and $0 < \beta < 0.5$, the term $0 < (1 - \min\{2\gamma\beta, 2\ell c \gamma \beta/dL\}) < 1$, the convergences $f(\boldsymbol{x}_k) \xrightarrow{a.s.} f^*$ and $f(\boldsymbol{x}_k) \xrightarrow{L^1} f^*$ are guaranteed. □

D.2.2 Assuming Convexity

Assumption D.2.3. For the non-strongly convex objective function f^{HF} and the algoritihm, we make the following assumptions

(1) $\boldsymbol{P}_k \in \mathbb{R}^{d\times\ell}$ are independent random matrices such that $\mathbb{E}[\boldsymbol{P}_k\boldsymbol{P}_k^T] = \boldsymbol{I}_d$ and $\boldsymbol{P}_k^T\boldsymbol{P}_k = (d/\ell)\boldsymbol{I}_\ell$ with $d > \ell$;

(2) Objective function $f^{\mathrm{HF}} : \mathbb{R}^d \to \mathbb{R}$ is convex and ∇f^{HF} is L-Lipschitz continuous;

(3) The function f^{HF} attains its minimum(s) f^* at x^* so that there exists a known R satisfying $\max_{\boldsymbol{x},\boldsymbol{x}^*}\{\|\boldsymbol{x} - \boldsymbol{x}^*\| : f^{\mathrm{HF}}(\boldsymbol{x}) \le f^{\mathrm{HF}}(\boldsymbol{x}_0)\} \le R$;

Given the above convex-but-not-strongly-convex assumption, we have the following L^1 convergence result:

Theorem D.2.4. *With backtracking implemented for line search, we have*

$$\mathbb{E}[f(\boldsymbol{x}_k)] - f^* \le \max\left\{\frac{2R^2}{k\beta\alpha_{\max}}, \frac{2dLR^2}{k\ell c\beta}\right\}. \tag{D.21}$$

Proof. We use the same notation as in the proof of Thm. D.2.2. Starting from Equation (D.18), we have

$$\mathbb{E}[f_e(\boldsymbol{x}_{k+1})|\mathcal{F}_k] = -\min\left\{\beta\alpha_{\max}, \frac{\ell c\beta}{dL}\right\}\|\nabla f(\boldsymbol{x}_k)\|^2 + f_e(\boldsymbol{x}_k). \tag{D.22}$$

By convexity and the Cauchy-Schwartz inequality, $\|\nabla f(\boldsymbol{x}_k)\| \ge f_e(\boldsymbol{x}_k)/R$. With the fact that $\mathbb{E}f_e(\boldsymbol{x}_{k+1}) \le \mathbb{E}f_e(\boldsymbol{x}_k)$, we have

$$\begin{aligned}
\mathbb{E}f_e(\boldsymbol{x}_{k+1}) - f_e(\boldsymbol{x}_k) &\le -\min\left\{\beta\alpha_{\max}, \frac{\ell c\beta}{dL}\right\}\frac{\mathbb{E}f_e^2(\boldsymbol{x}_k)}{2R^2} \\
&\le -\min\left\{\beta\alpha_{\max}, \frac{\ell c\beta}{dL}\right\}\frac{\mathbb{E}^2 f_e(\boldsymbol{x}_k)}{2R^2} \\
&\le -\min\left\{\beta\alpha_{\max}, \frac{\ell c\beta}{dL}\right\}\frac{\mathbb{E}f_e(\boldsymbol{x}_{k+1})\mathbb{E}f_e(\boldsymbol{x}_k)}{2R^2},
\end{aligned} \tag{D.23}$$

which further implies

$$\frac{1}{\mathbb{E}f_e(\boldsymbol{x}_{k+1})} \ge \frac{1}{\mathbb{E}f_e(\boldsymbol{x}_k)} + 2R^2\min\left\{\beta\alpha_{\max}, \frac{\ell c\beta}{dL}\right\}. \tag{D.24}$$

Applying (D.24) recursively, we obtain

$$\mathbb{E}f_e(\boldsymbol{x}_{k+1}) \leq \left(\min\left\{\beta\alpha_{\max}, \frac{\ell c\beta}{dL}\right\}\right)^{-1} \frac{2R^2}{k} = \max\left\{\frac{2R^2}{k\beta\alpha_{\max}}, \frac{2dLR^2}{k\ell c\beta}\right\}. \tag{D.25}$$

□

D.2.3 No convexity assumptions

Assumption D.2.5. We make the following assumptions:

(1) $\boldsymbol{P}_k \in \mathbb{R}^{d\times\ell}$ are independent random matrices such that $\mathbb{E}[\boldsymbol{P}_k\boldsymbol{P}_k^T] = \boldsymbol{I}_d$ and $\boldsymbol{P}_k^T\boldsymbol{P}_k = (d/\ell)\boldsymbol{I}_\ell$ with $d > \ell$;

(2) The objective function $f^{\text{HF}} : \mathbb{R}^d \to \mathbb{R}$ (or f^{HF}) attains its minimum f^* and ∇f^{HF} is L-Lipschitz continuous;

When Assumption D.2.5 holds, we have following L^2 convergence of the gradient norm result for SSD with line search:

Theorem D.2.6. *With Assumption D.2.5 holding and backtracking implemented for line search, we have*

$$\min_{k\in\{0,\ldots,K\}} \mathbb{E}[\|\nabla f(\boldsymbol{x}_k)\|^2] \leq \max\left\{\frac{(f(\boldsymbol{x}_0)-f^*)}{(K+1)\beta\alpha_{\max}}, \frac{dL(f(\boldsymbol{x}_0)-f^*)}{(K+1)\ell c\beta}\right\}. \tag{D.26}$$

That is, $k = \mathcal{O}(1/(\epsilon\beta\alpha_{\max}) + dL/(\epsilon\ell c\beta))$ iterations are required to achieve $\mathbb{E}\|\nabla f(\boldsymbol{x}_k)\|^2 \leq \epsilon$.

Proof. Following Equation (D.18)

$$\min\left\{\beta\alpha_{\max}, \frac{\ell c\beta}{dL}\right\} \|\nabla f(\boldsymbol{x}_k)\|^2 \leq f_e(\boldsymbol{x}_k) - \mathbb{E}[f_e(\boldsymbol{x}_{k+1})|\mathcal{F}_k], \tag{D.27}$$

which leads to the telescope series

$$\begin{aligned}\min\left\{\beta\alpha_{\max}, \frac{\ell c\beta}{dL}\right\} \sum_{k=0}^{K} \|\nabla f(\boldsymbol{x}_k)\|^2 &\leq \sum_{k=0}^{K} (f_e(\boldsymbol{x}_k) - \mathbb{E}[f_e(\boldsymbol{x}_{k+1})|\mathcal{F}_k]) \\ &= f(\boldsymbol{x}_0) - \mathbb{E}f(\boldsymbol{x}_{K+1}) \leq f(\boldsymbol{x}_0) - f^*.\end{aligned} \tag{D.28}$$

Therefore,

$$(K+1)\min_{k\in\{0,\ldots,K\}}\mathbb{E}\|\nabla f(\boldsymbol{x}_k)\|^2 \leq \left(\min\left\{\beta\alpha_{\max}, \frac{\ell c\beta}{dL}\right\}\right)^{-1}(f(\boldsymbol{x}_0)-f^*) \\ = \max\left\{\frac{(f(\boldsymbol{x}_0)-f^*)}{\beta\alpha_{\max}}, \frac{dL(f(\boldsymbol{x}_0)-f^*)}{\ell c\beta}\right\}. \tag{D.29}$$

A sufficient condition to let $\mathbb{E}\|\nabla f(\boldsymbol{x}_k)\|^2$ be ϵ-small is to let

$$k \geq \max\left\{\frac{(f(\boldsymbol{x}_0)-f^*)}{\epsilon\beta\alpha_{\max}}, \frac{dL(f(\boldsymbol{x}_0)-f^*)}{\epsilon\ell c\beta}\right\}. \tag{D.30}$$

□

D.3 Worst Function in the World: Additional Data

	$c=0.8$			$c=0.9$			$c=0.99$		
Method	$\ell=5$	$\ell=10$	$\ell=20$	$\ell=5$	$\ell=10$	$\ell=20$	$\ell=5$	$\ell=10$	$\ell=20$
GD	0.9026	0.9026	0.9026	0.9026	0.9026	0.9026	0.9026	0.9026	**0.9026**
CD	0.8984	0.8984	0.8984	0.8984	0.8984	0.8984	0.8984	0.8984	**0.8984**
FS-SSD	2.4495	2.4194	**2.3611**	2.4495	2.4196	2.3619	2.4497	2.4194	2.3622
SPSA	0.7713	0.7756	0.7502	0.6245	**0.4623**	0.5549	0.6046	0.6721	0.7006
GS	2.4598	2.4442	**2.4129**	2.4597	2.4447	2.4144	2.4596	2.4445	2.4150
HF-SSD	0.3620	0.2511	**0.2194**	0.8667	0.5109	0.3337	3.4582	1.7191	1.0331
BF-SSD	0.3177	0.2947	0.2932	0.2686	0.2526	0.2497	0.2316	0.2104	**0.1984**
VR-SSD	0.9885	0.9374	0.9178	0.9881	0.9464	0.9154	0.9925	0.9395	**0.9121**

Table D.1: Performance values for different optimization methods across various c and ℓ combinations at $N = 5{,}000$. The minimum value in each row is highlighted in bold.