Near-optimal hierarchical matrix approximation from matrix-vector products*

Tyler Chen† Feyza Duman Keles‡ Diana Halikias§
Cameron Musco¶ Christopher Musco‖ David Persson **

Abstract

We describe a randomized algorithm for producing a near-optimal hierarchical off-diagonal low-rank (HODLR) approximation to an $n \times n$ matrix $\mathbf{A}$, accessible only though matrix-vector products with $\mathbf{A}$ and $\mathbf{A}^\mathsf{T}$. We prove that, for the rank-k HODLR approximation problem, our method achieves a $(1+\beta)^{\log(n)}$-optimal approximation in expected Frobenius norm using $O(k\log(n)/\beta^3)$ matrix-vector products. In particular, the algorithm obtains a $(1+\varepsilon)$-optimal approximation with $O(k\log^4(n)/\varepsilon^3)$ matrix-vector products, and for any constant c, an n^c-optimal approximation with $O(k\log(n))$ matrix-vector products. Apart from matrix-vector products, the additional computational cost of our method is just $O(n\,\mathrm{poly}(\log(n), k, \beta))$. We complement the upper bound with a lower bound, which shows that any matrix-vector query algorithm requires at least $\Omega(k\log(n) + k/\varepsilon)$ queries to obtain a $(1+\varepsilon)$-optimal approximation.

Our algorithm can be viewed as a robust version of widely used "peeling" methods for recovering HODLR matrices and is, to the best of our knowledge, the first matrix-vector query algorithm to enjoy theoretical worst-case guarantees for approximation by any hierarchical matrix class. To control the propagation of error between levels of hierarchical approximation, we introduce a new perturbation bound for low-rank approximation, which shows that the widely used Generalized Nyström method enjoys inherent stability when implemented with noisy matrix-vector products. We also introduce a novel *randomly perforated* matrix sketching method to further control the error in the peeling algorithm.

1 Introduction

Many linear algebra tasks can be solved more efficiently in the presence of structure. As such, a fundamental framework for designing matrix algorithms is to first approximate relevant matrices with structured ones, and then to solve the resulting structured problem. For example, matrices are frequently approximated by *low-rank* matrices, which can be stored in less space, admit faster multiplication by vectors, and more. The power of low-rank structure has motivated significant work on finding *near-optimal* low-rank approximations of arbitrary matrices [DM05; DKM06; HMT11; CW13; NN13; Woo14; MM15; MM17].

For problems in computational science and data science that involve multi-scale phenomena, however, low-rank approximation is often insufficient. Frequently, *hierarchical low-rank structure* is more appropriate; long-range interactions at a given scale can be well-approximated by low-rank matrices, while shorter-range interactions can be recursively treated at a finer scale. Like low-rank matrices, hierarchical low-rank matrices are cheap to store and work with. For instance, matrix-vector multiplies, linear solves, and other operations can typically be done in linear or nearly linear time in the dimension of the matrix [AD13; Cha+07; VXCB16; OX22]. For this reason, hierarchical low-rank approximations lie at the heart of many numerical methods, including the Fast Multipole Method (FMM) [GR87; YDGD03; YBZ04; YIK17], and have applications ranging from partial differential equation (PDE) solvers and control problems to the non-uniform Fourier transform [BGH03; BK16; WEB24]. In fact, the study of hierarchical low-rank matrices is one of the most active topics of research in modern numerical linear algebra [Hac15; BK16].

*The full version of the paper can be accessed at https://arxiv.org/abs/2407.04686

†New York University (tyler.chen@nyu.edu)
‡New York University (fd2135@nyu.edu)
§Cornell University (dh736@cornell.edu)
¶UMass Amherst (cmusco@cs.umass.edu)
‖New York University (cmusco@nyu.edu)
**EPFL (david.persson@epfl.ch)

There are many different types of hierarchical low-rank matrices, including $\mathcal{H}$ matrices, $\mathcal{H}^2$ matrices, and hierarchical semiseparable (HSS) matrices [BK16]. In this work, we consider one of the most general classes: hierarchical off-diagonal low-rank (HODLR) matrices.[1]

DEFINITION 1.1. *Fix a rank parameter k. An $n \times n$ matrix $\mathbf{A}$ is* HODLR(k) *if $n \leq k$ or if $\mathbf{A}$ can be partitioned into $(n/2) \times (n/2)$ blocks*

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{bmatrix}$$

such that $\mathbf{A}_{1,2}$ and $\mathbf{A}_{2,1}$ are each of rank at most k and $\mathbf{A}_{1,1}$ and $\mathbf{A}_{2,2}$ are each HODLR(k).[2]

HODLR matrices are efficient to work with: they can be stored using just $O(nk \log(n/k))$ numbers, multiplied by vectors with $O(nk \log(n/k))$ operations, and admit linear system solves in $O(nk^3 \log(n/k) + nk^2 \log^2(n/k))$ time [BK16].

We are interested in finding a near-optimal HODLR approximation to a matrix $\mathbf{A}$ that can *only* be accessed through black-box matrix-vector product queries $\mathbf{x} \mapsto \mathbf{A}\mathbf{x}$ and matrix-transpose-vector product queries $\mathbf{x} \mapsto \mathbf{A}^\mathsf{T}\mathbf{x}$ (both of which will henceforth be referred to as "matvec queries to $\mathbf{A}$"). This problem has received significant attention due to both practical and theoretical importance [LLY11; Mar11; Mar16; LM24b; BHT23; BT23; LM24a]. For example, black-box HODLR approximation methods can be used to obtain fast direct solvers in settings where we have access to efficient matrix-vector products, e.g., via an FMM or iterative method [LLY11; Mar16]. Black-box methods are also central in the field of *operator or PDE learning*, where $\mathbf{A}$ is an unknown differential operator and one can access matrix-vector products through physical experiments or simulations involving different forcing functions [BT22; BHOT24; SO24].

In these applications and many others, matvec queries with $\mathbf{A}$ are expensive, and typically dominate other computational costs. As such, we hope to minimize the number of queries required to find a near-optimal HODLR approximation for $\mathbf{A}$. Matrix-vector query complexity has become central in theoretical work on numerical linear algebra due to the practical importance of the access model and the fact that it generalizes other models, such as the matrix sketching and Krylov subspace models [SWYZ21]. There has been recent progress giving tight query complexity bounds for central problems like linear system solving, eigenvector approximation, and trace estimation [SER18; BHSW20; MMMW21; DM21; Che+24; JPWZ24]. Closer to our setting, there has also been significant work on the query complexity of structured matrix approximation, with classes like low-rank matrices [BCW22; BN23], diagonal matrices [BKS07; TS11; BN22; DM23], sparse matrices [CPR74; CM83; CC86; WEV13; DSBN15; Ams+24], and beyond [WSB11; SKO21; HT23].

1.1 Formal problem setup

If $\mathbf{A}$ is exactly HODLR(k), it is well known that it can be recovered using $O(k \log(n/k))$ matvec queries using the so-called *peeling algorithm* of Lin, Lu, and Ying [LLY11; Mar16; LM24b].[3] We describe this algorithm in Section 2. However, in most practical situations, $\mathbf{A}$ is not *exactly* HODLR. This has resulted in broad concern about whether peeling algorithms applied to non-HODLR matrices might produce inaccurate approximations [LLY11; BTK19; HT23; BET22; BHT23]. Notably, a recent SIAM Linear Algebra Best Paper [BT22] poses an algorithmic challenge which roughly asks whether a near-HODLR matrix be approximated at nearly the same cost as algorithms for recovering an exactly HODLR matrix. Our work addresses this challenge. In particular, we study the following HODLR approximation problem, which makes no assumptions about $\mathbf{A}$.

PROBLEM 1.1. *Given an $n \times n$ matrix $\mathbf{A}$, accessible only by matvec queries, a rank parameter $k \geq 1$, and an accuracy parameter $\Gamma \geq 1$, find a* HODLR(k) *matrix $\widetilde{\mathbf{A}}$ such that*

$$\|\mathbf{A} - \widetilde{\mathbf{A}}\|_\mathsf{F} \leq \Gamma \cdot \min_{\mathbf{H} \in \mathrm{HODLR}(k)} \|\mathbf{A} - \mathbf{H}\|_\mathsf{F}.$$

[1] HSS matrices are special cases of HODLR matrices, as are other well-studied structured classes, like diagonal plus low-rank matrices [MMW21; TVX23; BCM17].

[2] Definition 1.1 applies to matrices with dimension $n = n_{\text{base}} \cdot 2^L$ for some $n_{\text{base}} \leq k$ and integer $L \geq 0$. Throughout this paper, we will always assume that n and k are related in this way. More general definitions of HODLR matrices are possible, but Definition 1.1 is the most standard [BK16].

[3] For exact recovery, the number of queries used by the peeling algorithm is within a constant factor of optimal; see Theorem 1.2 and Section 6 for a formal lower bound.

Interesting parameter regimes for Problem 1.1 can range from $\Gamma = (1+\varepsilon)$ for $\varepsilon \in (0,1)$ to larger approximation factors when $\mathbf{A}$ is very close to HODLR(k), e.g., $\Gamma = O(\log n)$ or $\Gamma = O(n^c)$ for a small constant c.

Problem 1.1 can be trivially solved for $\Gamma = 1$ using n matrix-vector products. Via multiplication by an identity matrix, we can recover all entries of $\mathbf{A}$ and then compute an exactly optimal HODLR approximation by computing optimal rank-k approximations to $\mathbf{A}$'s top right and bottom left $(n/2) \times (n/2)$ blocks, and recursing on the top left and bottom right blocks. However, outside this baseline and the case when $\mathbf{A}$ is exactly HODLR, we are unaware of any non-trivial results on Problem 1.1. This is despite the vast literature on HODLR matrix approximation and on efficient matrix-vector query methods for vanilla low-rank approximation [CW09; RST09; Coh+15; TYUC17; BCW22; TW23; BN23].

1.2 Main results Our main contribution is an efficient algorithm for solving Problem 1.1.

THEOREM 1.1. *Fix a rank parameter k and accuracy parameter β. Let $L = \lceil \log_2(n/k) \rceil$. There exists a non-adaptive[4] algorithm (Algorithm 3) that uses $O(k/\beta^3 \cdot L)$ matvec queries to $\mathbf{A}$ and solves Problem 1.1 to accuracy $\Gamma = (1+\beta)^L$ with probability at least $99/100$[5]. Apart from matvec queries, the algorithm requires $O(n \cdot \mathrm{poly}(\log(n), k, \beta))$ additional runtime.*

We highlight two interesting instantiations of Theorem 1.1. For any $\varepsilon > 0$, we obtain accuracy $(1+\varepsilon)$ with $O(k \log^4(n/k)/\varepsilon^3)$ total matvec queries by setting $\beta = O(\varepsilon/\log(n/k))$. Alternatively, for any constant $c > 0$, we obtain accuracy n^c with $O(k \log(n/k))$ total matvec queries by setting $\beta = 2^c - 1 = \Theta(1)$. This matches the complexity of existing methods for solving the *recovery problem*, i.e., the case when $\mathbf{A}$ is exactly HODLR.

Theorem 1.1 is obtained through a variant of the popular peeling algorithm for exact HODLR matrix recovery. This algorithm obtains an approximation via a "top down" approach. Specifically, it computes low-rank approximations to $\mathbf{A}$'s top right and bottom left blocks via randomized sketching, implicitly subtracts the results from $\mathbf{A}$, and then recurses on the upper left and lower right blocks. A major technical challenge for applying the peeling algorithm to Problem 1.1 is how to control error that can accumulate across levels of recursion when $\mathbf{A}$'s top right and lower left blocks are not *exactly* rank-k (i.e., when the matrix is not exactly HODLR). This potential accumulation of error has been highlighted in several prior works, including in the earliest work on the peeling method [LLY11; BTK19; BET22], and has been the key challenge in extending peeling to the solve the HODLR approximation problem.

We resolve this long-standing challenge using two new techniques. First, we prove that if peeling is implemented with the so-called *Generalized Nyström Method* for low-rank approximation [CW09; Nak20], sufficient oversampling leads to an algorithm that requires $kL/\mathrm{poly}(\beta)$ matrix-vector products to achieve error $\Gamma = (1+\beta)^{L+1}$. Our proof requires a novel perturbation analysis of sketching for low-rank approximation, and brings to light a surprising fact: the same result cannot be obtained if the more standard *Randomized SVD* method [HMT11] is used for low-rank approximation within peeling. Second, we obtain our final result (with a better polynomial dependence on β) by combining peeling with a new "randomly perforated" sketching distribution that allows for even stronger control of error buildup across recursive levels. Details of the existing peeling algorithm are given in Section 2, and our improvements are described in Section 3.

We complement our upper bound from Theorem 1.1 with a nearly matching lower-bound.

THEOREM 1.2. *For any $\varepsilon > 0$, any (possibly adaptive and randomized) algorithm that solves Problem 1.1 with error $\Gamma = (1+\varepsilon)$ and probability $\geq 1/25$ requires $\Omega(k \log(n/k) + k/\varepsilon)$ matvec queries to $\mathbf{A}$.*[6] *Moreover, any algorithm that solves Problem 1.1 for any finite Γ and any non-zero probability, requires $\Omega(k \log(n/k))$ queries.*

Theorem 1.2 establishes that our $O(k \log(n/k))$ query result to achieve error n^c from Theorem 1.1 cannot be improved in terms of the number of matvec queries, although it might be possible to obtain better error (e.g.,

[4] An algorithm that computes matrix-vector products $\mathbf{B}_1\mathbf{x}_1, \ldots, \mathbf{B}_t\mathbf{x}_t$ where $\mathbf{B}_i = \mathbf{A}$ or $\mathbf{B}_i = \mathbf{A}^\mathsf{T}$ for all $i \in [t]$ is *adaptive* if the choice of $\mathbf{B}_i$ and $\mathbf{x}_i$ depends on the results of prior products $\mathbf{B}_1\mathbf{x}_1, \ldots, \mathbf{B}_{i-1}\mathbf{x}_{i-1}$. The algorithm is *non-adaptive* (also called a *sketching algorithm*) if the queries are all chosen in advance. Non-adaptive methods, like those studied in this paper, are often preferred, as they allow the queries to be evaluated in parallel.

[5] More generally, we show that Algorithm 3 outputs $\widetilde{\mathbf{A}}$ with $\mathbb{E}[\|\mathbf{A} - \widetilde{\mathbf{A}}\|_\mathsf{F}] \leq \Gamma \cdot \min_{\mathbf{H} \in \mathrm{HODLR}(k)} \|\mathbf{A} - \mathbf{H}\|_\mathsf{F}$. The high probability bound stated in Theorem 1.1 can derived from this expectation bound via Markov's inequality – see Section 5.

[6] Formally, for a fixed constant c, we show that there is a distribution over inputs $\mathbf{A} \in \mathbb{R}^{n \times n}$ for $n \geq ck/\epsilon$ such that any (possibly randomized) algorithm using $O(k \log(n/k) + k/\epsilon)$ matvecs succeeds with probability $< 1/25$ over the randomness of the algorithm and the choice of input.

a $\log(n)$ or even constant factor approximation) with the same number of matvecs. Additionally, Theorem 1.2 establishes that our $O(k\log^4(n/k)/\varepsilon^3)$ query result for $(1+\varepsilon)$ error cannot be improved by more than log factors and a $1/\varepsilon^2$ factor. Interestingly, the lower bound shows a separation between the complexity of vanilla low-rank approximation and hierarchical low-rank approximation in terms of dependence on ε. The best known low-rank approximation algorithms use just $\tilde{O}(k/\varepsilon^{1/3})$ matrix-vector products to achieve relative error $(1+\varepsilon)$ [BCW22; MMM24].[7] In contrast, Theorem 1.2 shows that a sublinear dependence on $1/\varepsilon$ *cannot* be achieved for HODLR matrix approximation.

The proof of Theorem 1.2 builds on a growing body of work on lower bounds for adaptive matrix-vector product algorithms [SER18; BHSW20; Che+23], which require techniques beyond those used to prove lower bounds, e.g., in the non-adaptive sketching setting. Our proof draws specifically on a recent result on the number of matrix-vector products required to obtain an optimal block diagonal approximation to a matrix $\mathbf{A}$ [Ams+24].

Organization. The remainder of the paper is organized as follows. In Section 2, we describe the classic peeling algorithm for HODLR matrix recovery, and discuss the challenges in applying it to the HODLR matrix approximation problem (Problem 1.1). In Section 3, we discuss the techniques we use to analyze our variant of peeling, including our new perturbation bound for low-rank approximation. Then, in Section 4, we describe the exact implementation of our algorithm and introduce notation required for our main analysis. The final proof of Theorem 1.1 is given in Section 5. In Section 6, we prove our lower bound, Theorem 1.2. Finally, in Section 7 we show the results of some numerical experiments.

2 The Peeling Algorithm

In this section, we describe the classic peeling method for HODLR matrix recovery, on which our algorithm is based. As discussed, existing analyses of this method apply only in the setting when the input matrix $\mathbf{A}$ is exactly HODLR. We illustrate here how errors can arise when peeling is applied to solve Problem 1.1 in the general case when $\mathbf{A}$ is only approximated by a HODLR matrix. The precise notation used in the figures will be fully described in Section 4.2.

2.1 Low-rank approximation The peeling algorithm makes use of matrix-vector query algorithms for low-rank approximation: given a matrix $\mathbf{B}\in\mathbb{R}^{m_1\times m_2}$, accessible only via matrix-vector queries, we would like to find a rank-k approximation to $\mathbf{B}$, or to exactly recover $\mathbf{B}$ when $\operatorname{rank}(\mathbf{B})\le k$. It is well-known that the best low-rank approximation to $\mathbf{B}$ in the Frobenius norm is $[\![\mathbf{B}]\!]_k$, the rank-k truncated SVD of $\mathbf{B}$.

Perhaps the most well-known algorithms for this task are the Randomized SVD (RSVD) [HMT11] and the Generalized Nyström Method [CW09; TYUC17; Nak20], summarized below:

Algorithm 1 Randomized SVD

procedure RSVD($\mathbf{B}, k, s_{\text{R}}$)
  $\mathbf{\Omega}\sim\text{Gaussian}(m_2, s_{\text{R}})$
  $\mathbf{Q}=\text{orth}(\mathbf{B\Omega})$
  $\mathbf{X}=\mathbf{Q}^{\mathsf{T}}\mathbf{B}$ ▷ $\operatorname{argmin}_{\mathbf{Z}}\|\mathbf{B}-\mathbf{QZ}\|_{\mathsf{F}}$
  return $\mathbf{Q}[\![\mathbf{X}]\!]_k$

Algorithm 2 Generalized Nyström Method

procedure GNM($\mathbf{B}, k, s_{\text{R}}, s_{\text{L}}$)
  $\mathbf{\Omega}\sim\text{Gaussian}(m_2, s_{\text{R}})$, $\mathbf{\Psi}\sim\text{Gaussian}(m_1, s_{\text{L}})$
  $\mathbf{Q}=\text{orth}(\mathbf{B\Omega})$
  $\mathbf{X}=(\mathbf{\Psi}^{\mathsf{T}}\mathbf{Q})^{\dagger}\mathbf{\Psi}^{\mathsf{T}}\mathbf{B}$ ▷ $\operatorname{argmin}_{\mathbf{Z}}\|\mathbf{\Psi}^{\mathsf{T}}\mathbf{B}-\mathbf{\Psi}^{\mathsf{T}}\mathbf{QZ}\|_{\mathsf{F}}$
  return $\mathbf{Q}[\![\mathbf{X}]\!]_k$

Both methods first compute an orthonormal basis $\mathbf{Q}$ for the column span of the matrix $\mathbf{B\Omega}$, formed by multiplying $\mathbf{B}$ by a random sketching matrix $\mathbf{\Omega}$ with s_{R} columns. Throughout, we take this matrix to be Gaussian for simplicity, although most other popular sketching distributions can also be employed; see e.g. [Woo14; HMT11]. RSVD then projects $\mathbf{B}$ onto this column span and forms the best rank-k approximation of the result. Generalized Nyström follows the same idea, but instead computes an approximate projection of $\mathbf{B}$ onto $\mathbf{Q}$ by using a second sketch $\mathbf{\Psi}$ on the left.

If $\operatorname{rank}(\mathbf{B})\le k$, RSVD and Generalized Nyström both recover $\mathbf{B}$ exactly (with probability one) if, respectively, $s_{\text{R}}\ge k$ or $s_{\text{R}}, s_{\text{L}}\ge k$. More generally, these methods can obtain a near-optimal low-rank approximation for

[7] The best known lower bound for vanilla k-rank approximation is $\Omega(k+1/\varepsilon^{1/3})$ matrix-vector products [BCW22]. Combining the k and ε terms, as we do in our Theorem 1.2 for hierarchical approximation, is an open question.

arbitrary $\mathbf{B}$. In particular, if $s_{\text{R}} = O(k/\beta)$, then the output of RSVD satisfies $\|\mathbf{B}-\mathbf{Q}[\![\mathbf{X}]\!]_k\|_{\text{F}} \leq (1+\beta)\cdot\|\mathbf{B}-[\![\mathbf{B}]\!]_k\|_{\text{F}}$ with high probability [Sar06]. The output of Generalized Nyström gives the same guarantee when $s_{\text{R}} = O(k/\beta)$ and $s_{\text{L}} = O(k/\beta^3)$ [TYUC16].

2.2 Exact HODLR recovery Recall from Definition 1.1 that any HODLR(k) matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is composed to four $(n/2)\times(n/2)$ sized blocks. The off-diagonal blocks, $\mathbf{A}_{1,2}$ and $\mathbf{A}_{2,1}$, are rank-k and the on-diagonal blocks, $\mathbf{A}_{1,1}$ and $\mathbf{A}_{2,2}$, are themselves HODLR(k). The key idea of the peeling algorithm is to first recover the low-rank off-diagonal blocks $\mathbf{A}_{1,2}$ and $\mathbf{A}_{2,1}$, to implicitly subtract them from the matrix, and to then continue on to recursively recover the on-diagonal HODLR(k) blocks. More formally, peeling relies on the following observations:

Observation 1. We can perform matrix-vector products with the off-diagonal blocks $\mathbf{A}_{1,2}$ and $\mathbf{A}_{2,1}$ (and their transposes) using matrix-vector products with $\mathbf{A}$ and $\mathbf{A}^{\mathsf{T}}$. In particular, we can compute products with $\mathbf{A}_{2,1}$ and $(\mathbf{A}_{2,1})^{\mathsf{T}}$ by

$$\begin{bmatrix}\mathbf{A}_{1,1} & \mathbf{A}_{1,2}\\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Omega}\\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\sim\\ \mathbf{A}_{2,1}\boldsymbol{\Omega}\end{bmatrix}, \quad \begin{bmatrix}\mathbf{A}_{1,1} & \mathbf{A}_{1,2}\\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2}\end{bmatrix}^{\mathsf{T}}\begin{bmatrix}\mathbf{0}\\ \boldsymbol{\Psi}\end{bmatrix} = \begin{bmatrix}(\mathbf{A}_{2,1})^{\mathsf{T}}\boldsymbol{\Psi}\\ \sim\end{bmatrix}, \tag{2.1}$$

and analogously with $\mathbf{A}_{1,2}$ and $(\mathbf{A}_{1,2})^{\mathsf{T}}$ by

$$\begin{bmatrix}\mathbf{A}_{1,1} & \mathbf{A}_{1,2}\\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2}\end{bmatrix}\begin{bmatrix}\mathbf{0}\\ \boldsymbol{\Omega}\end{bmatrix} = \begin{bmatrix}\mathbf{A}_{1,2}\boldsymbol{\Omega}\\ \sim\end{bmatrix}, \quad \begin{bmatrix}\mathbf{A}_{1,1} & \mathbf{A}_{1,2}\\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2}\end{bmatrix}^{\mathsf{T}}\begin{bmatrix}\boldsymbol{\Psi}\\ \mathbf{0}\end{bmatrix} = \begin{bmatrix}\sim\\ (\mathbf{A}_{1,2})^{\mathsf{T}}\boldsymbol{\Psi}\end{bmatrix}. \tag{2.2}$$

Here, "$\sim$" indicates a block of the output that we ignore in our computations. As discussed in Section 2.1, algorithms like RSVD and Generalized Nyström can exactly recover a rank-k matrix (with probability one) using k products with each the matrix and its transpose. Thus, when $\mathbf{A}$ is exactly HODLR(k), we can fully recover both off-diagonal blocks using just $4k$ total queries to $\mathbf{A}$ (implementing k queries with each of $\mathbf{A}_{1,2}, (\mathbf{A}_{1,2})^{\mathsf{T}}, \mathbf{A}_{2,1}$, and $(\mathbf{A}_{2,1})^{\mathsf{T}}$).[8]

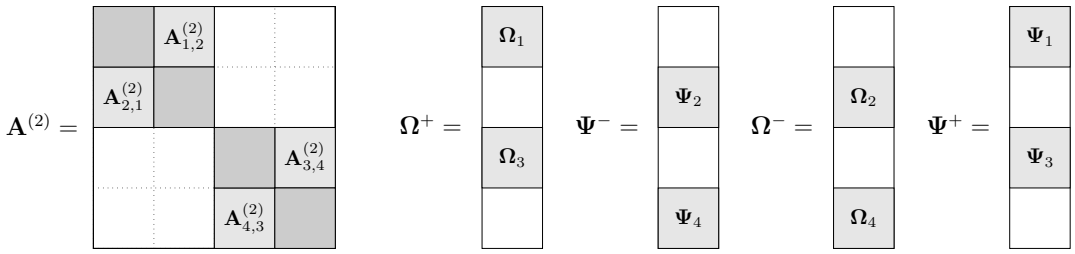

Figure 1: State of the hierarchical matrix at the start of the $\ell = 2$ level of peeling. $\mathbf{A}^{(2)}$ denotes the matrix $\mathbf{A}$ after subtracting the low-rank off-diagonal blocks recovered at the first level – these blocks are zero in $\mathbf{A}^{(2)}$ and shown as white in the figure. For $j = 1, 3$, we can simultaneously obtain the products $\mathbf{A}^{(2)}_{j+1,j}\boldsymbol{\Omega}_j$ and the transpose-products $(\mathbf{A}^{(2)}_{j+1,j})^{\mathsf{T}}\boldsymbol{\Psi}_{j+1}$ from the sketches $\mathbf{A}\boldsymbol{\Omega}^{+}$ and $\mathbf{A}^{\mathsf{T}}\boldsymbol{\Psi}^{-}$. Letting $\boldsymbol{\Omega}^{+}$ and $\boldsymbol{\Psi}^{-}$ have k columns and blocks chosen to be appropriate sketching matrices, these products can be used to exactly recover the rank-k blocks $\mathbf{A}^{(2)}_{j+1,j}$ for $j = 1, 3$. Obtaining products with and recovering $\mathbf{A}^{(2)}_{j-1,j}$ for $j = 2, 4$ can be done analogously using $\boldsymbol{\Omega}^{-}$ and $\boldsymbol{\Psi}^{+}$. Overall, we can recover all four rank-k off diagonal blocks at level $\ell = 2$ using just $4k$ queries to $\mathbf{A}$ (k for each of $\boldsymbol{\Omega}^{+}, \boldsymbol{\Omega}^{-}, \boldsymbol{\Psi}^{+}$, and $\boldsymbol{\Psi}^{-}$).

Observation 2. After exactly obtaining the blocks $\mathbf{A}_{2,1}$ and $\mathbf{A}_{1,2}$, we can *simultaneously* perform matrix-vector products with the on-diagonal blocks $\mathbf{A}_{1,1}$ and $\mathbf{A}_{2,2}$ (and their transposes) using matrix-vector queries to $\mathbf{A}$. In

[8] Algorithms in the literature [LLY11; LM24b; HT23] commonly use a sketch size $k+p$, where p is a small "oversampling" parameter (e.g., $p = 2$). This is important for adding robustness when recovering matrices which are numerically, but not exactly, rank-k. This is not necessary if we assume exact computations and exactly rank-k blocks.

particular, we can compute for any $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_1$,

$$\begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{bmatrix} \begin{bmatrix} \mathbf{\Omega}_1 \\ \mathbf{\Omega}_2 \end{bmatrix} - \begin{bmatrix} \mathbf{A}_{1,2}\mathbf{\Omega}_2 \\ \mathbf{A}_{2,1}\mathbf{\Omega}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{1,1}\mathbf{\Omega}_1 \\ \mathbf{A}_{2,2}\mathbf{\Omega}_2 \end{bmatrix},$$

and analogously

$$\begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \mathbf{\Psi}_1 \\ \mathbf{\Psi}_2 \end{bmatrix} - \begin{bmatrix} (\mathbf{A}_{1,2})^{\mathsf{T}}\mathbf{\Psi}_2 \\ (\mathbf{A}_{2,1})^{\mathsf{T}}\mathbf{\Psi}_1 \end{bmatrix} = \begin{bmatrix} (\mathbf{A}_{1,1})^{\mathsf{T}}\mathbf{\Psi}_1 \\ (\mathbf{A}_{2,2})^{\mathsf{T}}\mathbf{\Psi}_2 \end{bmatrix}.$$

Since $\mathbf{A}_{1,1}$ and $\mathbf{A}_{2,2}$ are themselves HODLR(k), by Observation 1, we can recover their low-rank off diagonal blocks (each of size $(n/4) \times (n/4)$) using just $4k$ queries to $\mathbf{A}$ (see Figure 1).

Observation 3. We can repeat this process, recursing toward the diagonal, to recover smaller and smaller off-diagonal low-rank blocks. Critically, the number of matrix-vector products used to recover the off-diagonal blocks at a given level depends only on the rank parameter k and is *independent* of the level itself. After $L = \lceil \log_2(n/k) \rceil$ recursive steps, the blocks will be of size at most k. At this point, the diagonal blocks can be recovered all at once using k matrix-vector products. Overall, $4k$ queries to $\mathbf{A}$ are used at each level, and k are used at the final level, giving total cost of $O(k \cdot (L+1))$ matvec queries.

2.3 Peeling in the presence of error The peeling algorithm described in Section 2.2 assumes that the matrix $\mathbf{A}$ is exactly HODLR(k). In particular, Observation 2 does not hold if the off-diagonal blocks at the previous level are not recovered exactly. This is illustrated in Figure 2.

As noted in [LLY11], "it is a natural concern that whether the error from low-rank decompositions on top levels accumulates in the peeling steps." In particular, if all of the error at a given level propagated to the next level through perturbations on the desired sketches, the error could double at each level; i.e., result in an *exponential* blow-up of the error with respect to the number of levels. In fact, as we demonstrate in Section 7.3, certain variants of the peeling algorithm can exhibit such a failure mode on hard instances. Thus, understanding the propagation of error from one level to the next is critical in the design and analysis of peeling algorithms. In Section 3 we discuss the techniques we use to control and analyze this error propagation.

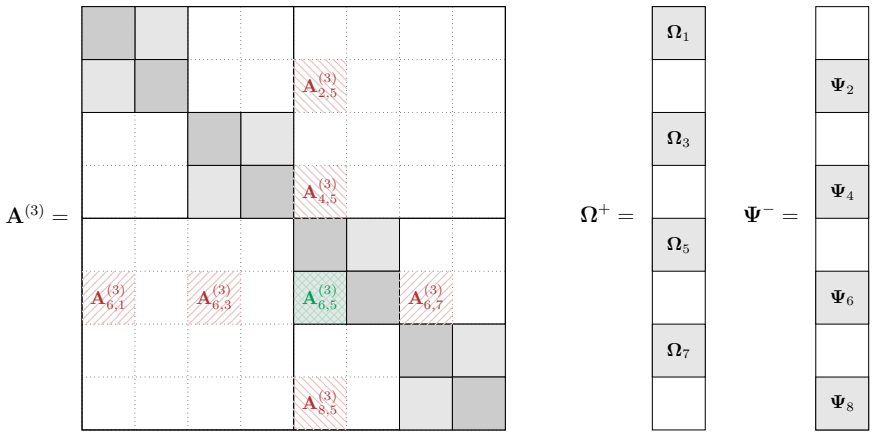

Figure 2: State of the hierarchical matrix at the start of the $\ell = 3$ level of peeling. $\mathbf{A}^{(3)}$ denotes the matrix $\mathbf{A}$ after subtracting off-diagonal blocks that were (approximately) recovered at the first two levels. We would like to use the sketches $\mathbf{A}^{(3)}_{6,5}\mathbf{\Omega}_5$ and $(\mathbf{A}^{(3)}_{6,5})^{\mathsf{T}}\mathbf{\Psi}_6$ to obtain a low-rank approximation to the off-diagonal block $\mathbf{A}^{(3)}_{6,5}$. However, if the off-diagonal blocks at previous levels were not recovered exactly (since these blocks may not be exactly rank-k), then after subtraction, these blocks will not be exactly zero in $\mathbf{A}^{(3)}$. We thus obtain perturbed sketches of the form $\mathbf{A}^{(3)}_{6,5}\mathbf{\Omega}_5 + \mathbf{A}^{(3)}_{6,1}\mathbf{\Omega}_1 + \mathbf{A}^{(3)}_{6,3}\mathbf{\Omega}_3 + \mathbf{A}^{(3)}_{6,7}\mathbf{\Omega}_7$ and $(\mathbf{A}^{(3)}_{6,5})^{\mathsf{T}}\mathbf{\Psi}_6 + (\mathbf{A}^{(3)}_{2,5})^{\mathsf{T}}\mathbf{\Psi}_2 + (\mathbf{A}^{(3)}_{4,5})^{\mathsf{T}}\mathbf{\Psi}_4 + (\mathbf{A}^{(3)}_{8,5})^{\mathsf{T}}\mathbf{\Psi}_8$.

3 Techniques

As discussed in Section 2.3, when the peeling algorithm is applied to a matrix $\mathbf{A}$ that is not exactly HODLR, errors at previous levels can pollute the matrix-vector products at a given level. The aim of this paper is to show that these errors can be controlled in such a way that the peeling algorithm can still solve Problem 1.1 to high accuracy. To do this we must understand: (1) how matrix-vector query algorithms for low-rank approximation are impacted by noise in their matvecs and (2) how this noise can be controlled in the setting of peeling.[9] In the next two subsections we outline the high-level ideas we use to address each question.

3.1 A perturbation bound for low-rank approximation

To understand how matvec query errors impact low-rank approximation algorithms, including RSVD and Generalized Nyström (see Section 2.1), we develop the following perturbation bound, which we believe will be of general interest. The bound is proven in Section 5.1.

THEOREM 3.1. *Let* $\mathbf{B} \in \mathbb{R}^{m_1 \times m_2}$, $\mathbf{\Omega} \in \mathbb{R}^{m_2 \times s}$, $\mathbf{E}_1 \in \mathbb{R}^{m_1 \times s}$, *and* $\mathbf{E}_2 \in \mathbb{R}^{s \times m_2}$, *and let* $\mathbf{Q} = \mathrm{orth}(\mathbf{B\Omega} + \mathbf{E}_1)$. *Write* $\mathbf{B}$ *in its singular value decomposition as*

$$\mathbf{B} = \mathbf{U\Sigma V}^\mathsf{T} = \begin{bmatrix} \mathbf{U}_{\text{top}} & \mathbf{U}_{\text{bot}} \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_{\text{top}} & \\ & \mathbf{\Sigma}_{\text{bot}} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{\text{top}}^\mathsf{T} \\ \mathbf{V}_{\text{bot}}^\mathsf{T} \end{bmatrix},$$

where $\mathbf{U}$ *and* $\mathbf{V}$ *are orthonormal and* $\mathbf{\Sigma}$ *is diagonal, and the "top" blocks represent the top* k *columns; i.e.,* $\mathbf{U}_{\text{top}} \in \mathbb{R}^{m_1 \times k}, \mathbf{V}_{\text{top}} \in \mathbb{R}^{m_2 \times k}$, *and* $\mathbf{\Sigma}_{\text{top}} \in \mathbb{R}^{k \times k}$. *Define also*

$$\mathbf{\Omega}_{\text{top}} = \mathbf{V}_{\text{top}}^\mathsf{T}\mathbf{\Omega}, \quad \mathbf{\Omega}_{\text{bot}} = \mathbf{V}_{\text{bot}}^\mathsf{T}\mathbf{\Omega}.$$

Then, assuming $\mathrm{rank}(\mathbf{\Omega}_{\text{top}}) = k$,

$$\|\mathbf{B} - \mathbf{Q}[\![\mathbf{Q}^\mathsf{T}\mathbf{B} + \mathbf{E}_2]\!]_k\|_\mathsf{F} \le \|\mathbf{E}_1\mathbf{\Omega}_{\text{top}}^\dagger\|_\mathsf{F} + 2\|\mathbf{E}_2\|_\mathsf{F} + \left(\|\mathbf{\Sigma}_{\text{bot}}\|_\mathsf{F}^2 + \|\mathbf{\Sigma}_{\text{bot}}\mathbf{\Omega}_{\text{bot}}\mathbf{\Omega}_{\text{top}}^\dagger\|_\mathsf{F}^2\right)^{1/2}.$$

Theorem 3.1 is most naturally a perturbation bound for the RSVD method, which, as discussed in Section 2.1, first computes an orthonormal basis $\mathbf{Q}$ for $\mathbf{B\Omega}$ where $\mathbf{\Omega}$ is a random sketching matrix, and then outputs $\mathbf{Q}[\![\mathbf{Q}^\mathsf{T}\mathbf{B}]\!]_k$ – the best rank-k approximation to $\mathbf{B}$ lying in the span of $\mathbf{Q}$. The error term $\mathbf{E}_1$ captures error that occurs during the initial sketching step, i.e., due to noise in computing the matrix-vector products $\mathbf{B\Omega}$. Similarly, $\mathbf{E}_2$ captures noise in the second projection step. Theorem 3.1 can be used to recover the standard bounds for RSVD implemented with exact matrix-vector products (i.e., where $\mathbf{E}_1$ and $\mathbf{E}_2$ are zero). In particular, in this case the RSVD Frobenius error is bounded by:

$$\|\mathbf{B} - \mathbf{Q}[\![\mathbf{Q}^\mathsf{T}\mathbf{B}]\!]_k\|_\mathsf{F} \le \left(\|\mathbf{\Sigma}_{\text{bot}}\|_\mathsf{F}^2 + \|\mathbf{\Sigma}_{\text{bot}}\mathbf{\Omega}_{\text{bot}}\mathbf{\Omega}_{\text{top}}^\dagger\|_\mathsf{F}^2\right)^{1/2} = \left(\|\mathbf{B} - [\![\mathbf{B}]\!]_k\|_\mathsf{F}^2 + \|\mathbf{\Sigma}_{\text{bot}}\mathbf{\Omega}_{\text{bot}}\mathbf{\Omega}_{\text{top}}^\dagger\|_\mathsf{F}^2\right)^{1/2}.$$

When the sketching matrix $\mathbf{\Omega}$ is Gaussian, we can observe that $\mathbf{\Omega}_{\text{bot}}$ and $\mathbf{\Omega}_{\text{top}}$ are independent Gaussian matrices since $\mathbf{V}_{\text{top}}$ and $\mathbf{V}_{\text{bot}}$ are orthogonal to each other. This allows us to apply a standard bound (see Theorem 5.1) to argue that when $s = O(k/\beta)$, $\mathbb{E}[\|\mathbf{\Sigma}_{\text{bot}}\mathbf{\Omega}_{\text{bot}}\mathbf{\Omega}_{\text{top}}^\dagger\|_\mathsf{F}^2] \le \beta\|\mathbf{\Sigma}_{\text{bot}}\|_\mathsf{F}^2 = \beta\|\mathbf{B} - [\![\mathbf{B}]\!]_k\|_\mathsf{F}^2$, which ultimately gives that $\mathbb{E}[\|\mathbf{B} - \mathbf{Q}[\![\mathbf{Q}^\mathsf{T}\mathbf{B}]\!]_k\|_\mathsf{F}] \le (1+\beta)\|\mathbf{B} - [\![\mathbf{B}]\!]_k\|_\mathsf{F}$.

Theorem 3.1 can also be used as a perturbation bound for the Generalized Nyström Method. In this case, $\mathbf{E}_2$ is used to account for the error due to using an approximate projection onto $\mathbf{Q}$ via a sketched regression problem, as well as the errors in the matvecs with $\mathbf{A}^\mathsf{T}$ used to set up the regression problem. We discuss this further in Section 3.2. As with RSVD, Theorem 3.1 matches the current best known bounds for Generalized Nyström implemented with exact matvecs.

[9] In [BHT23], a similar approach is taken to analyze an infinite dimensional variant of the peeling algorithm for the task of approximating the Green's function of an elliptic partial differential (PDE) operator by a HODLR operator. However, [BHT23] relies strongly on the fact that the off-diagonal blocks of the Green's function have rapid spectral decay and are thus low-rank, up to exponentially small error [BH03]. This allows the overall error to be controlled, even if the error can grow exponentially across levels. In contrast, we make no assumptions about $\mathbf{A}$.

3.2 Low-rank approximation in the peeling algorithm We now consider how Theorem 3.1 can be applied to control the propagation of error in the peeling algorithm implemented with different base algorithms for low-rank approximation of the off-diagonal blocks..

Randomized SVD. First consider implementing peeling with RSVD as the base low-rank approximation algorithm. In this case, $\mathbf{E}_1$ and $\mathbf{E}_2$ will be nonzero since we cannot compute exact matrix-vector products with an off-diagonal low-rank block $\mathbf{B}$ due to noise from previous levels (i.e., we cannot exactly compute $\mathbf{B}\boldsymbol{\Omega}$ and $\mathbf{Q}^{\mathsf{T}}\mathbf{B}$). Fortunately, the error $\mathbf{E}_1$ that arises in peeling has a particular structure that can be used to bound the $\|\mathbf{E}_1\boldsymbol{\Omega}_{\text{top}}^{\dagger}\|_{\mathsf{F}}$ term in Theorem 3.1. In particular, recalling Figure 2, we will have

$$\mathbf{E}_1 = \sum_j \mathbf{M}_j\boldsymbol{\Omega}_j = \mathbf{M}\widetilde{\boldsymbol{\Omega}}, \qquad \mathbf{M} = \begin{bmatrix}\mathbf{M}_1 & \mathbf{M}_2 & \cdots\end{bmatrix}, \qquad \widetilde{\boldsymbol{\Omega}} = \begin{bmatrix}\boldsymbol{\Omega}_1 \\ \boldsymbol{\Omega}_2 \\ \vdots\end{bmatrix}, \tag{3.3}$$

where the $\mathbf{M}_j$ are fixed matrices that depend on the recovery error at the previous levels of peeling, and the $\boldsymbol{\Omega}_j$ are independent Gaussian matrices.[10] $\widetilde{\boldsymbol{\Omega}}$ is itself a Gaussian matrix that is independent from $\boldsymbol{\Omega}_{\text{top}}$, allowing us to bound the $\mathbb{E}[\|\mathbf{E}_1\boldsymbol{\Omega}_{\text{top}}^{\dagger}\|_{\mathsf{F}}^2]$ term in Theorem 3.1 by $\mathbb{E}[\|\mathbf{E}_1\boldsymbol{\Omega}_{\text{top}}^{\dagger}\|_{\mathsf{F}}^2] = \mathbb{E}[\|\mathbf{M}\widetilde{\boldsymbol{\Omega}}\boldsymbol{\Omega}_{\text{top}}^{\dagger}\|_{\mathsf{F}}^2]] \le \beta\|\mathbf{M}\|_{\mathsf{F}}^2$ when $s = O(k/\beta)$, again using Theorem 5.1. I.e., with a large enough sketch size, we can drive down the error term $\mathbf{E}_1$ to be arbitrarily small, and ensure that it does not accumulate too much across the levels of peeling.

Unfortunately, we cannot handle the $\|\mathbf{E}_2\|_{\mathsf{F}}$ term in Theorem 3.1 in the same way. We have $\mathbf{E}_2 = \sum_j \mathbf{M}_j\mathbf{Q}_j$ for orthogonal matrices $\mathbf{Q}_j$ computed for each off-diagonal block at the current level. The $\mathbf{Q}_j$ and $\mathbf{M}_j$ matrices may be arbitrarily correlated, and so it is not clear that $\|\mathbf{E}_2\|_{\mathsf{F}}$ can be made small compared to the noise from the previous levels, $\|\mathbf{M}\|_{\mathsf{F}}$. In fact, as discussed in Section 7.3, theoretical and empirical evidence suggests the existence of hard instances where standard peeling with RSVD can suffer exponential error blow-up across levels.

Generalized Nyström. Our first key observation is that when peeling is implemented with the Generalized Nyström method as the base low-rank approximation algorithm, rather than RSVD, the $\|\mathbf{E}_2\|_{\mathsf{F}}$ error term in Theorem 3.1 can in fact be bounded. For peeling implemented with Generalized Nyström, $\mathbf{E}_2$ encapsulates two sources of error: the first is due to the fact that Generalized Nyström does not exactly return $\mathbf{Q}[\![\mathbf{Q}^{\mathsf{T}}\mathbf{B}]\!]_k$, but rather computes $\mathbf{X} = \operatorname{argmin}_{\mathbf{Z}} \|\boldsymbol{\Psi}^{\mathsf{T}}\mathbf{B} - \boldsymbol{\Psi}^{\mathsf{T}}\mathbf{Q}\mathbf{Z}\|_{\mathsf{F}}$ for a random sketching matrix $\boldsymbol{\Psi}$ and returns $\mathbf{Q}[\![\mathbf{X}]\!]_k$. We can think of $\mathbf{X}$ as an approximation to $\mathbf{Q}^{\mathsf{T}}\mathbf{B} = \operatorname{argmin}_{\mathbf{Z}} \|\mathbf{B} - \mathbf{Q}\mathbf{Z}\|_{\mathsf{F}}$. It is well known that this source of error can be controlled by setting the size of the sketch $\boldsymbol{\Psi}$ large enough – this leads to the standard analysis of Generalized Nyström from the literature [TYUC16].

The second source of error in $\mathbf{E}_2$ is due to noise in computing the matrix-vector products $\boldsymbol{\Psi}^{\mathsf{T}}\mathbf{B}$. Fortunately, analogously to how we bounded $\mathbf{E}_1$ for RSVD in Section 3.1, we can leverage the fact that in peeling, this noise has the structure of equation (3.3) – it is the product of a fixed matrix $\mathbf{M}$ (the recovery error at previous levels of peeling) and a random Gaussian matrix $\widetilde{\boldsymbol{\Psi}}$ that is independent of $\boldsymbol{\Psi}$. Using this structure, we are able to bound its impact on the final approximation quality. In particular, we use Theorem 3.1 to give the following bound for the Generalized Nyström Method with Gaussian errors as in (3.3). This bound is proven in Section 5.1.

THEOREM 3.2. *Let* $\mathbf{B} \in \mathbb{R}^{m_1\times m_2}$, $\mathbf{M} \in \mathbb{R}^{m_1\times p}$, *and* $\mathbf{N} \in \mathbb{R}^{q\times m_2}$ *be fixed matrices, and let* $\boldsymbol{\Omega} \sim \text{Gaussian}(m_2, s_{\text{R}})$, $\widetilde{\boldsymbol{\Omega}} \sim \text{Gaussian}(p, s_{\text{R}})$, $\boldsymbol{\Psi} \sim \text{Gaussian}(m_1, s_{\text{L}})$, *and* $\widetilde{\boldsymbol{\Psi}} \sim \text{Gaussian}(q, s_{\text{L}})$ *be independent. Define*

$$\mathbf{Q} := \operatorname{orth}(\mathbf{B}\boldsymbol{\Omega} + \mathbf{M}\widetilde{\boldsymbol{\Omega}}), \qquad \mathbf{X} := (\boldsymbol{\Psi}^{\mathsf{T}}\mathbf{Q})^{\dagger}(\boldsymbol{\Psi}^{\mathsf{T}}\mathbf{B} + \widetilde{\boldsymbol{\Psi}}^{\mathsf{T}}\mathbf{N}).$$

Then, provided $s_{\text{R}} > 2k+1$ *and* $s_{\text{L}} > 2s_{\text{R}}+1$

$$\mathbb{E}\left[\|\mathbf{B} - \mathbf{Q}[\![\mathbf{X}]\!]_k\|_{\mathsf{F}}^2\right] \le E_1 + E_2 + 2\sqrt{E_1E_2},$$

where

$$E_1 := \left(1 + \frac{k}{s_{\text{R}} - k - 1}\right)\|\mathbf{B} - [\![\mathbf{B}]\!]_k\|_{\mathsf{F}}^2$$

$$E_2 := \frac{18k}{s_{\text{R}} - k - 1}\|\mathbf{M}\|_{\mathsf{F}}^2 + \frac{8s_{\text{R}}}{s_{\text{L}} - s_{\text{R}} - 1}\|\mathbf{N}\|_{\mathsf{F}}^2 + \frac{32s_{\text{R}}}{s_{\text{L}} - s_{\text{R}} - 1}\|\mathbf{B} - [\![\mathbf{B}]\!]_k\|_{\mathsf{F}}^2.$$

[10] In Figure 2, the $\mathbf{M}_j\boldsymbol{\Omega}_j$ terms when $\mathbf{B} = \mathbf{A}_{6,5}^{(3)}$ are $\mathbf{A}_{6,1}^{(3)}\boldsymbol{\Omega}_1$, $\mathbf{A}_{6,3}^{(3)}\boldsymbol{\Omega}_3$, and $\mathbf{A}_{6,7}^{(3)}\boldsymbol{\Omega}_7$.

We can see from Theorem 3.2 that, for fixed noise matrices $\mathbf{M}$ and $\mathbf{N}$, if we set s_{R} large enough compared to k and s_{L} large enough compared to s_{R}, we can drive the expected error of the rank-k approximation $\mathbf{Q}[\![\mathbf{X}]\!]_k$ arbitrarily close to the error of the best possible rank-k approximation $[\![\mathbf{B}]\!]_k$. Formally, in Section 5 (Theorem 5.2) we use Theorem 3.2 to show that for any $\beta \in (0,1)$, if we set $s_{\text{R}} = O(k/\beta^2)$ and $s_{\text{L}} = O(s_{\text{R}}/\beta^2) = O(k/\beta^4)$, then at each level of peeling, our approximation error blows up by at most a $(1+\beta)$ factor. Over $L+1$ levels, we obtain final accuracy $\Gamma = (1+\beta)^{L+1}$. This matches our main result, Theorem 1.1, but with a slightly worse dependence on β: we require $O(k/\beta^4 \cdot L)$ rather than $O(k/\beta^3 \cdot L)$ total matvecs. In Section 3.3 we show how to give the improved β dependence using a novel sketching approach to further control the noise terms $\mathbf{M}$ and $\mathbf{N}$.

We note that the algorithm described above, which simply implements standard peeling with Generalized Nyström is particularly simple and performs well experimentally – see Section 7. It would be interesting to understand if our bounds on s_{R} and s_{L} are tight – we suspect that they are not. In particular, even in the case of exact matrix-vector products, we suspect that $s_{\text{L}} = O(k/\beta^2)$ suffices to achieve a $(1+\beta)$ approximation, even though current best known bound is $O(k/\beta^3)$. Giving an improved bound in this setting would very likely yield an improved bound in our setting.

3.3 Randomly perforated Gaussian sketching We next discuss how we can further improve the query complexity of peeling with Generalized Nyström-based low-rank approximation by altering the sketching distribution to explicitly reduce matvec query errors that arise due to inexact recovery at previous levels.

Referring back to Figures 1 and 2, we observe that at each level the peeling algorithm employs two random sketching matrices on the right: $\mathbf{\Omega}^+$ and $\mathbf{\Omega}^-$ and two on the left: $\mathbf{\Psi}^+$ and $\mathbf{\Psi}^-$. For sake of exposition we focus here just on the right sketches, $\mathbf{\Omega}^+$ and $\mathbf{\Omega}^-$, which are both block matrices with 2^ℓ blocks, alternating between random Gaussian blocks (i.e., $\mathbf{\Omega}_1$, $\mathbf{\Omega}_2$, $\mathbf{\Omega}_3$, ...) and zero blocks. In particular, $\mathbf{\Omega}^+$ has even blocks set to zero, while $\mathbf{\Omega}^-$ has odd blocks set to zero.

This "perforated" structure arises naturally from Observation 1 and Observation 2. Intuitively, since $\mathbf{\Omega}^+$ has even blocks set to zero, it has no interaction with the diagonal blocks at level ℓ with even indices. Thus, it can be used to simultaneously produce right sketches of every bottom-left off-diagonal block at level ℓ, without incurring any error due to the unrecovered on-diagonal blocks.

Of course, as discussed, when the peeling algorithm is applied to a non-HODLR matrix, the sketch does incur error due to inexact recovery at the previous levels. In particular, referring to Figure 2, when approximating a given off-diagonal block at level ℓ (e.g., $\mathbf{A}^{(3)}_{6,5}$ in the figure), the sketch incurs error from $2^{\ell-1}-1$ blocks that were only approximately recovered in previous levels of peeling (i.e., $\mathbf{A}^{(3)}_{6,1}, \mathbf{A}^{(3)}_{6,3}$, and $\mathbf{A}^{(3)}_{6,7}$ in the figure).

Our key idea to reduce this error is simple: we will increase the sparsity of $\mathbf{\Omega}^+$ and $\mathbf{\Omega}^-$, so that a higher fraction of blocks are set to zero. Thus, when recovering each block at level ℓ, we will incur error due to a smaller number of inexactly recovered blocks from the previous levels. We still need non-zero blocks for each of the 2^ℓ off-diagonal blocks that are recovered at level ℓ, so our sparser matrices will have a large number of block columns than before. See Figure 3 for an illustration.

Of course, just decreasing the number of inexactly recovered blocks that introduce error into each of our sketches may not decrease the magnitude of this error if the error is all concentrated on a few blocks. Thus, we will chose the nonzero blocks of our sketches randomly, ensuring that the expected error when recovering each block at level ℓ is small. Formally, our modified sketches $\mathbf{\Omega}^+, \mathbf{\Omega}^-, \mathbf{\Psi}^+$, and $\mathbf{\Psi}^-$ will be $2^\ell \times t$ block matrices, with a single Gaussian block placed randomly in each odd (resp. even) block row. They can be thought of as block Kronecker products of Gaussian matrices with *Count-sketch-like* sparse sketching matrices – see Section 4.1 for a more complete description. We call our sketches *randomly perforated Gaussian sketches*.

By increasing the number of block columns t in our randomly perforated sketches, we decrease the expected matvec error introduced by inexactly recovered blocks at each level. In particular, we show that the expected noise added to each matvec query is decreased by a factor of $1/t$ in the squared Frobenius norm. We describe our variant of Generalized Nyström-based peeling with perforated sketching in Algorithm 3 and analyze it in Section 5 (Theorem 5.2), using the perturbation bound for Generalized Nyström described in Section 3.2. Ultimately, this approach gives our main result Theorem 1.1.

We note that perforated sketching can also be combined with vanilla RSVD-based peeling, giving a similar bound to Theorem 1.1. We describe the resulting algorithm in Appendix A. We numerically compare our main Generalized Nyström Method-based algorithm with the RSVD-based algorithm in Section 7.

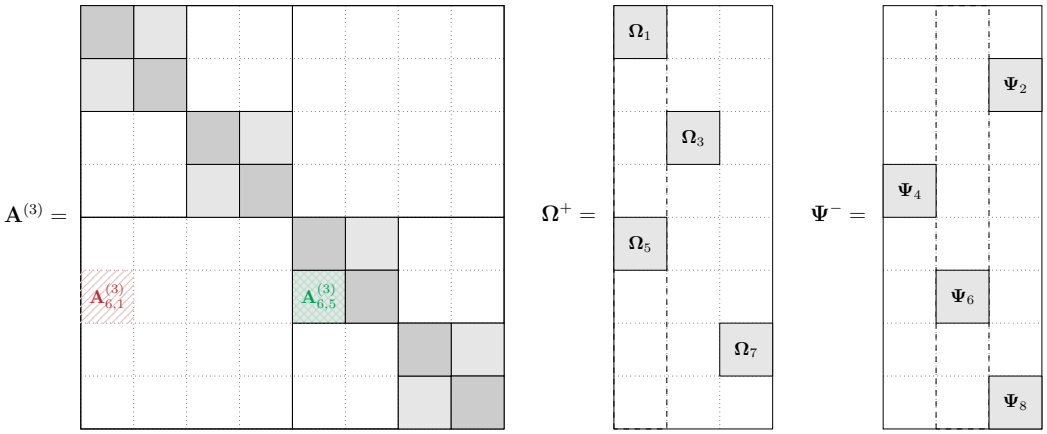

Figure 3: As in Figures 1 and 2, we aim to obtain the sketches $\mathbf{A}^{(3)}_{j\pm1,j}\boldsymbol{\Omega}^+_j$ and $(\mathbf{A}^{(3)}_{j\pm1,j})^\mathsf{T}\boldsymbol{\Psi}^-_{j\pm1}$ from the sketches $\mathbf{A}^{(3)}\boldsymbol{\Omega}^+$ and $(\mathbf{A}^{(3)})^\mathsf{T}\boldsymbol{\Psi}^-$. The key observation is that by using *randomly perforated Gaussian sketches*, which decrease the number of nonzero blocks per block-column (while increasing the number of column. This results in less error. We illustrate the error for block $\mathbf{A}^{(3)}_{6,5}$. Note that we obtain sketches $\mathbf{A}^{(3)}_{6,5}\boldsymbol{\Omega}_5 + \mathbf{A}^{(3)}_{6,1}\boldsymbol{\Omega}_1$ and $(\mathbf{A}^{(3)}_{6,5})^\mathsf{T}\boldsymbol{\Psi}_6$. Thus, the error is decreased compared to Figure 2.

4 The Generalized Nyström Method Peeling Algorithm

We now describe our variant of the peeling algorithm, which is summarized as Algorithm 3. In Sections 4.1 and 4.2 we describe preliminary notation and give pseudocode for the algorithm. Then, in Sections 4.3 and 4.4 we describe how the two key stages of the Generalized Nyström Method are approximately implemented in the peeling algorithm and introduce notation we will use in the analysis. Finally, in Section 4.5 we discuss how to recover the diagonal blocks at the final level. Analysis of the method appears in Section 5.

4.1 Notation for sketching distributions

We begin by defining notation needed to describe the "randomly perforated Gaussian" sketches used in our variant of the peeling algorithm.

DEFINITION 4.1. *Let* $\mathbf{X} \in \mathbb{R}^{p\times v}$ *and* $\mathbf{Y} \in \mathbb{R}^{pu\times t}$. *We define the product* $\mathbf{X} \bullet \mathbf{Y} \in \mathbb{R}^{pu\times vt}$ *by*

$$\underbrace{\begin{bmatrix} x_{1,1} & \cdots & x_{1,v} \\ \vdots & & \vdots \\ x_{p,1} & \cdots & x_{p,v} \end{bmatrix}}_{\mathbf{X}} \bullet \underbrace{\begin{bmatrix} \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_p \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\begin{bmatrix} x_{1,1}\mathbf{Y}_1 & \cdots & x_{1,v}\mathbf{Y}_1 \\ \vdots & & \vdots \\ x_{p,1}\mathbf{Y}_p & \cdots & x_{p,v}\mathbf{Y}_p \end{bmatrix}}_{\mathbf{X}\bullet\mathbf{Y}},$$

where $\mathbf{Y}_1, \ldots, \mathbf{Y}_p \in \mathbb{R}^{u\times t}$.

Note that " $\bullet$ " is a block row-wise Kronecker product; i.e. the block-rows of $\mathbf{X}\bullet\mathbf{Y}$ are obtained as the Kronecker product of the rows of $\mathbf{X}$ and the block-rows of $\mathbf{Y}$.

DEFINITION 4.2. *We say* $\boldsymbol{\Omega} \sim \text{Gaussian}(p, q)$ *if each entry of* $\boldsymbol{\Omega} \in \mathbb{R}^{p\times q}$ *is independently a standard normal random variable.*

DEFINITION 4.3. *We say* $\boldsymbol{\xi} \sim \text{CountSketch}(d, t)$ *if* $\boldsymbol{\xi} \in \{0,1\}^{d\times t}$ *is such that each row of* $\boldsymbol{\xi}$ *independently has exactly one nonzero entry in a uniformly random column. We respectively denote the* (j, i) *entry of* $\boldsymbol{\xi}$ *as* $\xi_{j,i}$.

DEFINITION 4.4. *We say* $\boldsymbol{\xi}^+, \boldsymbol{\xi}^- \sim \text{PerfCountSketch}(d, t)$ *if*

$$\boldsymbol{\xi}^+ = \begin{bmatrix} 1 & 0 & 1 & 0 & \cdots \end{bmatrix}^\mathsf{T} \bullet \boldsymbol{\xi}, \qquad \boldsymbol{\xi}^- = \begin{bmatrix} 0 & 1 & 0 & 1 & \cdots \end{bmatrix}^\mathsf{T} \bullet \boldsymbol{\xi},$$

where $\boldsymbol{\xi} \sim \text{CountSketch}(d, t)$. *We respectively denote the* (j, i) *entry of* $\boldsymbol{\xi}^+$ *and* $\boldsymbol{\xi}^-$ *as* $\xi^+_{j,i}$ *and* $\xi^-_{j,i}$.

We are now ready to formally define the randomly perforated Gaussian sketching distribution.

DEFINITION 4.5. *We say* $\boldsymbol{\xi}^+, \boldsymbol{\xi}^-, \boldsymbol{\Omega}^+, \boldsymbol{\Omega}^- \sim \text{RandPerfGaussian}(n, d, s, t)$ *if*

$$\boldsymbol{\Omega}^+ = \boldsymbol{\xi}^+ \bullet \begin{bmatrix} \boldsymbol{\Omega}_1 \\ \vdots \\ \boldsymbol{\Omega}_d \end{bmatrix}, \qquad \boldsymbol{\Omega}^- = \boldsymbol{\xi}^- \bullet \begin{bmatrix} \boldsymbol{\Omega}_1 \\ \vdots \\ \boldsymbol{\Omega}_d \end{bmatrix},$$

where $\boldsymbol{\xi}^+, \boldsymbol{\xi}^- \sim \text{PerfCountSketch}(d, t)$ *and each* $\boldsymbol{\Omega}_i \sim \text{Gaussian}(n/d, s)$ *independently.*

The sketching matrices $\boldsymbol{\Omega}^+$ and $\boldsymbol{\Psi}^-$ shown in Figure 3 illustrate the sparsity structure of the randomly perforated Gaussian sketching distribution (Definition 4.5) with $t = 3$. In the case that $t = 1$, then the randomly perforated Gaussian sketching matrices are of the form used by the standard peeling algorithm – see Figure 1.

4.2 Notation for iteration and pseudocode At level ℓ, the peeling algorithm aims to approximate the 2^ℓ low-rank off-diagonal blocks of $\mathbf{A}$, each of which is of size $n/2^\ell$.

The resulting approximation to these blocks is placed in a matrix $\mathbf{H}^{(\ell)}$. This is repeated for $L := \lceil \log_2(n/k) \rceil \leq O(\log(n/k))$ levels, at which points the blocks are of size at most k. At the final level $\ell = L$, the algorithm also obtains an approximation $\widehat{\mathbf{H}}$ to the 2^L diagonal blocks. The final approximation is then expressed in terms of the approximations at each level as

$$\widetilde{\mathbf{A}} = \mathbf{H}^{(1)} + \cdots + \mathbf{H}^{(L)} + \widehat{\mathbf{H}}. \tag{4.4}$$

Suppose we are at level ℓ, and let $\mathbf{A}^{(\ell)}$ be the matrix $\mathbf{A}$ at step ℓ after subtracting the approximations $\mathbf{H}^{(\ell-1)}, \ldots, \mathbf{H}^{(1)}$ from the previous levels; that is

$$\mathbf{A}^{(\ell)} := \mathbf{A} - (\mathbf{H}^{(\ell-1)} + \cdots + \mathbf{H}^{(1)}).$$

Partition $\mathbf{A}^{(\ell)}$ into a $2^\ell \times 2^\ell$ block matrix with blocks of size $(n/2^\ell) \times (n/2^\ell)$:

$$\mathbf{A}^{(\ell)} = \begin{bmatrix} \mathbf{A}^{(\ell)}_{1,1} & \mathbf{A}^{(\ell)}_{1,2} & \cdots & \mathbf{A}^{(\ell)}_{1,d} \\ \mathbf{A}^{(\ell)}_{2,1} & \mathbf{A}_{2,2} & \cdots & \mathbf{A}^{(\ell)}_{2,2^\ell} \\ \vdots & \vdots & & \vdots \\ \mathbf{A}^{(\ell)}_{d,1} & \mathbf{A}^{(\ell)}_{2^\ell,2} & \cdots & \mathbf{A}^{(\ell)}_{2^\ell,2^\ell} \end{bmatrix}. \tag{4.5}$$

We will use the Generalized Nyström Method to simultaneously approximate the relevant blocks

$$\mathbf{A}^{(\ell)}_{2,1},\ \mathbf{A}^{(\ell)}_{1,2},\ \mathbf{A}^{(\ell)}_{4,3},\ \mathbf{A}^{(\ell)}_{3,4},\ \ldots,\ \mathbf{A}^{(\ell)}_{2^\ell,2^\ell-1}, \mathbf{A}^{(\ell)}_{2^\ell-1,2^\ell}, \tag{4.6}$$

with low-rank matrices $\mathbf{H}^{(\ell)}_1 \approx \mathbf{A}^{(\ell)}_{2,1}$, $\mathbf{H}^{(\ell)}_2 \approx \mathbf{A}^{(\ell)}_{1,2}$, ..., $\mathbf{H}^{(\ell)}_d \approx \mathbf{A}^{(\ell)}_{2^\ell-1,2^\ell}$. The low-rank matrices obtained will then be placed in the block tridiagonal matrix

$$\mathbf{H}^{(\ell)} = \text{block-tridiag}\left(\begin{matrix} \mathbf{H}^{(\ell)}_2 & \mathbf{0} & \mathbf{H}^{(\ell)}_4 & \mathbf{0} & \cdots & \mathbf{H}^{(\ell)}_{2^\ell} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots \\ \mathbf{H}^{(\ell)}_1 & \mathbf{0} & \mathbf{H}^{(\ell)}_3 & \mathbf{0} & \cdots & \mathbf{H}^{(\ell)}_{2^\ell-1} \end{matrix} \right), \tag{4.7}$$

and the algorithm proceeds to the next level.

Note that the sketching matrices used by the peeling algorithm to recover the matrices in (4.6) depends on the parity of the column index j: when the first index is even the second index is one smaller and when the first index is odd the second index is one larger. To handle the two cases simultaneously in our analysis, we will introduce the following notation. Fix a value $j \in \{1, \ldots, d\}$. We will use "$\pm$" and "$\mp$" to indicate addition or subtraction depending on the parity of j:

$$\pm := \begin{cases} + & j \text{ odd} \\ - & j \text{ even} \end{cases}, \qquad \mp := \begin{cases} - & j \text{ odd} \\ + & j \text{ even} \end{cases}. \tag{4.8}$$

For instance example $j \pm 1$ means $j+1$ if j is odd and $j-1$ if j is even, and similarly $\xi_j^{\pm}$ means ξ_j^{+} if j is odd and ξ_j^{-} if j is even.

The diagonal blocks $\mathbf{A}_{j,j}^{(\ell)}$ may have large norm, and the sketching matrices used by the peeling algorithm are constructed explicitly to avoid touching these diagonal blocks. Assuming the algorithm has not accumulated much error, then besides the diagonal blocks and sub/super diagonal blocks we aim to recover, the other blocks in (4.6) will be on the scale of the best possible error. In particular, as described in Section 2, if $\mathbf{A}$ is exactly HODLR, then these blocks are all zero.

At the final level $\ell = L$, we must also approximate the on-diagonal blocks. Let $\widehat{\mathbf{A}}^{(L)}$ be the matrix after removing our approximation to the off-diagonal blocks of $\mathbf{A}^{(L)}$; i.e. $\widehat{\mathbf{A}}^{(L)} := \mathbf{A} - (\mathbf{H}^{(L)} + \cdots + \mathbf{H}^{(1)})$. Partition $\widehat{\mathbf{A}}^{(L)}$ into a $2^L \times 2^L$ block matrix with blocks of size $(n/2^L) \times (n/2^L)$:

$$\widehat{\mathbf{A}}^{(L)} = \begin{bmatrix} \widehat{\mathbf{A}}_{1,1}^{(L)} & \widehat{\mathbf{A}}_{1,2}^{(L)} & \cdots & \widehat{\mathbf{A}}_{1,2^L}^{(L)} \\ \widehat{\mathbf{A}}_{2,1}^{(L)} & \widehat{\mathbf{A}}_{2,2} & \cdots & \widehat{\mathbf{A}}_{2,2^L}^{(L)} \\ \vdots & \vdots & & \vdots \\ \widehat{\mathbf{A}}_{2^L,1}^{(L)} & \widehat{\mathbf{A}}_{2^L,2}^{(L)} & \cdots & \widehat{\mathbf{A}}_{2^L,2^L}^{(L)} \end{bmatrix}. \tag{4.9}$$

We note that $\widehat{\mathbf{A}}_{i,j}^{(L)} = \mathbf{A}_{i,j}^{(L)}$ for all $i \neq j \pm 1$; indeed, the approximation $\mathbf{H}^{(L)}$ to the off-diagonal low-rank blocks at level L only updates the blocks $\mathbf{A}_{j\pm1,j}^{(L)}$. We will approximate the blocks $\widehat{\mathbf{A}}_{j,j}^{(L)}$ with matrices $\widehat{\mathbf{H}}_j$ which we place in the matrix

$$\widehat{\mathbf{H}} = \text{block-diag}\begin{pmatrix} \widehat{\mathbf{H}}_1 & \widehat{\mathbf{H}}_2 & \widehat{\mathbf{H}}_3 & \cdots & \widehat{\mathbf{H}}_{2^L} \end{pmatrix}. \tag{4.10}$$

We now have the notation needed to fully describe Algorithm 3. In Sections 4.3 to 4.5 we describe in more detail the main conceptual pieces of the algorithm and introduce additional notation used in our analysis.

4.3 Range approximation We first describe how to obtain an approximate range $\mathbf{Q}_j^{(\ell)}$ for each $\mathbf{A}_{j\pm1,j}$ at level ℓ. Towards this end, let

$$\boldsymbol{\xi}^{+}, \boldsymbol{\xi}^{-}, \boldsymbol{\Omega}^{+}, \boldsymbol{\Omega}^{-} \sim \text{RandPerfGaussian}(n, d, s_{\text{R}}, s_{\text{R}})$$

as defined in Definition 4.5. We will access $\mathbf{A}$ via the sketches $\mathbf{A}^{(\ell)}\boldsymbol{\Omega}^{+}$ and $\mathbf{A}^{(\ell)}\boldsymbol{\Omega}^{-}$, each of which can be computed using $s_{\text{R}} t_{\text{R}}$ matvecs with $\mathbf{A}$.

Fix j and let $\rho = \rho(j)$ be the (unique) index such that $\xi_{j,\rho}^{\pm} = 1$. We will try to recover a good approximation to the range of $\mathbf{A}_{j\pm1,j}^{(\ell)}$ using the information from the $(j \pm 1, \rho)$ block of the sketch $\mathbf{A}\boldsymbol{\Omega}^{\pm}$. This is illustrated in Figure 3. In particular, we will use the sketch

$$\mathbf{Y}_j^{(\ell)} := \mathbf{A}_{j\pm1,:}^{(\ell)} \boldsymbol{\Omega}_{:,\rho}^{\pm} = \sum_{i=1}^{d} \mathbf{A}_{j\pm1,i}^{(\ell)} (\xi_{i,\rho}^{\pm} \boldsymbol{\Omega}_i).$$

It will be useful to write

$$\mathbf{Y}_j^{(\ell)} = \mathbf{A}_{j\pm1,j}^{(\ell)} \boldsymbol{\Omega}_j + \mathbf{E}_j^{(\ell)}, \qquad \mathbf{E}_j^{(\ell)} := \sum_{i \neq j} \xi_{i,\rho}^{\pm} \mathbf{A}_{j\pm1,i}^{(\ell)} \boldsymbol{\Omega}_i. \tag{4.11}$$

We will then set

$$\mathbf{Q}_j^{(\ell)} = \text{orth}(\mathbf{Y}_j^{(\ell)}) = \text{orth}(\mathbf{A}_{j\pm1,j}^{(\ell)} \boldsymbol{\Omega}_j + \mathbf{E}_j^{(\ell)}), \tag{4.12}$$

which will still be an approximate top subspace of $\mathbf{A}_{j\pm1,j}^{(\ell)}$, provided the noise term $\mathbf{E}_j^{(\ell)}$ is not too large. In particular, note that if $\mathbf{A}$ is exactly HODLR, then $\mathbf{E}_j^{(\ell)}$ is zero. This is because in this setting, $\mathbf{A}_{j\pm1,i}^{(\ell)} = \mathbf{0}$ except when $i = j$ or $i = j \pm 1$, and $\xi_{j\pm1,\rho}^{\pm} = 0$ so that there is no contribution from the on-diagonal block $\mathbf{A}_{j\pm1,j\pm1}^{(\ell)}$.

Algorithm 3 Generalized Nyström Method Peeling algorithm for HODLR approximation

procedure GeneralizedNyströmPeeling($\mathbf{A}, k, s_{\text{R}}, t_{\text{R}}, s_{\text{L}}, t_{\text{L}}$)
  Set $L = \lceil \log_2(n/k) \rceil$ ▷ Final level blocks of size at most k
  for $\ell = 1, 2, \ldots, L$ **do**
    Allocate and partition $\mathbf{H}^{(\ell)}$ as in (4.7) ▷ blocks of size $n/2^\ell \times n/2^\ell$
    $\boldsymbol{\xi}^+, \boldsymbol{\xi}^-, \boldsymbol{\Omega}^+, \boldsymbol{\Omega}^- \sim \text{RandPerfGaussian}(n, 2^\ell, s_{\text{R}}, t_{\text{R}})$ ▷ as in Definition 4.5
    $\boldsymbol{\zeta}^+, \boldsymbol{\zeta}^-, \boldsymbol{\Psi}^+, \boldsymbol{\Psi}^- \sim \text{RandPerfGaussian}(n, 2^\ell, s_{\text{L}}, t_{\text{L}})$ ▷ as in Definition 4.5
    Compute $\mathbf{A}^{(\ell)}\boldsymbol{\Omega}^\pm$ ▷ $2s_{\text{R}}t_{\text{R}}$ matvecs with $\mathbf{A}$
    Compute $(\boldsymbol{\Psi}^\pm)^\mathsf{T}\mathbf{A}^{(\ell)}$ ▷ $2s_{\text{L}}t_{\text{L}}$ matvecs with $\mathbf{A}^\mathsf{T}$
    for $j = 1, 2, \ldots, 2^\ell$ **do**
      Set ρ such that $\xi^\pm_{j,\rho} = 1$
      Set σ such that $\zeta^\mp_{j\pm1,\sigma} = 1$
      $\mathbf{Q}_j^{(\ell)} = \text{orth}(\mathbf{Y}_j^{(\ell)})$ ▷ $\mathbf{Y}_j^{(\ell)}$ is $(j \pm 1, \rho)$ block of $\mathbf{A}^{(\ell)}\boldsymbol{\Omega}^\pm$
      $\mathbf{X}_j^{(\ell)} = \big((\boldsymbol{\Psi}^\mp_{j\pm1})^\mathsf{T}\mathbf{Q}_j^{(\ell)}\big)^\dagger \mathbf{Z}_j^{(\ell)}$ ▷ $\mathbf{Z}_j^{(\ell)}$ is (σ, j) block of $(\boldsymbol{\Psi}^\pm)^\mathsf{T}\mathbf{A}$
      $\mathbf{H}_j^{(\ell)} = \mathbf{Q}_j^{(\ell)}[\![\mathbf{X}_j^{(\ell)}]\!]_k$ ▷ $\mathbf{H}_j^{(\ell)}$ is $(j \pm 1, j)$-th block of $\mathbf{H}^{(\ell)}$
  Allocate and partition $\widehat{\mathbf{H}}$ as in (4.10) ▷ blocks of size $n/2^L \times n/2^L$
  $\widehat{\boldsymbol{\zeta}}^+, \widehat{\boldsymbol{\zeta}}^-, \widehat{\boldsymbol{\Psi}}^+, \widehat{\boldsymbol{\Psi}}^- \sim \text{RandPerfGaussian}(n, 2^L, s_{\text{L}}, t_{\text{L}})$ ▷ as in Definition 4.5
  $\widehat{\boldsymbol{\zeta}} = \widehat{\boldsymbol{\zeta}}^+ + \widehat{\boldsymbol{\zeta}}^-$, $\widehat{\boldsymbol{\Psi}} = \widehat{\boldsymbol{\Psi}}^+ + \widehat{\boldsymbol{\Psi}}^-$
  Compute $\widehat{\boldsymbol{\Psi}}^\mathsf{T}\widehat{\mathbf{A}}^{(L)}$ ▷ $s_{\text{L}}t_{\text{L}}$ matvecs with $\mathbf{A}^\mathsf{T}$
  for $j = 1, 2, \ldots, 2^L$ **do**
    Set σ such that $\widehat{\zeta}_{j,\sigma} = 1$
    $\widehat{\mathbf{X}}_j = \big((\boldsymbol{\Psi}_j)^\mathsf{T}\big)^\dagger \widehat{\mathbf{Z}}_j$ ▷ $\widehat{\mathbf{Z}}_j$ is (σ, j) block of $(\widehat{\boldsymbol{\Psi}})^\mathsf{T}\widehat{\mathbf{A}}^{(L)}$
    $\widehat{\mathbf{H}}_j = \widehat{\mathbf{X}}_j$ ▷ $\widehat{\mathbf{H}}_j$ is (j, j)-th block of $\widehat{\mathbf{H}}$
  return $\widetilde{\mathbf{A}} = \mathbf{H}^{(1)} + \cdots + \mathbf{H}^{(L)} + \widehat{\mathbf{H}}$

4.4 Low-rank approximation from given subspace We now describe how to obtain the low-rank approximations $\mathbf{H}_j^{(\ell)}$ to each $\mathbf{A}^{(\ell)}_{j\pm1,j}$ at level ℓ, given the approximate left subspaces $\mathbf{Q}_j^{(\ell)}$ computed by the procedure described in Section 4.3. Let

$$\boldsymbol{\zeta}^+, \boldsymbol{\zeta}^-, \boldsymbol{\Psi}^+, \boldsymbol{\Psi}^- \sim \text{RandPerfGaussian}(n, d, s_{\text{L}}, t_{\text{L}})$$

as defined in Definition 4.5. We will access $\mathbf{A}$ via the sketches $(\mathbf{A}^{(\ell)})^\mathsf{T}\boldsymbol{\Psi}^+$ and $(\mathbf{A}^{(\ell)})^\mathsf{T}\boldsymbol{\Psi}^-$, each of which can be computed using $s_{\text{L}}t_{\text{L}}$ matvecs with $\mathbf{A}^\mathsf{T}$.

Fix j and let $\sigma = \sigma(j)$ be the (unique) index such that $\zeta^\mp_{j\pm1,\sigma} = 1$. We will try to recover a low-rank approximation of $\mathbf{A}^{(\ell)}_{j\pm1,j}$ whose column space is $\mathbf{Q}_j^{(\ell)}$ from the (σ, j) block of the sketch $(\boldsymbol{\Psi}^\mp)^\mathsf{T}\mathbf{A}^{(\ell)}$. This is illustrated in Figure 3. In particular, we will use the sketch

$$\mathbf{Z}_j^{(\ell)} := (\boldsymbol{\Psi}^\mp_{:,\sigma})^\mathsf{T}\mathbf{A}^{(\ell)}_{:,j} = \sum_{i=1}^{d} (\zeta^\mp_{i,\sigma}\boldsymbol{\Psi}_i)^\mathsf{T}\mathbf{A}^{(\ell)}_{i,j}.$$

Similar to above, we therefore have

$$\mathbf{Z}_j^{(\ell)} = (\boldsymbol{\Psi}^\mp_{j\pm1})^\mathsf{T}\mathbf{A}^{(\ell)}_{j\pm1,j} + \mathbf{F}_j^{(\ell)}, \qquad \mathbf{F}_j^{(\ell)} := \sum_{i \neq j\pm1} \zeta^\mp_{i,\sigma}(\boldsymbol{\Psi}_i)^\mathsf{T}\mathbf{A}^{(\ell)}_{i,j}. \tag{4.13}$$

Now, we obtain obtain the right factors by

$$\mathbf{X}_j^{(\ell)} := \big((\boldsymbol{\Psi}^\mp_{j\pm1})^\mathsf{T}\mathbf{Q}_j^{(\ell)}\big)^\dagger \mathbf{Z}_j^{(\ell)} = \operatorname*{argmin}_{\mathbf{X}} \|(\boldsymbol{\Psi}_{j\pm1})^\mathsf{T}\mathbf{Q}_j^{(\ell)}\mathbf{X} - \mathbf{Z}_j^{(\ell)}\|_\mathsf{F}. \tag{4.14}$$

Finally, we obtain an approximation $\mathbf{H}_j^{(\ell)} = \mathbf{Q}_j^{(\ell)} [\![\mathbf{X}_j^{(\ell)}]\!]_k$ to $\mathbf{A}_{j\pm1,j}^{(\ell)}$.

If $\mathbf{F}_j^{(\ell)}$ is small, then $\mathbf{Q}_j^{(\ell)}\mathbf{X}_j^{(\ell)}$ will be nearly the best approximation to $\mathbf{A}_{j\pm1,j}^{(\ell)}$ with range equal to the column span of $\mathbf{Q}_j^{(\ell)}$. Similar to before, if $\mathbf{A}$ is exactly HODLR, the noise term $\mathbf{F}_j^{(\ell)}$ is zero.

4.5 Recovering the diagonal blocks The strategy we use to recover the diagonal blocks is similar to the off-diagonal blocks. However, since the blocks have at most k columns, there is no need to obtain the left subspace.[11] We let

$$\widehat{\boldsymbol{\zeta}}^{+}, \widehat{\boldsymbol{\zeta}}^{-}, \widehat{\boldsymbol{\Psi}}^{+}, \widehat{\boldsymbol{\Psi}}^{-} \sim \text{RandPerfGaussian}(n, d, s_{\text{L}}, t_{\text{L}})$$

as defined in Definition 4.5. Next, define

$$\widehat{\boldsymbol{\zeta}} = \widehat{\boldsymbol{\zeta}}^{+} + \widehat{\boldsymbol{\zeta}}^{-}, \qquad \widehat{\boldsymbol{\Psi}} = \widehat{\boldsymbol{\Psi}}^{+} + \widehat{\boldsymbol{\Psi}}^{-}$$

Fix j and let $\sigma = \sigma(j)$ be the (unique) index such that $\widehat{\zeta}_{j,\sigma} = 1$. We will try to recover an approximation to $\widehat{\mathbf{A}}_{j,j}^{(L)}$ using the (σ, j) block of the sketch $(\mathbf{A}^{(L+1)})^{\mathsf{T}}\widehat{\boldsymbol{\Psi}}$. In particular, we will use the sketch

$$\widehat{\mathbf{Z}}_j := (\widehat{\boldsymbol{\Psi}}_{:,\sigma})^{\mathsf{T}} \widehat{\mathbf{A}}_{:,j}^{(L)} = \sum_{i=1}^{d} \xi_{i,\sigma} (\boldsymbol{\Psi}_i)^{\mathsf{T}} \widehat{\mathbf{A}}_{i,j}^{(L)}$$

Similar to above, we therefore have

$$\widehat{\mathbf{Z}}_j = (\boldsymbol{\Psi}_j)^{\mathsf{T}} \widehat{\mathbf{A}}_{j,j}^{(L)} + \widehat{\mathbf{F}}_j, \qquad \widehat{\mathbf{F}}_j := \sum_{i \neq j} \xi_{i,\sigma} (\boldsymbol{\Psi}_i)^{\mathsf{T}} \widehat{\mathbf{A}}_{i,j}^{(L)}. \tag{4.15}$$

We obtain the diagonal blocks by

$$\widehat{\mathbf{X}}_j := \left((\boldsymbol{\Psi}_j)^{\mathsf{T}}\right)^{\dagger} \widehat{\mathbf{Z}}_j = \operatorname*{argmin}_{\mathbf{H}} \|(\boldsymbol{\Psi}_j)^{\mathsf{T}}\mathbf{H} - \widehat{\mathbf{Z}}_j\|_{\mathsf{F}}. \tag{4.16}$$

Finally, we obtain an approximation $\widehat{\mathbf{H}}_j = \widehat{\mathbf{X}}_j$ to $\widehat{\mathbf{A}}_{j,j}^{(L)}$.

5 Analysis

We are now prepared to prove our main accuracy guarantee for Algorithm 3, stated as Theorem 5.2. The simplified Theorem 1.1 stated in Section 1.2 will follow as a direct corollary.

In Section 5.1 we first prove Theorem 3.1, a deterministic perturbation bound for RSVD and Generalized Nyström implemented with inexact matvec queries (see discussion in Section 3.1). We then use this bound to prove Theorem 3.2, is a probabilistic error bound that holds for Generalized Nyström implemented with random Gaussian sketches. In Section 5.2 we use this bound to prove our main approximation guarantees. In particular, our analysis applies Theorem 3.2 to each off-diagonal block at each level. Assuming we have obtained near-optimal low-rank approximations to the off-diagonal blocks at the previous levels, we show that we can obtain near-optimal low-rank approximations to the off-diagonal blocks at the current level.

5.1 Perturbation bound for the RSVD We begin with proving Theorem 3.1, which we restate below.

THEOREM 3.1. *Let* $\mathbf{B} \in \mathbb{R}^{m_1 \times m_2}$, $\boldsymbol{\Omega} \in \mathbb{R}^{m_2 \times s}$, $\mathbf{E}_1 \in \mathbb{R}^{m_1 \times s}$, *and* $\mathbf{E}_2 \in \mathbb{R}^{s \times m_2}$, *and let* $\mathbf{Q} = \text{orth}(\mathbf{B}\boldsymbol{\Omega} + \mathbf{E}_1)$. *Write* $\mathbf{B}$ *in its singular value decomposition as*

$$\mathbf{B} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathsf{T}} = \begin{bmatrix}\mathbf{U}_{\text{top}} & \mathbf{U}_{\text{bot}}\end{bmatrix} \begin{bmatrix}\boldsymbol{\Sigma}_{\text{top}} & \\ & \boldsymbol{\Sigma}_{\text{bot}}\end{bmatrix} \begin{bmatrix}\mathbf{V}_{\text{top}}^{\mathsf{T}} \\ \mathbf{V}_{\text{bot}}^{\mathsf{T}}\end{bmatrix},$$

where $\mathbf{U}$ *and* $\mathbf{V}$ *are orthonormal and* $\boldsymbol{\Sigma}$ *is diagonal, and the "top" blocks represent the top* k *columns; i.e.,* $\mathbf{U}_{\text{top}} \in \mathbb{R}^{m_1 \times k}, \mathbf{V}_{\text{top}} \in \mathbb{R}^{m_2 \times k}$, *and* $\boldsymbol{\Sigma}_{\text{top}} \in \mathbb{R}^{k \times k}$. *Define also*

$$\boldsymbol{\Omega}_{\text{top}} = \mathbf{V}_{\text{top}}^{\mathsf{T}}\boldsymbol{\Omega}, \quad \boldsymbol{\Omega}_{\text{bot}} = \mathbf{V}_{\text{bot}}^{\mathsf{T}}\boldsymbol{\Omega}.$$

[11] This is also true for the off-diagonal blocks in levels where $n/2^{\ell} \leq s_{\text{R}}$. In such cases $\mathbf{Q}_j^{(\ell)}$ will be (square) orthogonal, so it could be set to the identity a priori. We do not separate this case to simplify the notation in our analysis.

Then, assuming $\operatorname{rank}(\mathbf{\Omega}_{\text{top}}) = k$,

$$\|\mathbf{B} - \mathbf{Q}[\![\mathbf{Q}^\mathsf{T}\mathbf{B} + \mathbf{E}_2]\!]_k\|_\mathsf{F} \leq \|\mathbf{E}_1\mathbf{\Omega}_{\text{top}}^\dagger\|_\mathsf{F} + 2\|\mathbf{E}_2\|_\mathsf{F} + \left(\|\mathbf{\Sigma}_{\text{bot}}\|_\mathsf{F}^2 + \|\mathbf{\Sigma}_{\text{bot}}\mathbf{\Omega}_{\text{bot}}\mathbf{\Omega}_{\text{top}}^\dagger\|_\mathsf{F}^2\right)^{1/2}.$$

Proof. Consider the matrix $\mathbf{C} := (\mathbf{I} - \mathbf{Q}\mathbf{Q}^\mathsf{T})\mathbf{B} + \mathbf{Q}(\mathbf{Q}^\mathsf{T}\mathbf{B} + \mathbf{E}_2)$. The best rank-$k$ Frobenius-norm approximation to $\mathbf{C}$ with range contained in range$(\mathbf{Q})$ is $\mathbf{Q}[\![\mathbf{Q}^\mathsf{T}\mathbf{C}]\!]_k$ [Gu15, Theorem 3.5]. Using this, the triangle inequality, and the relations $\|\mathbf{B} - \mathbf{C}\|_\mathsf{F} = \|\mathbf{Q}\mathbf{E}_2\|_\mathsf{F} = \|\mathbf{E}_2\|_\mathsf{F}$ and $\mathbf{Q}^\mathsf{T}\mathbf{C} = \mathbf{Q}^\mathsf{T}\mathbf{B} + \mathbf{E}_2$ we obtain

$$\begin{aligned}
\|\mathbf{B} - \mathbf{Q}[\![\mathbf{Q}^\mathsf{T}\mathbf{B} + \mathbf{E}_2]\!]_k\|_\mathsf{F} &= \|\mathbf{B} - \mathbf{Q}[\![\mathbf{Q}^\mathsf{T}\mathbf{C}]\!]_k\|_\mathsf{F} \\
&\leq \|\mathbf{B} - \mathbf{C}\|_\mathsf{F} + \|\mathbf{C} - \mathbf{Q}[\![\mathbf{Q}^\mathsf{T}\mathbf{C}]\!]_k\|_\mathsf{F} \\
&\leq \|\mathbf{B} - \mathbf{C}\|_\mathsf{F} + \|\mathbf{C} - \mathbf{Q}[\![\mathbf{Q}^\mathsf{T}\mathbf{B}]\!]_k\|_\mathsf{F} \\
&\leq 2\|\mathbf{B} - \mathbf{C}\|_\mathsf{F} + \|\mathbf{B} - \mathbf{Q}[\![\mathbf{Q}^\mathsf{T}\mathbf{B}]\!]_k\|_\mathsf{F} \\
&= 2\|\mathbf{E}_2\|_\mathsf{F} + \|\mathbf{B} - \mathbf{Q}[\![\mathbf{Q}^\mathsf{T}\mathbf{B}]\!]_k\|_\mathsf{F}. \qquad (5.17)
\end{aligned}$$

Now let $\mathbf{M} := \mathbf{\Omega}_{\text{top}}^\dagger\mathbf{V}_{\text{top}}^\mathsf{T}$. Note that $\operatorname{rank}((\mathbf{B\Omega}+\mathbf{E}_1)\mathbf{M}) \leq k$ and let $\mathbf{P} \in \mathbb{R}^{m_1 \times m_1}$ denote the orthogonal projector onto range$((\mathbf{B\Omega} + \mathbf{E}_1)\mathbf{M})$. Again, using [Gu15, Theorem 3.5] and the triangle inequality, we obtain

$$\begin{aligned}
\|\mathbf{B} - \mathbf{Q}[\![\mathbf{Q}^\mathsf{T}\mathbf{B}]\!]_k\|_\mathsf{F} &\leq \|(\mathbf{I} - \mathbf{P})\mathbf{B}\|_\mathsf{F} \\
&\leq \|(\mathbf{I} - \mathbf{P})\mathbf{B\Omega M}\|_\mathsf{F} + \|(\mathbf{I} - \mathbf{P})\mathbf{B}(\mathbf{I} - \mathbf{\Omega M})\|_\mathsf{F} \\
&\leq \|(\mathbf{I} - \mathbf{P})\mathbf{B\Omega M}\|_\mathsf{F} + \|\mathbf{B}(\mathbf{I} - \mathbf{\Omega M})\|_\mathsf{F}. \qquad (5.18)
\end{aligned}$$

Now, following [Con23, Chapter 5] we can bound each term (5.18). First note that

$$(\mathbf{I} - \mathbf{P})\mathbf{B\Omega M} = (\mathbf{I} - \mathbf{P})(\mathbf{B\Omega} + \mathbf{E}_1)\mathbf{M} - (\mathbf{I} - \mathbf{P})\mathbf{E}_1\mathbf{M} = -(\mathbf{I} - \mathbf{P})\mathbf{E}_1\mathbf{M}.$$

Hence,

$$\|(\mathbf{I} - \mathbf{P})\mathbf{B\Omega M}\|_\mathsf{F} \leq \|\mathbf{E}_1\mathbf{M}\|_\mathsf{F} = \|\mathbf{E}_1\mathbf{\Omega}_{\text{top}}^\dagger\|_\mathsf{F}. \qquad (5.19)$$

For the second term, observe that since we have assumed $\operatorname{rank}(\mathbf{\Omega}_{\text{top}}) = k$, $\mathbf{V}_{\text{top}}\mathbf{V}_{\text{top}}^\mathsf{T}\mathbf{\Omega M} = \mathbf{V}_{\text{top}}\mathbf{V}_{\text{top}}^\mathsf{T}\mathbf{\Omega}(\mathbf{V}_{\text{top}}^\mathsf{T}\mathbf{\Omega})^\dagger\mathbf{V}_{\text{top}}^\mathsf{T} = \mathbf{V}_{\text{top}}\mathbf{V}_{\text{top}}^\mathsf{T}$. We thus have:

$$\mathbf{B}(\mathbf{I} - \mathbf{\Omega M}) = \mathbf{B}\mathbf{V}_{\text{bot}}\mathbf{V}_{\text{bot}}^\mathsf{T}(\mathbf{I} - \mathbf{\Omega M}) = \mathbf{B}\mathbf{V}_{\text{bot}}\mathbf{V}_{\text{bot}}^\mathsf{T} - \mathbf{B}\mathbf{V}_{\text{bot}}\mathbf{V}_{\text{bot}}^\mathsf{T}\mathbf{\Omega M}.$$

By the Pythagorean theorem then we have

$$\begin{aligned}
\|\mathbf{B}(\mathbf{I} - \mathbf{\Omega M})\|_\mathsf{F} &= \left(\|\mathbf{B}\mathbf{V}_{\text{bot}}\mathbf{V}_{\text{bot}}^\mathsf{T}\|_\mathsf{F}^2 + \|\mathbf{B}\mathbf{V}_{\text{bot}}\mathbf{V}_{\text{bot}}^\mathsf{T}\mathbf{\Omega M}\|_\mathsf{F}^2\right)^{1/2} \\
&= \left(\|\mathbf{\Sigma}_{\text{bot}}\|_\mathsf{F}^2 + \|\mathbf{\Sigma}_{\text{bot}}\mathbf{\Omega}_{\text{bot}}\mathbf{\Omega}_{\text{top}}^\dagger\|_\mathsf{F}^2\right)^{1/2}. \qquad (5.20)
\end{aligned}$$

Combining (5.17) and (5.20) yields the desired inequality. ∎

5.1.1 Probabilistic bounds To apply Theorem 3.1 to a setting with Gaussian sketching matrices, we make use of the following well-known fact about the Frobenius norm of Gaussian matrices and their pseudoinverses, proofs of which can be found throughout the literature; see for instance [HMT11, Proposition A.5].

THEOREM 5.1. *Let* $\mathbf{X}$ *be* $u \times v$ *and* $\mathbf{G} \sim \text{Gaussian}(v, q)$ *and* $\mathbf{H} \sim \text{Gaussian}(p, q)$ *independently. Then, if* $q > p+1$,

$$\mathbb{E}\left[\|\mathbf{X}\mathbf{G}\mathbf{H}^\dagger\|_\mathsf{F}^2\right] = \|\mathbf{X}\|_\mathsf{F}^2 \cdot \mathbb{E}\left[\|\mathbf{H}^\dagger\|_\mathsf{F}^2\right] = \frac{p}{q-p-1}\|\mathbf{X}\|_\mathsf{F}^2.$$

We proceed with proving the following lemma, which relates to the error of the approximate projection step of the Generalized Nyström Method, when implemented with approximate matvecs as they arise in the peeling algorithm – see discussion in Section 3.2.

LEMMA 5.1. *Let* $\mathbf{B} \in \mathbb{R}^{m_1 \times m_2}$, $\mathbf{Q} \in \mathbb{R}^{m_1 \times s_R}$ *with orthonormal columns, and* $\mathbf{N} \in \mathbb{R}^{q \times m_2}$ *be fixed matrices and* $\mathbf{\Psi} \sim \text{Gaussian}(m_1, s_L)$ *and* $\widetilde{\mathbf{\Psi}} \sim \text{Gaussian}(q, s_L)$ *be independent. Then, provided* $s_L > s_R + 1$,

$$\mathbb{E}\Big[\|\mathbf{Q}^\mathsf{T}\mathbf{B} - (\mathbf{\Psi}^\mathsf{T}\mathbf{Q})^\dagger(\mathbf{\Psi}^\mathsf{T}\mathbf{B} + \widetilde{\mathbf{\Psi}}^\mathsf{T}\mathbf{N})\|_\mathsf{F}^2\Big] = \frac{s_R}{s_L - s_R - 1}\Big(\|\mathbf{B} - \mathbf{Q}\mathbf{Q}^\mathsf{T}\mathbf{B}\|_\mathsf{F}^2 + \|\mathbf{N}\|_\mathsf{F}^2\Big).$$

Proof. Extend $\mathbf{Q}$ to a square orthogonal matrix $[\mathbf{Q}\ \widetilde{\mathbf{Q}}]$. Write $\mathbf{\Psi}_1 = \mathbf{\Psi}^\mathsf{T}\mathbf{Q}$ and $\mathbf{\Psi}_2 = \mathbf{\Psi}^\mathsf{T}\widetilde{\mathbf{Q}}$. Since $\mathbf{Q}$ and $\widetilde{\mathbf{Q}}$ are orthogonal, $\mathbf{\Psi}_1 \sim \text{Gaussian}(s_L, s_R)$ and $\mathbf{\Psi}_2 \sim \text{Gaussian}(s_L, m_1 - s_R)$ are independent.

Further, since $[\mathbf{Q}\ \widetilde{\mathbf{Q}}]$ is orthogonal, $\mathbf{Q}\mathbf{Q}^\mathsf{T} + \widetilde{\mathbf{Q}}\widetilde{\mathbf{Q}}^\mathsf{T}$ is the identity, and

$$\mathbf{\Psi}^\mathsf{T}\mathbf{B} = \mathbf{\Psi}^\mathsf{T}(\mathbf{Q}\mathbf{Q}^\mathsf{T} + \widetilde{\mathbf{Q}}\widetilde{\mathbf{Q}}^\mathsf{T})\mathbf{B} = \mathbf{\Psi}_1\mathbf{Q}^\mathsf{T}\mathbf{B} + \mathbf{\Psi}_2\widetilde{\mathbf{Q}}^\mathsf{T}\mathbf{B}.$$

Therefore, recalling our definition of $\mathbf{\Psi}_1 = \mathbf{\Psi}^\mathsf{T}\mathbf{Q}$,

$$\mathbf{\Psi}_1^\dagger(\mathbf{\Psi}^\mathsf{T}\mathbf{B} + \widetilde{\mathbf{\Psi}}^\mathsf{T}\mathbf{N}) = \mathbf{\Psi}_1^\dagger\mathbf{\Psi}_1\mathbf{Q}^\mathsf{T}\mathbf{B} + \mathbf{\Psi}_1^\dagger\mathbf{\Psi}_2\widetilde{\mathbf{Q}}^\mathsf{T}\mathbf{B} + \mathbf{\Psi}_1^\dagger\widetilde{\mathbf{\Psi}}^\mathsf{T}\mathbf{N}.$$

Since $s_L \geq s_R$, with probability 1, $\mathbf{\Psi}_1^\dagger\mathbf{\Psi}_1 = \mathbf{I}$. Thus, rearranging the above equation, we find that

$$\mathbf{Q}^\mathsf{T}\mathbf{B} - \mathbf{\Psi}_1^\dagger(\mathbf{\Psi}^\mathsf{T}\mathbf{B} + \widetilde{\mathbf{\Psi}}^\mathsf{T}\mathbf{N}) = -\mathbf{\Psi}_1^\dagger\mathbf{\Psi}_2\widetilde{\mathbf{Q}}^\mathsf{T}\mathbf{B} - \mathbf{\Psi}_1^\dagger\widetilde{\mathbf{\Psi}}^\mathsf{T}\mathbf{N}. \tag{5.21}$$

Next, as shown in [PBK24, Lemma A.2(i)], for deterministic matrices $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ and a Gaussian matrix $\mathbf{\Psi}_2$,

$$\mathbb{E}\Big[\|\mathbf{X}\mathbf{\Psi}_2\mathbf{Y} + \mathbf{Z}\|_\mathsf{F}^2\Big] = \|\mathbf{X}\|_\mathsf{F}^2\|\mathbf{Y}\|_\mathsf{F}^2 + \|\mathbf{Z}\|_\mathsf{F}^2. \tag{5.22}$$

Therefore, using that $\mathbf{\Psi}_1, \mathbf{\Psi}_2$ and $\widehat{\mathbf{\Psi}}$ are independent and applying (5.22) and Theorem 5.1 to bound the righthand side of (5.21), we obtain

$$\begin{aligned}
\mathbb{E}\Big[\|\mathbf{Q}^\mathsf{T}\mathbf{B} - \mathbf{\Psi}_1^\dagger(\mathbf{\Psi}^\mathsf{T}\mathbf{B} + \widetilde{\mathbf{\Psi}}^\mathsf{T}\mathbf{N})\|_\mathsf{F}^2\Big] &= \mathbb{E}\Big[\|\mathbf{\Psi}_1^\dagger\mathbf{\Psi}_2\widetilde{\mathbf{Q}}^\mathsf{T}\mathbf{B} + \mathbf{\Psi}_1^\dagger\widetilde{\mathbf{\Psi}}^\mathsf{T}\mathbf{N}\|_\mathsf{F}^2\Big] \\
&= \mathbb{E}\Big[\mathbb{E}\Big[\|\mathbf{\Psi}_1^\dagger\mathbf{\Psi}_2\widetilde{\mathbf{Q}}^\mathsf{T}\mathbf{B} + \mathbf{\Psi}_1^\dagger\widetilde{\mathbf{\Psi}}^\mathsf{T}\mathbf{N}\|_\mathsf{F}^2\Big|\mathbf{\Psi}_1, \widetilde{\mathbf{\Psi}}\Big]\Big] \\
&= \mathbb{E}\Big[\|\mathbf{\Psi}_1^\dagger\|_\mathsf{F}^2\|\widetilde{\mathbf{Q}}^\mathsf{T}\mathbf{B}\|_\mathsf{F}^2 + \|\mathbf{\Psi}_1^\dagger\widetilde{\mathbf{\Psi}}^\mathsf{T}\mathbf{N}\|_\mathsf{F}^2\Big]. \\
&= \frac{s_R}{s_L - s_R - 1}\|\widetilde{\mathbf{Q}}^\mathsf{T}\mathbf{B}\|_\mathsf{F}^2 + \frac{s_R}{s_L - s_R - 1}\|\mathbf{N}\|_\mathsf{F}^2.
\end{aligned}$$

Noting that $\|\widetilde{\mathbf{Q}}^\mathsf{T}\mathbf{B}\|_\mathsf{F}^2 = \|\mathbf{B} - \mathbf{Q}\mathbf{Q}^\mathsf{T}\mathbf{B}\|_\mathsf{F}^2$ yields the desired result. ∎

Finally, we use Theorem 3.1 and Lemma 5.1 to prove Theorem 3.2.

THEOREM 3.2. *Let* $\mathbf{B} \in \mathbb{R}^{m_1 \times m_2}$, $\mathbf{M} \in \mathbb{R}^{m_1 \times p}$, *and* $\mathbf{N} \in \mathbb{R}^{q \times m_2}$ *be fixed matrices, and let* $\mathbf{\Omega} \sim \text{Gaussian}(m_2, s_R)$, $\widetilde{\mathbf{\Omega}} \sim \text{Gaussian}(p, s_R)$, $\mathbf{\Psi} \sim \text{Gaussian}(m_1, s_L)$, *and* $\widetilde{\mathbf{\Psi}} \sim \text{Gaussian}(q, s_L)$ *be independent. Define*

$$\mathbf{Q} := \text{orth}(\mathbf{B}\mathbf{\Omega} + \mathbf{M}\widetilde{\mathbf{\Omega}}), \qquad \mathbf{X} := (\mathbf{\Psi}^\mathsf{T}\mathbf{Q})^\dagger(\mathbf{\Psi}^\mathsf{T}\mathbf{B} + \widetilde{\mathbf{\Psi}}^\mathsf{T}\mathbf{N}).$$

Then, provided $s_R > 2k + 1$ *and* $s_L > 2s_R + 1$

$$\mathbb{E}\Big[\|\mathbf{B} - \mathbf{Q}[\![\mathbf{X}]\!]_k\|_\mathsf{F}^2\Big] \leq E_1 + E_2 + 2\sqrt{E_1 E_2},$$

where

$$\begin{aligned}
E_1 &:= \left(1 + \frac{k}{s_R - k - 1}\right)\|\mathbf{B} - [\![\mathbf{B}]\!]_k\|_\mathsf{F}^2 \\
E_2 &:= \frac{18k}{s_R - k - 1}\|\mathbf{M}\|_\mathsf{F}^2 + \frac{8s_R}{s_L - s_R - 1}\|\mathbf{N}\|_\mathsf{F}^2 + \frac{32s_R}{s_L - s_R - 1}\|\mathbf{B} - [\![\mathbf{B}]\!]_k\|_\mathsf{F}^2.
\end{aligned}$$

Proof. Let $\mathbf{B}$ have SVD as partitioned in Theorem 3.1 and define $\mathbf{\Omega}_{\text{top}} = \mathbf{V}_{\text{top}}^{\mathsf{T}}\mathbf{\Omega}$ and $\mathbf{\Omega}_{\text{bot}} = \mathbf{V}_{\text{bot}}^{\mathsf{T}}\mathbf{\Omega}$. Note that, by the unitary invariance of Gaussian random matrices $\mathbf{\Omega}_{\text{top}} \sim \text{Gaussian}(k, s_{\text{R}})$ and $\mathbf{\Omega}_{\text{bot}} \sim \text{Gaussian}(m_2 - k, s_{\text{R}})$ independently of one another and $\widetilde{\mathbf{\Omega}}$. Note that $\mathbf{\Omega}_{\text{top}}$ has rank k with probability one, and Theorem 3.1 asserts

$$\|\mathbf{B} - \mathbf{Q}[\![\mathbf{X}]\!]_k\|_{\mathsf{F}} \le \underbrace{\|\mathbf{M}\widetilde{\mathbf{\Omega}}\mathbf{\Omega}_{\text{top}}^{\dagger}\|_{\mathsf{F}}}_{A} + \underbrace{2\|\mathbf{Q}^{\mathsf{T}}\mathbf{B} - \mathbf{X}\|_{\mathsf{F}}}_{B} + \underbrace{\left(\|\mathbf{\Sigma}_{\text{bot}}\|_{\mathsf{F}}^2 + \|\mathbf{\Sigma}_{\text{bot}}\mathbf{\Omega}_{\text{bot}}\mathbf{\Omega}_{\text{top}}^{\dagger}\|_{\mathsf{F}}^2\right)^{1/2}}_{C}. \tag{5.23}$$

We will set $E_1 = \mathbb{E}[C^2]$ and E_2 as an upper bound for for $2\mathbb{E}[A^2 + B^2]$ which is itself an upper bound for $\mathbb{E}[(A+B)^2]$. Then, by the linearity of expectation and Cauchy–Schwarz,

$$\begin{aligned}\mathbb{E}[(A+B+C)^2] &= \mathbb{E}[(A+B)^2] + \mathbb{E}[C^2] + 2\,\mathbb{E}[(A+B)C] \\ &\le \mathbb{E}[(A+B)^2] + \mathbb{E}[C^2] + 2\sqrt{\mathbb{E}[(A+B)^2]\,\mathbb{E}[C^2]} \\ &\le E_2 + E_1 + 2\sqrt{E_2 E_1}.\end{aligned}$$

It now remains to compute $\mathbb{E}[A^2]$, $\mathbb{E}[B^2]$, and $\mathbb{E}[C^2]$.

First, using Theorem 5.1 we have that

$$\mathbb{E}[A^2] = \mathbb{E}\left[\|\mathbf{M}\widetilde{\mathbf{\Omega}}\mathbf{\Omega}_{\text{top}}^{\dagger}\|_{\mathsf{F}}^2\right] = \frac{k}{s_{\text{R}} - k - 1}\|\mathbf{M}\|_{\mathsf{F}}^2; \tag{5.24}$$

$$\mathbb{E}[C^2] = \left(1 + \frac{k}{s_{\text{R}} - k - 1}\right)\|\mathbf{\Sigma}_{\text{bot}}\|_{\mathsf{F}}^2 = \left(1 + \frac{k}{s_{\text{R}} - k - 1}\right)\|\mathbf{B} - [\![\mathbf{B}]\!]_k\|_{\mathsf{F}}^2. \tag{5.25}$$

Since $\mathbf{Q}\mathbf{Q}^{\mathsf{T}}\mathbf{B}$ is the best Frobenius-norm approximation to $\mathbf{B}$ with range contained in range($\mathbf{Q}$), using Theorem 3.1 along with the inequality $(x+y)^2 \le 2x^2 + 2y^2$ gives

$$\|\mathbf{B} - \mathbf{Q}\mathbf{Q}^{\mathsf{T}}\mathbf{B}\|_{\mathsf{F}}^2 \le \|\mathbf{B} - \mathbf{Q}[\![\mathbf{Q}^{\mathsf{T}}\mathbf{B}]\!]_k\|_{\mathsf{F}}^2 \le 2\left(\|\mathbf{M}\mathbf{\Omega}\mathbf{\Omega}_{\text{top}}^{\dagger}\|_{\mathsf{F}}^2 + \|\mathbf{\Sigma}_{\text{bot}}\|_{\mathsf{F}}^2 + \|\mathbf{\Sigma}_{\text{bot}}\mathbf{\Omega}_{\text{bot}}\mathbf{\Omega}_{\text{top}}^{\dagger}\|_{\mathsf{F}}^2\right). \tag{5.26}$$

Applying Lemma 5.1, (5.26), and Theorem 5.1 yields

$$\begin{aligned}\mathbb{E}[B^2] &= \frac{4s_{\text{R}}}{s_{\text{L}} - s_{\text{R}} - 1}\left(\mathbb{E}\left[\|\mathbf{B} - \mathbf{Q}\mathbf{Q}^{\mathsf{T}}\mathbf{B}\|_{\mathsf{F}}^2\right] + \|\mathbf{N}\|_{\mathsf{F}}^2\right) \\ &\le \frac{4s_{\text{R}}}{s_{\text{L}} - s_{\text{R}} - 1}\left(2\,\mathbb{E}\left[\|\mathbf{M}\widetilde{\mathbf{\Omega}}\mathbf{\Omega}_{\text{top}}^{\dagger}\|_{\mathsf{F}}^2\right] + 2\|\mathbf{\Sigma}_{\text{bot}}\|_{\mathsf{F}}^2 + 2\,\mathbb{E}\left[\|\mathbf{\Sigma}_{\text{bot}}\mathbf{\Omega}_{\text{bot}}\mathbf{\Omega}_{\text{top}}^{\dagger}\|_{\mathsf{F}}^2\right] + \|\mathbf{N}\|_{\mathsf{F}}^2\right) \\ &= \frac{4s_{\text{R}}}{s_{\text{L}} - s_{\text{R}} - 1}\left(\frac{2k}{s_{\text{R}} - k - 1}\|\mathbf{M}\|_{\mathsf{F}}^2 + 2\|\mathbf{\Sigma}_{\text{bot}}\|_{\mathsf{F}}^2 + \frac{2k}{s_{\text{R}} - k - 1}\|\mathbf{\Sigma}_{\text{bot}}\|_{\mathsf{F}}^2 + \|\mathbf{N}\|_{\mathsf{F}}^2\right).\end{aligned}$$

Our choices of k, s_{R}, and s_{L} ensure $\frac{k}{s_{\text{R}} - k - 1} \le 1$ and $\frac{s_{\text{R}}}{s_{\text{L}} - s_{\text{R}} - 1} \le 1$. Hence,

$$\mathbb{E}[B^2] \le \frac{8k}{s_{\text{R}} - k - 1}\|\mathbf{M}\|_{\mathsf{F}}^2 + \frac{16s_{\text{R}}}{s_{\text{L}} - s_{\text{R}} - 1}\|\mathbf{\Sigma}_{\text{bot}}\|_{\mathsf{F}}^2 + \frac{4s_{\text{R}}}{s_{\text{L}} - s_{\text{R}} - 1}\|\mathbf{N}\|_{\mathsf{F}}^2. \tag{5.27}$$

Combining (5.24) and (5.27) and noting $\|\mathbf{\Sigma}_{\text{bot}}\|_{\mathsf{F}}^2 = \|\mathbf{B} - [\![\mathbf{B}]\!]_k\|_{\mathsf{F}}^2$ yields

$$2\,\mathbb{E}[A^2 + B^2] \le \frac{18k}{s_{\text{R}} - k - 1}\|\mathbf{M}\|_{\mathsf{F}}^2 + \frac{32s_{\text{R}}}{s_{\text{L}} - s_{\text{R}} - 1}\|\mathbf{\Sigma}_{\text{bot}}\|_{\mathsf{F}}^2 + \frac{8s_{\text{R}}}{s_{\text{L}} - s_{\text{R}} - 1}\|\mathbf{N}\|_{\mathsf{F}}^2 := E_2,$$

as required. ∎

5.2 Analysis of Algorithm 3

We are now prepared to analyze Algorithm 3.

DEFINITION 5.1. *For each $\ell = 1, 2, \ldots, L$, let $\mathcal{F}_\ell$ be the sigma algebra representing the information known to the algorithm at the start of the ℓ-th step.*

We begin by proving the key approximation guarantee for an off-diagonal low-rank block at level ℓ. This shows that, even in the presence of noise, we can obtain a near-optimal low-rank approximation to each of the off-diagonal blocks (and the on-diagonal blocks at the final level $\ell = L$). This can be viewed as a version of Theorem 3.2 adapted to the errors appearing within the peeling algorithm.

LEMMA 5.2. *Fix a rank parameter k and let $\beta \in (0,1)$. Suppose $t_{\mathrm{L}} \geq 1$ and s_{R}, t_{L}, and s_{R} are such that*

$$\frac{k}{s_{\mathrm{R}}-k-1} \leq \frac{\beta}{30}, \qquad \frac{k}{s_{\mathrm{R}}-k-1}\cdot\frac{1}{t_{\mathrm{R}}} \leq \frac{\beta^2}{30^2}, \qquad \frac{s_{\mathrm{R}}}{s_{\mathrm{L}}-s_{\mathrm{R}}-1} \leq \frac{\beta^2}{30^2}.$$

Then for each $\ell = 1, 2, \ldots, L$ the off-diagonal low-rank blocks obtained by Algorithm 3 are nearly optimal in the sense that

$$\mathbb{E}\Big[\|\mathbf{A}^{(\ell)}_{j\pm1,j} - \mathbf{Q}^{(\ell)}_j[\![\mathbf{X}^{(\ell)}_j]\!]_k\|_{\mathsf{F}}^2\Big|\mathcal{F}_\ell\Big] \leq \|\mathbf{A}^{(\ell)}_{j\pm1,j} - [\![\mathbf{A}^{(\ell)}_{j\pm1,j}]\!]_k\|_{\mathsf{F}}^2 + \beta E^{(\ell)}_j.$$

where

$$E^{(\ell)}_j := \max\left\{\|\mathbf{A}^{(\ell)}_{j\pm1,j} - [\![\mathbf{A}^{(\ell)}_{j\pm1,j}]\!]_k\|_{\mathsf{F}}^2,\ \sum_{\substack{i=1\\ i\neq j,j\pm1}}^{2^\ell}\frac{1}{2}\|\mathbf{A}^{(\ell)}_{i,j}\|_{\mathsf{F}}^2,\ \sum_{\substack{i=1\\ i\neq j,j\pm1}}^{2^\ell}\frac{1}{2}\|\mathbf{A}^{(\ell)}_{j\pm1,i}\|_{\mathsf{F}}^2\right\}.$$

Moreover, at the final level $\ell = L$, the on-diagonal blocks obtained by Algorithm 3 are nearly exact in that

$$\mathbb{E}\Big[\|\widehat{\mathbf{A}}^{(L)}_{j,j} - \widehat{\mathbf{X}}_j\|_{\mathsf{F}}^2\Big|\mathcal{F}_{L+1}\Big] \leq \beta\sum_{\substack{i=1\\ i\neq j}}^{2^L}\|\widehat{\mathbf{A}}^{(L)}_{i,j}\|_{\mathsf{F}}^2.$$

Proof. We begin with the first result. Let $d = 2^\ell$ and let $\boldsymbol{\Omega}^{\pm}$ and $\boldsymbol{\Psi}^{\pm}$ be the sketches used at level ℓ of Algorithm 3. Fix j and recall that ρ and σ are the unique indices so that $\xi^{\pm}_{j,\rho} = 1$ and $\zeta^{\mp}_{j\pm1,\sigma} = 1$. Define

$$\mathbf{M} = \begin{bmatrix}\xi^{\pm}_{1,\rho}\mathbf{A}^{(\ell)}_{j\pm1,1} & \cdots & \xi^{\pm}_{j-1,\rho}\mathbf{A}^{(\ell)}_{j\pm1,j-1} & \xi^{\pm}_{j+1,\rho}\mathbf{A}^{(\ell)}_{j\pm1,j+1} & \cdots & \xi^{\pm}_{d,\rho}\mathbf{A}^{(\ell)}_{j\pm1,d},\end{bmatrix}$$
$$\mathbf{N} = \begin{bmatrix}\zeta^{\mp}_{1,\sigma}\mathbf{A}^{(\ell)}_{1,j} & \cdots & \zeta^{\mp}_{j\pm1-1,\sigma}\mathbf{A}^{(\ell)}_{j\pm1-1,j} & \zeta^{\mp}_{j\pm1+1,\sigma}\mathbf{A}^{(\ell)}_{j\pm1+1,j} & \cdots & \zeta^{\mp}_{d,\sigma}\mathbf{A}^{(\ell)}_{d,j},\end{bmatrix}$$

and the random Gaussian matrices

$$\widetilde{\boldsymbol{\Omega}} = \begin{bmatrix}(\boldsymbol{\Omega}_1)^{\mathsf{T}} & \cdots & (\boldsymbol{\Omega}_{j-1})^{\mathsf{T}} & (\boldsymbol{\Omega}_{j+1})^{\mathsf{T}} & \cdots & (\boldsymbol{\Omega}_d)^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}$$
$$\widetilde{\boldsymbol{\Psi}} = \begin{bmatrix}(\boldsymbol{\Psi}_1)^{\mathsf{T}} & \cdots & (\boldsymbol{\Psi}_{j\pm1-1})^{\mathsf{T}} & (\boldsymbol{\Psi}_{j\pm1+1})^{\mathsf{T}} & \cdots & (\boldsymbol{\Psi}_d)^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}.$$

Then we can write

$$\mathbf{E}^{(\ell)}_j = \mathbf{M}\widetilde{\boldsymbol{\Omega}}, \qquad \mathbf{F}^{(\ell)}_j = \widetilde{\boldsymbol{\Psi}}^{\mathsf{T}}\mathbf{N},$$

where $\mathbf{E}^{(\ell)}_j$ and $\mathbf{F}^{(\ell)}_j$ are are the additive perturbation terms as defined in (4.11) and (4.13). Furthermore, note that $\widetilde{\boldsymbol{\Omega}}$ and $\widetilde{\boldsymbol{\Psi}}$ are independent of $\boldsymbol{\Omega}_j$ and $\boldsymbol{\Psi}_{j\pm1}$. Recall that $\mathbf{Q}^{(\ell)}_j$ is an orthonormal basis for range$(\mathbf{A}_{j\pm1,j}\boldsymbol{\Omega}_j + \mathbf{E}^{(\ell)}_j)$ and $\mathbf{X}^{(\ell)}_j = (\boldsymbol{\Psi}^{\mathsf{T}}_{j\pm1}\mathbf{Q}^{(\ell)}_j)^\dagger(\boldsymbol{\Psi}^{\mathsf{T}}_j\mathbf{A}_{j\pm1,j} + \widetilde{\boldsymbol{\Psi}}^{\mathsf{T}}\mathbf{N})$.

The specified conditions on k, s_{R}, and s_{L} ensure that $s_{\mathrm{L}} \geq 2s_{\mathrm{R}}+1$ and $s_{\mathrm{R}} \geq 2k+1$. Therefore, Theorem 3.2 yields

$$\mathbb{E}\Big[\|\mathbf{A}^{(\ell)}_{j\pm1,j} - \mathbf{Q}^{(\ell)}_j[\![\mathbf{X}^{(\ell)}_j]\!]_k\|_{\mathsf{F}}^2\Big|\mathcal{F}_\ell, \boldsymbol{\xi}, \boldsymbol{\zeta}\Big] \leq E_1 + E_2 + 2\sqrt{E_1E_2}, \tag{5.28}$$

where

$$E_1 := \left(1 + \frac{k}{s_{\mathrm{R}}-k-1}\right)\|\mathbf{A}^{(\ell)}_{j\pm1,j} - [\![\mathbf{A}^{(\ell)}_{j\pm1,j}]\!]_k\|_{\mathsf{F}}^2,$$
$$E_2 := \frac{18k}{s_{\mathrm{R}}-k-1}\|\mathbf{M}\|_{\mathsf{F}}^2 + \frac{8s_{\mathrm{R}}}{s_{\mathrm{L}}-s_{\mathrm{R}}-1}\|\mathbf{N}\|_{\mathsf{F}}^2 + \frac{32s_{\mathrm{R}}}{s_{\mathrm{L}}-s_{\mathrm{R}}-1}\|\mathbf{A}^{(\ell)}_{j\pm1,j} - [\![\mathbf{A}^{(\ell)}_{j\pm1,j}]\!]_k\|_{\mathsf{F}}^2.$$

By our choice of $\boldsymbol{\xi}^{\pm}$, the $\xi^{\pm}_{i,\rho}$ are independent for different i and $\mathbb{E}[\xi^{\pm}_{i,\rho}]$ is zero or $1/t_{\mathrm{R}}$ depending on the parity of ρ. Thus, $\mathbb{E}[\xi^{\pm}_{i,\rho}] \leq 1/t_{\mathrm{R}}$. Furthermore, note that $\mathbb{E}[\xi^{\pm}_{j\pm1,\rho}] = 0$ by construction. Thus,

$$\mathbb{E}\Big[\|\mathbf{M}\|_{\mathsf{F}}^2\Big] = \sum_{i\neq j}\mathbb{E}[\xi^{\pm}_{i,\rho}]\|\mathbf{A}^{(\ell)}_{j\pm1,i}\|_{\mathsf{F}}^2 \leq \sum_{\substack{i=1\\ i\neq j,j\pm1}}^{d}\|\mathbf{A}^{(\ell)}_{j\pm1,i}\|_{\mathsf{F}}^2.$$

By a similar argument,

$$\mathbb{E}\Big[\|\mathbf{N}\|_{\mathsf{F}}^2\Big] = \sum_{i\neq j\pm 1} \mathbb{E}[\zeta_{i,\sigma}^{\mp}]\|\mathbf{A}_{i,j}^{(\ell)}\|_{\mathsf{F}}^2 \leq \frac{1}{t_{\mathrm{L}}} \sum_{\substack{i=1\\ i\neq j,j\pm 1}}^{d} \|\mathbf{A}_{i,j}^{(\ell)}\|_{\mathsf{F}}^2.$$

Hence, by the law of total expectation, the Cauchy-Schwarz inequality, and (5.28) we have

$$\mathbb{E}\Big[\|\mathbf{A}_{j\pm1,j}^{(\ell)} - \mathbf{Q}_j^{(\ell)}[\![\mathbf{X}_j^{(\ell)}]\!]_k\|_{\mathsf{F}}^2 \Big| \mathcal{F}_\ell\Big] \leq E_1 + E_2' + 2\sqrt{E_1 E_2'}, \tag{5.29}$$

where

$$E_2' := \frac{18k}{s_{\mathrm{R}} - k - 1} \cdot \frac{1}{t_{\mathrm{R}}} \sum_{\substack{i=1\\ i\neq j,j\pm1}}^{d} \|\mathbf{A}_{j\pm1,i}^{(\ell)}\|_{\mathsf{F}}^2 + \frac{8s_{\mathrm{R}}}{s_{\mathrm{L}} - s_{\mathrm{R}} - 1} \cdot \frac{1}{t_{\mathrm{L}}} \sum_{\substack{i=1\\ i\neq j,j\pm1}}^{d} \|\mathbf{A}_{i,j}^{(\ell)}\|_{\mathsf{F}}^2 + \frac{32s_{\mathrm{R}}}{s_{\mathrm{L}} - s_{\mathrm{R}} - 1}\|\mathbf{A}_{j\pm1,j}^{(\ell)} - [\![\mathbf{A}_{j\pm1,j}^{(\ell)}]\!]_k\|_{\mathsf{F}}^2.$$

Using that

$$\max\left\{\|\mathbf{A}_{j\pm1,j}^{(\ell)} - [\![\mathbf{A}_{j\pm1,j}^{(\ell)}]\!]_k\|_{\mathsf{F}}^2, \frac{1}{2}\sum_{\substack{i=1\\ i\neq j,j\pm1}}^{d} \|\mathbf{A}_{j\pm1,i}^{(\ell)}\|_{\mathsf{F}}^2, \frac{1}{2}\sum_{\substack{i=1\\ i\neq j,j\pm1}}^{d} \|\mathbf{A}_{i,j}^{(\ell)}\|_{\mathsf{F}}^2\right\} \leq E_j^{(\ell)}$$

and our assumptions on t_{L}, t_{R}, s_{L}, and s_{R} allows us to get bounds

$$\frac{k}{s_{\mathrm{R}} - k - 1}\|\mathbf{A}_{j\pm1,j}^{(\ell)} - [\![\mathbf{A}_{j\pm1,j}^{(\ell)}]\!]_k\|_{\mathsf{F}}^2 \leq \frac{\beta}{30}E_j^{(\ell)} \qquad \frac{18k}{s_{\mathrm{R}} - k - 1} \cdot \frac{1}{t_{\mathrm{R}}} \sum_{\substack{i=1\\ i\neq j,j\pm1}}^{d} \|\mathbf{A}_{j\pm1,i}^{(\ell)}\|_{\mathsf{F}}^2 \leq 36\frac{\beta^2}{30^2}E_j^{(\ell)}$$

$$\frac{8s_{\mathrm{R}}}{s_{\mathrm{L}} - s_{\mathrm{R}} - 1} \cdot \frac{1}{t_{\mathrm{L}}} \sum_{\substack{i=1\\ i\neq j,j\pm1}}^{d} \|\mathbf{A}_{i,j}^{(\ell)}\|_{\mathsf{F}}^2 \leq 16\frac{\beta^2}{30^2}E_j^{(\ell)} \qquad \frac{32s_{\mathrm{R}}}{s_{\mathrm{L}} - s_{\mathrm{R}} - 1}\|\mathbf{A}_{j\pm1,j}^{(\ell)} - [\![\mathbf{A}_{j\pm1,j}^{(\ell)}]\!]_k\|_{\mathsf{F}}^2 \leq 32\frac{\beta^2}{30^2}E_j^{(\ell)}.$$

Therefore, since $E_1 \leq 2E_j^{(\ell)}$ and $1/30 + (36+16+32)/30^2 + 2\sqrt{2(36+16+32)/30^2} \approx 0.991 \leq 1$, we get a bound

$$E_1 + E_2' + 2\sqrt{E_1 E_2'} \leq \|\mathbf{A}_{j\pm1,j}^{(\ell)} - [\![\mathbf{A}_{j\pm1,j}^{(\ell)}]\!]_k\|_{\mathsf{F}}^2 + \beta E_j^{(\ell)}. \tag{5.30}$$

Combining (5.30) with (5.29) yields the first inequality.

The proof of the second result is similar to the analysis of the Generalized Nyström method Theorem 3.2. First define

$$\widehat{\mathbf{N}} = \begin{bmatrix}\widehat{\zeta}_{1,\rho}\widehat{\mathbf{A}}_{1,j}^{(L)} & \cdots & \widehat{\zeta}_{j-1,\rho}\widehat{\mathbf{A}}_{j-1,j}^{(L)} & \widehat{\zeta}_{j+1,\rho}\widehat{\mathbf{A}}_{j+1,j}^{(L)} & \cdots & \widehat{\zeta}_{2^L,\rho}\widehat{\mathbf{A}}_{2^L,j}^{(L)}\end{bmatrix}.$$

and the random Gaussian matrix

$$\widetilde{\boldsymbol{\Psi}} = \begin{bmatrix}(\widehat{\boldsymbol{\Psi}}_1)^{\mathsf{T}} & \cdots & (\widehat{\boldsymbol{\Psi}}_{j-1})^{\mathsf{T}} & (\widehat{\boldsymbol{\Psi}}_{j+1})^{\mathsf{T}} & \cdots & (\widehat{\boldsymbol{\Psi}}_{2^L})^{\mathsf{T}}\end{bmatrix}^{\mathsf{T}}.$$

Similar to above we have

$$\mathbb{E}[\|\widehat{\mathbf{N}}\|_{\mathsf{F}}^2] \leq \frac{1}{t_{\mathrm{L}}} \sum_{\substack{i=1\\ i\neq j}}^{2^L} \|\widehat{\mathbf{A}}_{i,j}^{(L)}\|_{\mathsf{F}}^2.$$

Since $s_{\text{L}} \geq s_{\text{R}} > 2k+1$ and $k \geq n/2^L$ we can apply Theorem 5.1 to obtain

$$\begin{aligned}
\mathbb{E}\Big[\|\widehat{\mathbf{A}}_{j,j}^{(L)} - \widehat{\mathbf{H}}_j\|_{\mathsf{F}}^2 \Big| \mathcal{F}_{L+1}\Big] &= \mathbb{E}\Big[\|(\widehat{\mathbf{\Psi}}_j^{\mathsf{T}})^\dagger \widetilde{\mathbf{\Psi}}^{\mathsf{T}} \widehat{\mathbf{N}}\|_{\mathsf{F}}^2 \Big| \mathcal{F}_{L+1}\Big] \\
&= \frac{n/2^L}{s_{\text{L}} - n/2^L - 1} \mathbb{E}\big[\|\widehat{\mathbf{N}}\|_{\mathsf{F}}^2\big] \\
&\leq \frac{k}{s_{\text{R}} - k - 1} \frac{1}{t_{\text{L}}} \sum_{\substack{i=1 \\ i \neq j}}^{2^L} \|\widehat{\mathbf{A}}_{i,j}^{(L)}\|_{\mathsf{F}}^2 \\
&\leq \frac{\beta}{30} \sum_{\substack{i=1 \\ i \neq j}}^{2^L} \|\widehat{\mathbf{A}}_{i,j}^{(L)}\|_{\mathsf{F}}^2 \leq \beta \sum_{\substack{i=1 \\ i \neq j}}^{2^L} \|\widehat{\mathbf{A}}_{i,j}^{(L)}\|_{\mathsf{F}}^2,
\end{aligned}$$

as required. ∎

Finally, we prove our main theorem.

THEOREM 5.2. *Fix a rank parameter k and let $\beta \in (0,1)$. Suppose $t_{\text{L}} \geq 1$ and s_{R}, t_{L}, and s_{R} are such that*

$$\frac{k}{s_{\text{R}} - k - 1} \leq \frac{\beta}{30}, \qquad \frac{k}{s_{\text{R}} - k - 1} \cdot \frac{1}{t_{\text{R}}} \leq \frac{\beta^2}{30^2}, \qquad \frac{s_{\text{R}}}{s_{\text{L}} - s_{\text{R}} - 1} \leq \frac{\beta^2}{30^2}.$$

Let $L = \lceil \log_2(n/k) \rceil$. Then, using $2Ls_{\text{R}}t_{\text{R}}$ products with $\mathbf{A}$ and $(2L+1)s_{\text{L}}t_{\text{L}}$ products with $\mathbf{A}^{\mathsf{T}}$, Algorithm 3 outputs a HODLR(k) *matrix $\widetilde{\mathbf{A}}$ such that*

$$\mathbb{E}\Big[\|\mathbf{A} - \widetilde{\mathbf{A}}\|_{\mathsf{F}}^2\Big] \leq (1+\beta)^{L+1} \cdot \min_{\mathbf{H} \in \text{HODLR}(k)} \|\mathbf{A} - \mathbf{H}\|_{\mathsf{F}}^2.$$

Theorem 1.1 is an immediate consequence of Theorem 5.2.

Proof of Theorem 1.1. We can instantiate Theorem 5.2 with

$$s_{\text{R}} = O(k/\beta), \quad t_{\text{R}} = O(1/\beta), \quad s_{\text{L}} = O(s_{\text{R}}/\beta^2) = O(k/\beta^3), \quad t_{\text{L}} = 1. \tag{5.31}$$

Write $\mathsf{err}^2 := \|\mathbf{A} - \widetilde{\mathbf{A}}\|_{\mathsf{F}}^2$ and $\mathsf{opt}^2 := \min_{\mathbf{H} \in \text{HODLR}(k)} \|\mathbf{A} - \mathbf{H}\|_{\mathsf{F}}^2$. Rearanging Theorem 1.1 we see $\mathbb{E}[\mathsf{err}^2 - \mathsf{opt}^2] \leq ((1+\beta)^{L+1} - 1)\mathsf{opt}^2$. Then, since $\mathsf{err}^2 - \mathsf{opt}^2 \geq 0$, we can apply Markov's inequality and obtain a bound

$$\mathbb{P}\Big[\mathsf{err}^2 - \mathsf{opt}^2 > 100\big((1+\beta)^{L+1} - 1\big)\mathsf{opt}^2\Big] \leq \frac{1}{100}.$$

Note that $1 + 100((1+\beta)^{L+1} - 1) \leq (1+100\beta)^{L+1}$ for all $\beta, L > 0$. Therefore, $\mathsf{err}^2 \leq (1+100\beta)^{L+1}\mathsf{opt}^2$, except with probability at most $1/100$. Taking a square root and adjusting β by a constant factor gives the result. ∎

The parameters in (5.31) are chosen to minimize the beta dependencies in the resulting bound. Another interesting parameter setting is

$$s_{\text{R}} = O(k/\beta^2), \quad t_{\text{R}} = 1, \quad s_{\text{L}} = O(k/\beta^4), \quad t_{\text{L}} = 1, \tag{5.32}$$

which corresponds to an implementation of the standard peeling algorithm with the Generalized Nyström Method (i.e. does not use randomly perforated sketching) attaining the same $(1+\beta)^{L+1}$ approximation factor. In the experiments in Section 7, we find that it does not seem necessary to set $s_{\text{L}} = O(k/\beta^4)$, and that $s_{\text{L}} = O(k/\beta^2)$ (with $t_{\text{L}} = 1$) seems sufficient.

Proof of Theorem 5.2. First, note that at each of the L levels we use $2s_{\text{R}}t_{\text{R}}$ matrix-vector products with $\mathbf{A}$ and $2s_{\text{L}}t_{\text{L}}$ matrix-vector products with $\mathbf{A}^{\mathsf{T}}$. When recovering the diagonal we use an additional $s_{\text{L}}t_{\text{L}}$ products with $\mathbf{A}^{\mathsf{T}}$. This results in the specified cost.

We now analyze the error. For each $\ell = 1, 2, \ldots, L$ partition $\mathbf{A}^{(\ell)}$ as in (4.5) and $\mathbf{H}^{(\ell)}$ as in (4.7). Define,

$$\mathsf{opt}(\ell) := \left(\sum_{j=1}^{2^\ell} \|\mathbf{A}^{(\ell)}_{j\pm 1,j} - [\![\mathbf{A}^{(\ell)}_{j\pm 1,j}]\!]_k\|_{\mathsf{F}}^2 \right)^{1/2}.$$

By the definition of HODLR matrices the on-diagonal blocks at level L are of size at most k. Therefore, since the different levels are disjoint,

$$\min_{\mathbf{H}\in \mathrm{HODLR}(k)} \|\mathbf{A} - \mathbf{H}\|_{\mathsf{F}}^2 = \mathsf{opt}(1)^2 + \cdots + \mathsf{opt}(L)^2. \tag{5.33}$$

Next, define the error incurred in the off-diagonal low-rank blocks at level ℓ,

$$\mathsf{err}(\ell) := \left(\sum_{j=1}^{2^\ell} \|\mathbf{A}^{(\ell)}_{j\pm 1,j} - \mathbf{H}^{(\ell)}_j\|_{\mathsf{F}}^2 \right)^{1/2}.$$

Define also the error of the on-diagonal blocks at the final level,

$$\widehat{\mathsf{err}} := \left(\sum_{j=1}^{2^L} \|\mathbf{A}^{(L)}_{j,j} - \widehat{\mathbf{H}}_j\|_{\mathsf{F}}^2 \right)^{1/2}.$$

Since the approximations at each level are disjoint (see (4.4)) we have

$$\|\mathbf{A} - \widetilde{\mathbf{A}}\|_{\mathsf{F}}^2 = \mathsf{err}(1)^2 + \cdots + \mathsf{err}(L)^2 + \widehat{\mathsf{err}}^2. \tag{5.34}$$

We claim that it suffices to prove that for each $\ell = 1, 2, \ldots, L$,

$$\mathbb{E}\big[\mathsf{err}(\ell)^2\big] \le \mathsf{opt}(\ell)^2 + \beta \cdot (1+\beta)^{\ell-1} \cdot \big(\mathsf{opt}(1)^2 + \cdots + \mathsf{opt}(\ell)^2\big). \tag{5.35}$$

and that at the final level

$$\mathbb{E}\big[\widehat{\mathsf{err}}^2\big] \le \beta \cdot (1+\beta)^{L} \cdot \big(\mathsf{opt}(1)^2 + \cdots + \mathsf{opt}(L)^2\big). \tag{5.36}$$

Indeed, since $\mathsf{err}(\ell')^2 \ge 0$, together (5.35) and (5.36) imply that

$$\begin{aligned}\mathbb{E}\big[\mathsf{err}(1)^2 + \cdots + \mathsf{err}(L)^2 + \widehat{\mathsf{err}}^2\big] &\le \Big(1 + \beta \cdot \big(1 + (1+\beta) + \cdots + (1+\beta)^L\big)\Big) \cdot \Big(\mathsf{opt}(1)^2 + \cdots + \mathsf{opt}(L)^2\Big) \\ &= (1+\beta)^{L+1} \cdot \Big(\mathsf{opt}(1)^2 + \cdots + \mathsf{opt}(L)^2\Big).\end{aligned}$$

In light of (5.33) and (5.34), this is the desired result.

We begin by proving (5.35) by induction. First note that (5.35) at level $\ell = 1$ we have incurrred no error from previous levels and Lemma 5.2 gives $\mathbb{E}[\mathsf{err}(1)] \le (1+\beta)\mathsf{opt}(1)$. Now suppose the analog of (5.35) holds for each level up to $\ell - 1$. Then, since $\mathsf{err}(\ell')^2 \ge 0$,

$$\begin{aligned}\mathbb{E}\big[\mathsf{err}(1)^2 + \cdots + \mathsf{err}(\ell-1)^2\big] &\le \big(1 + \beta \cdot (1 + (1+\beta) + \cdots + (1+\beta)^{\ell-2})\big) \cdot \big(\mathsf{opt}(1)^2 + \cdots + \mathsf{opt}(\ell-1)^2\big) \\ &= (1+\beta)^{\ell-1} \cdot \big(\mathsf{opt}(1)^2 + \cdots + \mathsf{opt}(\ell-1)^2\big). \end{aligned} \tag{5.37}$$

Applying Lemma 5.2 for each $j = 1, 2, \ldots 2^\ell$ and summing over j we find

$$\mathbb{E}\big[\mathsf{err}(\ell)^2\big] = \mathbb{E}\big[\mathbb{E}\big[\mathsf{err}(\ell)^2 \big| \mathcal{F}_\ell\big]\big] \le \mathbb{E}\left[\mathsf{opt}(\ell)^2 + \beta \sum_{j=1}^{2^\ell} E_j^{(\ell)}\right]. \tag{5.38}$$

Observe that the sum over all blocks of $\mathbf{A}^{(\ell)}$ except the on-diagonal blocks and the off-diagonal low-rank blocks at level ℓ is expressed as

$$\sum_{j=1}^{2^\ell} \sum_{\substack{i=1 \\ i\neq j, j\pm 1}}^{2^\ell} \|\mathbf{A}^{(\ell)}_{i,j}\|_{\mathsf{F}}^2 = \sum_{j=1}^{2^\ell} \sum_{\substack{i=1 \\ i\neq j, j\pm 1}}^{2^\ell} \|\mathbf{A}^{(\ell)}_{j\pm 1,i}\|_{\mathsf{F}}^2.$$

Therefore,

$$\sum_{j=1}^{2^\ell} E_j^{(\ell)} \le \sum_{j=1}^{2^\ell} \left(\|\mathbf{A}_{j\pm1,j}^{(\ell)} - [\![\mathbf{A}_{j\pm1,j}^{(\ell)}]\!]_k\|_{\mathsf{F}}^2 + \sum_{\substack{i=1 \\ i\neq j,j\pm1}}^{2^\ell} \frac{1}{2}\|\mathbf{A}_{i,j}^{(\ell)}\|_{\mathsf{F}}^2 + \sum_{\substack{i=1 \\ i\neq j,j\pm1}}^{2^\ell} \frac{1}{2}\|\mathbf{A}_{j\pm1,i}^{(\ell)}\|_{\mathsf{F}}^2 \right)$$

$$= \mathsf{opt}(\ell)^2 + \sum_{j=1}^{2^\ell} \sum_{\substack{i=1 \\ i\neq j,j\pm1}}^{2^\ell} \|\mathbf{A}_{i,j}^{(\ell)}\|_{\mathsf{F}}^2$$

$$= \mathsf{opt}(\ell)^2 + \big(\mathsf{err}(1)^2 + \cdots + \mathsf{err}(\ell-1)^2\big). \tag{5.39}$$

In the final equality above we have used that the (i,j) blocks for $i \neq j, j\pm1$ contain exactly the errors made at previous levels. Hence, taking the expectation of (5.39), using (5.37), and plugging into (5.38) gives

$$\begin{aligned}\mathbb{E}\big[\mathsf{err}(\ell)^2\big] &\le \mathsf{opt}(\ell)^2 + \beta\big(\mathsf{opt}(\ell)^2 + (1+\beta)^{\ell-1}\big(\mathsf{opt}(1)^2 + \cdots + \mathsf{opt}(\ell-1)^2\big)\big) \\ &\le \mathsf{opt}(\ell)^2 + \beta(1+\beta)^{\ell-1}\big(\mathsf{opt}(1)^2 + \cdots + \mathsf{opt}(\ell)^2\big),\end{aligned}$$

which gives (5.35).

The proof of (5.36) follows analogously. By Lemma 5.2 we have

$$\mathbb{E}\big[\widehat{\mathsf{err}}^2\big] \le \beta \sum_{j=1}^{2^L} \sum_{\substack{i=1 \\ i\neq j}}^{2^L} \mathbb{E}\big[\|\widehat{\mathbf{A}}_{i,j}^{(L)}\|_{\mathsf{F}}^2\big] = \beta \sum_{\ell=1}^{L} \mathbb{E}\left[\mathsf{err}(\ell)^2\right] \le \beta(1+\beta)^L(\mathsf{opt}(1)^2 + \cdots + \mathsf{opt}(L)^2),$$

where we used that $\sum_{j=1}^{2^L} \sum_{i=1,i\neq j}^{2^L} \|\widehat{\mathbf{A}}_{i,j}^{(L)}\|_{\mathsf{F}}^2$ is precisely the errors made at previous levels and (5.37). This gives the desired inequality. ∎

6 Lower Bounds

In this section, we prove that our main result on HOLDR matrix approximation (Theorem 1.1) is close to optimal in terms of the number of matrix-vector products needed to solve Problem 1.1.

6.1 Lower bound for exact recovery We begin with a simple lower bound in the setting where $\mathbf{A}$ is exactly HODLR, in which case solving Problem 1.1 requires *exactly* recovering $\mathbf{A}$. For this setting, we can appeal to prior work by Halikias and Townsend [HT23] on the query complexity of recovering matrices from *linearly parameterized matrix families*. In particular, any set $\mathbf{B}_1, \ldots, \mathbf{B}_m$ of *linearly independent* base matrices induces a linearly parameterized family $\mathcal{L}$ consisting of linear combinations of the base matrices; i.e.

$$\mathcal{L} = \left\{ \sum_{i=1}^{m} \theta_i \mathbf{B}_i : \boldsymbol{\theta} = [\theta_1, \ldots, \theta_m] \in \mathbb{R}^m \right\}$$

Halikias and Townsend observe (see [HT23], Lemma 2.2) that recovering a matrix $\mathbf{A} \in \mathcal{L}$ requires at least $\lceil m/n \rceil$ matvec queries with $\mathbf{A}$. In particular, any matrix-vector product with $\mathbf{A}$ or $\mathbf{A}^\mathsf{T}$ yields n linear equations in entries of $\boldsymbol{\theta}$. Since recovering $\mathbf{A}$ amounts to determining the corresponding $\boldsymbol{\theta}$, at least m such equations are needed to uniquely determine $\mathbf{A}$. We leverage this observation to prove the following:

THEOREM 6.1. *Any algorithm that can recover any $n \times n$ matrix $\mathbf{A} \in \mathrm{HODLR}(k)$ from adaptive matvec queries must use at least $\lceil k \log_2(n/k) \rceil$ queries. Therefore, solving Problem 1.1 requires $\Omega(k \log(n/k))$ matvec queries for any finite approximation factor Γ.*

Proof. The proof follows from observing that, while $\mathrm{HODLR}(k)$ is not itself a linearly parameterized family, it contains a linearly parameterized family $\mathcal{L}$. In particular, consider the subset of $\mathrm{HODLR}(k)$ matrices where every rank-k block in the matrix is zero everywhere except in its first k columns. The first k columns in each block can be chosen to have entries with any value such that the block is rank-k, and the matrix is therefore $\mathrm{HODLR}(k)$.

As such, $\mathcal{L}$ is equal to the set of matrices that can be written as a linear combination of $\mathbf{B}_1, \ldots, \mathbf{B}_m$, where each $\mathbf{B}_i$ is a matrix that is zero everywhere, but has a single 1 in one of the first k columns in one of the rank-k blocks of the HODLR structure. Recall that for an $n \times n$ matrix to be HODLR(k), it must be that $n = n_{\text{base}} \cdot 2^p$ for integers $n_{\text{base}} \in (k/2, k]$ and $p = \log_2(n/n_{\text{base}})$.[12] It can be checked that the total number of base matrices, m, equals:

$$m = 2n_{\text{base}}n + nk(p-1) > nkp \geq nk\log_2(n/k)$$

So, by Lemma 2.2 in [HT23], we require $\lceil nk\log_2(n/k)\rceil/n\rceil = \lceil k\log_2(n/k)\rceil$ matvec queries to exactly recover a given $\mathbf{A} \in \mathcal{L}$. Since $\mathcal{L} \subset$ HODLR(k), the theorem follows. We note that to solve Problem 1.1 for finite Γ, we must exactly recover all $\mathbf{A} \in$ HODLR(k). ∎

6.2 Lower bound for approximation Next, we prove a lower bound in the setting where $\mathbf{A}$ is not exactly HODLR, and we seek to solve Problem 1.1 with error $\Gamma = (1+\varepsilon)$ for some $\varepsilon \in (0,1)$. A natural approach for such a lower bound might be to leverage lower bounds for near-optimal rank-k approximation, since HODLR approximation is a strictly harder problem. However, as discussed in Section 1.2, the best known lower bound for Frobenius-norm error rank-k approximation in the matrix-vector query model is just $O(k+1/\varepsilon^{1/3})$ [BN23]. Such a lower bound does not show that the multiplicative relationship between k and poly$(1/\varepsilon)$ in our upper bound, Theorem 1.1, is necessary.

We obtain a stronger lower bound by instead proving a reduction from the problem of *fixed-pattern sparse matrix approximation*, which was recently studied in [Ams+24]. That work proves tight lower bounds in the matrix-vector query model for approximating matrices by a wide variety of matrix classes with fixed sparsity patterns, including diagonal matrices, banded matrices, and more. Below, we state a special case of the lower bound from [Ams+24] for approximation by *block diagonal matrices*. This lower bound will be used to obtain our lower bound for HODLR approximation. For integers n and b, such that b divides n, we let $\mathcal{B}(n,b)$ denote the set of $n \times n$ block-diagonal matrices with blocks of size $b \times b$. We have the following:

LEMMA 6.1. (COROLLARY OF THM. 2 FROM [AMS+24][13]) *There are absolute constants $c, C > 0$ such that the following holds: For any $\varepsilon > 0$ and any positive integers b, n such that b divides n and $n \geq cb/\varepsilon$, there is a distribution over $n \times n$ matrices $\mathbf{A}$ such that any algorithm that accesses $\mathbf{A}$ with $\leq Cb/\varepsilon$ adaptive matvec queries and returns an approximation $\widetilde{\mathbf{B}} \in \mathcal{B}(n,b)$ must have $\|\mathbf{A} - \widetilde{\mathbf{B}}\|_{\mathsf{F}} \geq (1+\varepsilon)\min_{\mathbf{B}\in\mathcal{B}(n,b)} \|\mathbf{B} - \mathbf{A}\|_{\mathsf{F}}$ with probabilty $\geq 24/25$. The probability is taken over the randomness of $\mathbf{A}$, as well as possible randomness in the algorithm.*

Lemma 6.1 establishes that, even to succeed with small positive probability, any algorithm for computing a near-optimal block-diagonal approximation to an arbitrary matrix $\mathbf{A}$ requires $\Omega(b/\varepsilon)$ matvec queries to $\mathbf{A}$. Intuitively this result is useful because block diagonal approximation is an *easier* problem than HODLR approximation. In particular, it is not hard to verify that $\mathcal{B}(n, 2k) \subset$ HODLR(k); matrices in $\mathcal{B}(n,2k)$ are zero except on the diagonal and off-diagonal low-rank blocks of the final level of HODLR(k) matrices. Formally, we prove that, if we had an algorithm for finding a near-optimal HODLR approximation to a given matrix, the result could be post-processed via projection onto $\mathcal{B}(n,2k)$ to obtain a near-optimal block-diagonal approximation. If we found the near-optimal HODLR approximation with $o(k/\epsilon)$ matvec queries, we would violate Lemma 6.1. This approach results in the following lower bound:

THEOREM 6.2. *There are absolute constants $c, C > 0$ such that the following holds: For any k, $\varepsilon > 0$, and $n \geq ck/\varepsilon$,[14] there is a distribution over $n \times n$ matrices $\mathbf{A}$ such that any algorithm which accesses $\mathbf{A}$ with $\leq Ck/\varepsilon$ adaptive matvec queries fails to solve Problem 1.1 with probability $\geq 24/25$.*

Proof. First note that we may assume $\epsilon \leq 1$, as the result already holds by Theorem 6.1 for larger values of ϵ. Let $n = n_{\text{base}}2^p$ for $n_{\text{base}} \in \lfloor k/2+1\rfloor, \ldots, k$ and $p \geq 0$. Let $b = 2n_{\text{base}}$ and observe that $b \geq k$, and for $c \geq 2$, $b \leq n$. Let $\mathbf{S} \in \{0,1\}^{n\times n}$ be an indicator matrix for a block-diagonal sparsity pattern with n/b blocks of size b. I.e., $\mathbf{S}$ is zero everywhere except that, for each $t \in 0, \ldots, n/b-1$, $\mathbf{S}_{i,j} = 1$ for $i,j \in \{bt+1, \ldots, bt+b\}$. Observe

[12] If $n \leq k/2$ then the bound is vacuously true.

[14] Recall that HODLR(k) matrices are only defined for dimensions n of the form $n = n_{\text{base}}2^p$ for $n_{\text{base}} \leq k$ and $p \geq 0$. Formally, the theorem holds for any such n that is $\geq ck/\varepsilon$.

that for any block diagonal matrix $\mathbf{B} \in \mathcal{B}(n,b)$, $\mathbf{B} \circ \mathbf{S} = \mathbf{B}$, where "$\circ$" denotes the entrywise product. Let $\bar{\mathbf{S}}$ denote the compliment of $\mathbf{S}$: for all i,j, $\bar{\mathbf{S}}_{i,j} = 1 - \mathbf{S}_{i,j}$. Observe that for any matrix $\mathbf{H}$, we can write:

$$\|\mathbf{A} - \mathbf{H}\|_{\mathsf{F}}^2 = \|\mathbf{A} \circ \mathbf{S} - \mathbf{H} \circ \mathbf{S}\|_{\mathsf{F}}^2 + \|\mathbf{A} \circ \bar{\mathbf{S}} - \mathbf{H} \circ \bar{\mathbf{S}}\|_{\mathsf{F}}^2. \tag{6.40}$$

Define

$$\mathbf{B}^* = \mathbf{A} \circ \mathbf{S} \operatorname*{argmin}_{\mathbf{B} \in \mathcal{B}(n,b)} \|\mathbf{A} - \mathbf{B}\|_{\mathsf{F}}^2, \qquad \mathbf{H}^* = \operatorname*{argmin}_{\mathbf{H} \in \mathrm{HODLR}(k)} \|\mathbf{A} - \mathbf{H}\|_{\mathsf{F}}^2.$$

Since $\mathcal{B}(n,b) \subset \mathrm{HODLR}(k)$, we have that $\|\mathbf{A} - \mathbf{H}^*\|_{\mathsf{F}} \leq \|\mathbf{A} - \mathbf{B}^*\|_{\mathsf{F}}$.

Our proof approach is to show that, if we can solve Problem 1.1 with error $\Gamma = (1+\varepsilon)$ using m matrix-vector products, i.e. if we can find some $\widetilde{\mathbf{H}} \in \mathrm{HODLR}(k)$ that satisfies

$$\|\mathbf{A} - \widetilde{\mathbf{H}}\|_{\mathsf{F}} \leq (1+\varepsilon)\|\mathbf{A} - \mathbf{H}^*\|_{\mathsf{F}}, \tag{6.41}$$

then $\widetilde{\mathbf{H}} \circ \mathbf{S}$ is a near-optimal block diagonal approximation to $\mathbf{A}$. Concretely, we will show that

$$\|\mathbf{A} - \widetilde{\mathbf{H}} \circ \mathbf{S}\|_{\mathsf{F}} \leq (1+\varepsilon)\|\mathbf{A} - \mathbf{B}^*\|_{\mathsf{F}}. \tag{6.42}$$

Accordingly, we reach a contradiction to Lemma 6.1 unless $m = \Omega(k/\varepsilon) = \Omega(b/\varepsilon)$ matrix-vector products were used to compute $\widetilde{\mathbf{H}}$.

To see why (6.41) implies (6.42), the main observation is that:

$$\|\mathbf{A} \circ \bar{\mathbf{S}} - \widetilde{\mathbf{H}} \circ \bar{\mathbf{S}}\|_{\mathsf{F}}^2 \geq \|\mathbf{A} - \mathbf{H}^*\|_{\mathsf{F}}^2. \tag{6.43}$$

(6.43) follows from the fact that the entries in the bottom level of a HODLR(k) matrix (those corresponding to the ones in $\mathbf{S}$) can be set arbitrarily without violating the HODLR property. Accordingly, by adjusting those entries to match $\mathbf{A}$ exactly, we can obtain a HODLR approximation with error equal to $\|\mathbf{A} \circ \bar{\mathbf{S}} - \widetilde{\mathbf{H}} \circ \bar{\mathbf{S}}\|_{\mathsf{F}}^2$. The optimal approximation $\mathbf{H}^*$ cannot have larger error.

Then, using (6.41), (6.40), and (6.43), we have:

$$\begin{aligned}(1+\varepsilon)^2\|\mathbf{A} - \mathbf{H}^*\|_{\mathsf{F}}^2 \geq \|\mathbf{A} - \widetilde{\mathbf{H}}\|_{\mathsf{F}}^2 &= \|\mathbf{A} \circ \mathbf{S} - \widetilde{\mathbf{H}} \circ \mathbf{S}\|_{\mathsf{F}}^2 + \|\mathbf{A} \circ \bar{\mathbf{S}} - \widetilde{\mathbf{H}} \circ \bar{\mathbf{S}}\|_{\mathsf{F}}^2 \\ &\geq \|\mathbf{A} \circ \mathbf{S} - \widetilde{\mathbf{H}} \circ \mathbf{S}\|_{\mathsf{F}}^2 + \|\mathbf{A} - \mathbf{H}^*\|_{\mathsf{F}}^2.\end{aligned}$$

We conclude that

$$\|\mathbf{A} \circ \mathbf{S} - \widetilde{\mathbf{H}} \circ \mathbf{S}\|_{\mathsf{F}}^2 \leq (2\varepsilon + \varepsilon^2)\|\mathbf{A} - \mathbf{H}^*\|_{\mathsf{F}}^2 \leq (2\varepsilon + \varepsilon^2)\|\mathbf{A} - \mathbf{B}^*\|_{\mathsf{F}}^2.$$

Moreover, $\|\mathbf{A} - \widetilde{\mathbf{H}} \circ \mathbf{S}\|_{\mathsf{F}}^2 = \|\mathbf{A} \circ \mathbf{S} - \widetilde{\mathbf{H}} \circ \mathbf{S}\|_{\mathsf{F}}^2 + \|\mathbf{A} - \mathbf{A} \circ \mathbf{S}\|_{\mathsf{F}}^2$, and recall that $\mathbf{A} \circ \mathbf{S} = \mathbf{B}^*$, So:

$$\|\mathbf{A} - \widetilde{\mathbf{H}} \circ \mathbf{S}\|_{\mathsf{F}}^2 \leq (2\varepsilon + \varepsilon^2)\|\mathbf{A} - \mathbf{B}^*\|_{\mathsf{F}}^2 + \|\mathbf{A} - \mathbf{B}^*\|_{\mathsf{F}}^2 = (1+\varepsilon)^2\|\mathbf{A} - \mathbf{B}^*\|_{\mathsf{F}}^2.$$

Taking a square root on both sides proves (6.42), and as explained above, Theorem 6.2 follows. ∎

We conclude the section by noting that Theorem 1.2 (stated in Section 1) follows from combining Theorem 6.1 and Theorem 6.2. An interesting question for future work is to prove a lower bound that multiplicatively combines k, $\log(n/k)$, and $1/\varepsilon$; e.g., to prove that $m = \Omega(k\log(n/k)/\varepsilon)$ matrix-vector product queries are necessary to solve Problem 1.1. This is currently beyond the reach of our current approach and that of [Ams+24]. In particular, the lower bound instance in [Ams+24] is based on a random Wishart matrix, which has found applications in a number of prior results on lower bounds for adaptive matrix-vector query algorithms [BHSW20]. It can be checked that a constant factor near-optimal HODLR approximation for such a matrix can be found by simply returning a near-optimal block diagonal approximation for block size $O(k)$. Since that can be done with $O(k)$ matrix-vector products using the algorithm from [Ams+24], we cannot hope to use the Wishart instance to prove a lower bound that combines k and $\log(n/k)$.

7 Numerical experiments and examples

In this section we provide several examples which provide insight into the behavior of peeling-based algorithms. The code used to generate our figures is available at https://github.com/tchen-research/HODLR_approx.

In our experiments we consider two parameter choices for the Generalized Nyström Method based method Algorithm 3 and two parameter choices for a Randomized SVD based method Algorithm 4 described in Appendix A. The parameter values are summarized in Table 1. The aim of our numerical experiments is to provide some basic insight into how the various parameters of peeling algorithms impact their performance. There are many reasonable parameter combinations, and a comprehensive understanding of the best choice of parameters is far beyond the scope of this paper, but would be an interesting topic for future work.

name	style	algorithm	s_{R}	t_{R}	s_{L}	t_{L}	# matvecs
GN1	——	Algorithm 3	k/β	1	k/β^2	1	$O(k/\beta^2)$
GN2	.....	Algorithm 3	k/β	$1/\beta$	k/β^2	1	$O(k/\beta^2)$
RSVD1	- - -	Algorithm 4	k/β	1	$\sim$	1	$O(k/\beta)$
RSVD2	—·-	Algorithm 4	k/β	$1/\beta$	$\sim$	$1/\beta$	$O(k/\beta^2)$

Table 1: Parameter choices used in our numerical experiments.

The RSVD1 and GN1 methods do not use random perforated sketches and can therefore be viewed as standard implementations of the peeling algorithm (e.g. as described in [LLY11; Mar16]). The RSVD1 method uses the same sketching dimensions as required to obtain a $(1+O(\beta))$-optimal rank-k approximation to a matrix [HMT11]. Similarly, the GN1 method uses the sketching dimensions required for the Generalized Nyström Method (without truncation to rank-k) to produce a low-rank approximation with error less than $(1+O(\beta))$ times the optimal rank-k approximation to a matrix [TYUC17].[15] The choice of paramaters for GN1 is also equivalent (after rescaling β) to the parameters (5.32) needed for Theorem 5.2 to guarantee convergence for Generalized Nyström peeling method without perforation.

Both RSVD2 and GN2 use the randomly perforated sketches described in Section 3. In particular, the GN2 method adds perforation to the sketch used to obtain the approximate range. Compared to GN1, this does not increase the asymptotic cost but improves the bound for Γ which can be obtained from Theorem 5.2. Interestingly, in our numerical experiments, the accuracy of GN1 and GN2 is very similar. The RSVD2 method adds perforation to both sketches. This increases the asymptotic cost over RSVD1. However, as illustrated in Section 7.3, regardless of β, RSVD1 cannot solve Problem 1.1 for arbitrary $\Gamma > 1$. On the other hand, RSVD2 can.

For an approximation $\widetilde{\mathbf{A}}$ to $\mathbf{A}$, we will consider the relative and absolute errors defined by:

$$\text{relative error: } \frac{\|\mathbf{A}-\widetilde{\mathbf{A}}\|_{\mathsf{F}} - \|\mathbf{A}-\mathbf{A}^\star\|_{\mathsf{F}}^2}{\|\mathbf{A}-\mathbf{A}^\star\|_{\mathsf{F}}^2}, \qquad \text{absolute error: } \|\mathbf{A}-\widetilde{\mathbf{A}}\|_{\mathsf{F}},$$

where $\mathbf{A}^\star := \min_{\mathbf{H}\in\text{HODLR}(k)} \|\mathbf{A}-\mathbf{H}\|_{\mathsf{F}}^2$ is the best possible HODLR(k) approximation to $\mathbf{A}$. When $\widetilde{\mathbf{A}}$ is HODLR(k), the relative error corresponds to the smallest value of ε for which Problem 1.1 is solved with $\Gamma = (1+\varepsilon)$.

7.1 Poisson's equation In this example, we take $\mathbf{A}$ as the discretized solution operator to a differential equation. In particular, we consider the 2-dimensional Poisson's equation

$$\frac{\partial^2}{\partial x^2}u(x,y) + \frac{\partial^2}{\partial y^2}u(x,y) = f(x,y)$$

on a periodic domain $(x,y) \in [0,1]^2$ (i.e., with boundary conditions $u(x,0) = u(x,1)$ and $u(0,y) = u(1,y)$. We discretize the problem on a uniform $t\times t$ grid $x_i = i/t$, $y_j = j/t$ for $i,j = 0,1,\dots,t-1$ (so that $n = t^2$). The matrix $\mathbf{A}$ is defined as the $n\times n$ linear map taking forcing data $\mathbf{f} = \{f(x_i,y_j)\}_{i,j=1}^t \in \mathbb{R}^n$ to the solution approximate

[15] As we discuss in Section 3.2, it is believed that this also produces a $(1+O(\beta))$-optimal approximation with truncation.

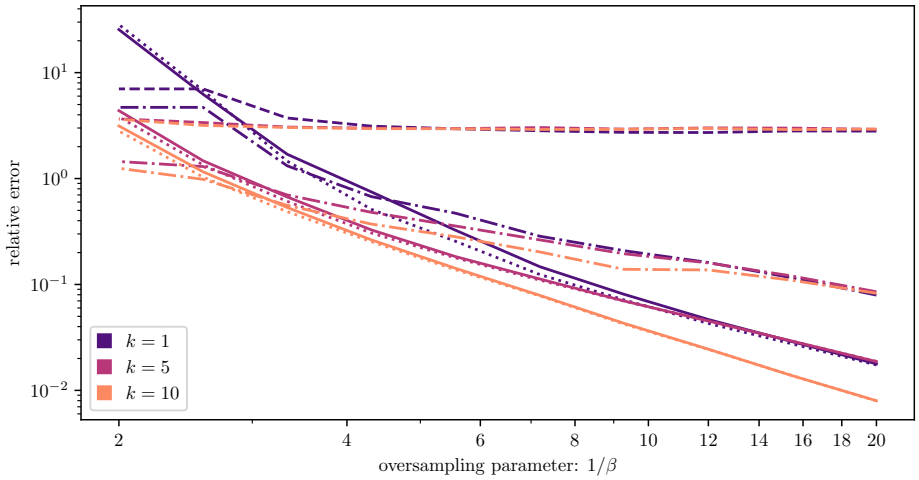

Figure 4: Relative error of peeling algorithms from Table 1 on discrete inverse Poisson matrix described in (7.44) as a function of the the oversampling parameter β for several choices of the rank-parameter k. *Legend*: GN1 (——), GN2 (·····), RSVD1 (- - -), RSVD2 (—·-). *Takeaway*: Both Generalized Nyström Method-based variants seem to perform similarly, and produce an increasingly accurate output as $1/\beta$ increases. On the other hand, the standard randomized SVD-based variant, RSVD1, stagnates. The RSVD2 variant, which uses randomly perforated sketches, converges.

$\mathbf{u} = \{u_{i,j}\}_{i,j=1}^{t}$. Matrix-vector products with $\mathbf{A}$ (and $\mathbf{A}^\mathsf{T}$) can be efficiently computed using a FFT-based 2D Poisson solver. Specifically, given $\mathbf{f} \in \mathbb{R}^n = \mathbb{R}^{t\times t}$, we define $\mathbf{A}$ by

$$\mathbf{A}\mathbf{f} = \mathrm{IDFT}_2(\mathbf{D}\,\mathrm{DFT}_2(\mathbf{f})), \tag{7.44}$$

where $\mathbf{D} : \mathbb{R}^{t\times t} \to \mathbb{R}^{t\times t}$ is the diagonal map with diagonal entries $\mathbf{D}_{i,j} = -1/(\kappa_i^2 + \kappa_j^2)$, where the harmonics κ_i are defined by $\kappa_i = i$ if $i \le t/2 - 1$ and $\kappa_i = m - t$ if $i > t/2 - 1$. Here DFT_2 and IDFT_2 are respectively the 2-dimensional Discrete Fourier Transform and Inverse Discrete Fourier Transform. In our experiment we set $t = 32$ so that $n = t^2 = 1024$.

In Figure 4 we show the relative error of the algorithms from Table 1 for various ranks k as a function of the oversampling parameters β (averaged over 20 trials). As expected, as $\beta \to 0$, the relative errors of GN1, GN2, and RSVD1 all converge to zero. On the other hand, RSVD1 stagnates. In the left panel of Figure 5 we show the absolute error. Here we see that while the RSVD1 algorithm has a much higher relative error than the other algorithms, the absolute error is not significantly larger. Finally, in the right panel of Figure 5 we show the absolute error of the algorithms without truncation to rank-k. This results in a much better approximations for the same number of matrix-vector products, but the resulting approximations have HODLR rank larger than k. The best tradeoff between HODLR rank and matvecs depends on the problem at hand and the computing environment.

7.2 Kernel Matrix In this example we consider a kernel matrix $\mathbf{A}$ defined by

$$[\mathbf{A}]_{i,j} = \begin{cases} 1/\|\mathbf{x}_i - \mathbf{x}_j\|_2 & i \neq j \\ 0 & i = j \end{cases}, \tag{7.45}$$

where $\{\mathbf{x}_i\}_{i=1}^n$ are a collection of points. Matrix-vector products with matrices such as $\mathbf{A}$ can be efficiently performed using algorithms such as the Fast Multipole Method [GR87].

In our numerical experiments, we set $n = 16384$ and sample points $\mathbf{x}_i = (x_i, y_i, z_i)$ by taking x_i as uniformly spaced points in $[-4, 4]$, $y_i = \sin(2\pi x_i) + 0.05\xi_i$, and $z_i = \cos(2\pi x_i) + 0.05\zeta_i$, where ξ_i and ζ_i are independent

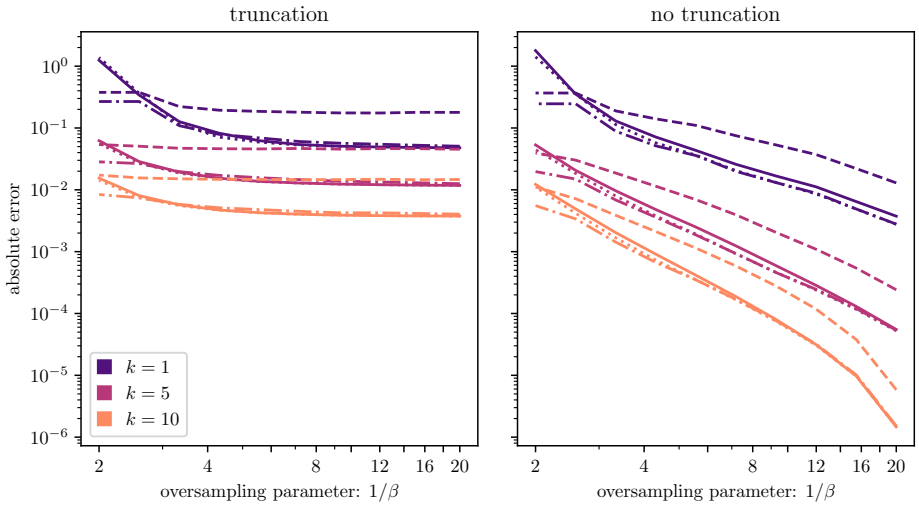

Figure 5: Absolute error of peeling algorithms from Table 1 on discrete inverse Poisson matrix described in (7.44) as a function of the oversampling parameter β with and without truncation to rank-k for several choices of the rank-parameter k. *Legend*: GN1 (——), GN2 (·····), RSVD1 (- - -), RSVD2 (—·–). *Takeaway*: Without truncation, all of the algorithms can perform significantly better. However, the resulting approximations are not HODLR(k), and are therefore more expensive to store and work with.

standard normal random variables. Products with $\mathbf{A}$ are performed using the Flatiron Institute's FMM3D code [Ask+]. The left panel of Figure 6 shows a sample of the points we use, and the right panel shows the magnitude of the entries of the kernel matrix (7.45) induced by these points. In Figure 7 we show the absolute error of the GN1 and RSVD1 methods as a function of the target rank, for several values of β.

7.3 Hard instances for RSVD-based peeling In this section we provide two examples which illustrate that a RSVD-based implementation of the peeling algorithm, which truncates the low-rank approximations to rank-k at every level, cannot solve Problem 1.1 for certain values of Γ.

We emphasize that the purpose of this section is simply to illustrate a potential failure mode which must be addressed by an algorithm provably solving Problem 1.1. In particular, these hard instances are not hard instances for more practical RSVD-based variants which do not truncate.

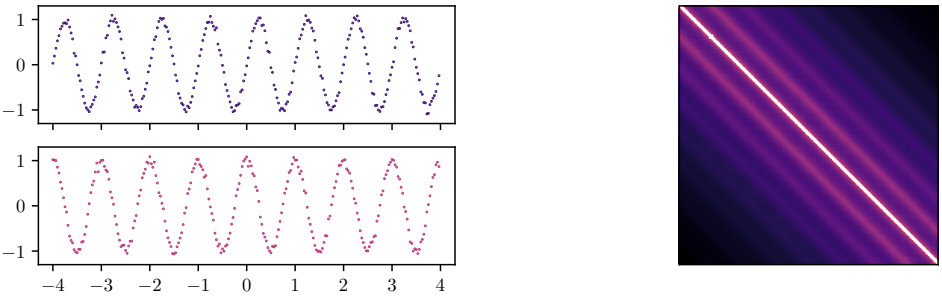

Figure 6: *Top/Bottom Left*: (x_i, y_i) and (x_i, z_i), ordered by x-value. *Right*: Log-magnitude of entries of $\mathbf{A}$ defined in (7.45). In both plots we subsample the data to 256 points for visual clarity.

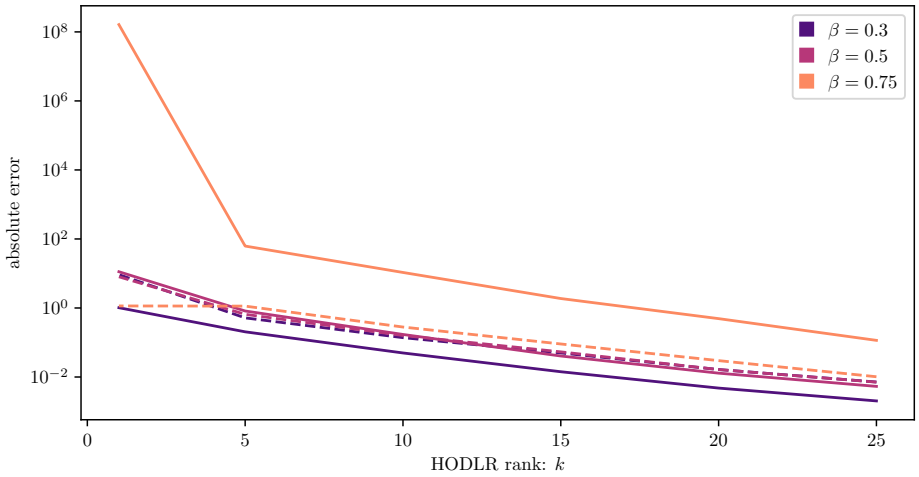

Figure 7: Absolute error of peeling algorithms from Table 1 on the kernel matrix described in (7.45) as a function of the rank-parameter k for several values of β. *Legend*: GN1 (——), RSVD1 (- - -). *Takeaway*: While RSVD1 may not produce near-optimal low-rank approximations, it can sometimes produce good approximations. In addition, when the sketch size used for the regression problem is too large, the Generalized Nyström Method does not produce highly accurate low-rank approximations.

7.3.1 An illustrative example Our first hard instance illustrates that all of the error from a one level can propagate to the next level. This example provides clear intuition for why the Randomized SVD-based peeling algorithm (with sketches of size s_{R} and truncation to rank-k at every level) cannot solve Problem 1.1 for arbitrary $\Gamma > 1$. We emphasize that the hard instance depends on the truncation rank-k used by the algorithm. If we allow the algorithm to use an adaptively chosen rank or do not use truncation (both of which are often done in practice [LLY11; LM24b]) then it would no longer be a hard instance.

Define, for some $\eta \gg 1$,

$$\mathbf{X} = \begin{bmatrix} \mathbf{I}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \qquad \mathbf{Y} = \eta \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_k \end{bmatrix}.$$

Construct the matrix

$$\mathbf{A} = \left[\begin{array}{cc:cc} \mathbf{0} & \mathbf{X} & \mathbf{Y} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \\ \hdashline \mathbf{Y} & \mathbf{X} & \mathbf{0} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \end{array}\right].$$

For notational convenience, we write equality for the limits as $\eta \to \infty$. In particular, when $\eta \to \infty$, Randomized SVD (with any $s_{\text{R}} \geq k$) will recover optimal rank-k approximations to the bottom left and top right blocks.

We then remove the low-rank components we found at the first level to obtain

$$\left[\begin{array}{cc:cc} \mathbf{0} & \mathbf{X} & \mathbf{Y} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \\ \hdashline \mathbf{Y} & \mathbf{X} & \mathbf{0} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \end{array}\right] - \left[\begin{array}{cc:cc} \mathbf{0} & \mathbf{0} & \mathbf{Y} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hdashline \mathbf{Y} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right] = \left[\begin{array}{cc:cc} \mathbf{0} & \mathbf{X} & \mathbf{0} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \\ \hdashline \mathbf{0} & \mathbf{X} & \mathbf{0} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \end{array}\right].$$

Note, however, that since the off-diagonal low-rank blocks of $\mathbf{A}$ are not rank-k, we fail to zero out the off-diagonal blocks at the first level.

At the next level, the randomized SVD (with any $s_{\text{R}} \geq k$) will exactly recover an orthonormal basis $\mathbf{Q}$

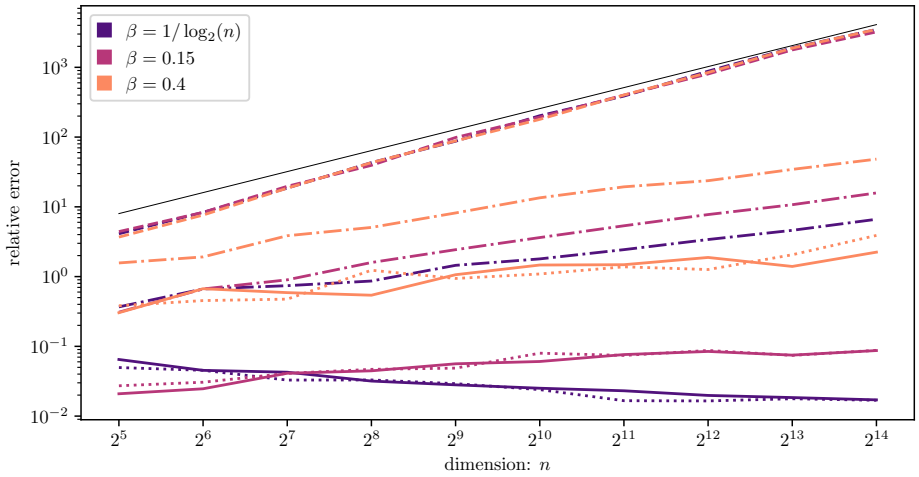

Figure 8: Relative error of peeling algorithms on hard instance described in (7.46) for several values of β. *Legend*: GN1 (——), GN2 (·····), RSVD1 (- - -), RSVD2 (—·-). *Takeaway*: The RSVD1 variant has error growing as n (thin solid reference line); i.e. exponential growth at every level. When the sketch size increases sufficiently quickly with n, the Generalized Nyström Method-based variants produce an error bounded independent of n.

containing the range of $\mathbf{X}$. In particular, we perform a product

$$\left[\begin{array}{cc:cc} \mathbf{0} & \mathbf{X} & \mathbf{0} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \\ \hdashline \mathbf{0} & \mathbf{X} & \mathbf{0} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \end{array}\right] \left[\begin{array}{c} \boldsymbol{\Omega}_1 \\ \mathbf{0} \\ \hdashline \boldsymbol{\Omega}_3 \\ \mathbf{0} \end{array}\right] = \left[\begin{array}{c} \mathbf{0} \\ \mathbf{X}(\boldsymbol{\Omega}_1 + \boldsymbol{\Omega}_3) \\ \hdashline \mathbf{0} \\ \mathbf{X}(\boldsymbol{\Omega}_1 + \boldsymbol{\Omega}_3) \end{array}\right].$$

We then compute[16] $\mathbf{Q} = \operatorname{orth}(\mathbf{X}(\boldsymbol{\Omega}_1 + \boldsymbol{\Omega}_3))$ and

$$\left[\begin{array}{cc:cc} \mathbf{0} & \mathbf{Q}^\mathsf{T} & \mathbf{0} & \mathbf{Q}^\mathsf{T} \end{array}\right] \left[\begin{array}{cc:cc} \mathbf{0} & \mathbf{X} & \mathbf{0} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \\ \hdashline \mathbf{0} & \mathbf{X} & \mathbf{0} & \mathbf{X} \\ \mathbf{X} & \mathbf{0} & \mathbf{X} & \mathbf{0} \end{array}\right] = \left[\begin{array}{cc:cc} 2\mathbf{Q}^\mathsf{T}\mathbf{X} & \mathbf{0} & 2\mathbf{Q}^\mathsf{T}\mathbf{X} & \mathbf{0} \end{array}\right],$$

and analogously for the super-diagonal off-diagonal blocks. Since $\mathbf{Q}\mathbf{Q}^\mathsf{T}\mathbf{X} = \mathbf{X}$, our rank-$k$ approximation to each of the off-diagonal blocks at the second level is $2\mathbf{X}$. In particular, we see that all of the error from the first level propagates to the second level. Even assuming we exactly recover the diagonals (if we do not, this can only increase the error), the final approximation is thus

$$\widetilde{\mathbf{A}} = \left[\begin{array}{cc:cc} \mathbf{0} & 2\mathbf{X} & \mathbf{Y} & \mathbf{0} \\ 2\mathbf{X} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hdashline \mathbf{Y} & \mathbf{0} & \mathbf{0} & 2\mathbf{X} \\ \mathbf{0} & \mathbf{0} & 2\mathbf{X} & \mathbf{0} \end{array}\right].$$

As long as $\eta > 1$, the best HODLR approximation to $\mathbf{A}$ is

$$\mathbf{A}^\star = \left[\begin{array}{cc:cc} \mathbf{0} & \mathbf{X} & \mathbf{Y} & \mathbf{0} \\ \mathbf{X} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hdashline \mathbf{Y} & \mathbf{0} & \mathbf{0} & \mathbf{X} \\ \mathbf{0} & \mathbf{0} & \mathbf{X} & \mathbf{0} \end{array}\right].$$

Therefore we find $\|\mathbf{A} - \mathbf{A}^\star\|_\mathsf{F}^2 = 4\|\mathbf{X}\|_\mathsf{F}^2$ while $\|\mathbf{A} - \widetilde{\mathbf{A}}\|_\mathsf{F}^2 = 8\|\mathbf{X}\|_\mathsf{F}^2$.

[16] Here we assume $\operatorname{orth}(\cdot)$ is a deterministic function.

7.3.2 Exponential growth The example in Section 7.3.1 shows that there are problems for which a Randomized SVD-based peeling algorithm cannot solve Problem 1.1 for small (constant) Γ. We now exhibit a problem instance for a Randomized SVD-based peeling algorithm which suggests that such algorithms cannot even guarantee better than an $O(n)$-factor approximation. This is exponentially large in the number of levels, and to the best of our knowledge, is the first instance demonstrating an exponential instability in the algorithm.

This is the first problem instance we are aware of for which a variant of the peeling algorithm actually incurs a significant propagation of error from level-to-level. It remains an open question whether other variants of the peeling algorithm (e.g., using Randomized SVD without truncation) can fail for the approximation problem.

For any integer $L > 0$ let $n = 2^L$ and define the $n \times n$ matrix $\mathbf{A}_n$ by

$$[\mathbf{A}_n]_{i,j} = \begin{cases} 1 & j = 0,\ i \text{ odd} \\ \eta & j = 1,\ i = 2^1, 2^2, \ldots, 2^L \\ 0 & \text{otherwise} \end{cases}. \tag{7.46}$$

For each of the nonzero off-diagonal blocks, note that the second column is orthogonal to the first column. Hence, the optimal rank-1 HODLR approximation to $\mathbf{A}_n$ is just $\mathbf{A}_n$ with the first column set to zero. This means the error of the optimal approximation is from the $n/2 - 1$ ones in the first column excluding the $(1,1)$ entry on the diagonal.

We set $\eta = 10^8$ and run an implementation of the peeling algorithm using the Randomized SVD with truncation to rank-1 at each level to obtain a HODLR rank-1 matrix $\widehat{\mathbf{A}}_n$. This is repeated for increasing values of n. In Figure 8 we plot (as a function of n) the quantity $\|\mathbf{A}_n - \widehat{\mathbf{A}}_n\|_{\mathsf{F}}/\sqrt{n/2-1} - 1$, which is the smallest value of Γ for which the output $\widehat{\mathbf{A}}_n$ solves Problem 1.1 (averaged over 20 trials). This experiment suggests that $\|\mathbf{A}_n - \widehat{\mathbf{A}}_n\|_{\mathsf{F}}/\sqrt{n/2-1} - 1 = \Theta(n)$ as $n \to \infty$. Since $n = 2^L$, this is exponentially bad (with a constant base 2) in the number of levels L.

8 Outlook

We have presented an algorithm provably solving the HODLR approximation problem in the matrix-vector query model. As far as we can tell, this is the first result on the approximation problem in the matrix-vector query model for any hierarchical matrix family. While we focus on HODLR approximation for clarity of exposition, we expect the ideas used in our analysis can be extended to algorithms for other hierarchical families. The most natural would extension would be to the coloring-based variant of the peeling algorithm for $\mathcal{H}$-matrices introduced in [LM24b]. The $\mathcal{H}$ format is a generalization of HODLR which allows for a more general tree structure and recursive blocks off of the diagonal, and is significantly more efficient than HODLR for multi-dimensional problems.

Our work raises a number of interesting questions on approximation algorithms for hierarchical matrices:

- For any constant $c > 0$, we show that Problem 1.1 can be solved to accuracy $\Gamma = n^c$ with $O(k \log(n/k))$ matrix-vector queries. Up to constants, this is the best possible query complexity; as we prove in Theorem 1.2, exactly recovering a HODLR matrix requires $O(k \log(n/k))$ queries. It remains open whether there exist matvec query algorithms which solve Problem 1.1 to higher accuracy (e.g. $\Gamma = \log(n)$) with $O(k \log(n/k))$ queries.

- Hierarchical Semi-Separable (HSS) matrices are an important subfamily of HODLR matrices. The low-rank factors of HSS matrices at different levels are related in such a way that they can be stored and manipulated more efficiently than general HODLR matrices. In particular, there are a number of algorithms for recovering an exactly HSS matrix that require $O(k)$ matrix-vector products [LM24a; HT23]. It is therefore natural to ask whether there exists an HSS *approximation* algorithm producing $(1+\varepsilon)$-optimal HSS approximation using $O(\text{poly}(k, 1/\varepsilon))$ matvecs. A major difficulty is that, in contrast with HODLR matrices, we are unaware of a simple characterization of the best HSS approximation to a given matrix.

- Operator learning aims to learn representations of operators mapping functions to functions [Lu+21; KLS24; BT23; Li+21; GSBK21]. A number of recent works study the problem of approximating certain classes of infinite dimensional linear operators [BT22; BHT23] by hierarchically structured operators. Understanding how our analysis and theoretical techniques extend to the infinite dimensional setting may yield stronger theoretical guarantees for some problems in operator learning.

- Our structural perturbation bound Theorem 3.1 for low-rank approximation suggests fundamental differences between the behaviors of the Randomized SVD and Generalized Nyström Method in the presence of noisy matrix-vector products. The bound also highlights limitations in our current understanding of the impact of truncation on the Generalized Nyström Method. The best known bounds for the method with truncation are worse than without truncation, but we are unaware of any convincing evidence that such bounds are sharp. Further theoretical and numerical studies of these methods would be of interest.

Acknowledgements

Diana Halikias was supported by the Office of Naval Research (ONR) under grant N00014-23-1-2729. Cameron Musco was partially supported by NSF Grants 2046235 and 2427363. Tyler Chen and Christopher Musco were partially supported by NSF Grants 2045590 and 2427363. David Persson was supported by the SNSF research project *Fast algorithms from low-rank updates*, grant number: 200020_178806.

We also thank Noah Amsel for useful discussions in the early stages of this work.

A A randomized-SVD peeling algorithm

As discussed in Section 7.3, a simple RSVD-based peeling algorithm with truncation cannot perform well, as there is no way of controlling errors made in the projection step. However, by using a similar randomized perforation technique as described in Section 3.3, one can implement an RSVD-based algorithm which can solve Problem 1.1 for arbitrary $\Gamma > 1$. We now define the additional notation needed. The full algorithm is described in Algorithm 4.

At level ℓ, the RSVD-based peeling algorithm will obtain left subspaces $\mathbf{Q}_j^{(\ell)}$ as in Section 4.3. However, rather than solving a regression problem like Algorithm 3, the algorithm will attempt to directly compute $(\mathbf{Q}_j^{(\ell)})^{\mathsf{T}}\mathbf{A}_{j\pm 1,j}^{(\ell)}$. To control the error, we first sample $\boldsymbol{\zeta}^+, \boldsymbol{\zeta}^- \sim \text{PerfCountSketch}(d, t_{\text{R}})$ and define sketching matrices

$$\boldsymbol{\Psi}^+ = \boldsymbol{\xi}^+ \bullet \mathbf{Q}^{(\ell)}, \qquad \boldsymbol{\Psi}^- = \boldsymbol{\xi}^- \bullet \mathbf{Q}^{(\ell)}, \qquad \mathbf{Q}^{(\ell)} := \begin{bmatrix} (\mathbf{Q}_1^{(\ell)})^{\mathsf{T}} & \cdots & (\mathbf{Q}_d^{(\ell)})^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$$

By setting t_{R} large, the expected squared error for each block can be driven arbitrarily small.

As with Algorithm 3, at the final level, we do not need to sketch $\mathbf{A}$. However, to limit the error when trying to perform the projections (onto the identity), we sample $\widehat{\boldsymbol{\xi}} \sim \text{CountSketch}(d, t_{\text{R}})$ and use the sketching matrix

$$\widehat{\boldsymbol{\Psi}} = \widehat{\boldsymbol{\xi}} \bullet (\mathbf{1} \otimes \mathbf{I}).$$

Here $\mathbf{1}$ is the all-ones vector and "$\otimes$" is denotes the Kronecker product so that $\mathbf{1} \otimes \mathbf{I}$ is the stacked identity matrix.

THEOREM A.1. *Fix a rank parameter k and let $\beta \in (0,1)$. Suppose $t_{\text{L}} \geq 1$ and s_{R}, t_{L}, and s_{R} are such that*

$$\frac{k}{s_{\text{R}} - k - 1} \leq \frac{\beta}{10}, \qquad \frac{k}{s_{\text{R}} - k - 1} \cdot \frac{1}{t_{\text{R}}} \leq \frac{\beta^2}{10^2}, \qquad \frac{1}{t_{\text{L}}} \leq \frac{\beta^2}{10^2}.$$

Let $L = \lceil \log_2(n/k) \rceil$. Then, using $2Ls_{\text{R}}t_{\text{R}}$ products with $\mathbf{A}$ and $(2L+1)s_{\text{L}}t_{\text{L}}$ products with $\mathbf{A}^{\mathsf{T}}$, Algorithm 4 outputs a HODLR(k) *matrix $\widetilde{\mathbf{A}}$ such that*

$$\mathbb{E}\left[\|\mathbf{A} - \widetilde{\mathbf{A}}\|_{\mathsf{F}}^2\right] \leq (1+\beta)^{L+1} \cdot \min_{\mathbf{H} \in \text{HODLR}(k)} \|\mathbf{A} - \mathbf{H}\|_{\mathsf{F}}^2.$$

Proof. The proof is analagous to Theorem 5.2.

First, note that in Theorem 3.2 we change the definition of $\mathbf{X}$ to

$$\mathbf{X} := \mathbf{Q}^{\mathsf{T}}\mathbf{B} + \mathbf{E},$$

where $\mathbf{E} \in \mathbb{R}^{s_{\text{L}} \times m_2}$. The resulting bound now has

$$E_2 := \frac{2k}{s_{\text{R}} - k - 1}\|\mathbf{M}\|_{\mathsf{F}}^2 + 2\|\mathbf{E}\|_{\mathsf{F}}^2.$$

Algorithm 4 Randomized SVD Peeling algorithm for HODLR approximation

procedure RANDOMIZEDSVDPEELING($\mathbf{A}, k, s_{\text{L}}, t_{\text{L}}, s_{\text{R}}$)
  Set $L = \lceil \log_2(n/k) \rceil$ ▷ Final level blocks of size at most k
  for $\ell = 1, 2, \ldots, L$ **do**
    Allocate and partition $\mathbf{H}^{(\ell)}$ as in (4.7) ▷ blocks of size $n/2^\ell \times n/2^\ell$
    Sample $\boldsymbol{\xi}^+, \boldsymbol{\xi}^-, \boldsymbol{\Omega}^+, \boldsymbol{\Omega}^- \sim \text{RandPerfGaussian}(n, 2^\ell, s_{\text{R}}, t_{\text{R}})$ ▷ as in Definition 4.5
    Compute $\mathbf{A}^{(\ell)}\boldsymbol{\Omega}^\pm$ ▷ $2s_{\text{R}}s_{\text{R}}$ matvecs with $\mathbf{A}$
    for $j = 1, 2, \ldots, 2^\ell$ **do**
      $\mathbf{Q}_j^{(\ell)} = \text{orth}(\mathbf{Y}_j^{(\ell)})$ ▷ $\mathbf{Y}_j^{(\ell)}$ is $(j \pm 1, \rho)$ block of $\mathbf{A}^{(\ell)}\boldsymbol{\Omega}^\pm$
    Sample $\boldsymbol{\zeta}^+, \boldsymbol{\zeta}^- \sim \text{PerfCountSketch}(d, t_{\text{L}})$ ▷ as in Definition 4.4
    Set $\boldsymbol{\Psi}^\pm = \boldsymbol{\xi}^\pm \bullet \mathbf{Q}^{(\ell)}$
    Compute $(\boldsymbol{\Psi}^\pm)^{\mathsf{T}}\mathbf{A}^{(\ell)}$ ▷ $2s_{\text{L}}t_{\text{L}}$ matvecs with $\mathbf{A}^{\mathsf{T}}$
    for $j = 1, 2, \ldots, 2^\ell$ **do**
      Extract $\mathbf{X}_j^{(\ell)}$ ▷ $\mathbf{X}_j^{(\ell)}$ is (j, σ) block of $(\boldsymbol{\Psi}^\pm)^{\mathsf{T}}\mathbf{A}$
      $\mathbf{H}_j^{(\ell)} = \mathbf{Q}_j^{(\ell)}[\![\mathbf{X}_j^{(\ell)}]\!]_k$ ▷ $\mathbf{H}_j^{(\ell)}$ is $(j \pm 1, j)$-th block of $\mathbf{H}^{(\ell)}$
  Allocate and partition $\widehat{\mathbf{H}}$ as in (4.10) ▷ blocks of size $n/2^L \times n/2^L$
  Sample $\widehat{\boldsymbol{\zeta}} \sim \text{CountSketch}(2^\ell, t_{\text{L}})$ ▷ as in Definition 4.3
  Set $\widehat{\boldsymbol{\Psi}} = \widehat{\boldsymbol{\zeta}} \bullet ([1, \ldots, 1]^{\mathsf{T}} \otimes \mathbf{I})$
  $\widehat{\boldsymbol{\Psi}}^{\mathsf{T}}\widehat{\mathbf{A}}^{(L)} = \mathbf{A}^{\mathsf{T}}\widehat{\boldsymbol{\Psi}} - (\mathbf{H}^{(L)} + \cdots + \mathbf{H}^{(1)})\boldsymbol{\Psi}$ ▷ $s_{\text{L}}t_{\text{L}}$ matvecs with $\mathbf{A}^{\mathsf{T}}$
  for $j = 1, 2, \ldots, 2^L$ **do**
    Extract $\widehat{\mathbf{X}}_j$ ▷ $\widehat{\mathbf{X}}_j$ is (j, σ) block of $\boldsymbol{\Psi}^{\mathsf{T}}\widehat{\mathbf{A}}^{(L)}$
    $\widehat{\mathbf{H}}_j = \widehat{\mathbf{X}}_j$ ▷ $\widehat{\mathbf{H}}_j$ is (j, j)-th block of $\widehat{\mathbf{H}}$
  return $\widetilde{\mathbf{A}} = \mathbf{H}^{(1)} + \cdots + \mathbf{H}^{(L)} + \widehat{\mathbf{H}}$

Next, in Lemma 5.2 we now plug in $\mathbf{E} := (\mathbf{Q}^{(\ell)})^{\mathsf{T}}\mathbf{N}$ and note that $\|\mathbf{E}\|_{\mathsf{F}}^2 \le \|\mathbf{N}\|_{\mathsf{F}}^2$. The bound (5.28) therefore becomes

$$\mathbb{E}\left[\|\mathbf{A}_{j\pm1,j}^{(\ell)} - \mathbf{Q}_j^{(\ell)}[\![\mathbf{X}_j^{(\ell)}]\!]_k\|_{\mathsf{F}}^2 \middle| \mathcal{F}_\ell, \boldsymbol{\xi}, \boldsymbol{\zeta}\right] \le E_1 + E_2 + 2\sqrt{E_1 E_2}, \tag{A.1}$$

where E_1 is as before and

$$E_2 := \frac{2k}{s_{\text{R}} - k - 1}\|\mathbf{M}\|_{\mathsf{F}}^2 + 2\|\mathbf{N}\|_{\mathsf{F}}^2.$$

Therefore, we get

$$E_2' := \frac{2k}{s_{\text{R}} - k - 1} \cdot \frac{1}{t_{\text{R}}} \sum_{\substack{i=1 \\ i \ne j, j\pm1}}^{d} \|\mathbf{A}_{j\pm1,i}^{(\ell)}\|_{\mathsf{F}}^2 + \frac{2}{t_{\text{L}}} \sum_{\substack{i=1 \\ i \ne j, j\pm1}}^{d} \|\mathbf{A}_{i,j}^{(\ell)}\|_{\mathsf{F}}^2.$$

and our assumptions on t_{L}, t_{R}, and s_{L} allow us to bound

$$\frac{k}{s_{\text{R}} - k - 1}\|\mathbf{A}_{j\pm1,j}^{(\ell)} - [\![\mathbf{A}_{j\pm1,j}^{(\ell)}]\!]_k\|_{\mathsf{F}}^2 \le \frac{\beta}{10}E_j^{(\ell)},$$

$$\frac{2k}{s_{\text{R}} - k - 1} \cdot \frac{1}{t_{\text{R}}} \sum_{\substack{i=1 \\ i \ne j, j\pm1}}^{d} \|\mathbf{A}_{j\pm1,i}^{(\ell)}\|_{\mathsf{F}}^2 \le 4\frac{\beta^2}{10^2}E_j^{(\ell)}, \qquad \frac{2}{t_{\text{L}}} \sum_{\substack{i=1 \\ i \ne j, j\pm1}}^{d} \|\mathbf{A}_{i,j}^{(\ell)}\|_{\mathsf{F}}^2 \le 4\frac{\beta^2}{10^2}E_j^{(\ell)}.$$

Finally, since $1/10 + (4+4)/10^2 + 2\sqrt{2(4+4)/10^2} \approx 0.980 \le 1$ we get the desired bound. ∎

References

[AD13] S. Ambikasaran and E. Darve. An $\mathcal{O}(N \log N)$ Fast Direct Solver for Partial Hierarchically Semi-Separable Matrices: With Application to Radial Basis Function Interpolation. In: *Journal of Scientific Computing* 57.3 (2013), pp. 477–501 (cited on page 1).

[Ams+24] N. Amsel, T. Chen, F. D. Keles, D. Halikias, C. Musco, and C. Musco. Fixed-sparsity matrix approximation from matrix-vector products. 2024. arXiv: 2402.09379 [cs.DS] (cited on pages 2, 4, 23, 24).

[Ask+] T. Askham, A. Barnett, Z. Gimbutas, L. Greengard, L. Lu, J. Magland, D. Malhotra, M. Rachh, V. Rokhlin, M. O'Neil, and F. Vico. Flatiron institute fast multipole libraries. https://github.com/flatironinstitute/FMM3D (cited on page 27).

[BCM17] D. Bertsimas, M. S. Copenhaver, and R. Mazumder. Certifiably optimal low rank factor analysis. In: *Journal of Machine Learning Research* 18.1 (2017), pp. 907–959 (cited on page 2).

[BCW22] A. Bakshi, K. L. Clarkson, and D. P. Woodruff. Low-Rank Approximation with $1/\epsilon^{1/3}$ Matrix-Vector Products. In: *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*. STOC 2022. Rome, Italy: Association for Computing Machinery, 2022, pp. 1130–1143 (cited on pages 2–4).

[BET22] N. Boullé, C. J. Earls, and A. Townsend. Data-driven discovery of Green's functions with human-understandable deep learning. In: *Sci. Rep.* 12.1 (2022), p. 4824 (cited on pages 2, 3).

[BGH03] S. Börm, L. Grasedyck, and W. Hackbusch. Introduction to hierarchical matrices with applications. In: *Engineering Analysis with Boundary Elements* 27.5 (2003), pp. 405–422 (cited on page 1).

[BH03] M. Bebendorf and W. Hackbusch. Existence of $\mathcal{H}$-matrix approximants to the inverse FE-matrix of elliptic operators with L^∞-coefficients. In: *Numer. Math.* 95.1 (2003), pp. 1–28 (cited on page 7).

[BHOT24] N. Boullé, D. Halikias, S. E. Otto, and A. Townsend. Operator learning without the adjoint. 2024. arXiv: 2401.17739 [math.NA] (cited on page 2).

[BHSW20] M. Braverman, E. Hazan, M. Simchowitz, and B. Woodworth. The Gradient Complexity of Linear Regression. In: *Proceedings of Thirty Third Conference on Learning Theory*. Vol. 125. Proceedings of Machine Learning Research. PMLR, 2020, pp. 627–647 (cited on pages 2, 4, 24).

[BHT23] N. Boullé, D. Halikias, and A. Townsend. Elliptic PDE learning is provably data-efficient. In: *Proceedings of the National Academy of Sciences* 120.39 (2023) (cited on pages 2, 7, 30).

[BK16] J. Ballani and D. Kressner. Matrices with Hierarchical Low-Rank Structures. In: *Exploiting Hidden Structure in Matrix Computations: Algorithms and Applications*. Springer International Publishing, 2016, pp. 161–209 (cited on pages 1, 2).

[BKS07] C. Bekas, E. Kokiopoulou, and Y. Saad. An estimator for the diagonal of a matrix. In: *Applied Numerical Mathematics* 57.11–12 (2007), pp. 1214–1229 (cited on page 2).

[BN22] R. A. Baston and Y. Nakatsukasa. Stochastic diagonal estimation: probabilistic bounds and an improved algorithm. 2022. arXiv: 2201.10684 [cs.DS] (cited on page 2).

[BN23] A. Bakshi and S. Narayanan. Krylov Methods are (nearly) Optimal for Low-Rank Approximation. In: *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2023 (cited on pages 2, 3, 23).

[BT22] N. Boullé and A. Townsend. Learning Elliptic Partial Differential Equations with Randomized Linear Algebra. In: *Foundations of Computational Mathematics* 23.2 (2022), pp. 709–739 (cited on pages 2, 30).

[BT23] N. Boullé and A. Townsend. A Mathematical Guide to Operator Learning. 2023. arXiv: 2312.14688 [math.NA] (cited on pages 2, 30).

[BTK19] W. Boukaram, G. Turkiyyah, and D. Keyes. Hierarchical Matrix Operations on GPUs: Matrix-Vector Multiplication and Compression. In: *ACM Transactions on Mathematical Software* 45.1 (2019), pp. 1–28 (cited on pages 2, 3).

[CC86] T. F. Coleman and J.-Y. Cai. The Cyclic Coloring Problem and Estimation of Sparse Hessian Matrices. In: *SIAM Journal on Algebraic Discrete Methods* 7.2 (1986), pp. 221–235 (cited on page 2).

[Cha+07] S. Chandrasekaran, P. Dewilde, M. Gu, W. Lyons, and T. Pals. A Fast Solver for HSS Representations via Sparse Matrices. In: *SIAM Journal on Matrix Analysis and Applications* 29.1 (2007), pp. 67–81 (cited on page 1).

[Che+23] S. Chewi, J. De Dios Pont, J. Li, C. Lu, and S. Narayanan. Query lower bounds for log-concave sampling. In: *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2023 (cited on page 4).

[Che+24] S. Chewi, J. de Dios Pont, J. Li, C. Lu, and S. Narayanan. Query lower bounds for log-concave sampling. In: *J. ACM* (2024) (cited on page 2).

[CM83] T. F. Coleman and J. J. Moré. Estimation of Sparse Jacobian Matrices and Graph Coloring Blems. In: *SIAM Journal on Numerical Analysis* 20.1 (1983), pp. 187–209 (cited on page 2).

[Coh+15] M. Cohen, S. Elder, C. Musco, C. Musco, and M. Persu. Dimensionality reduction for k-means clustering and low rank approximation. In: *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*. 2015, pp. 163–172 (cited on page 3).

[Con23] M. Connolly. Probabilistic rounding error analysis for numerical linear algebra. Available at `https://research.manchester.ac.uk/en/studentTheses/probabilistic-rounding-error-analysis-for-numerical-linear-algebr`. PhD thesis. University of Manchester, 2023 (cited on page 15).

[CPR74] A. R. Curtis, M. J. D. Powell, and J. K. Reid. On the Estimation of Sparse Jacobian Matrices. In: *IMA Journal of Applied Mathematics* 13.1 (1974), pp. 117–119 (cited on page 2).

[CW09] K. L. Clarkson and D. P. Woodruff. Numerical linear algebra in the streaming model. In: *Proceedings of the forty-first annual ACM symposium on Theory of computing*. STOC '09. ACM, 2009 (cited on pages 3, 4).

[CW13] K. L. Clarkson and D. P. Woodruff. Low Rank Approximation and Regression in Input Sparsity Time. In: *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*. 2013, pp. 81–90 (cited on page 1).

[DKM06] P. Drineas, R. Kannan, and M. W. Mahoney. Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. In: *SIAM Journal on Computing* 36.1 (2006), pp. 158–183 (cited on page 1).

[DM05] P. Drineas and M. W. Mahoney. On the Nyström method for approximating a Gram matrix for improved kernel-based learning. In: *Journal of Machine Learning Research* 6 (2005), pp. 2153–2175 (cited on page 1).

[DM21] P. Dharangutte and C. Musco. Dynamic trace estimation. In: *Advances in Neural Information Processing Systems 34 (NeurIPS)*. 2021 (cited on page 2).

[DM23] P. Dharangutte and C. Musco. "A Tight Analysis of Hutchinson's Diagonal Estimator". In: *Symposium on Simplicity in Algorithms (SOSA)*. Society for Industrial and Applied Mathematics, 2023, pp. 353–364 (cited on page 2).

[DSBN15] G. Dasarathy, P. Shah, B. N. Bhaskar, and R. D. Nowak. Sketching Sparse Matrices, Covariances, and Graphs via Tensor Products. In: *IEEE Transactions on Information Theory* 61.3 (2015), pp. 1373–1388 (cited on page 2).

[GR87] L. Greengard and V. Rokhlin. A Fast Algorithm for Particle Simulations. In: *Journal of Computational Physics* 135.2 (1987), pp. 280–292 (cited on pages 1, 26).

[GSBK21] C. Gin, D. Shea, S. Brunton, and J. Kutz. DeepGreen: deep learning of Green's functions for nonlinear boundary value problems. In: *Scientific Reports* 11 (2021) (cited on page 30).

[Gu15] M. Gu. Subspace iteration randomization and singular value problems. In: *SIAM J. Sci. Comput.* 37.3 (2015), A1139–A1173 (cited on page 15).

[Hac15] W. Hackbusch. Hierarchical Matrices: Algorithms and Analysis. Springer Publishing Company, Incorporated, 2015 (cited on page 1).

[HMT11] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. In: *SIAM Review* 53.2 (2011), pp. 217–288 (cited on pages 1, 3, 4, 15, 25).

[HT23] D. Halikias and A. Townsend. Structured matrix recovery from matrix-vector products. In: *Numerical Linear Algebra with Applications* 31.1 (2023) (cited on pages 2, 5, 22, 23, 30).

[JPWZ24] S. Jiang, H. Pham, D. P. Woodruff, and Q. Zhang. Optimal sketching for trace estimation. In: *Advances in Neural Information Processing Systems 37 (NeurIPS)*. 2024 (cited on page 2).

[KLS24] N. B. Kovachki, S. Lanthaler, and A. M. Stuart. Operator Learning: Algorithms and Analysis. 2024. arXiv: 2402.15715 [cs.LG] (cited on page 30).

[Li+21] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. In: *International Conference on Learning Representations*. 2021 (cited on page 30).

[LLY11] L. Lin, J. Lu, and L. Ying. Fast construction of hierarchical matrix representation from matrix–vector multiplication. In: *Journal of Computational Physics* 230.10 (2011), pp. 4071–4087 (cited on pages 2, 3, 5, 6, 25, 28).

[LM24a] J. Levitt and P.-G. Martinsson. Linear-Complexity Black-Box Randomized Compression of Rank-Structured Matrices. In: *SIAM Journal on Scientific Computing* 46.3 (2024), A1747–A1763 (cited on pages 2, 30).

[LM24b] J. Levitt and P.-G. Martinsson. Randomized compression of rank-structured matrices accelerated with graph coloring. In: *Journal of Computational and Applied Mathematics* 451 (2024), p. 116044 (cited on pages 2, 5, 28, 30).

[Lu+21] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. In: *Nature Machine Intelligence* 3 (2021), pp. 218–229 (cited on page 30).

[Mar11] P. G. Martinsson. A Fast Randomized Algorithm for Computing a Hierarchically Semiseparable Representation of a Matrix. In: *SIAM Journal on Matrix Analysis and Applications* 32.4 (2011), pp. 1251–1274 (cited on page 2).

[Mar16] P.-G. Martinsson. Compressing Rank-Structured Matrices via Randomized Sampling. In: *SIAM Journal on Scientific Computing* 38.4 (2016), A1959–A1986 (cited on pages 2, 25).

[MM15] C. Musco and C. Musco. Randomized Block Krylov Methods for Stronger and Faster Approximate Singular Value Decomposition. In: *Advances in Neural Information Processing Systems 28 (NeurIPS)*. 2015, pp. 1396–1404 (cited on page 1).

[MM17] C. Musco and C. Musco. Recursive Sampling for the Nyström Method. In: *Advances in Neural Information Processing Systems 30 (NeurIPS)*. 2017, pp. 3833–3845 (cited on page 1).

[MMM24] R. Meyer, C. Musco, and C. Musco. On the Unreasonable Effectiveness of Single Vector Krylov Methods for Low-Rank Approximation. In: *Proceedings of the 35th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2024 (cited on page 4).

[MMMW21] R. A. Meyer, C. Musco, C. Musco, and D. Woodruff. Hutch++: Optimal Stochastic Trace Estimation. In: *Proceedings of the 4th Symposium on Simplicity in Algorithms (SOSA)* (2021) (cited on page 2).

[MMW21] C. Musco, C. Musco, and D. P. Woodruff. Simple Heuristics Yield Provable Algorithms for Masked Low-Rank Approximation. In: *Proceedings of the 12th Conference on Innovations in Theoretical Computer Science (ITCS)*. 2021, 6:1–6:20 (cited on page 2).

[Nak20] Y. Nakatsukasa. Fast and stable randomized low-rank matrix approximation. 2020. arXiv: 2009.11392 [math.NA] (cited on pages 3, 4).

[NN13] J. Nelson and H. L. Nguyen. OSNAP: Faster Numerical Linear Algebra Algorithms via Sparser Subspace Embeddings. In: *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 2013, pp. 117–126 (cited on page 1).

[OX22] X. Ou and J. Xia. SuperDC: Superfast Divide-And-Conquer Eigenvalue Decomposition With Improved Stability for Rank-Structured Matrices. In: *SIAM Journal on Scientific Computing* 44.5 (2022), A3041–A3066 (cited on page 1).

[PBK24] D. Persson, N. Boullé, and D. Kressner. Randomized Nyström approximation of non-negative self-adjoint operators. 2024. arXiv: 2404.00960 [math.NA] (cited on page 16).

[RST09] V. Rokhlin, A. Szlam, and M. Tygert. A Randomized Algorithm for Principal Component Analysis. In: *SIAM Journal on Matrix Analysis and Applications* 31.3 (2009), pp. 1100–1124 (cited on page 3).

[Sar06] T. Sarlos. Improved approximation algorithms for large matrices via random projections. In: *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 2006, pp. 143–152 (cited on page 5).

[SER18] M. Simchowitz, A. El Alaoui, and B. Recht. Tight query complexity lower bounds for PCA via finite sample deformed wigner law. In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. STOC '18. ACM, 2018 (cited on pages 2, 4).

[SKO21] F. Schäfer, M. Katzfuss, and H. Owhadi. Sparse Cholesky Factorization by Kullback–Leibler Minimization. In: *SIAM Journal on Scientific Computing* 43.3 (2021), A2019–A2046 (cited on page 2).

[SO24] F. Schäfer and H. Owhadi. Sparse Recovery of Elliptic Solvers from Matrix-Vector Products. In: *SIAM Journal on Scientific Computing* 46.2 (2024), A998–A1025 (cited on page 2).

[SWYZ21] X. Sun, D. P. Woodruff, G. Yang, and J. Zhang. Querying a Matrix through Matrix-Vector Products. In: *ACM Trans. Algorithms* 17.4 (2021) (cited on page 2).

[TS11] J. M. Tang and Y. Saad. A probing method for computing the diagonal of a matrix inverse. In: *Numerical Linear Algebra with Applications* 19.3 (2011), pp. 485–501 (cited on page 2).

[TVX23] L. Tunçel, S. A. Vavasis, and J. Xu. Computational Complexity of Decomposing a Symmetric Matrix as a Sum of Positive Semidefinite and Diagonal Matrices. In: *Foundations of Computational Mathematics* (2023) (cited on page 2).

[TW23] J. A. Tropp and R. J. Webber. Randomized algorithms for low-rank matrix approximation: Design, analysis, and applications. 2023. arXiv: 2306.12418 [math.NA] (cited on page 3).

[TYUC16] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher. Randomized single-view algorithms for low-rank matrix approximation. 2016. arXiv: 1609.00048v1 [cs.NA] (cited on pages 5, 8).

[TYUC17] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher. Practical Sketching Algorithms for Low-Rank Matrix Approximation. In: *SIAM Journal on Matrix Analysis and Applications* 38.4 (2017), pp. 1454–1485 (cited on pages 3, 4, 25).

[VXCB16] J. Vogel, J. Xia, S. Cauley, and V. Balakrishnan. Superfast Divide-and-Conquer Method and Perturbation Analysis for Structured Eigenvalue Solutions. In: *SIAM Journal on Scientific Computing* 38.3 (2016), A1358–A1382 (cited on page 1).

[WEB24] H. Wilber, E. N. Epperly, and A. H. Barnett. A superfast direct inversion method for the nonuniform discrete Fourier transform. 2024. arXiv: 2404.13223 [math.NA] (cited on page 1).

[WEV13] T. Wimalajeewa, Y. C. Eldar, and P. K. Varshney. Recovery of Sparse Matrices via Matrix Sketching. 2013. arXiv: 1311.2448 [cs.NA] (cited on page 2).

[Woo14] D. P. Woodruff. Sketching as a Tool for Numerical Linear Algebra. In: *Foundations and Trends in Theoretical Computer Science* 10.1–2 (2014), pp. 1–157 (cited on pages 1, 4).

[WSB11] A. Waters, A. Sankaranarayanan, and R. Baraniuk. SpaRCS: Recovering low-rank and sparse matrices from compressive measurements. In: *Advances in Neural Information Processing Systems*. Vol. 24. Curran Associates, Inc., 2011 (cited on page 2).

[YBZ04] L. Ying, G. Biros, and D. Zorin. A kernel-independent adaptive fast multipole algorithm in two and three dimensions. In: *Journal of Computational Physics* 196.2 (2004), pp. 591–626 (cited on page 1).

[YDGD03] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis. Improved fast Gauss transform and efficient kernel density estimation. In: *Proceedings of the 9th IEEE International Conference on Computer Vision*. 2003, p. 464 (cited on page 1).

[YIK17] R. Yokota, H. Ibeid, and D. Keyes. Fast Multipole Method as a Matrix-Free Hierarchical Low-Rank Approximation. In: *Eigenvalue Problems: Algorithms, Software and Applications in Petascale Computing*. 2017, pp. 267–286 (cited on page 1).