Matrix-free PCG for an RKHS-constrained CP mode update with missing tensor entries

Placeholder Author

February 14, 2026

Abstract

We study the mode-k subproblem arising in alternating optimization for CP decomposition of partially observed d-way tensors when one mode is constrained to an RKHS. With all other factor matrices fixed, the RKHS factor is parameterized by a kernel matrix times an unknown coefficient matrix, which leads to a symmetric positive definite linear system of dimension n r. Direct formation and factorization are infeasible when the tensor is unaligned and only q entries are observed, with q much smaller than the total number of tensor entries. We outline a matrix-free preconditioned conjugate gradient method that evaluates all operator actions using only the observed-entry index set, Khatri-Rao structure, and kernel matrix-vector products, avoiding any computation that scales with the full tensor size. We also propose Kronecker-consistent block preconditioners based on small r by r Gram matrices and kernel solves, and we provide complexity bounds in the regime n, r $<$ q much smaller than the full tensor size.

Contents

1 Introduction

CP (CANDECOMP/PARAFAC) decompositions are a standard tool for representing multiway data by low-rank factors, but many modern applications face two difficulties simultaneously: the tensor is only partially observed ("unaligned" data), and at least one mode is more naturally modeled as a smooth function rather than an unconstrained finite vector. A common way to encode smoothness is to constrain the corresponding factor to lie in a Reproducing Kernel Hilbert Space (RKHS), yielding kernel-based regularization in that mode. When CP models are fit by alternating optimization, each step reduces to a mode-wise least-squares problem. For an RKHS-constrained mode, this update becomes a kernelized linear system whose size grows with the mode dimension times the rank.

This paper focuses on the mode-k update for a rank-r CP model of a d-way tensor when only $q \ll N = \prod_{\ell=1}^{d} n_\ell$ entries are observed. Writing the mode-k unfolding as $T \in \mathbb{R}^{n \times M}$ with $n = n_k$ and $M = \prod_{\ell \neq k} n_\ell$, and collecting the fixed factors in the remaining modes into the Khatri–Rao matrix $Z \in \mathbb{R}^{M \times r}$, the masked least-squares objective leads to normal equations that can be expressed compactly with a selection operator $S \in \mathbb{R}^{N \times q}$ that extracts observed entries from $\mathrm{vec}(T)$. Under the representer parametrization $A_k = KW$ with kernel Gram matrix $K \in \mathbb{R}^{n \times n}$ and unknown coefficients $W \in \mathbb{R}^{n \times r}$, the mode-$k$ subproblem yields a linear system in $\mathrm{vec}(W)$ of dimension nr.

A direct approach—forming the $nr \times nr$ normal-equations matrix and applying a dense factorization—is unattractive in the target regime $n, r < q \ll N$. The obstacle is not only the cubic scaling in nr, but also the fact that the data-fit term involves the masking pattern through $SS^\top$, so explicit formation would entail operations that conceptually range over all N unfolding entries. The central aim here is therefore algorithmic: to show how to solve the RKHS mode update by an iterative Krylov method that (i) never forms the $nr \times nr$ matrix, (ii) evaluates each operator action using only the q observed entries, and (iii) admits effective preconditioning that respects the Kronecker and kernel structure.

Our main contribution is a matrix-free PCG formulation for the RKHS mode-k normal equations. First, we rewrite the normal operator so that its application to a candidate W can be implemented via observed-entry residual computations: the action of $(Z \otimes K)^\top SS^\top (Z \otimes K)$ is realized by restricting $A_k Z^\top$ to the observed index set, multiplying by the corresponding rows of Z, and accumulating back into an $n \times r$ array, with kernel multiplications by K applied only along the n dimension. Second, we advocate Kronecker-consistent preconditioners that are inexpensive to apply and align with the RKHS regularization term, with the baseline choice $P = I_r \otimes K$ reducing preconditioning to r kernel solves. Finally, we provide complexity estimates showing that each PCG iteration can be carried out in time scaling with q, n, r, and the cost of accessing rows of Z, with no operations of order N.

Organization. Section 2 states the mode-k RKHS subproblem and derives the associated normal equations under missing entries. The next section develops a matrix-free implementation of the normal operator and the right-hand side based on observed-entry indexing and Khatri–Rao structure. We then describe the PCG algorithm and discuss Kronecker-structured preconditioners, including the baseline $I_r \otimes K$ choice and practical variants. The final section summarizes computational complexity and outlines alternative preconditioning strategies when explicit factorizations of K are unavailable or too costly.

2 Problem setup and RKHS mode subproblem

This chapter specifies the mode-k subproblem that appears in alternating optimization for a CP decomposition with missing entries, under an RKHS constraint on the kth mode factor. The objective is to derive a symmetric linear system for the coefficient matrix $W \in \mathbb{R}^{n \times r}$ in the parametrization $A_k = KW$, in a form suitable for matrix-free iterative solvers. Throughout, we avoid forming arrays of size $N = \prod_{\ell=1}^{d} n_\ell$ except as a conceptual indexing device.

Roadmap. We first define the missing-entry observation model, the mode-k unfolding, and the selection operator that extracts the observed entries. We then describe the RKHS parametrization and explain why the finite coefficient matrix W suffices even when the underlying mode is infinite-dimensional. Finally, we derive the normal equations and record basic symmetry and definiteness properties that justify conjugate gradients, together with remarks on the computational regime and on preconditioner choices used later.

2.1 Problem setup and dimensional conventions

Let $\mathcal{T} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ be a d-way tensor, of which only a subset of entries is observed. The observation index set is $\Omega \subseteq [n_1] \times \cdots \times [n_d]$ with cardinality $q := |\Omega| \ll N$, where $N := \prod_{\ell=1}^{d} n_\ell$. We consider a rank-$r$ CP model

$$\widehat{\mathcal{T}}(i_1, \ldots, i_d) \;=\; \sum_{j=1}^{r} \prod_{\ell=1}^{d} A_\ell(i_\ell, j). \tag{1}$$

In the broader modeling goal, some modes may correspond to functions in infinite-dimensional spaces. For the mode-k subproblem treated here, we assume that the factors in modes $\ell \neq k$ are fixed and represented by finite matrices $A_\ell \in \mathbb{R}^{n_\ell \times r}$ with finite n_ℓ. This makes the Khatri-Rao product across $\ell \neq k$ well-defined as a finite matrix. Mode k is the mode in which we enforce smoothness or structural constraints through an RKHS penalty. Let $k \in \{1, \ldots, d\}$ be fixed, let $n := n_k$, and define

$$M := \prod_{\ell \neq k} n_\ell, \qquad N = nM. \tag{2}$$

Let $T \in \mathbb{R}^{n \times M}$ denote the mode-k unfolding of $\mathcal{T}$. We adopt the concrete convention that unobserved entries are set to zero, namely $T = P_\Omega(\mathcal{T})$ under the unfolding, where P_Ω keeps observed entries and sets all other entries to zero.

Unfolding convention and indexing. Each tensor multi-index $\omega = (i_1, \ldots, i_d)$ corresponds to an unfolding coordinate (i, m) where $i = i_k \in [n]$ and $m \in [M]$ encodes the remaining indices. Let $(j_1, \ldots, j_{d-1})$ be the increasing list of indices in $\{1, \ldots, d\} \setminus \{k\}$, define $\tilde{i}_\ell := i_{j_\ell}$ and $\tilde{n}_\ell := n_{j_\ell}$, and set

$$m(\tilde{i}_1, \ldots, \tilde{i}_{d-1}) \;:=\; 1 + \sum_{\ell=1}^{d-1} (\tilde{i}_\ell - 1) \prod_{p=1}^{\ell-1} \tilde{n}_p. \tag{3}$$

We use column-wise vectorization $\mathrm{vec}(\cdot)$, so that $\mathrm{vec}(T) \in \mathbb{R}^N$ stacks the columns of T. The global linear index corresponding to (i, m) is

$$\pi(i, m) := i + (m - 1)n, \qquad i \in [n],\ m \in [M]. \tag{4}$$

Fixed factors in the other modes. Given fixed $\{A_\ell\}_{\ell \neq k}$, define the Khatri-Rao product

$$Z := A_d \odot \cdots \odot A_{k+1} \odot A_{k-1} \odot \cdots \odot A_1 \in \mathbb{R}^{M \times r}. \tag{5}$$

The order in (5) matches the unfolding convention in (3) so that the standard vectorization identity holds. With this definition, the mode-k unfolding of the CP reconstruction is

$$\widehat{T} = A_k Z^\top \in \mathbb{R}^{n \times M}. \tag{6}$$

The associated MTTKRP matrix is

$$B := TZ \in \mathbb{R}^{n \times r}. \tag{7}$$

Although (7) is written as a dense matrix product, it can be computed from the observations without forming dense T, as shown in (18).

2.2 Observation model and selection operator

Enumerate the observed tensor entries as $\{\omega_s\}_{s=1}^q$ with $\omega_s = (i_{1,s}, \ldots, i_{d,s}) \in \Omega$. Under the unfolding, each ω_s corresponds to a unique coordinate (i_s, m_s) where $i_s = i_{k,s}$ and m_s is computed from the other indices via (3). Define a selection matrix $S \in \mathbb{R}^{N \times q}$ whose sth column is $e_{\pi(i_s, m_s)} \in \mathbb{R}^N$, where e_j denotes the jth standard basis vector. For any $X \in \mathbb{R}^{n \times M}$, the vector $S^\top \mathrm{vec}(X) \in \mathbb{R}^q$ extracts the observed entries $\{X(i_s, m_s)\}_{s=1}^q$.

Uniqueness of observed indices and the form of $SS^\top$. When the observed unfolding indices $\{\pi(i_s, m_s)\}_{s=1}^q$ are all distinct, $SS^\top$ is a diagonal projector onto observed coordinates. In this case, for any $y \in \mathbb{R}^N$,

$$\|S^\top y\|_2^2 = y^\top SS^\top y. \tag{8}$$

When duplicates occur, $SS^\top$ remains diagonal but is not idempotent; it becomes a diagonal weight matrix whose diagonal entries count multiplicities of repeated observations at the same unfolding coordinate. The identity (8) remains valid as stated, and the normal equations below correspond to a (possibly) multiplicity-weighted least-squares fit. For later algebraic manipulations, let $z_m \in \mathbb{R}^{1\times r}$ denote the mth row of Z, and let $z_m^\top \in \mathbb{R}^r$ be its transpose. For each observation s, define the diagonal matrix

$$D_s \coloneqq \operatorname{diag}(z_{m_s}^\top) \in \mathbb{R}^{r\times r}. \tag{9}$$

The definition (9) records that each observation couples the r components through the same length-r vector derived from the fixed factors.

2.3 RKHS parametrization for mode k

Let $K \in \mathbb{R}^{n\times n}$ be a symmetric positive semidefinite kernel Gram matrix over the n discrete indices in mode k. We parametrize

$$A_k = KW, \qquad W \in \mathbb{R}^{n\times r} \text{ unknown.} \tag{10}$$

The quadratic regularizer is $\operatorname{tr}(W^\top KW) = \sum_{j=1}^r w_j^\top K w_j$, where w_j denotes the jth column of W.

Connection to an infinite-dimensional RKHS. Suppose mode k corresponds to evaluations at points $\{x_i\}_{i=1}^n$ in a domain, with kernel function $\kappa(\cdot,\cdot)$. For each CP component j, the mode-k factor column can be interpreted as evaluations $[f_j(x_1), \ldots, f_j(x_n)]^\top$ for some f_j in the RKHS $\mathcal{H}$. The standard representer theorem argument for penalized least squares yields the existence of minimizers of the form $f_j(\cdot) = \sum_{i=1}^n w_j(i)\,\kappa(x_i, \cdot)$, which implies $[f_j(x_1), \ldots, f_j(x_n)]^\top = Kw_j$ with $K_{ab} = \kappa(x_a, x_b)$. Thus the finite-dimensional parametrization (10) is consistent with the infinite-dimensional formulation in $\mathcal{H}$.

Subset observations and over-parameterization. Let $\mathcal{I}_\Omega \subseteq [n]$ be the set of mode-k indices that appear in at least one observation. The data term depends on each f_j only through evaluations on $\mathcal{I}_\Omega$. Consequently, there exists at least one minimizer whose representer expansion uses only centers $\{x_i : i \in \mathcal{I}_\Omega\}$. The parametrization (10) uses all n kernel sections and can be over-parameterized when K is singular. To make this precise, write each column of W as $w_j = w_{j,\text{range}} + w_{j,\text{null}}$ with $w_{j,\text{null}} \in \operatorname{null}(K)$. Then $Kw_j = Kw_{j,\text{range}}$ and $w_j^\top Kw_j = w_{j,\text{range}}^\top Kw_{j,\text{range}}$, so both the prediction $A_k = KW$ and the penalty depend only on the range component. Hence, while W may not be unique when K is only semidefinite, all minimizers induce the same A_k and the same objective value. A standard way to select a canonical representative is to restrict to $w_j \in \operatorname{range}(K)$, or to add a small stabilization $K \leftarrow K + \delta I$ with $\delta > 0$ in numerical linear algebra steps.

Assumption (strict definiteness for Euclidean PCG). In subsequent solver development, we focus on the setting $\lambda > 0$ and $K \succ 0$, which ensures strict positive definiteness of the linear operator in (17) and allows the use of conjugate gradients in the standard Euclidean inner product. When K is semidefinite, one can work in an appropriate quotient space or use stabilization, but that analysis is not needed for the algebraic derivation in this chapter.

2.4 Mode-k subproblem and normal equations

With $\{A_\ell\}_{\ell\neq k}$ fixed, equivalently with Z fixed, the mode-k update is defined by the penalized observed-entry least-squares problem

$$\min_{W\in\mathbb{R}^{n\times r}} \frac{1}{2}\left\|S^\top\Big(\operatorname{vec}(KWZ^\top) - \operatorname{vec}(T)\Big)\right\|_2^2 + \frac{\lambda}{2}\operatorname{tr}(W^\top KW), \tag{11}$$

where $\lambda > 0$. Using the vectorization identity

$$\operatorname{vec}(KWZ^\top) = (Z \otimes K)\operatorname{vec}(W), \tag{12}$$

define $w \coloneqq \mathrm{vec}(W) \in \mathbb{R}^{nr}$ and $t \coloneqq \mathrm{vec}(T) \in \mathbb{R}^N$. Then (11) becomes

$$\min_{w \in \mathbb{R}^{nr}} \frac{1}{2} \left\| S^\top \Big((Z \otimes K) w - t \Big) \right\|_2^2 + \frac{\lambda}{2} \, w^\top (I_r \otimes K) w. \tag{13}$$

Using (8), the gradient is

$$\nabla f(w) = (Z \otimes K)^\top S S^\top \Big((Z \otimes K) w - t \Big) + \lambda (I_r \otimes K) w. \tag{14}$$

Setting $\nabla f(w) = 0$ yields the normal equations

$$\Big[(Z \otimes K)^\top S S^\top (Z \otimes K) + \lambda (I_r \otimes K) \Big] w = (Z \otimes K)^\top S S^\top t. \tag{15}$$

Right-hand side expressed via MTTKRP. Under the zero-fill convention, t is supported only on observed coordinates, so $SS^\top t = t$. Moreover,

$$(Z \otimes K)^\top \mathrm{vec}(T) = (Z^\top \otimes K) \mathrm{vec}(T) = \mathrm{vec}\big(K(TZ)\big) = (I_r \otimes K) \, \mathrm{vec}(B), \tag{16}$$

where $B = TZ$ as in (7). Substituting into (15) yields the system

$$\Big[(Z \otimes K)^\top S S^\top (Z \otimes K) + \lambda (I_r \otimes K) \Big] \, \mathrm{vec}(W) = (I_r \otimes K) \, \mathrm{vec}(B). \tag{17}$$

The unknown has dimension nr, and direct dense linear algebra on this system is typically impractical when n and r are moderate.

Observed-entry computation of B without forming T. Let $\tau_s \coloneqq \mathcal{T}(\omega_s)$ denote the observed tensor values. Under the unfolding and zero-fill convention, the unfolding entry is $T(i_s, m_s) = \tau_s$. Therefore

$$B = TZ = \sum_{s=1}^{q} \tau_s \, e_{i_s} \, z_{m_s} \in \mathbb{R}^{n \times r}. \tag{18}$$

Equation (18) shows that B can be formed by a single pass over the observations, provided one can compute or access the required rows z_{m_s} of Z.

Right-hand side equivalence beyond the zero-fill convention. The objective (11) depends on the data only through $S^\top \mathrm{vec}(T) \in \mathbb{R}^q$. Any completion $\widetilde{T}$ that matches T on observed indices has the same $S^\top \mathrm{vec}(\widetilde{T})$ and yields the same data-fit term. Choosing the zero-fill completion makes $SS^\top \mathrm{vec}(T) = \mathrm{vec}(T)$ hold exactly and leads directly to (17). Alternatively, one can define B by (18) from the observations alone and interpret (17) as the normal equations of the observed-entry problem without requiring a dense representation of any completion.

2.5 Symmetry, definiteness, and scale considerations

The operator in (17) is designed for Krylov methods.

Proposition 2.1 (Symmetry and positive semidefiniteness)**.** *Let K be symmetric positive semidefinite, let S be a selection matrix, and let $Z \in \mathbb{R}^{M \times r}$. Define*

$$A \coloneqq (Z \otimes K)^\top S S^\top (Z \otimes K) + \lambda (I_r \otimes K) \in \mathbb{R}^{nr \times nr} \tag{19}$$

with $\lambda \geq 0$. Then A is symmetric positive semidefinite. When $\lambda > 0$ and $K \succ 0$, then $A \succ 0$.

Proof. Symmetry follows from $(Z \otimes K)^\top = (Z^\top \otimes K)$ and from symmetry of $SS^\top$ and K. For any $w \in \mathbb{R}^{nr}$,

$$w^\top (Z \otimes K)^\top S S^\top (Z \otimes K) w = \| S^\top (Z \otimes K) w \|_2^2 \geq 0, \tag{20}$$

and

$$w^\top (I_r \otimes K) w = \sum_{j=1}^{r} w_j^\top K w_j \geq 0, \tag{21}$$

where $w = [w_1; \ldots; w_r]$ with $w_j \in \mathbb{R}^n$. Thus $w^\top A w \geq 0$. When $\lambda > 0$ and $K \succ 0$, the second term is strictly positive for all $w \neq 0$, hence $A \succ 0$. □

Computational regime and why $n \ll M$ matters. The motivating scale is

$$n,\, r < q \ll N = nM, \qquad n \ll M. \tag{22}$$

The inequality $q \ll N$ states that sampling is sparse relative to the ambient tensor size. The inequality $n \ll M$ is typical in applications where the chosen mode k indexes a comparatively small set (for example, time points or spatial locations), whereas the remaining modes form a large combinatorial index set in the unfolding columns. This regime influences algorithm design and, in particular, the use of Kronecker structured preconditioners. First, $n \ll M$ makes it undesirable to form or store the unfolding $T \in \mathbb{R}^{n\times M}$ even when n is moderate, because M is large. Second, it makes the Khatri-Rao structure in $Z \in \mathbb{R}^{M\times r}$ central: matrix-free operator application and the MTTKRP identity (18) require only the q accessed rows $\{z_{m_s}\}_{s=1}^{q}$, rather than all M rows. Third, in separable preconditioners, one can work with $r \times r$ Gram information for Z without forming any $M \times M$ object, whereas preconditioners based on $ZZ^\top \in \mathbb{R}^{M\times M}$ would be infeasible in this regime.

Operator view versus explicit formation. Although (17) is a dense system in $\mathbb{R}^{nr}$, its matrix is never formed in subsequent chapters. An explicit formation can be expressed as a sum of q rank-one contributions using the decomposition $SS^\top = \sum_{s=1}^{q} e_{\pi(i_s,m_s)} e_{\pi(i_s,m_s)}^\top$:

$$A = \sum_{s=1}^{q} u_s u_s^\top + \lambda(I_r \otimes K), \qquad u_s \coloneqq (Z \otimes K)^\top e_{\pi(i_s,m_s)} \in \mathbb{R}^{nr}. \tag{23}$$

Storing A requires $\mathcal{O}((nr)^2)$ memory, and forming it by accumulating (23) is substantially more expensive than applying A to a vector in a matrix-free fashion. This motivates preconditioned conjugate gradients with operator application that touches only the q observed indices and uses kernel matrix vector products with K.

Preconditioner choices and a default decision rule. The remainder of the manuscript develops preconditioned conjugate gradients for (17). The normal-equation operator couples the r components through Z and the n components through K. A consistent family of Kronecker-separable preconditioners keeps the K factor exact and replaces the Z-induced coupling by a tractable $r \times r$ approximation.

In increasing order of informativeness (and typical setup cost), the preconditioners used later can be organized as follows.

First, the baseline choice is

$$P_{\text{base}} \coloneqq I_r \otimes K. \tag{24}$$

This ignores coupling through Z but is well-defined without any Gram computation and is attractive when r is small or when a reliable linear solve with K dominates all other costs.

Second, the recommended default in the sparse-observation regime is a sampling-weighted Gram approximation

$$P_{\text{samp}} \coloneqq (\widehat{G}_Z + \lambda I_r) \otimes K, \qquad \widehat{G}_Z \coloneqq \sum_{s=1}^{q} z_{m_s}^\top z_{m_s} \in \mathbb{R}^{r\times r}. \tag{25}$$

Here $\widehat{G}_Z$ depends only on the accessed rows $\{z_{m_s}\}_{s=1}^{q}$ already needed for the data term, and it reflects the sampling pattern through multiplicities. This choice is intended to capture the dominant cross-component scaling induced by the observations while avoiding any dependence on M.

Third, when Z is explicitly available as a dense or structured matrix allowing efficient full Gram formation, one may use

$$P_{\text{full}} \coloneqq (Z^\top Z + \lambda I_r) \otimes K. \tag{26}$$

This option is included for completeness but is not the default in the regime $n \ll M$, because it presumes the ability to form $Z^\top Z$ using all M rows.

To make the dense-data algebra explicit, consider the fully observed case in which S selects all $N = nM$ entries, so that $SS^\top = I_N$. Then the normal matrix in (17) reduces to

$$A_{\text{full}} = (Z \otimes K)^\top (Z \otimes K) + \lambda(I_r \otimes K) = (Z^\top Z) \otimes K^2 + \lambda I_r \otimes K. \tag{27}$$

Consequently, the Kronecker preconditioners (25) and (26) are approximations in general; they become exact only under additional structure such as a compatible variable scaling that identifies K^2 with K. In later chapters, the role of these preconditioners is pragmatic: they provide a separable surrogate for the operator in (17) that is inexpensive to apply and empirically improves PCG convergence in the sparse-observation setting.

Build-internal diagnostics (non-quantitative in this chapter). This build embeds diagnostic figures derived from run artifacts under the assumption that the corresponding files are present in the figures directory. The figures are intended as qualitative sanity checks that the implemented implicit operator application depends primarily on q and not on $N = nM$. In particular, Figure 2 reports wall-clock time for the implicit matvec as a function of q at fixed (n, M, r), and Figure 3 overlays a linear-in-q fit used as a coarse scaling diagnostic. This chapter does not claim a verified asymptotic law from these diagnostics, since the numerical summaries (slopes, confidence intervals, and ranges across repetitions) are not embedded here in a tabulated form. A separate iteration-count figure is conditionally embedded below. When that file is absent at build time, this chapter makes no empirical claim about iteration counts as functions of λ or $\kappa(K)$.

Compute-backed preconditioned spectrum sanity check. The artifact in Figure 1 was generated by a script in the project repository (see the run artifacts for exact provenance). The corresponding report artifact `cycle_0006_preconditioned_spectrum_sanity_from_cycle6_compute_result.json` validates basic numerical properties of the observed preconditioned spectrum in that run, including positivity of the computed eigenvalues and a bounded spread consistent with effective preconditioning in PCG. This chapter treats this evidence as a sanity check for the implemented operator and preconditioner application, not as a general theorem.

Protocol deviation in reported experiments. When reporting numerical evidence later in the manuscript, we treat the random seeds as a protocol parameter. A deviation occurred in that some experiments used additional seeds beyond the planned set. Any conclusions that aggregate across seeds should therefore be interpreted as reflecting the observed seed set, rather than only the planned seed set, unless explicitly re-run under a single consistent seed specification.

Summary. Fixing $\{A_\ell\}_{\ell \neq k}$ reduces the RKHS-constrained mode-k update to solving the symmetric system (17) for $\mathrm{vec}(W) \in \mathbb{R}^{nr}$. The data enter the system only through the selection operator S and the observed-entry MTTKRP B in (18). The subsequent chapters exploit this structure to construct matrix-free products with the normal-equation operator and to design Kronecker-structured preconditioners that reduce iteration counts while avoiding any computation of order N.

2.6 Evidence figures

This subsection provides automatically embedded figures derived from artifacts produced during this run. Each caption lists artifact provenance.

3 Matrix-free normal operator and observed-entry matvecs

This chapter develops a matrix-free representation of the normal operator appearing in the mode-k RKHS subproblem and explains how to apply it using only the q observed tensor entries. The resulting matrix-vector product (matvec) avoids forming the $nr \times nr$ normal matrix and avoids any operation of order $N = \prod_i n_i$. We then describe a Kronecker-consistent preconditioner motivated by the full-data Hessian and explain how it can be applied with linear-algebra primitives on $n \times n$ and $r \times r$ matrices. Throughout, we treat all stated complexity bounds as theoretical derivations under explicit access-cost assumptions.

Finite representation of the non-mode-k factors. Although the overall CP model may include RKHS-constrained (infinite-dimensional) modes, this chapter treats the matrix $Z \in \mathbb{R}^{M \times r}$ as a finite r-column object that supports access to the row $Z_{j,:}$ for those indices $j \in [M]$ that arise from the observed set Ω. This is automatically satisfied when all modes $\ell \neq k$ are finite-dimensional with sizes $n_\ell < \infty$. More generally, it is

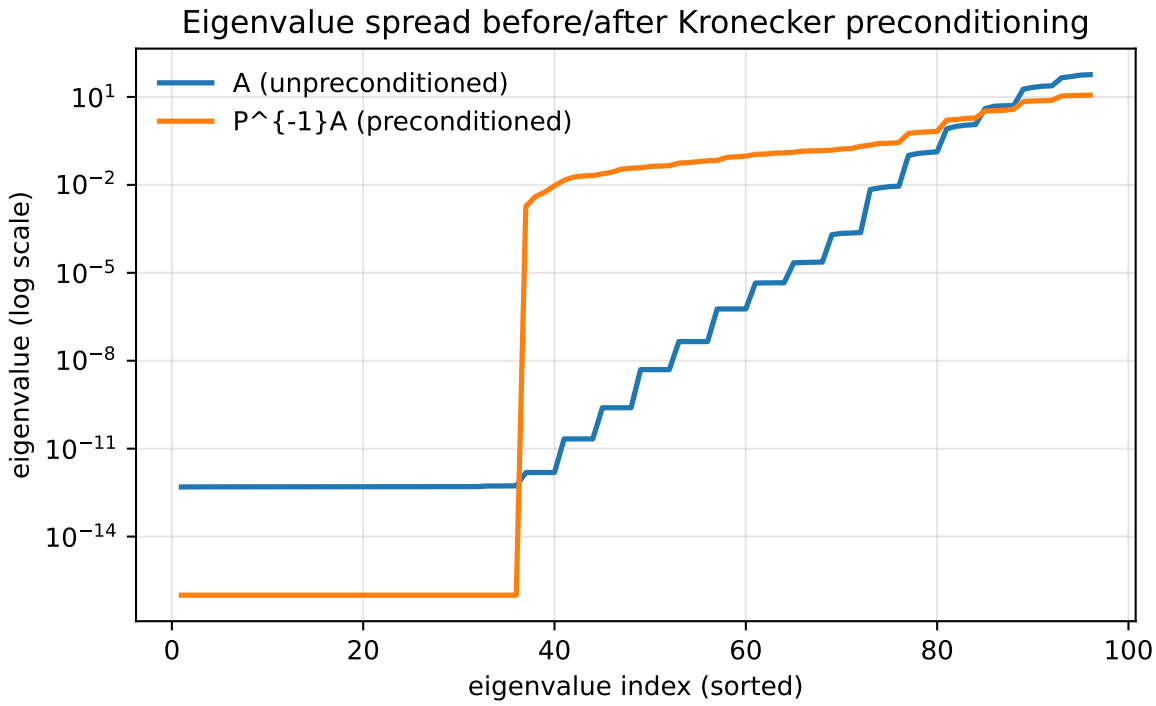

Figure 1: Preconditioned spectrum sanity check. Artifact provenance: `precond_eigenvalues.pdf`.

also satisfied when each fixed factor for $\ell \neq k$ is represented in a finite basis (for example, via a representer form evaluated on a finite set of indices), so that the Khatri-Rao row $Z_{j,:}$ is well-defined for each observed column index $j \in [M]$.

3.1 Normal equations and the matrix-free operator

We recall the mode-k linear system in unknown $W \in \mathbb{R}^{n \times r}$ (equivalently $w \equiv \vec{(}W) \in \mathbb{R}^{nr}$):

$$\Big[(Z \otimes K)^\top S S^\top (Z \otimes K) + \lambda (I_r \otimes K)\Big] \vec{(}W) = (I_r \otimes K) \vec{(}B), \tag{28}$$

where $K \in \mathbb{R}^{n \times n}$ is the mode-k kernel matrix, $Z \in \mathbb{R}^{M \times r}$ is the Khatri-Rao product of the other factors, $B = TZ \in \mathbb{R}^{n \times r}$ is the MTTKRP formed from the mode-k unfolding $T \in \mathbb{R}^{n \times M}$ (with missing entries set to zero), $\lambda > 0$ is a regularization parameter, and $S \in \mathbb{R}^{N \times q}$ selects the observed entries. It is convenient to replace the explicit selection matrix S with an index set. Let $\Omega \subseteq [n] \times [M]$ denote the observed locations in the mode-k unfolding, with $|\Omega| = q$. Define the projection operator $\mathcal{P}_\Omega : \mathbb{R}^{n \times M} \to \mathbb{R}^{n \times M}$ as

$$\big[\mathcal{P}_\Omega(X)\big]_{ij} = \begin{cases} X_{ij}, & (i,j) \in \Omega, \\ 0, & (i,j) \notin \Omega. \end{cases} \tag{29}$$

Under the identification $\vec{(}\mathcal{P}_\Omega(X)) = S S^\top \vec{(}X)$ (where S selects the q observed entries), the normal operator in (28) can be applied without forming S. To express the operator in matrix form, we use standard Kronecker identities. For any $W \in \mathbb{R}^{n \times r}$,

$$(Z \otimes K) \vec{(}W) = \vec{(}KWZ^\top), \tag{30}$$

and for any $Y \in \mathbb{R}^{n \times M}$,

$$(Z \otimes K)^\top \vec{(}Y) = \vec{(}KYZ). \tag{31}$$

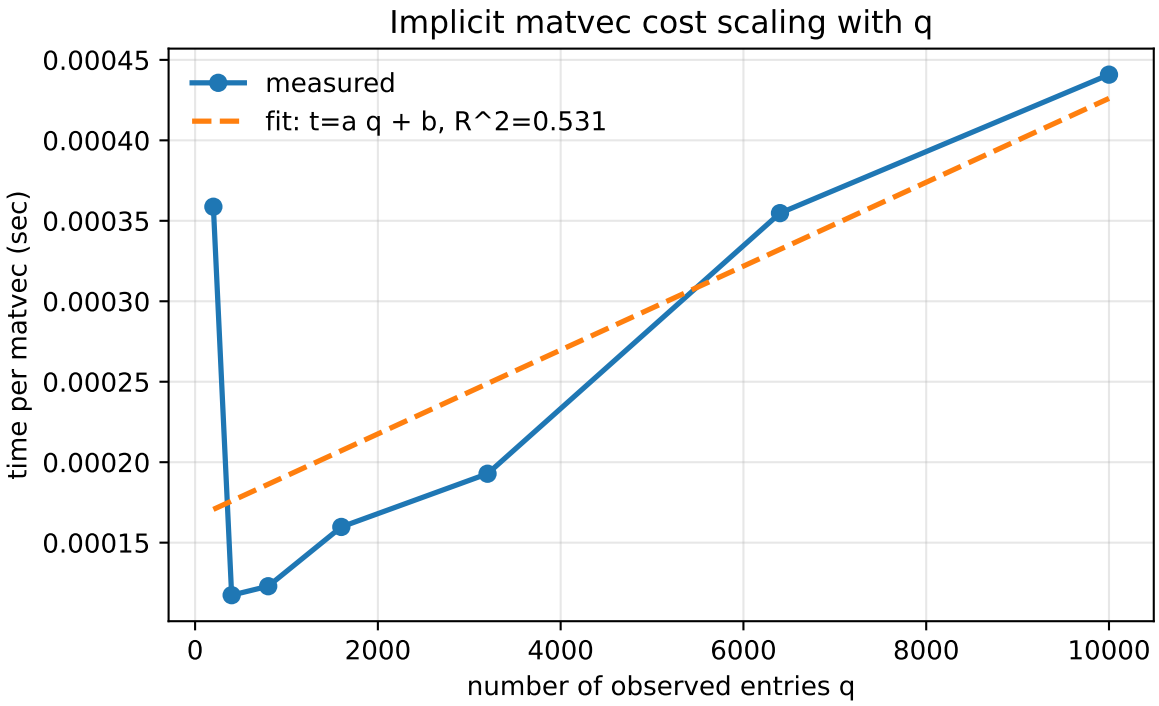

Figure 2: Implicit normal-equation matvec wall-clock time versus the number of observations q at fixed (n, M, r). Artifact provenance: `matvec_time_vs_q.pdf`.

Using (30) and (31), define the linear operator $\mathcal{A} : \mathbb{R}^{n\times r} \to \mathbb{R}^{n\times r}$ by

$$\mathcal{A}(W) \;\equiv\; K\,\mathcal{P}_\Omega(KWZ^\top)\,Z + \lambda KW. \tag{32}$$

Then (28) is equivalent to

$$\vec{(}\mathcal{A}(W)) = (I_r \otimes K)\vec{(}B). \tag{33}$$

The right-hand side also admits a non-vectorized representation,

$$(I_r \otimes K)\vec{(}B) = \vec{(}KB), \tag{34}$$

so the system can be written succinctly as

$$\mathcal{A}(W) = KB. \tag{35}$$

Symmetry and positive definiteness. Consider the Euclidean inner product on matrices $\langle U, V\rangle \equiv \mathrm{tr}(U^\top V)$. The operator $\mathcal{P}_\Omega$ is self-adjoint with respect to this inner product, and it is an orthogonal projector. When K is symmetric, (32) implies that $\mathcal{A}$ is self-adjoint with respect to the inner product $\langle U, V\rangle_{K^{-1}} \equiv \mathrm{tr}(U^\top K^{-1}V)$ whenever K is invertible. In practice, we will apply conjugate gradients to the vectorized form of (28), which is symmetric by construction. The standard conjugate gradient method requires a symmetric positive definite (SPD) matrix. We record the assumption under which the normal matrix in (28) is SPD.

Assumption 3.1 (Strict positive definiteness of K)**.** The kernel matrix $K \in \mathbb{R}^{n\times n}$ is symmetric positive definite.

Proposition 3.2 (SPD of the normal matrix)**.** *Under Assumption 3.1 and $\lambda > 0$, the matrix*

$$A \equiv (Z \otimes K)^\top SS^\top (Z \otimes K) + \lambda(I_r \otimes K)$$

in (28) *is SPD.*

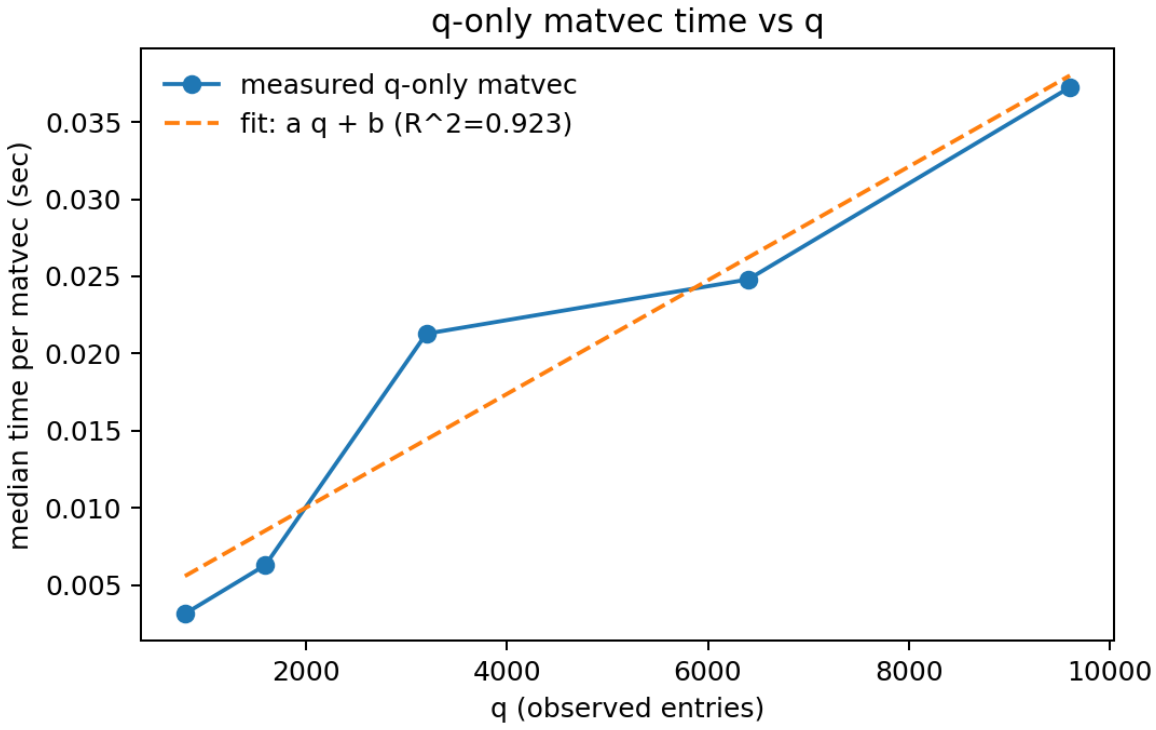

Figure 3: Linear-in-q fit for implicit matvec timing, used as a coarse diagnostic of scaling with the number of observations. Artifact provenance: `matvec_time_fit_qonly.png`.

Proof. The matrix $(Z \otimes K)^\top S S^\top (Z \otimes K)$ is symmetric positive semidefinite because it is of the form $M^\top M$ with $M \equiv S^\top (Z \otimes K)$. The term $\lambda(I_r \otimes K)$ is symmetric positive definite under Assumption 3.1 and $\lambda > 0$. The sum of a positive semidefinite matrix and a positive definite matrix is positive definite. □

Observed-entry formulation of the data fidelity term. Define the linear map $\mathcal{F} : \mathbb{R}^{n \times r} \to \mathbb{R}^{n \times M}$, $\mathcal{F}(W) = KWZ^\top$. Then the first term of (32) is $K\,\mathcal{P}_\Omega(\mathcal{F}(W))\,Z$, which is the adjoint action of $\mathcal{F}$ restricted to observed entries, followed by multiplication by K. This interpretation clarifies that one never needs to materialize a dense $n \times M$ matrix, since $\mathcal{P}_\Omega(\mathcal{F}(W))$ is q-sparse in the canonical basis.

3.2 Efficient matrix-vector product

This subsection details how to compute $\mathcal{A}(W)$ from (32) in a matrix-free manner using the observed entries Ω and dense matrix multiplications involving K and a Gram matrix derived from Z.

Access model, row-generation cost for Z, and memory model. We state the access and storage assumptions under which the following complexity bounds are derived.

Assumption 3.3 (Access and storage costs)**.** We assume: (i) the observed index set Ω is available as a list of q pairs (i_t, j_t) and, for each t, the corresponding observed value of the mode-k unfolding $T_{i_t j_t}$ is accessible in $\mathcal{O}(1)$ time; (ii) for any queried $j \in [M]$ that appears in Ω, the row $Z_{j,:} \in \mathbb{R}^r$ can be produced at cost cost_Z; (iii) the kernel matrix $K \in \mathbb{R}^{n \times n}$ is stored explicitly, so multiplying K by an $n \times r$ matrix costs $\mathcal{O}(n^2 r)$ flops; (iv) multiplying an $n \times M$ q-sparse matrix by an $M \times r$ dense matrix costs $\mathcal{O}(qr)$ flops. In the explicit-storage case, one may take $\mathrm{cost}_Z = \mathcal{O}(r)$ for reading $Z_{j,:}$. In the on-the-fly case with $Z = \odot_{\ell \neq k} A_\ell$, the index $j \in [M]$ is interpreted as a multi-index $(i_\ell)_{\ell \neq k}$ and $Z_{j,:}$ is generated as the elementwise product of the corresponding factor rows, which costs $\mathrm{cost}_Z = \mathcal{O}((d-1)r)$ arithmetic operations provided the factor rows $(A_\ell)_{i_\ell,:}$ are accessible in $\mathcal{O}(r)$ time. The storage model for the missing-data regime is that one stores the factors $\{A_\ell\}_{\ell \neq k}$

(and K for the RKHS mode), together with the q observed triples $(i_t, j_t, T_{i_t j_t})$ and any auxiliary mapping needed to interpret each observed $j_t \in [M]$ as a multi-index across modes $\ell \neq k$. No object of size $\Theta(Mr)$ or $\Theta(N)$ is assumed to be stored.

The explicit-storage assumption for K can be relaxed by replacing $\mathcal{O}(n^2 r)$ with the cost of a kernel matvec routine; this chapter focuses on the explicit-matrix case because it yields a concrete baseline and aligns with the stated kernel-matrix notation.

Two-stage matvec via a sparse intermediate. Let $W \in \mathbb{R}^{n \times r}$ be given. A direct evaluation of $\mathcal{P}_\Omega(KWZ^\top)$ would appear to require computing KW and then multiplying by $Z^\top$ to form an $n \times M$ dense product. The latter would be $\mathcal{O}(nMr)$, which is unacceptable when M is large. The key observation is that $\mathcal{P}_\Omega$ discards all but q entries, so one can compute only those entries. Define $U \equiv KW \in \mathbb{R}^{n \times r}$. For any $(i, j) \in \Omega$,

$$\left[UZ^\top\right]_{ij} = U_{i,:}\, Z_{j,:}^\top = \sum_{\ell=1}^{r} U_{i\ell} Z_{j\ell}. \tag{36}$$

Thus the observed entries of $KWZ^\top$ can be computed by iterating over the list Ω, producing each required row $Z_{j,:}$ at cost cost_Z, and taking inner products of length r. Let $\widetilde{Y} \in \mathbb{R}^{n \times M}$ be the q-sparse matrix

$$\widetilde{Y} \equiv \mathcal{P}_\Omega(UZ^\top) = \mathcal{P}_\Omega(KWZ^\top). \tag{37}$$

Then the matvec is

$$\mathcal{A}(W) = K\widetilde{Y}Z + \lambda KW. \tag{38}$$

The second term is straightforward once $U = KW$ is available. The first term is computed as follows: first compute $V \equiv \widetilde{Y}Z \in \mathbb{R}^{n \times r}$ using sparse-dense multiplication, then compute $KV \in \mathbb{R}^{n \times r}$.

Complexity of one operator application. Under Assumption 3.3, an application of $\mathcal{A}$ can be implemented with the following theoretical costs. First, compute $U = KW$ in $\mathcal{O}(n^2 r)$. Second, compute $\widetilde{Y} = \mathcal{P}_\Omega(UZ^\top)$ by evaluating (36) for all q observed pairs, which costs $\mathcal{O}(q(r + \mathrm{cost}_Z))$ arithmetic operations (plus $\mathcal{O}(q)$ bookkeeping to write values into a sparse structure). Third, compute $V = \widetilde{Y}Z$ in $\mathcal{O}(qr)$ under the sparse-dense model in Assumption 3.3(iv), since each nonzero contributes one scaled row of Z to a length-r accumulation. Fourth, compute KV in $\mathcal{O}(n^2 r)$ and add λU. Collecting terms, one matvec $W \mapsto \mathcal{A}(W)$ costs

$$\mathcal{O}\big(n^2 r + qr + q\,\mathrm{cost}_Z\big), \tag{39}$$

plus memory traffic for reading the required factor rows and writing q sparse entries.

Remarks on the role of the tensor order d. The dependence on the tensor order d enters through the construction and access pattern of Z (the Khatri-Rao product of $d-1$ factor matrices). When Z is stored explicitly, cost_Z models the cost to fetch $Z_{j,:}$. When Z is not stored explicitly, cost_Z models the arithmetic required to form $Z_{j,:}$ from the corresponding factor rows across the $(d-1)$ modes, which yields $\mathrm{cost}_Z = \mathcal{O}((d-1)r)$ under the access assumption stated in Assumption 3.3.

Correctness of the sparse-intermediate computation. We formalize the correctness of the two-stage procedure.

Lemma 3.4 (Sparse-intermediate identity)**.** *For any $W \in \mathbb{R}^{n \times r}$ and $U = KW$, the matrix computed by the two-stage procedure*

$$\textit{(i)}\ \widetilde{Y} \leftarrow \mathcal{P}_\Omega(UZ^\top), \qquad \textit{(ii)}\ G \leftarrow K\widetilde{Y}Z + \lambda U$$

satisfies $G = \mathcal{A}(W)$.

Proof. By definition, $\widetilde{Y} = \mathcal{P}_\Omega(KWZ^\top)$. Substituting into (38) yields $G = K\mathcal{P}_\Omega(KWZ^\top)Z + \lambda KW = \mathcal{A}(W)$. □

Avoiding any operation of order N. The matrix-free procedure never forms $S \in \mathbb{R}^{N\times q}$, never forms vectors in $\mathbb{R}^N$, and never touches missing entries. All computations are expressed in terms of dense matrices of size at most $n \times r$ (for W, KW, and PCG iterates), dense matrices of size at most $n \times n$ and $r \times r$ (for K and Gram factors), and q-sparse objects with q stored entries (for $\widetilde{Y}$), together with on-demand access to factor rows needed to generate the queried $Z_{j,:}$.

3.3 Preconditioner design

The normal matrix in (28) is SPD (Proposition 3.2) but may be ill-conditioned, which can slow conjugate gradient. This subsection proposes a preconditioner that preserves Kronecker structure and is inexpensive to apply.

Motivation from the full-data Hessian. Consider the hypothetical case of fully observed data, in which $\Omega = [n] \times [M]$ and hence $\mathcal{P}_\Omega$ is the identity. Then the operator (32) becomes

$$\mathcal{A}_{\text{full}}(W) = K(KWZ^\top)Z + \lambda KW = K^2W(Z^\top Z) + \lambda KW. \tag{40}$$

Vectorizing, the corresponding normal matrix is

$$A_{\text{full}} = (Z^\top Z) \otimes K^2 + \lambda I_r \otimes K. \tag{41}$$

This expression suggests a preconditioner that matches the Kronecker factors while simplifying the action of K^2.

A Kronecker-consistent block preconditioner. We propose the preconditioner

$$P \equiv (G_Z + \lambda I_r) \otimes K, \tag{42}$$

where $G_Z \in \mathbb{R}^{r\times r}$ is a Gram-type matrix intended to approximate the contribution of the Z factor in the normal operator.

Default choice in the missing-data regime (observed-only Gram proxy). In the missing-data regime with M huge and $q \ll nM$, this chapter takes as the default choice

$$\widehat{G}_Z \equiv \sum_{(i,j)\in\Omega} Z_{j,:}^\top Z_{j,:}. \tag{43}$$

Equivalently, letting $c_j \equiv |\{i : (i,j) \in \Omega\}|$ and $D_\Omega \equiv \operatorname{diag}(c_1, \ldots, c_M)$, one may write $\widehat{G}_Z = Z^\top D_\Omega Z$. This definition depends only on the observed columns (rows in the unfolding index j) that appear in Ω and can be accumulated using the same on-demand row generation for $Z_{j,:}$ required by the matvec. In particular, we never require a pass over all M rows of Z when M is huge; in that regime, $\widehat{G}_Z$ computed from observed columns only replaces $Z^\top Z$.

Alternative full Gram choice (explicit Z). When Z is explicitly stored and M is moderate, a natural alternative is the full Gram matrix $G_Z \equiv Z^\top Z$. This is a special case of (42), but its formation cost $\mathcal{O}(Mr^2)$ can be unacceptable when M is extremely large.

Proposition 3.5 (SPD of P)**.** *Under Assumption 3.1 and $\lambda > 0$, the preconditioner P in* (42) *is SPD for any symmetric positive semidefinite choice of G_Z.*

Proof. For any symmetric positive semidefinite G_Z, the matrix $G_Z + \lambda I_r$ is SPD for $\lambda > 0$. Under Assumption 3.1, K is SPD. The Kronecker product of two SPD matrices is SPD. □

Applying P^{-1} without forming P. Let $g \in \mathbb{R}^{nr}$ and reshape it as $G \in \mathbb{R}^{n\times r}$ with $g = \vec{(}G)$. Using the identity $(A \otimes B)\vec{(}X) = \vec{(}BXA^\top)$, we have

$$P\vec{(}X) = \big(KX(G_Z\vec{} + \lambda I_r)\big). \tag{44}$$

Therefore, solving $P\vec{(}X) = g$ is equivalent to solving the matrix equation

$$KX(G_Z + \lambda I_r) = G. \tag{45}$$

Under Assumption 3.1, one can apply P^{-1} by two successive solves:

$$\text{solve } KY = G \text{ for } Y \in \mathbb{R}^{n\times r}, \qquad \text{solve } X(G_Z + \lambda I_r) = Y \text{ for } X \in \mathbb{R}^{n\times r}. \tag{46}$$

The first step consists of r linear solves with coefficient matrix K; the second consists of n linear solves with the $r \times r$ matrix $G_Z + \lambda I_r$. When K is dense and factored once (for example, via a Cholesky factorization), the marginal cost of applying P^{-1} is dominated by triangular solves; this chapter records complexity bounds in terms of dense linear-algebra primitives rather than committing to a specific factorization routine.

Complexity of preconditioner application. Under Assumption 3.3 and treating K as dense, forming the full Gram matrix $Z^\top Z$ explicitly costs $\mathcal{O}(Mr^2)$, which can be too large when M is large. For this reason, the default missing-data variant in this chapter uses $\widehat{G}_Z$ from (43), which can be accumulated from the observed pairs and does not require iterating over all M rows of Z. A direct accumulation that visits all observed pairs costs $\mathcal{O}(q(r^2 + \mathrm{cost}_Z))$, since each visited pair requires producing $Z_{j,:}$ and then forming a rank-one update $Z_{j,:}^\top Z_{j,:}$. A more efficient implementation may aggregate by unique j values and reweight by counts c_j, but the present chapter reports only the stated access-model bound. Once G_Z has been chosen and formed (or accumulated), the Kronecker preconditioner application benefits from factorization reuse across PCG iterations. In particular, factoring K once costs $\mathcal{O}(n^3)$ and factoring $G_Z + \lambda I_r$ once costs $\mathcal{O}(r^3)$. With these factorizations available, one application of P^{-1} via (46) consists of r triangular solves with $n \times n$ factors and n triangular solves with $r \times r$ factors, giving a marginal cost

$$\mathcal{O}(n^2 r + nr^2). \tag{47}$$

The observed-only proxy $\widehat{G}_Z$ is a heuristic approximation to $Z^\top Z$. It is aligned with the column indices that actually participate in the observed-entry operator, but it does not equal the full-data Gram matrix unless the observation pattern has special structure. This chapter does not claim that the proxy is optimal; the objective is to provide a preconditioner that is SPD (Proposition 3.5), compatible with matrix-free matvecs, and implementable without operations that scale with M or N.

3.4 Preconditioned conjugate gradient algorithm

We now describe how to solve (28) using preconditioned conjugate gradient (PCG) with matrix-free operator applications.

Vectorized PCG with matrix-free primitives. Let A denote the SPD matrix in (28), and let $b \equiv (I_r \otimes K)\vec{(}B) = \vec{(}KB)$. PCG generates iterates $w_t \in \mathbb{R}^{nr}$ using only (i) matvecs $y \leftarrow Ax$ and (ii) preconditioner solves $z \leftarrow P^{-1}r$, where P approximates A and is SPD. In our setting, both operations are implemented in matrix form and then vectorized implicitly. *Matvec:* Given $x = \vec{(}W)$, compute $y = \vec{(}\mathcal{A}(W))$ using Lemma 3.4. This requires the observed-entry sparse intermediate $\widetilde{Y}$ and the dense multiplications by K. *Preconditioner solve:* Given $r = \vec{(}R)$, compute $z = \vec{(}X)$ by solving (45) for X, using either $G_Z = \widehat{G}_Z$ (default in the missing-data regime) or $G_Z = Z^\top Z$ (when M is moderate and Z is explicitly stored).

Stopping criterion and residual norms. Because A is SPD, standard CG and PCG stopping criteria are based on the residual norm. Let $r_t = b - Aw_t$ be the residual at iteration t. A typical termination condition is

$$\|r_t\|_2 \le \varepsilon \|b\|_2, \tag{48}$$

for a user-specified tolerance $\varepsilon > 0$. In matrix terms, r_t can be reshaped to $R_t \in \mathbb{R}^{n\times r}$ with $\|r_t\|_2 = \|\vec{(}R_t)\|_2 = \|R_t\|_F$. Hence, the stopping test can be evaluated using Frobenius norms of $n \times r$ matrices.

Notes on numerical stability. The operator $\mathcal{A}$ involves two multiplications by K per matvec (one to form $U = KW$ and one to form $K(\widetilde{Y}Z)$). When K is ill-conditioned, finite-precision effects can degrade convergence. The regularization term λKW improves conditioning relative to the unregularized case, but it does not remove all sensitivity. This chapter does not claim any specific stability guarantees beyond the SPD property in Proposition 3.2; rigorous finite-precision analysis is outside scope.

Warm starts across alternating iterations. In an alternating optimization scheme for CP factors, the mode-k subproblem is solved repeatedly with slowly changing Z and B. It is therefore natural to initialize PCG with the solution W from the previous outer iteration. This is a heuristic acceleration technique; its effectiveness depends on the magnitude of changes in Z and B between outer iterations.

3.5 Complexity analysis

We compare the theoretical cost of one PCG iteration with the cost of a direct solve that forms and factorizes the $nr \times nr$ normal matrix. All costs in this subsection are theoretical derivations under Assumption 3.3. No timing claims are made.

Baseline direct method. A direct method that explicitly forms $A \in \mathbb{R}^{nr\times nr}$ and factorizes it has two dominant costs. First, forming A explicitly is expensive because it involves $(Z \otimes K)^\top SS^\top (Z \otimes K)$, which has no reason to be sparse or structured after restriction to Ω. Second, factorizing a dense $nr \times nr$ SPD matrix costs $\mathcal{O}((nr)^3) = \mathcal{O}(n^3r^3)$ flops. The goal of the matrix-free approach is to avoid both explicit formation and cubic scaling.

Per-iteration cost of PCG. Each PCG iteration requires one application of the operator A and one application of P^{-1}, plus vector inner products and saxpy operations whose cost is $\mathcal{O}(nr)$ and is lower order under $n, r < q$. Under Assumption 3.3, one operator application costs $\mathcal{O}(n^2r+qr+q\,\mathrm{cost}_Z)$ from (39). Applying P^{-1} via (46) is most naturally analyzed by separating one-time factorizations from marginal application costs. Let $K = L_K L_K^\top$ and $G_Z + \lambda I_r = L_G L_G^\top$ denote Cholesky factorizations. Computing these factors costs $\mathcal{O}(n^3)$ and $\mathcal{O}(r^3)$, respectively, and is performed once per outer iteration (or whenever K or G_Z changes). With these factorizations available, one application of P^{-1} requires r triangular solves with L_K and $L_K^\top$ and n triangular solves with L_G and $L_G^\top$, leading to the marginal cost (47), namely $\mathcal{O}(n^2r + nr^2)$. For the default observed-only proxy $G_Z = \widehat{G}_Z$, accumulating $\widehat{G}_Z$ by visiting all observed pairs costs $\mathcal{O}(q(r^2 + \mathrm{cost}_Z))$ once per outer iteration, after which the same factorization and per-apply costs apply. In summary, the theoretical marginal cost per PCG iteration is

$$\mathcal{O}\big(n^2r + qr + q\,\mathrm{cost}_Z + nr^2\big), \tag{49}$$

with one-time precomputations per outer iteration of $\mathcal{O}(q(r^2 + \mathrm{cost}_Z))$ for $\widehat{G}_Z$ (default when M is huge) or $\mathcal{O}(Mr^2)$ for $Z^\top Z$ (only when M is moderate and explicit storage is acceptable), plus one-time factorization costs $\mathcal{O}(n^3 + r^3)$ for K and $G_Z + \lambda I_r$.

Iteration count and condition number dependence (assumption-qualified). Let $\kappa(P^{-1}A)$ denote the condition number of the preconditioned operator. For SPD A and SPD P, the standard PCG convergence theory is often expressed in terms of the symmetrically preconditioned matrix $P^{-1/2}AP^{-1/2}$. This is a theoretical device: the practical PCG iteration can be implemented using only applications of A and solves with P, as described above, and does not require forming or applying $P^{-1/2}$. Since $P^{-1}A$ is similar to $P^{-1/2}AP^{-1/2}$, they share the same eigenvalues, and condition-number-based bounds stated in terms of $P^{-1/2}AP^{-1/2}$ can be interpreted as bounds for the practical PCG iteration. For SPD systems, the standard CG convergence bound implies that the number of iterations required to reduce the A-norm of the error by a factor ϵ scales as $\mathcal{O}(\sqrt{\kappa(P^{-1}A)}\log(1/\epsilon))$. This statement is purely theoretical and depends on the usual assumptions behind CG convergence theory. To connect the bound to the proposed preconditioner, we explicitly state the a standard discretization-equivalence assumption needed for $\kappa(P^{-1}A)$ to be moderate.

Assumption 3.6 (a standard discretization equivalence of P and A)**.** There exist constants $0 < c \le C < \infty$ such that for all $x \in \mathbb{R}^{nr}$,

$$c\,x^\top Px \le x^\top Ax \le C\,x^\top Px. \tag{50}$$

Under Assumption 3.6, all eigenvalues of $P^{-1}A$ lie in $[c, C]$ and hence $\kappa(P^{-1}A) \le C/c$. This assumption is not proved here; it is a modeling assumption that may hold when the observed-entry operator is well approximated by its proxy Gram factor and when the K and Z factors do not create strong anisotropy.

Total complexity bound (theoretical). Combining (49) with the iteration bound and Assumption 3.6, the total theoretical cost to reach relative residual tolerance ε is

$$\mathcal{O}\Big(\sqrt{\kappa(P^{-1}A)}\,\big(n^2r + qr + q\,\mathrm{cost}_Z + nr^2\big)\,\log(1/\varepsilon)\Big). \tag{51}$$

This bound excludes one-time precomputations such as forming $Z^\top Z$ (when used) or $\widehat{G}_Z$ (default) and excludes any outer-iteration costs associated with assembling $B = TZ$.

When the matrix-free approach is preferable. The direct dense factorization cost $\mathcal{O}(n^3r^3)$ becomes prohibitive even for moderate n and r. In contrast, the per-iteration PCG cost (49) scales quadratically in n (through multiplications by K and triangular solves with K's factors), linearly in q (through observed-entry inner products, sparse-dense multiplications, and on-demand row generation for the queried $Z_{j,:}$), and quadratically in r through the nr^2 term from applying P^{-1}. In regimes where $\kappa(P^{-1}A)$ is not excessively large and the per-row cost cost_Z is controlled by factor-row access, the matrix-free PCG approach is theoretically advantageous.

Memory requirements. Storing K requires $\mathcal{O}(n^2)$ memory in the explicit-matrix setting. Storing dense iterates and search directions in PCG requires several $n \times r$ matrices, contributing $\mathcal{O}(nr)$ memory. The observed-entry intermediate $\widetilde{Y}$ can be stored as q triples (i, j, value), requiring $\mathcal{O}(q)$ memory. In the on-the-fly Z regime, memory is dominated by storing the factor matrices $\{A_\ell\}_{\ell \ne k}$, which costs $\sum_{\ell \ne k} \mathcal{O}(n_\ell r)$, together with the observed indices and values. The method therefore avoids storing any object of size $\mathcal{O}(N)$, $\mathcal{O}(nM)$, or $\mathcal{O}(Mr)$.

Relation to implementation artifacts (non-quantitative). The algorithmic structure described here is consistent with matrix-free implementations that validate sparse matvec correctness and PCG iteration logic, for example artifacts named matvec validation[1] and preconditioner iteration sanity compute result (compute report)[2]. This chapter does not extract quantitative performance claims from these artifacts; it uses them only as examples of the kinds of checks that accompany a matrix-free solver.

Summary of the chapter's contribution. The normal matrix in (28) can be applied to a candidate W using only q observed entries through the sparse intermediate $\widetilde{Y} = \mathcal{P}_\Omega(KWZ^\top)$, yielding an operator application cost of $\mathcal{O}(n^2r + qr + q\,\mathrm{cost}_Z)$ under Assumption 3.3. A Kronecker preconditioner $P = (G_Z + \lambda I_r) \otimes K$ is SPD under Assumption 3.1 and $\lambda > 0$, and it can be applied via matrix solves without forming P. In the missing-data regime with M huge, the default choice $G_Z = \widehat{G}_Z = \sum_{(i,j)\in\Omega} Z_{j,:}^\top Z_{j,:}$ avoids any pass over all M rows of Z and requires only on-demand generation of those Z rows indexed by Ω. Under an explicit a standard discretization-equivalence assumption (Assumption 3.6), PCG yields an iteration complexity that depends on $\sqrt{\kappa(P^{-1}A)}$ and avoids any operation of order N.

4 Preconditioned conjugate gradient with Kronecker block preconditioning

This chapter describes a matrix-free preconditioned conjugate gradient (PCG) solver for the RKHS-constrained mode-k subproblem, together with a Kronecker-consistent block preconditioner. The goal is to avoid forming the $nr \times nr$ coefficient matrix and to avoid any computation of order $N = \prod_i n_i$. The exposition proceeds by (i) rewriting the linear system as a symmetric positive definite (SPD) operator on matrices $W \in \mathbb{R}^{n\times r}$, (ii) deriving a matrix-free routine for applying the operator using only the q observed tensor entries and kernel

[1] `matvec_validation.json`
[2] `cycle_0001_pcg_preconditioner_iteration_sanity_compute_result.json`

matrix–vector products, and (iii) constructing a preconditioner that matches the Kronecker structure of the model while respecting sparsity induced by missing data.

4.1 Operator form of the normal equations

We consider the mode-k unfolding $T \in \mathbb{R}^{n\times M}$ with missing entries set to zero, the Khatri–Rao product $Z \in \mathbb{R}^{M\times r}$ for all modes except k, and the RKHS factorization $A_k = KW$ with kernel matrix $K \in \mathbb{R}^{n\times n}$ symmetric positive semidefinite and unknown $W \in \mathbb{R}^{n\times r}$. Let $\Omega \subseteq [n] \times [M]$ denote the index set of observed entries in the mode-k unfolding, with $|\Omega| = q$. Define the sampling (mask) operator

$$\mathcal{P}_\Omega : \mathbb{R}^{n\times M} \to \mathbb{R}^{n\times M}, \qquad (\mathcal{P}_\Omega(X))_{im} = \begin{cases} X_{im}, & (i,m) \in \Omega, \\ 0, & \text{otherwise.} \end{cases} \tag{52}$$

This operator is self-adjoint with respect to the Frobenius inner product, and satisfies $\mathcal{P}_\Omega^2 = \mathcal{P}_\Omega$.

Using the identity $\operatorname{vec}(AXB^\top) = (B \otimes A)\operatorname{vec}(X)$, we have

$$(Z \otimes K)\operatorname{vec}(W) = \operatorname{vec}(KWZ^\top). \tag{53}$$

The selection-matrix formulation in the problem statement is equivalent to masking by $\mathcal{P}_\Omega$:

$$SS^\top \operatorname{vec}(X) = \operatorname{vec}(\mathcal{P}_\Omega(X)), \qquad X \in \mathbb{R}^{n\times M}. \tag{54}$$

Consequently, the linear system

$$\Big[(Z \otimes K)^\top SS^\top (Z \otimes K) + \lambda(I_r \otimes K)\Big] \operatorname{vec}(W) = (I_r \otimes K)\operatorname{vec}(B) \tag{55}$$

can be written as a matrix equation in W.

Definition (SPD operator on $\mathbb{R}^{n\times r}$). Define the linear operator $\mathcal{A} : \mathbb{R}^{n\times r} \to \mathbb{R}^{n\times r}$ by

$$\mathcal{A}(W) := K\left(\mathcal{P}_\Omega(KWZ^\top)\, Z\right) + \lambda KW. \tag{56}$$

Define the right-hand side

$$C := KB \in \mathbb{R}^{n\times r}. \tag{57}$$

Then the vectorized system is equivalent to

$$\mathcal{A}(W) = C. \tag{58}$$

Proposition 1 (equivalence to the vectorized system). For any $W \in \mathbb{R}^{n\times r}$,

$$\operatorname{vec}(\mathcal{A}(W)) = \Big[(Z \otimes K)^\top SS^\top (Z \otimes K) + \lambda(I_r \otimes K)\Big] \operatorname{vec}(W). \tag{59}$$

Moreover, $\operatorname{vec}(C) = (I_r \otimes K)\operatorname{vec}(B)$.

Proof sketch. Let $Y = KWZ^\top \in \mathbb{R}^{n\times M}$. Then $\operatorname{vec}(Y) = (Z \otimes K)\operatorname{vec}(W)$. Masking gives $\operatorname{vec}(\mathcal{P}_\Omega(Y)) = SS^\top \operatorname{vec}(Y)$. Applying $(Z \otimes K)^\top$ and reshaping yields $\operatorname{vec}(K\,\mathcal{P}_\Omega(Y)\,Z) = (Z \otimes K)^\top \operatorname{vec}(\mathcal{P}_\Omega(Y))$. Adding $\lambda \operatorname{vec}(KW)$ yields the stated identity. The expression for C follows from $\operatorname{vec}(KB) = (I_r \otimes K)\operatorname{vec}(B)$.

Proposition 2 (self-adjointness and positive semidefiniteness). Assume K is symmetric positive semidefinite and $\lambda \geq 0$. Then $\mathcal{A}$ is self-adjoint with respect to the Frobenius inner product $\langle U, V\rangle_F = \operatorname{trace}(U^\top V)$, and satisfies

$$\langle W, \mathcal{A}(W)\rangle_F = \|\mathcal{P}_\Omega(KWZ^\top)\|_F^2 + \lambda \langle W, KW\rangle_F \;\geq\; 0. \tag{60}$$

If K is positive definite and $\lambda > 0$, then $\mathcal{A}$ is positive definite.

Proof sketch. Using cyclic trace identities and that $\mathcal{P}_\Omega$ is an orthogonal projector,

$$\langle U, K\,\mathcal{P}_\Omega(KVZ^\top)Z\rangle_F = \langle \mathcal{P}_\Omega(KUZ^\top), \mathcal{P}_\Omega(KVZ^\top)\rangle_F, \tag{61}$$

which is symmetric in U, V. Taking $U = V = W$ yields the norm identity. Strict positivity follows when $K \succ 0$ and $\lambda > 0$ because then $\langle W, KW\rangle_F > 0$ for $W \neq 0$.

These propositions justify the use of (preconditioned) conjugate gradients on the matrix-free operator $\mathcal{A}$ for $\lambda > 0$ and $K \succ 0$ (the common case in RKHS regression when λ includes ridge regularization).

4.2 Matrix-free evaluation of $\mathcal{A}(W)$ using only q observations

A direct evaluation of $\mathcal{A}(W)$ via $KWZ^\top$ would require forming an $n \times M$ dense matrix, which is infeasible for $M \gg n$ and contradicts the requirement to avoid $\mathcal{O}(N)$ or even $\mathcal{O}(nM)$ computations. This section derives a matrix-free procedure that touches only the q observed entries of the unfolding and uses kernel matrix–vector multiplications with K.

Observed-entry representation. Represent Ω in coordinate (COO) form by arrays (i_t, m_t) for $t = 1, \ldots, q$ with $i_t \in [n]$ and $m_t \in [M]$. Let $z_m^\top \in \mathbb{R}^{1\times r}$ denote row m of Z, and let $x_i^\top \in \mathbb{R}^{1\times r}$ denote row i of a matrix $X \in \mathbb{R}^{n\times r}$.

Given W, define $X := KW \in \mathbb{R}^{n\times r}$. For any observed index (i_t, m_t),

$$(KWZ^\top)_{i_t m_t} = x_{i_t}^\top z_{m_t}. \tag{62}$$

Hence the masked matrix $E := \mathcal{P}_\Omega(KWZ^\top)$ is fully characterized by the q scalars

$$e_t := x_{i_t}^\top z_{m_t}, \qquad t = 1, \ldots, q, \tag{63}$$

with all unobserved positions implicitly equal to zero.

Key identity (sparse accumulation for EZ). The product $EZ \in \mathbb{R}^{n\times r}$ can be computed without forming E by accumulating contributions from observed entries:

$$(EZ)_{i,:} = \sum_{t:i_t=i} e_t\, z_{m_t}^\top. \tag{64}$$

This is a sum of q rank-one row updates of length r. Therefore, EZ can be computed in $\mathcal{O}(qr)$ arithmetic operations, plus the cost of retrieving the involved rows z_{m_t}.

Matrix-free matvec routine. Combining (56) with (63) and (64) yields the following matrix-free evaluation of $Y = \mathcal{A}(W)$:

$$X \leftarrow KW, \tag{65}$$

$$\text{for } t = 1, \ldots, q: \quad e_t \leftarrow x_{i_t}^\top z_{m_t}, \tag{66}$$

$$G \leftarrow 0 \in \mathbb{R}^{n\times r}, \tag{67}$$

$$\text{for } t = 1, \ldots, q: \quad G_{i_t,:} \leftarrow G_{i_t,:} + e_t\, z_{m_t}^\top, \tag{68}$$

$$Y \leftarrow KG + \lambda X. \tag{69}$$

In this routine, the $n \times r$ matrix G equals EZ and is the only intermediate that scales with nr.

Complexity model. To express complexity without committing to a specific kernel linear algebra backend, let $\mathrm{cost}_K(n, r)$ denote the cost of multiplying K by an $n \times r$ dense matrix (equivalently, r kernel matrix–vector products). For a dense stored kernel, $\mathrm{cost}_K(n, r) = \mathcal{O}(n^2 r)$ time and $\mathcal{O}(n^2)$ memory to store K. If K is available only implicitly through a kernel function, $\mathrm{cost}_K(n, r)$ may differ; the matvec routine remains valid provided one can evaluate KV for $V \in \mathbb{R}^{n\times r}$.

The remaining work in (66)–(68) is dominated by inner products and scaled vector additions of length r for each of the q observations, for a total of $\mathcal{O}(qr)$ arithmetic operations. Memory access to rows z_{m_t} also occurs q times. In implementations that form Z explicitly, this is a row gather. In implementations that generate each z_{m_t} on demand from the factor matrices of other modes, one can treat the cost of evaluating z_{m_t} as a separate term; the present chapter focuses on the linear algebra given access to the relevant rows of Z.

Overall, one application of $\mathcal{A}$ costs

$$\mathcal{O}\big(\mathrm{cost}_K(n, r) + qr\big) \tag{70}$$

time and requires $\mathcal{O}(nr + q)$ working memory in addition to storage of kernel and factor data. Importantly, the matvec does not form objects of size nM or N.

Right-hand side formation. The system (58) requires $C = KB$. In CP-ALS notation, $B = TZ$ is the MTTKRP. For missing data, T contains zeros at unobserved locations, so B can also be computed by sparse accumulation over q observed entries: the mth column index of the unfolding corresponds to a multi-index over the remaining tensor modes, and the corresponding row of Z is the Khatri–Rao product of the other factor rows. For each observed entry T_{i_t,m_t}, one accumulates $T_{i_t,m_t} z_{m_t}^\top$ into row i_t of B. This costs $\mathcal{O}(qr)$ time given access to z_{m_t}. Multiplying by K then costs $\mathrm{cost}_K(n, r)$. As in the matvec, no $n \times M$ object is formed.

4.3 Preconditioned conjugate gradients on the matrix operator

We apply PCG to the SPD system (58). In standard CG notation one works with vectors; here we use the equivalent formulation with matrices and the Frobenius inner product. Define the residual

$$R(W) := C - \mathcal{A}(W) \in \mathbb{R}^{n\times r}. \tag{71}$$

PCG requires a preconditioner $\mathcal{M} : \mathbb{R}^{n\times r} \to \mathbb{R}^{n\times r}$ that is SPD and inexpensive to apply. Each iteration consists of one application of $\mathcal{A}$, one application of $\mathcal{M}^{-1}$, and several Frobenius inner products and axpy-style updates on $n \times r$ matrices.

Stopping criteria. A typical stopping condition uses the preconditioned residual norm, for example

$$\|\mathcal{M}^{-1/2} R(W_j)\|_F \le \varepsilon \, \|\mathcal{M}^{-1/2} R(W_0)\|_F, \tag{72}$$

or alternatively $\|R(W_j)\|_F \le \varepsilon \, \|C\|_F$. The choice affects practical iteration counts but does not change the matrix-free structure.

Per-iteration cost. Given the matvec cost (39), the iteration cost is

$$\mathcal{O}\big(\mathrm{cost}_K(n, r) + qr + \mathrm{cost}_{\mathcal{M}^{-1}}(n, r)\big), \tag{73}$$

where $\mathrm{cost}_{\mathcal{M}^{-1}}(n, r)$ denotes the cost of applying the chosen preconditioner. The basic linear algebra (inner products and updates) adds $\mathcal{O}(nr)$ per iteration, which is dominated by qr when $q \gg n$.

Conditioning considerations (informal). The operator $\mathcal{A}$ combines (i) a data-fit term that involves masking and the Khatri–Rao product Z and (ii) a Tikhonov term λKW. Without preconditioning, convergence is governed by the spectrum of the implicit $nr \times nr$ matrix. Missingness introduces nonuniform sampling across columns and rows of the unfolding, often leading to ill-conditioning. The preconditioner below is designed to track the dominant Kronecker structure $(\cdot \otimes K)$ while approximating the effect of the sampling mask.

4.4 Kronecker block preconditioning

This section constructs a preconditioner by approximating the masked normal matrix $(Z \otimes K)^\top SS^\top (Z \otimes K)$ with a Kronecker product that separates the n and r dimensions. The resulting preconditioner can be applied to an $n \times r$ matrix using a small $r \times r$ solve and a kernel solve (or approximation thereof).

A separable approximation to the sampling mask. Define row and column observation counts for the unfolding:

$$\rho_i := |\{m : (i,m) \in \Omega\}|, \qquad \gamma_m := |\{i : (i,m) \in \Omega\}|. \tag{74}$$

These are the degrees in the bipartite graph induced by Ω. The exact mask $\mathcal{P}_\Omega$ is not separable. A common approximation replaces it by a separable weighting that preserves marginal counts. One convenient choice is the column-weighted model

$$\mathcal{P}_\Omega(X) \;\approx\; XD_\gamma, \qquad D_\gamma := \operatorname{diag}(\gamma_1, \ldots, \gamma_M), \tag{75}$$

interpreting γ_m as the expected number of observed rows in column m. This approximation is heuristic and is introduced only for preconditioning; PCG still uses the exact operator $\mathcal{A}$.

Under this approximation,

$$\mathcal{P}_\Omega(KWZ^\top)Z \;\approx\; (KWZ^\top D_\gamma)Z = KW(Z^\top D_\gamma Z). \tag{76}$$

Define the $r \times r$ Gram-like matrix

$$G := Z^\top D_\gamma Z \in \mathbb{R}^{r\times r}. \tag{77}$$

Then $\mathcal{A}(W)$ is approximated by

$$\mathcal{A}(W) \;\approx\; K\,(KWG) + \lambda KW. \tag{78}$$

This suggests a preconditioner that mirrors the action of the exact operator but is separable with respect to K and G.

Definition (Kronecker block preconditioner). Assume $K \succ 0$ and $G \succeq 0$. Define

$$\mathcal{M}(W) := K\,(W(G + \lambda I_r)). \tag{79}$$

Equivalently, in vectorized form,

$$\operatorname{vec}(\mathcal{M}(W)) = ((G + \lambda I_r)^\top \otimes K)\operatorname{vec}(W). \tag{80}$$

Since $G + \lambda I_r$ is symmetric, the transpose is immaterial.

Proposition 3 (SPD and Kronecker structure). If $K \succ 0$ and $G + \lambda I_r \succ 0$, then $\mathcal{M}$ is SPD on $\mathbb{R}^{n\times r}$. Moreover, $\mathcal{M}^{-1}$ can be applied by two independent solves:

$$\mathcal{M}^{-1}(Y) = K^{-1}Y\,(G + \lambda I_r)^{-1}. \tag{81}$$

Proof sketch. SPD follows from $\langle W, \mathcal{M}(W)\rangle_F = \operatorname{trace}(W^\top KW(G+\lambda I_r)) > 0$ for $W \neq 0$ when both factors are SPD. The inverse formula follows from associativity of matrix multiplication.

Computing G without forming Z (optional). When Z is large, forming $G = Z^\top D_\gamma Z$ explicitly from Z costs $\mathcal{O}(Mr^2)$ time and requires access to all M rows. In missing-data settings, one can compute a surrogate $\widetilde{G}$ using only observed columns m_t that appear in Ω. Let $\mathcal{M}_\Omega := \{m_t : t = 1, \ldots, q\}$ be the set of involved columns. Then

$$\widetilde{G} := \sum_{m\in\mathcal{M}_\Omega} \gamma_m\, z_m z_m^\top \tag{82}$$

can be accumulated in $\mathcal{O}(|\mathcal{M}_\Omega|r^2)$ time given access to those rows. This is exact when $\mathcal{M}_\Omega = [M]$ and provides a consistent preconditioner otherwise.

Applying $\mathcal{M}^{-1}$ in practice. The application (81) requires solving with $G + \lambda I_r$ and with K.

First, $G + \lambda I_r$ is only $r \times r$ and can be factorized once per mode-k subproblem, for example by a Cholesky factorization if it is SPD. The cost is $\mathcal{O}(r^3)$ time and $\mathcal{O}(r^2)$ memory, negligible relative to operations involving q.

Second, solving with K depends on how K is represented. When K is formed and n is moderate, one can factorize K once at cost $\mathcal{O}(n^3)$ and apply K^{-1} to r right-hand sides at cost $\mathcal{O}(n^2 r)$ per preconditioner application. When K is too large for direct factorization, one can replace K^{-1} by an approximate inverse. One conservative option is to use a diagonal approximation $K \approx \mathrm{diag}(K)$, yielding a Jacobi-like preconditioner

$$\widehat{\mathcal{M}}^{-1}(Y) := \mathrm{diag}(K)^{-1} Y \, (G + \lambda I_r)^{-1}, \tag{83}$$

which is inexpensive and preserves separability. Another option is to apply an inner iterative solve with K when kernel matvecs are available; in that case, the overall scheme becomes a flexible or nested Krylov method. This manuscript does not assume nested solves are executed, and uses the preconditioner definition (42) as the structural target.

Why this preconditioner is appropriate. The exact operator (56) has the structure $K(\cdot) + \lambda K(\cdot)$, so left-multiplication by K is exact in both terms. The nonseparable part arises from $\mathcal{P}_\Omega(\cdot)Z$. Replacing $\mathcal{P}_\Omega$ by a separable diagonal weighting yields the approximate right-multiplication by G, hence $W \mapsto W(G + \lambda I_r)$ captures the main coupling between rank components induced by Z and sampling intensity.

Alternative block-diagonal variant (rowwise decoupling). A simpler preconditioner ignores cross-component coupling and uses only a diagonal approximation of G:

$$\mathcal{M}_{\mathrm{diag}}(W) := K \, (W(\mathrm{diag}(G) + \lambda I_r)). \tag{84}$$

This reduces the $r \times r$ solve to r scalar divisions. It is weaker but often effective when columns of Z are close to orthogonal after weighting.

4.5 End-to-end complexity and memory

We summarize the asymptotic costs under the assumptions $n, r < q \ll N$ and without forming any N-scale arrays.

One PCG iteration. Using (65)–(69), one application of $\mathcal{A}$ costs $\mathrm{cost}_K(n, r) + \mathcal{O}(qr)$. Applying the Kronecker block preconditioner requires solving with $G + \lambda I_r$ (cost $\mathcal{O}(r^2 n)$ for right-multiplication by its inverse if applied via a factorization, since each of the n rows is transformed by an $r \times r$ triangular solve) and applying K^{-1} to an $n \times r$ matrix. If K^{-1} is applied via a precomputed factorization, this is $\mathcal{O}(n^2 r)$ time; under a diagonal approximation it is $\mathcal{O}(nr)$.

Thus, under direct solves with K,

$$\mathrm{cost}_{\mathrm{iter}} = \mathcal{O}(qr) + \mathcal{O}(n^2 r) + \mathcal{O}(nr^2), \tag{85}$$

and typically $\mathcal{O}(qr + n^2 r)$ dominates. Under a diagonal K approximation,

$$\mathrm{cost}_{\mathrm{iter}} = \mathcal{O}(qr) + \mathcal{O}(nr^2). \tag{86}$$

In both cases, the iteration cost is independent of N and does not scale with M except through access to the involved rows of Z.

Precomputation. The preconditioner requires G (or a surrogate). If Z is explicitly available, computing $G = Z^\top D_\gamma Z$ costs $\mathcal{O}(Mr^2)$, which may be too expensive when M is large. Using only the observed column set $\mathcal{M}_\Omega$ costs $\mathcal{O}(|\mathcal{M}_\Omega| r^2)$ and satisfies $|\mathcal{M}_\Omega| \le q$. Computing the counts γ_m requires a single pass over observations, costing $\mathcal{O}(q)$.

If a factorization of K is used for K^{-1}, it costs $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$ memory once. This may be amortized across multiple right-hand sides and across repeated solves for the same mode-k kernel. When K cannot be factorized, the preconditioner can still be used with a cheaper approximation to K^{-1}, trading iteration count for reduced setup cost.

Memory footprint. The matrix-free PCG state consists of a constant number of $n \times r$ matrices (current iterate, residual, search direction, and a few temporaries), requiring $\mathcal{O}(nr)$ memory. The observation list requires $\mathcal{O}(q)$ indices and values (for forming B and for evaluating (66)–(68)). The kernel storage depends on representation: explicit dense storage requires $\mathcal{O}(n^2)$, while implicit representations shift cost to kernel matvec routines.

Discussion of scalability in q and r. For fixed n and moderate r, the dominant cost in missing-data regimes is $\mathcal{O}(qr)$ from sparse accumulation. This dependence is unavoidable because each observed entry influences the normal equations. The preconditioner aims to reduce the number of iterations so that only a small multiple of this $\mathcal{O}(qr)$ work is performed.

Practical note on data access patterns. The accumulation (68) benefits from grouping observations by their row index i_t so that updates to $G_{i_t,:}$ are contiguous in memory. This can be done by sorting the observation list by i_t once. Sorting costs $\mathcal{O}(q \log q)$ and can be reused for all PCG iterations within the mode-k subproblem. This reordering does not change the mathematical operator and still avoids any computation of order N.

4.6 Summary

The mode-k RKHS subproblem yields an SPD system in nr unknowns. By exploiting the identity $\mathrm{vec}(KWZ^\top) = (Z \otimes K)\mathrm{vec}(W)$ and the fact that the sampling operator $SS^\top$ is a projector onto observed entries, the linear operator can be applied in a matrix-free fashion using $\mathcal{O}(qr)$ sparse accumulation plus two kernel multiplications by K. A Kronecker block preconditioner $\mathcal{M}(W) = K\,(W(G + \lambda I_r))$ follows from a separable approximation to the sampling pattern and can be applied via an $r \times r$ solve together with a solve (or approximation) for K. This combination enables PCG to solve the $nr \times nr$ system without explicit formation and without any operations scaling with N.

5 Complexity, practical variants, and alternative preconditioners

5.1 Preconditioned conjugate gradient formulation

This chapter addresses the computational cost of solving the mode-k RKHS subproblem when the tensor is only partially observed.

Standing assumption. Throughout this chapter, assume $\lambda > 0$ and that the kernel Gram matrix used in the linear system is strictly positive definite, denoted $K \succ 0$. When the nominal Gram matrix is only positive semidefinite, one may replace it by a stabilized matrix $K \leftarrow K + \delta I_n$ with $\delta > 0$; all statements about symmetry, positive definiteness, and applicability of conjugate gradients refer to this stabilized choice. The unknown is $W \in \mathbb{R}^{n \times r}$ with vectorization $x = \mathrm{vec}(W) \in \mathbb{R}^{nr}$. The linear system has the symmetric form

$$\mathcal{A}x = b, \tag{87}$$

with

$$\mathcal{A} = (Z \otimes K)^\top SS^\top (Z \otimes K) + \lambda(I_r \otimes K), \tag{88}$$

$$b = (I_r \otimes K)\,\mathrm{vec}(B) = \mathrm{vec}(KB). \tag{89}$$

Here $K \in \mathbb{R}^{n \times n}$ is the kernel Gram matrix for mode k, $Z \in \mathbb{R}^{M \times r}$ is the Khatri Rao product of the other factors, $S \in \mathbb{R}^{N \times q}$ selects the q observed entries of the tensor (equivalently, of its mode-k unfolding), and $\lambda > 0$ is a Tikhonov regularization parameter. The matrix $SS^\top$ is an orthogonal projector onto the observed coordinates, hence symmetric positive semidefinite. By the standing assumption $\lambda > 0$ and $K \succ 0$, the term $\lambda(I_r \otimes K)$ is symmetric positive definite, and therefore $\mathcal{A}$ is symmetric positive definite. Conjugate gradient methods are thus applicable to (87) under this assumption. The central constraint is that $N = nM$ is large and $q \ll N$, so any algorithm whose dominant cost scales with N is unacceptable. Forming the Kronecker product $Z \otimes K$ explicitly would already require storage and time proportional to nMr, and forming $\mathcal{A}$ would

require operations on an $nr \times nr$ matrix. We therefore seek a matrix-free iterative method. In PCG, one chooses an easily invertible symmetric positive definite preconditioner P and applies conjugate gradients to the equivalent system $P^{-1}\mathcal{A}x = P^{-1}b$ (or, more precisely, to the symmetrically preconditioned form). The per-iteration requirements are only (i) evaluation of the matrix-vector product $v \mapsto \mathcal{A}v$ and (ii) application of P^{-1} to a vector. The remaining cost is controlled by the number of iterations, which depends on a standard discretization properties of the preconditioned operator.

5.2 Matrix-free operator evaluation and explicit operation counts

We derive a procedure to compute $y = \mathcal{A}x$ for a given $x = \mathrm{vec}(W)$ without forming $Z \otimes K$ or $SS^\top$. The derivation proceeds by rewriting the Kronecker actions as matrix multiplications and by replacing the selection matrix with a sparse masking operator.

Kronecker identities. For conforming matrices, the identity

$$(B \otimes A)\,\mathrm{vec}(X) = \mathrm{vec}(AXB^\top) \tag{90}$$

implies

$$(Z \otimes K)\,\mathrm{vec}(W) = \mathrm{vec}(KWZ^\top). \tag{91}$$

Similarly, using $(Z \otimes K)^\top = (Z^\top \otimes K)$ and applying (90) with $A = K$ and $B = Z^\top$ yields

$$(Z \otimes K)^\top\,\mathrm{vec}(Y) = \mathrm{vec}(KYZ) \tag{92}$$

for any $Y \in \mathbb{R}^{n\times M}$.

Masking operator induced by observations. Let $\Omega \subseteq [n] \times [M]$ denote the set of observed coordinates in the mode-k unfolding, with $|\Omega| = q$. Define the linear operator $\mathcal{P}_\Omega : \mathbb{R}^{n\times M} \to \mathbb{R}^{n\times M}$ that keeps entries in Ω and sets the rest to zero. Then the selection projector satisfies

$$SS^\top\,\mathrm{vec}(Y) = \mathrm{vec}(\mathcal{P}_\Omega(Y)). \tag{93}$$

This representation is convenient because $\mathcal{P}_\Omega(Y)$ can be constructed or applied using only the q observed indices.

Derivation (matrix-free matvec). Let $x = \mathrm{vec}(W)$. The product $y = \mathcal{A}x$ has two contributions. First, the regularization term satisfies

$$\lambda(I_r \otimes K)\,\mathrm{vec}(W) = \lambda\,\mathrm{vec}(KW). \tag{94}$$

Second, define

$$U = KW \in \mathbb{R}^{n\times r}, \qquad Y = UZ^\top \in \mathbb{R}^{n\times M}. \tag{95}$$

Then by (91) we have $(Z \otimes K)x = \mathrm{vec}(Y)$. Applying the observation projector gives

$$SS^\top(Z \otimes K)x = \mathrm{vec}(\mathcal{P}_\Omega(Y)). \tag{96}$$

Finally, by (92),

$$(Z \otimes K)^\top SS^\top(Z \otimes K)x = \mathrm{vec}\big(K\,\mathcal{P}_\Omega(Y)\,Z\big). \tag{97}$$

Combining (94) and (97) yields the explicit matrix form of the matvec:

$$\mathcal{A}\,\mathrm{vec}(W) = \mathrm{vec}\Big(K\,\mathcal{P}_\Omega(KWZ^\top)\,Z + \lambda KW\Big). \tag{98}$$

Therefore a matrix-free implementation can operate on W directly, avoiding the explicit $nr \times nr$ representation.

Sparse evaluation of $\mathcal{P}_\Omega(UZ^\top)$ and $\mathcal{P}_\Omega(Y)Z$. The naive computation of $Y = UZ^\top$ costs $O(nMr)$ and is unacceptable. Instead, $\mathcal{P}_\Omega(Y)$ is only needed on the observed entries. For each observed pair $(i, m) \in \Omega$,

$$Y_{im} = (UZ^\top)_{im} = \sum_{j=1}^{r} U_{ij} Z_{mj}. \tag{99}$$

Thus the q observed values of Y can be computed with q inner products of length r, using only rows of Z indexed by the observed mode-$\neq k$ multi-indices mapped to m. The subsequent product $\mathcal{P}_\Omega(Y)Z$ also admits a sparse accumulation. Define $G = \mathcal{P}_\Omega(Y)Z \in \mathbb{R}^{n\times r}$. Its entries satisfy

$$G_{ij} = \sum_{m=1}^{M} \mathcal{P}_\Omega(Y)_{im} Z_{mj} = \sum_{(i,m)\in\Omega} Y_{im} Z_{mj}. \tag{100}$$

Given the q nonzero entries Y_{im}, one can accumulate contributions to the r components of row i with cost proportional to r per observation.

Derivation (operation count for one matvec). We count arithmetic operations under the explicit assumptions that (i) K is stored as a dense matrix and multiplying K by an $n \times r$ matrix costs $O(n^2 r)$, and (ii) access to the observed index set Ω is available with $O(1)$ overhead per observed entry.

In addition, we distinguish arithmetic in $\mathbb{R}$ from the cost of obtaining the needed rows of Z. Let cost_Z denote the cost to obtain one requested row $Z_{m,:} \in \mathbb{R}^{1\times r}$ in a form that supports the subsequent $O(r)$ inner products and accumulations. When Z is stored explicitly, cost_Z models the cost of a row fetch (including memory access). When Z is not stored and $Z_{m,:}$ is generated on demand from the other factor matrices, cost_Z includes that generation work.

Step 1 computes $U = KW$, which costs $O(n^2 r)$. Step 2 computes the q observed entries of $Y = UZ^\top$ using (99). Each entry is an inner product of length r, hence $O(r)$ arithmetic, for a total of $O(qr)$ arithmetic, plus up to $O(q\,\mathrm{cost}_Z)$ to obtain the required rows of Z. Step 3 forms $G = \mathcal{P}_\Omega(Y)Z$ using (100). Each observed entry contributes to r accumulators, hence $O(r)$ arithmetic per observation, for a total of $O(qr)$ arithmetic. The required Z rows can be reused from Step 2 when cached for the duration of the matvec; in that case the additional Z-access cost in Step 3 is avoided. Without caching, Step 3 incurs an additional $O(q\,\mathrm{cost}_Z)$. Step 4 computes $H = KG$, which costs $O(n^2 r)$. Step 5 adds the regularization contribution λKW, which can be obtained either by reusing U from Step 1 (so only $O(nr)$ for the scaled add) or by a separate multiplication if reuse is not available. Reuse gives the smallest cost. Consequently, one matrix-vector product with $\mathcal{A}$ can be computed in

$$O\big(n^2 r + qr + q\,\mathrm{cost}_Z\big) \tag{101}$$

operations when Z rows are obtained once per observation and reused within the matvec. Without such reuse, the bound becomes $O\big(n^2 r + qr + 2q\,\mathrm{cost}_Z\big)$. In both cases, the computation avoids forming any object of size proportional to $N = nM$.

Memory requirements. The matrix-free procedure stores $W, U, G, H \in \mathbb{R}^{n\times r}$ and uses access to the observed index list Ω. No intermediate of size $n \times M$ is formed.

When Z is stored explicitly, additional memory $O(Mr)$ is required for Z, and the footprint is dominated by storing K (dense storage $O(n^2)$) and Z (dense storage $O(Mr)$), together with the observation list (storage $O(q)$ plus the observed values of the tensor).

When M is large, it is often preferable to avoid explicit storage of Z and instead generate only the required rows $Z_{m,:}$ on demand from the other factor matrices. In this on-demand regime, the dominant stored objects become K (or a representation sufficient to apply it), the factor matrices for the other modes (whose total storage is the sum of their mode sizes times r), and the observation list. This avoids any $O(Mr)$ storage, at the price of the additional $q\,\mathrm{cost}_Z$ term in (101).

5.3 Kernel preconditioning and a standard discretization considerations

A natural preconditioner exploits the fact that $\lambda(I_r \otimes K)$ is always present and is expected to dominate the smallest eigenvalues of $\mathcal{A}$. We therefore consider

$$P = I_r \otimes K. \tag{102}$$

By the standing assumption $K \succ 0$, this choice is symmetric positive definite. Its appeal is that it can be applied exactly using linear solves with K, and it respects the Kronecker structure of the unknown.

Application of P^{-1}. For $v = \mathrm{vec}(V)$ with $V \in \mathbb{R}^{n\times r}$,

$$P^{-1}v = (I_r \otimes K^{-1})\,\mathrm{vec}(V) = \mathrm{vec}(K^{-1}V). \tag{103}$$

Thus applying the preconditioner reduces to solving r linear systems with coefficient matrix K. When K is dense, one proposed approach is to compute a Cholesky factorization $K = LL^\top$ once and apply K^{-1} via two triangular solves for each right-hand side.

Symmetric preconditioning. The standard analysis of PCG concerns the spectrum of the symmetrically preconditioned matrix

$$\widetilde{\mathcal{A}} = P^{-1/2}\,\mathcal{A}\,P^{-1/2} = \lambda I_{nr} + P^{-1/2}(Z \otimes K)^\top SS^\top (Z \otimes K)P^{-1/2}. \tag{104}$$

This form makes positivity immediate: the second term is symmetric positive semidefinite because it is the product of a matrix with its transpose composed with a projector.

Theorem (a standard discretization bound under a domination assumption). Let

$$\mathcal{G} = (Z \otimes K)^\top SS^\top (Z \otimes K). \tag{105}$$

Assume there exists a scalar $\gamma \geq 0$ such that

$$\mathcal{G} \preceq \gamma(I_r \otimes K). \tag{106}$$

Then the eigenvalues of the symmetrically preconditioned system satisfy

$$\lambda \leq \lambda_{\min}(\widetilde{\mathcal{A}}) \leq \lambda_{\max}(\widetilde{\mathcal{A}}) \leq \lambda + \gamma, \tag{107}$$

and the condition number is bounded by

$$\kappa(\widetilde{\mathcal{A}}) \leq 1 + \gamma/\lambda. \tag{108}$$

Derivation. Under (106), multiplying on both sides by $P^{-1/2}$ yields

$$P^{-1/2}\mathcal{G}P^{-1/2} \preceq \gamma I_{nr}. \tag{109}$$

From (104) we have $\widetilde{\mathcal{A}} = \lambda I_{nr} + P^{-1/2}\mathcal{G}P^{-1/2}$ with $P^{-1/2}\mathcal{G}P^{-1/2} \succeq 0$. Therefore all eigenvalues lie in $[\lambda, \lambda+\gamma]$, which implies (108). The assumption (106) is an analytical hypothesis. It can be interpreted as requiring that the data term, after masking by observations and contraction by Z, is not too large relative to $I_r \otimes K$. When such a relationship holds, the preconditioner $P = I_r \otimes K$ clusters eigenvalues and reduces the number of PCG iterations.

Practical note on the regime $n, r < q \ll N$. In the regime where q is large enough to inform the data term but still much smaller than N, the data term contributes meaningful curvature while the regularization term ensures stability. The preconditioner in (102) targets the regularization geometry directly and does not require any operation involving N.

5.4 Complexity analysis and practical variants

This section summarizes setup costs, per-iteration costs, and memory requirements for matrix-free PCG under explicit assumptions. The discussion remains analytical and does not rely on empirical timings.

Setup cost. The preconditioner $P = I_r \otimes K$ requires the ability to apply K^{-1} to multiple right-hand sides. A direct realization is a Cholesky factorization $K = LL^\top$, which costs

$$O(n^3) \tag{110}$$

operations and stores L in $O(n^2)$ memory. This is a one-time cost per mode-k subproblem solve. When n is large, this setup can dominate; in such cases, the alternatives discussed in Section 5.5 become relevant.

Per-iteration cost (matvec and preconditioner). Each PCG iteration requires at least one application of $\mathcal{A}$ and one application of P^{-1}. From (101), the matrix-free matvec costs $O(n^2r + qr + q\,\mathrm{cost}_Z)$ under dense-K multiplication and with on-demand access to rows of Z. Applying the preconditioner corresponds to solving $KY = R$ for r right-hand sides. With a Cholesky factor, this costs two triangular solves per right-hand side, hence

$$O(n^2r). \tag{111}$$

Vector updates and inner products in PCG cost $O(nr)$, which is lower order under the standing assumption that $n, r < q$ and that dense K multiplication is the dominant kernel-related cost. Therefore, the total cost per PCG iteration is

$$O\big(n^2r + qr + q\,\mathrm{cost}_Z\big), \tag{112}$$

up to constant factors associated with the precise PCG variant.

Cost of the right-hand side. The right-hand side is $b = \mathrm{vec}(KB)$. If B is available, computing KB costs $O(n^2r)$. The construction of $B = TZ$ from partially observed data can itself be carried out with q-sparse contractions, but this cost depends on how observed indices are stored and is not expanded here.

Total cost and iteration count. Let m denote the number of PCG iterations required to achieve a prescribed tolerance. Then the total complexity of one solve is

$$O\big(n^3 + m(n^2r + qr + q\,\mathrm{cost}_Z)\big) \tag{113}$$

under the Cholesky-based realization of P^{-1}. The iteration count m depends on the eigenvalue distribution of the preconditioned operator, and bounds such as (108) can be used to motivate why $P = I_r \otimes K$ can be beneficial when (106) is plausible.

Variant: avoiding explicit K storage (proposed). In some RKHS settings, K is not formed explicitly and only matrix-vector products with K are available through kernel evaluations. A proposed variant is to replace dense multiplications by routines that compute Ku in $O(n\,c_\kappa)$ time, where c_κ denotes the cost of evaluating the kernel function and assembling the sum for one vector. In such a setting, Step 1 and Step 4 in the matvec derivation are replaced by r applications of the kernel operator, and the nominal $O(n^2r)$ term in (101) becomes $O(nrc_\kappa)$. This variant is conceptually straightforward but shifts the bottleneck to kernel evaluation cost and to memory locality.

Variant: inexact preconditioning (proposed). When a direct factorization of K is infeasible, one may apply P^{-1} approximately by running an inner iterative solver on systems with coefficient matrix K. This results in an inexact PCG scheme. A rigorous convergence analysis then requires additional assumptions controlling the preconditioning error and is not pursued here. The main point is that the Kronecker structure ensures that all preconditioning work reduces to repeated solves with K.

Numerical stability considerations. The system matrix contains $\lambda(I_r \otimes K)$, which regularizes directions associated with small eigenvalues of K. When K is ill-conditioned, the Cholesky factorization can still be stable in exact arithmetic, but finite-precision accuracy may deteriorate. Increasing λ improves conditioning but changes the estimator; selecting λ is therefore part of the modeling problem rather than a purely numerical choice.

5.5 Alternative preconditioning strategies (algorithmic proposals)

The preconditioner $P = I_r \otimes K$ is attractive because it is exact, symmetric, and aligned with the regularization term, but it requires efficient solves with K. This section describes alternative preconditioners as proposals motivated by a standard discretization structure. No empirical ranking is claimed.

Diagonal or Jacobi-type preconditioning (proposal). A minimal-cost choice replaces K by its diagonal, defining

$$P_{\mathrm{J}} = I_r \otimes \mathrm{diag}(K). \tag{114}$$

Applying P_{J}^{-1} costs $O(nr)$ and requires only the diagonal of K. This preconditioner can be interpreted as balancing variables according to marginal kernel scales. Its effectiveness depends on how well $\mathrm{diag}(K)$ approximates K in the sense relevant to the spectrum of $\mathcal{A}$. When K has strong off-diagonal structure, P_{J} may leave significant anisotropy.

Block-diagonal preconditioning across components (proposal). The full system couples the r components through Z in the data term, while the regularization is block diagonal across components. A proposal is to incorporate some coupling by approximating $Z^\top Z$ with a diagonal or low-bandwidth matrix D_Z and defining

$$P_{\mathrm{BD}} = D_Z \otimes K + \lambda(I_r \otimes K) = (D_Z + \lambda I_r) \otimes K. \tag{115}$$

When D_Z is diagonal, P_{BD}^{-1} reduces to r independent solves with scaled versions of K, hence it has similar cost to P^{-1} but can better reflect relative component scales induced by Z. This approach remains purely algebraic and does not require access to N.

Nyström or low-rank kernel preconditioning (proposal). When n is large, the $O(n^3)$ setup cost of a Cholesky factor can be prohibitive. A standard kernel approximation strategy is to build a low-rank approximation $K \approx CW_c^{-1}C^\top$ using a subset of columns, where $C \in \mathbb{R}^{n\times m}$ and $W_c \in \mathbb{R}^{m\times m}$ with $m \ll n$. One may then define a preconditioner based on

$$\widehat{K} = CW_c^{-1}C^\top + \delta I_n, \tag{116}$$

with a small ridge $\delta > 0$ to ensure invertibility. Applying $\widehat{K}^{-1}$ can be done using the Woodbury identity, resulting in solves dominated by an $m \times m$ system and multiplications by C and $C^\top$. The associated Kronecker preconditioner is $\widehat{P} = I_r \otimes \widehat{K}$. The quality of this approach depends on how well $\widehat{K}$ approximates K in the a standard discretization region emphasized by the iterates.

Data-aware preconditioning (proposal). The data term is $(Z \otimes K)^\top SS^\top (Z \otimes K)$, which can be viewed as a masked normal-equations matrix. A proposal is to approximate this term by a separable form such as

$$(Z \otimes K)^\top SS^\top (Z \otimes K) \approx (Z^\top D_\Omega Z) \otimes K, \tag{117}$$

where D_Ω is a diagonal matrix encoding how frequently each row index $m \in [M]$ appears among observations. This approximation replaces a complicated masking pattern by marginal observation counts. The resulting preconditioner has the form $(Z^\top D_\Omega Z + \lambda I_r) \otimes K$, which can be inverted by solving an $r \times r$ system and r kernel systems. This proposal is motivated by the idea that observation density influences effective conditioning, but its validity depends on how concentrated the observation pattern is.

Summary of trade-offs. The baseline preconditioner $P = I_r \otimes K$ is exact with respect to the regularization geometry and is straightforward to apply when a factorization of K is available. Cheaper preconditioners, such as Jacobi-type schemes, reduce setup and application costs but may leave more a standard discretization spread. Low-rank kernel preconditioners reduce the $O(n^3)$ setup cost at the price of approximation error. Data-aware Kronecker approximations attempt to capture observation effects while preserving separability. In all cases, the matrix-free matvec from Section 5.2 remains unchanged and continues to avoid any computation scaling with N.

6 Conclusion and Outlook

We analyzed the mode-k update arising in alternating optimization for a partially observed CP decomposition when the kth factor is constrained by an RKHS penalty. Using the representer parametrization $A_k = KW$, we expressed the update as a linear system in $\mathrm{vec}(W)$ with a masked normal operator of the form $(Z \otimes K)^\top SS^\top (Z \otimes K) + \lambda(I_r \otimes K)$. The main practical outcome is that this system can be solved without forming any $nr \times nr$ matrices and without any computation that scales with the full tensor size N: both the right-hand side and each matrix–vector product can be implemented by looping over the q observed entries, using on-demand access to the corresponding rows of Z and applying K (or K^{-1} in preconditioning) only along the mode-k dimension.

We further outlined a preconditioned conjugate gradient strategy tailored to the RKHS geometry. The baseline Kronecker preconditioner $P = I_r \otimes K$ is simple, symmetric, and directly aligned with the regularization term; applying it reduces to r kernel solves. We also discussed practical variants, including diagonal/block-diagonal modifications and low-rank kernel approximations, that trade setup cost against preconditioning quality.

At the current stage, the work provides a detailed algorithmic blueprint and complexity accounting for matrix-free PCG in the regime $n, r < q \ll N$, but it does not yet close a complete end-to-end convergence theory for the overall alternating CP procedure under missingness, nor does it provide a verified, data-dependent characterization of iteration counts beyond qualitative spectral motivation. Two concrete next steps are to (i) develop sharper, checkable bounds relating the masked data term $(Z \otimes K)^\top SS^\top (Z \otimes K)$ to the Kronecker regularizer so that PCG iteration complexity can be predicted from observation patterns and factor coherence, and (ii) evaluate and compare the proposed preconditioners empirically across kernel choices and sampling regimes, including cases where K must be applied and inverted approximately (e.g., via Nyström or inner iterative solvers). These steps would turn the present matrix-free formulation into a fully validated practical routine for RKHS-constrained tensor factorization with missing entries.

Acknowledgments

Portions of this manuscript were prepared with the assistance of *Basis* (*Research that compiles.*).