Matrix-free Preconditioned Conjugate Gradient for Kernelized ALS with Sparse Observations

Author. Lior Isthmus
Date. 2026-02-13

Abstract

We present a matrix-free solution strategy for a structured normal equation arising in kernelized alternating least squares (ALS) for CP tensor decomposition [2] under sparse (incomplete) observations. The target linear system has dimension $nr \times nr$ and involves a Khatri–Rao product Z, a kernel matrix K (RKHS representation) [5,6], a selection operator encoding $q \ll N$ observed tensor entries, and a Tikhonov term $\lambda(I_r \otimes K)$. We show how to solve the system efficiently using a preconditioned conjugate gradient (PCG) method [3,4] without explicitly forming any $nr \times nr$ matrix and without performing any computation of order N. The key contribution is an explicit, rigorously justified matrix-free routine for the required matrix–vector products, implemented via two kernel multiplications and $O(qdr)$ gather–scatter operations using only the observation list. We also provide practical preconditioners that exploit the kernel structure and admit fast application. Detailed proofs are deferred to appendices.

Keywords. CP decomposition, ALS, sparse tensor completion, RKHS, kernel methods, matrix-free Krylov methods, preconditioned conjugate gradient, Khatri–Rao product.

Contents

1. Introduction

Kernelized tensor decomposition models arise when one or more tensor modes are endowed with a reproducing kernel Hilbert space (RKHS) representation [5,6]. In such settings, ALS updates for CP tensor decomposition can lead to large structured linear systems involving Kronecker products, selection operators for missing data, and kernel matrices [5,6]. A representative instance is the following: one seeks an unknown matrix $W \in \mathbb{R}^{n\times r}$ such that a normal equation of size $nr \times nr$ is satisfied, where n is a mode size and r is the CP rank.

When the tensor is sparsely observed (with q observed entries and $q \ll N$ where N is the total number of tensor entries), it is essential to avoid computations of order N and to refrain from forming dense $nr \times nr$ matrices. This paper develops a complete matrix-free PCG solver for such systems [3,4]. The design is guided by three principles:

1. **Operator-first formulation:** express the normal equation as an operator equation $\mathcal{A}(W) = \mathcal{F}$ on $\mathbb{R}^{n\times r}$, bypassing explicit $nr \times nr$ matrices.
2. **Modular matrix-free evaluation:** implement $W \mapsto \mathcal{A}(W)$ using only (i) kernel multiplications $X \mapsto KX$ and (ii) a single pass over the observation list, thereby ensuring the absence of any $O(N)$ work.
3. **Verifiability:** specify a minimal output package (residual checks and logs) enabling third-party re-computation of the final residual using the same matrix-free operator.

The main text presents the overall logic and complete algorithms; detailed algebraic and complexity proofs are delegated to appendices.

2. Notation and problem setup

We fix integers $d \geq 2$ (tensor order) and $r \geq 1$ (CP rank). For each mode $m \in \{1, \dots, d\}$ let $n_m \in \mathbb{N}$ denote the mode size, and define the total size

$$N := \prod_{m=1}^{d} n_m.$$

We fix a mode index $k \in \{1, \dots, d\}$ and write

$$n := n_k, \qquad M := \prod_{m\neq k} n_m,$$

so that the mode-k unfolding of a tensor has shape $n \times M$ and $N = nM$.

2.1 Sparse observations

We assume that only q entries of the tensor are observed, with $q \ll N$. We represent the observations as a list

$$\mathcal{O} := \{(i_1^{(e)}, \dots, i_d^{(e)}, t^{(e)}) : e = 1, \dots, q\}$$

where $i_m^{(e)} \in \{1, \dots, n_m\}$ is the mode-m index and $t^{(e)} \in \mathbb{R}$ is the observed value.

The mode-k unfolding index pair associated with observation e is

$$i^{(e)} := i_k^{(e)} \in \{1, \dots, n\}, \qquad j^{(e)} \in \{1, \dots, M\},$$

where $j^{(e)}$ is the column index corresponding to the multi-index $(i_1^{(e)}, \dots, i_{k-1}^{(e)}, i_{k+1}^{(e)}, \dots, i_d^{(e)})$. We will **not** require explicit evaluation of $j^{(e)}$; instead, we will generate the associated Khatri–Rao row directly from the multi-index.

Definition 2.1 (Observation set and projection). Let $\Omega \subset \{1, \dots, n\} \times \{1, \dots, M\}$ denote the set of observed unfolding locations:

$$\Omega := \{(i^{(e)}, j^{(e)}) : e = 1, \dots, q\}$$

Define the projection $P_\Omega : \mathbb{R}^{n \times M} \to \mathbb{R}^{n \times M}$ by

$$(P_\Omega(X))_{ij} := \begin{cases} X_{ij}, & (i, j) \in \Omega, \\ 0, & \text{otherwise.} \end{cases}$$

2.2 CP factors and Khatri–Rao structure

Let $A_m \in \mathbb{R}^{n_m \times r}$ denote the CP factor matrix for mode m when $m \neq k$. We define the Khatri–Rao product

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

where $\odot$ denotes the Khatri–Rao product (column-wise Kronecker product) [2].

Definition 2.2 (On-the-fly Khatri–Rao row). Given an observation multi-index $(i_1, \dots, i_d)$, define the associated row vector $z(i_1, \dots, i_d) \in \mathbb{R}^r$ by

$$z(i_1, \dots, i_d)_s := \prod_{m \neq k} (A_m)_{i_m, s}, \qquad s \in \{1, \dots, r\}.$$

Equivalently, if j is the unfolding column index corresponding to $(i_m)_{m \neq k}$, then $z(i_1, \dots, i_d)$ equals the row $Z_{j,:}$.

2.3 Kernelized mode and unknown variable

We assume that the mode-k factor is represented via an RKHS kernel matrix.

Definition 2.3 (Kernelized factor parametrization). Let $K \in \mathbb{R}^{n\times n}$ be a symmetric positive semidefinite kernel matrix [5,6]. The mode-k factor is parametrized as

$$A_k = KW,$$

where $W \in \mathbb{R}^{n\times r}$ is the unknown to be solved for.

We also fix a regularization parameter $\lambda > 0$.

3. Target linear system and objectives

Let $T \in \mathbb{R}^{n\times M}$ denote the mode-k unfolding of the tensor, with missing entries treated as zero (so that $T_{ij} = 0$ for $(i,j) \notin \Omega$). Define the matrix

$$B := TZ \in \mathbb{R}^{n\times r},$$

which is the matricized tensor times Khatri–Rao product (MTTKRP) associated with mode k [2].

The target system is the following normal equation in $\mathrm{vec}(W) \in \mathbb{R}^{nr}$ [1].

$$\Big((Z \otimes K)^{\mathsf{T}} SS^{\mathsf{T}}(Z \otimes K) + \lambda(I_r \otimes K)\Big)\mathrm{vec}(W) = (I_r \otimes K)\mathrm{vec}(B)$$

where $S \in \mathbb{R}^{N\times q}$ is a selection matrix encoding the observed entries, and I_r is the $r \times r$ identity matrix.

Objectives

We aim to provide a complete, implementable solution meeting the following requirements:

- Use an **iterative preconditioned conjugate gradient (PCG)** method.
- Specify at least one effective **preconditioner** and explain how to apply it.
- Explain in detail how to compute the required **matrix–vector products** without forming the $nr \times nr$ matrix.
- Provide a clear **complexity analysis** under the regime $n, r < q \ll N$ while **avoiding any computation of order** N.

4. Matrix-free operator formulation

The central step is to recast the $nr \times nr$ system into an operator equation on matrices $W \in \mathbb{R}^{n\times r}$.

4.1 Algebraic identities

We will use the standard vectorization identity (proved in Appendix A) [7].

$$(Z \otimes K)\mathrm{vec}(W) = \mathrm{vec}(KWZ^{\mathsf{T}}).$$

Moreover, the action of SS^{T} corresponds to zeroing out unobserved entries, which is equivalent to applying the projection P_Ω after reshaping into an $n \times M$ matrix (Appendix B).

4.2 Definition of the matrix-free operator

Definition 4.1 (Matrix-free operator and right-hand side). Define the linear operator $\mathcal{A} : \mathbb{R}^{n\times r} \to \mathbb{R}^{n\times r}$ by

$$\mathcal{A}(W) := K\Big(P_\Omega(KWZ^{\mathsf{T}})Z\Big) + \lambda KW.$$

Define the right-hand side $\mathcal{F} \in \mathbb{R}^{n\times r}$ by

$$\mathcal{F} := KB.$$

The operator equation of interest is

$$\mathcal{A}(W) = \mathcal{F}.$$

Proposition 4.2 (Equivalence to the Kronecker normal equation). For every $W \in \mathbb{R}^{n\times r}$,

$$\mathrm{vec}(\mathcal{A}(W)) = \Big((Z \otimes K)^{\mathsf{T}} SS^{\mathsf{T}} (Z \otimes K) + \lambda(I_r \otimes K)\Big)\mathrm{vec}(W)$$

and $\mathrm{vec}(\mathcal{F}) = (I_r \otimes K)\mathrm{vec}(B)$. Consequently, W satisfies the operator equation (equation) if and only if $\mathrm{vec}(W)$ satisfies the original system (equation).

Outline of proof. Combine the identity (equation) with the interpretation of SS^{T} as an observation projection (Appendix B), and then apply the adjoint identity for $(Z \otimes K)^{\mathsf{T}}$ (Appendix A). The regularization term follows from $\mathrm{vec}(KW) = (I_r \otimes K)\mathrm{vec}(W)$. Full details are given in Appendices A–D.

Remark 4.3 (Why the operator form is useful). Equation (equation) can be solved by PCG using only repeated evaluations of $W \mapsto \mathcal{A}(W)$ and applications of a preconditioner. We will implement $\mathcal{A}(W)$ without forming $Z \in \mathbb{R}^{M\times r}$, without forming $KWZ^{\mathsf{T}} \in \mathbb{R}^{n\times M}$, and without forming any $nr \times nr$ matrices.

5. Matrix-free algorithms for operator application and right-hand side construction

This section provides explicit routines to compute $\mathcal{A}(W)$ and $\mathcal{F}$ using only the observation list $\mathcal{O}$, the fixed factors $A_{m_{m \neq k}}$, and kernel multiplications by K.

5.1 Required primitives

We will treat kernel multiplication as an abstract primitive.

Definition 5.1 (Kernel multiplication primitive). Assume access to a routine `KernelMul(X)` returning KX for any $X \in \mathbb{R}^{n \times r}$. Let $\mathsf{Cost}_K(n, r)$ denote its cost. For a dense explicit K, one may take $\mathsf{Cost}_K(n, r) = O(n^2 r)$. We also assume access to a routine `SolveK(R)` returning $X = K^{-1}R$ for any $R \in \mathbb{R}^{n \times r}$, and denote its cost by $\mathsf{Cost}_{K^{-1}}(n, r)$.

We also use the row-generation rule (equation).

Algorithm 5.2 (On-the-fly Khatri–Rao row). Input: an observation $(i_1, \dots, i_d, t)$ and factor matrices $A_{m_{m \neq k}}$. Output: $z \in \mathbb{R}^r$ with $z = z(i_1, \dots, i_d)$.

```
z ← (1,1,...,1)   R^r
for m in {1,...,d} \ {k} do
    z ← z   A_m[i_m, :]
end for
return z
```

The cost is $O((d-1)r)$ arithmetic operations.

5.2 Matrix-free application of the operator

Algorithm 5.3 (ApplyA: matrix-free evaluation of $\mathcal{A}(W)$). Input: $W \in \mathbb{R}^{n \times r}$, observations $\mathcal{O}$, factors $A_{m_{m \neq k}}$, kernel routine `KernelMul`, regularization $\lambda > 0$. Output: $\mathcal{A}(W)$ as in (equation).

```
Y ← KernelMul(W)              # Y = K W
C ← 0_{n×r}
for e = 1,...,q do
    (i_1,...,i_d,t) ← O[e]
    i ← i_k
    z ← KRRow(i_1,...,i_d)    # z = Z_{j(e),:} (computed on the fly)
    u ←  Y[i,:], z           # scalar inner product
    C[i,:] ← C[i,:] + u z     # rank-1 scatter into row i
end for
return KernelMul(C) +  Y
```

The loop implements the projection $P_\Omega(KWZ^\top)$ implicitly by visiting only observed entries.

5.3 Matrix-free construction of the right-hand side

Algorithm 5.4 (BuildRHS: matrix-free evaluation of $\mathcal{F} = KB$). Input: observations $\mathcal{O}$, factors $A_{m_{m\neq k}}$, kernel routine `KernelMul`. Output: $\mathcal{F} \in \mathbb{R}^{n\times r}$.

```
B ← 0_{n×r}
for e = 1,...,q do
    (i_1,...,i_d,t) ← O[e]
    i ← i_k
    z ← KRRow(i_1,...,i_d)
    B[i,:] ← B[i,:] + t z
end for
return KernelMul(B)           # F = K B
```

Proposition 5.5 (Correctness and cost of Algorithms 5.3–5.4). Algorithm 5.3 returns $\mathcal{A}(W)$ as defined in (equation), and Algorithm 5.4 returns $\mathcal{F}$ as defined in (equation). Their costs are

$$\mathsf{Cost}(\text{ApplyA}) = 2\mathsf{Cost}_K(n, r) + O(qdr),$$

$$\mathsf{Cost}(\text{BuildRHS}) = \mathsf{Cost}_K(n, r) + O(qdr),$$

and neither algorithm performs any computation of order N.

Outline of proof. For ApplyA, set $Y = KW$. For each observation e at unfolding position $(i^{(e)}, j^{(e)})$, the scalar $u^{(e)} = \langle Y_{i^{(e)},:}, Z_{j^{(e)},:}\rangle$ equals $(KWZ^\mathsf{T})_{i^{(e)}j^{(e)}}$. The scatter accumulation $C[i^{(e)}, :] \mathrel{+}= u^{(e)} Z_{j^{(e)},:}$ therefore computes exactly

$$C = P_\Omega(KWZ^\mathsf{T})Z.$$

Multiplying by K and adding λKW yields $\mathcal{A}(W)$. The BuildRHS routine is analogous, using that $B = TZ$ where T is zero on unobserved entries. Complexity follows by counting per-observation work and kernel multiplications. Full details appear in Appendices E–F.

6. Preconditioned conjugate gradient solver and preconditioners

We now specify the PCG method for solving the operator equation (equation) and present preconditioners suitable for the kernelized structure.

6.1 Conditions for PCG applicability

The conjugate gradient method requires a symmetric positive definite (SPD) operator in the underlying Euclidean space [3,4]. Because the regularization involves K, the SPD property depends on definiteness of K.

Assumption 6.1 (Positive definiteness of the kernel). We assume K is symmetric positive definite (SPD). If K is only positive semidefinite, one may replace K by $K + \varepsilon I$ for any $\varepsilon > 0$ to enforce SPD; all statements below then apply to the perturbed system.

Proposition 6.2 (SPD of the normal equation). Under Assumption 6.1 and $\lambda > 0$, the coefficient matrix in (equation) is SPD. Equivalently, the operator $\mathcal{A}$ in (equation) is SPD with respect to the Frobenius inner product on $\mathbb{R}^{n\times r}$ induced by vectorization.

Outline of proof. Let $M := Z \otimes K$ and $P := SS^{\mathsf{T}}$, noting P is symmetric positive semidefinite. Then the coefficient matrix is $M^{\mathsf{T}}PM + \lambda(I_r \otimes K)$. The first term is symmetric positive semidefinite and the second is symmetric positive definite because $I_r \otimes K$ is SPD when K is SPD. The sum is therefore SPD. Full details are in Appendix G.

6.2 Matrix-free PCG

We present PCG in a form that operates on matrices $W \in \mathbb{R}^{n\times r}$ using the matrix-free ApplyA routine.

Algorithm 6.3 (Matrix-free preconditioned conjugate gradient). Input: routines `ApplyA(W)` and `ApplyPInv(R)` implementing $W \mapsto \mathcal{A}(W)$ and $R \mapsto P^{-1}(R)$; right-hand side $\mathcal{F}$; initial guess W_0; tolerance `tol`; maximum iterations `maxit`. Output: approximate solution W to $\mathcal{A}(W) = \mathcal{F}$.

```
W ← W0
R ← F - ApplyA(W)
Z ← ApplyPInv(R)
Pdir ← Z
  ←  R, Z _F
for t = 0,1,2,...,maxit-1 do
    Q ← ApplyA(Pdir)
      ←  / Pdir, Q _F
    W ← W +  Pdir
    R ← R -  Q
    if ||R||_F / ||F||_F  tol then
        break
    end if
    Z ← ApplyPInv(R)
     _new ←  R, Z _F
      ←  _new /
    Pdir ← Z +  Pdir
      ←  _new
end for
return W
```

Here $\langle X, Y\rangle_F := \mathrm{trace}(X^{\mathsf{T}}Y)$ is the Frobenius inner product.

6.3 Preconditioners

We provide two practical preconditioners. The first is recommended as a default because it is simple, cheap, and leverages the guaranteed regularization structure.

Definition 6.4 (Kernel regularization preconditioner). Define $P_1 : \mathbb{R}^{n\times r} \to \mathbb{R}^{n\times r}$ by

$$P_1(W) := \lambda K W.$$

Then $P_1^{-1}(R) = \lambda^{-1}K^{-1}R$ can be applied by solving $KX = R$ for X (columnwise).

This preconditioner captures the SPD component $\lambda(I_r \otimes K)$ of the system.

Definition 6.5 (Separated Kronecker preconditioner). Let $G \in \mathbb{R}^{r\times r}$ be any symmetric positive semidefinite matrix that approximates the coupling induced by Z (e.g., $G \approx Z^\mathsf{T} Z$ or a weighted variant derived from observation frequencies). Define

$$P_2(W) := KW(G + \lambda I_r).$$

When $G + \lambda I_r$ is SPD, one can apply P_2^{-1} by (i) solving $KX = R$ for X, then (ii) right-multiplying by $(G + \lambda I_r)^{-1}$:

$$P_2^{-1}(R) = K^{-1}R(G + \lambda I_r)^{-1}.$$

Remark 6.6 (Applying K^{-1} in practice). If K is stored explicitly and n is moderate, a Cholesky factorization $K = LL^\mathsf{T}$ can be computed once (cost $O(n^3)$) and reused to apply K^{-1} to multiple right-hand sides at cost $O(n^2 r)$ [7]. If K is too large or only available implicitly, one may replace K^{-1} with an approximate solve; this yields an inexact preconditioner and requires additional care, which we do not pursue here.

7. Complexity and memory analysis

We analyze the dominant costs under the regime $n, r < q \ll N$ and emphasize the absence of any $O(N)$ computation.

Proposition 7.1 (Per-iteration cost of PCG). Assume Algorithms 5.3 and 5.4 are used for $\mathcal{A}(W)$ and $\mathcal{F}$, and preconditioner P_1 (Definition 6.4) is applied by a routine of cost $\mathsf{Cost}_{K^{-1}}(n, r)$. Then each PCG iteration costs

$$O\big(\mathsf{Cost}_K(n, r) + qdr + \mathsf{Cost}_{K^{-1}}(n, r)\big),$$

up to constant factors, and requires no operations scaling with N.

Proof. Inspect Algorithm 6.3. In each loop iteration, the only calls that depend on the linear system are `Q ← ApplyA(Pdir)` and `Z ← ApplyPInv(R)`. By Proposition 5.5, `ApplyA` evaluates one kernel multiplication to form $Y = KPdir$, performs a single pass over the q observations with $O(qdr)$ arithmetic, and performs one further kernel multiplication, hence costs $2\mathsf{Cost}_K(n, r) + O(qdr)$ and avoids any computation of order N. For the preconditioner $P_1(W) = \lambda KW$, applying P_1^{-1} amounts to solving $KX = R$ for X column-wise and scaling by λ^{-1}, which costs $\mathsf{Cost}_{K^{-1}}(n, r)$ by definition. The remaining Frobenius inner products and axpy updates in Algorithm 6.3 act on $n \times r$ matrices and cost $O(nr)$, which is absorbed into the stated bound; under the regime $n, r < q$ and $d \geq 2$, one has $nr \leq qdr$. Summing these contributions yields the claimed per-iteration complexity and shows that no step scales with N.

Corollary 7.2 (Total cost to reach a target residual). Let m be the number of PCG iterations until $\|R_m\|_F / \|\mathcal{F}\|_F \leq \mathrm{tol}$. The total cost is

$$O\Big(\mathsf{Cost}(\mathrm{BuildRHS}) + m\big(\mathsf{Cost}_K(n, r) + qdr + \mathsf{Cost}_{K^{-1}}(n, r)\big)\Big),$$

where $\mathsf{Cost}(\mathrm{BuildRHS}) = \mathsf{Cost}_K(n, r) + O(qdr)$ by Proposition 5.5.

Proof. By Proposition 5.5, `BuildRHS` costs $\mathsf{Cost}_K(n, r) + O(qdr)$. Summing this one-time cost with m iterations of Proposition 7.1 gives the stated total complexity.

Remark 7.3 (Explicit avoidance of $O(N)$ work). No step requires forming the full unfolding $T \in \mathbb{R}^{n \times M}$, the full Khatri–Rao product $Z \in \mathbb{R}^{M \times r}$, or any $nr \times nr$ matrix. The only dependence on the tensor size is through the observation list of length q, which is assumed to satisfy $q \ll N$.

8. Verifiability package for third-party checking

Although the solver is iterative and matrix-free, it is straightforward to make the output independently checkable.

Definition 8.1 (Verifiability package). A verifiability package for a computed solution $\widehat{W}$ consists of:

1. The observation list $\mathcal{O}$ and the fixed factor matrices $A_{m_{m \neq k}}$.
2. The regularization λ and a specification of the kernel operator $X \mapsto KX$ (explicit matrix or a deterministic routine).
3. The tolerance `tol`, maximum iterations `maxit`, and the chosen preconditioner description (e.g., P_1 with a specified K^{-1} application method).
4. A log of PCG diagnostics: residual norms $\|R_t\|_F / \|\mathcal{F}\|_F$, iteration count, and termination reason.

Proposition 8.2 (Post-hoc residual verification). Given $\widehat{W}$ and the verifiability package, a third party can recompute

$$\widehat{R} := \mathcal{F} - \mathcal{A}(\widehat{W})$$

using Algorithms 5.3–5.4 and verify that $\|\widehat{R}\|_F / \|\mathcal{F}\|_F \leq \mathrm{tol}$, without any computation of order N.

Proof. This is immediate from the availability of matrix-free routines for $\mathcal{A}$ and $\mathcal{F}$ and the observation-only nature of their implementations.

Remark 8.3 (Determinism). If `KernelMul` and the observation iteration order are deterministic, then the residual check is deterministic. If randomized kernel approximations are used, the randomness seed should be included in the verifiability package.

9. Conclusion and appendix roadmap

We provided a rigorous matrix-free PCG solution strategy for the kernelized ALS normal equation (equation) under sparse observations. The essential technical ingredient is the matrix-free operator representation (equation) and its implementation via two kernel multiplications and a single pass over the observation list. This ensures the algorithm avoids any $O(N)$ computation in the regime $q \ll N$.

This manuscript includes the following appendices:

- **Appendix A:** vectorization and Kronecker/Khatri–Rao identities, including (equation).
- **Appendix B:** equivalence between the selection operator SS^T and the projection P_Ω.
- **Appendix C:** derivation of the normal equation from the kernelized ALS subproblem.
- **Appendix D:** full proof of Proposition 4.2 (operator equivalence).
- **Appendix E:** full proof of Proposition 5.5 for ApplyA.
- **Appendix F:** full proof of Proposition 5.5 for BuildRHS.
- **Appendix G:** full proof of Proposition 6.2 (SPD conditions for PCG).

References

1. *First Proof*. arXiv preprint arXiv:2602.05192, 2026.
2. Tamara G. Kolda and Brett W. Bader. “Tensor Decompositions and Applications.” *SIAM Review*, 51(3):455–500, 2009.
3. Magnus R. Hestenes and Eduard Stiefel. “Methods of Conjugate Gradients for Solving Linear Systems.” *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952.
4. Yousef Saad. *Iterative Methods for Sparse Linear Systems*, Second Edition. SIAM, 2003.
5. Nachman Aronszajn. “Theory of Reproducing Kernels.” *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

6. Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2002.
7. Gene H. Golub and Charles F. Van Loan. *Matrix Computations*, Fourth Edition. Johns Hopkins University Press, 2013.

Appendix A. Algebraic Identities for Vectorization, Kronecker, and Khatri–Rao Products

This appendix collects and proves algebraic identities used to derive the matrix-free operator formulation and to justify the matrix-free routines in the main text. Throughout, all vectorizations are column-major.

A.1 Vectorization and Kronecker products

Definition A.1 (Vectorization convention and Kronecker product). Let $X \in \mathbb{R}^{p\times n}$. We define $\mathrm{vec}(X) \in \mathbb{R}^{pn}$ by stacking the columns of X:

$$\mathrm{vec}(X) := \begin{bmatrix} X_{:,1} \\ X_{:,2} \\ \vdots \\ X_{:,n} \end{bmatrix}.$$

Let $A \in \mathbb{R}^{m\times p}$ and $B \in \mathbb{R}^{n\times q}$. The Kronecker product $B \otimes A \in \mathbb{R}^{(mn)\times(pq)}$ is the block matrix defined by

$$B \otimes A := \begin{bmatrix} b_{11}A & b_{12}A & \cdots & b_{1q}A \\ b_{21}A & b_{22}A & \cdots & b_{2q}A \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1}A & b_{n2}A & \cdots & b_{nq}A \end{bmatrix},$$

where b_{ij} denotes the (i,j) entry of B.

Lemma A.2 (Vectorization identity). Let $A \in \mathbb{R}^{m\times p}$, $X \in \mathbb{R}^{p\times n}$, and $B \in \mathbb{R}^{n\times q}$. Then

$$\mathrm{vec}(AXB) = (B^{\mathsf{T}} \otimes A)\mathrm{vec}(X).$$

Proof. Let $Y := AXB \in \mathbb{R}^{m\times q}$. Fix $i \in \{1,\dots,m\}$ and $j \in \{1,\dots,q\}$. By expanding the matrix product,

$$Y_{ij} = \sum_{\alpha=1}^{p}\sum_{\beta=1}^{n} A_{i\alpha}X_{\alpha\beta}B_{\beta j}.$$

Now consider the vector $(B^\mathsf{T} \otimes A)\mathrm{vec}(X) \in \mathbb{R}^{mq}$. Under column-major vectorization, the entry of $\mathrm{vec}(Y)$ at position $i + (j-1)m$ is Y_{ij}. Likewise, the entry of $\mathrm{vec}(X)$ corresponding to $X_{\alpha\beta}$ is at position $\alpha + (\beta - 1)p$.

By the block definition of the Kronecker product,

$$\big((B^\mathsf{T} \otimes A)\mathrm{vec}(X)\big)_{i+(j-1)m} = \sum_{\beta=1}^{n}\sum_{\alpha=1}^{p}(B^\mathsf{T})_{j\beta}A_{i\alpha}X_{\alpha\beta}.$$

Since $(B^\mathsf{T})_{j\beta} = B_{\beta j}$, the right-hand side equals

$$\sum_{\beta=1}^{n}\sum_{\alpha=1}^{p}B_{\beta j}A_{i\alpha}X_{\alpha\beta} = \sum_{\alpha=1}^{p}\sum_{\beta=1}^{n}A_{i\alpha}X_{\alpha\beta}B_{\beta j} = Y_{ij}.$$

Thus $\mathrm{vec}(Y) = (B^\mathsf{T} \otimes A)\mathrm{vec}(X)$ entrywise, proving the claim.

Corollary A.3 (Specializations used in the main text). Let $K \in \mathbb{R}^{n\times n}$, $W \in \mathbb{R}^{n\times r}$, and $Z \in \mathbb{R}^{M\times r}$. Then: 1.

$$(Z \otimes K)\mathrm{vec}(W) = \mathrm{vec}(KWZ^\mathsf{T}).$$

2.

$$(I_r \otimes K)\mathrm{vec}(W) = \mathrm{vec}(KW).$$

Proof. For (1), apply Lemma A.2 with $A = K$, $X = W$, and $B = Z^\mathsf{T}$. Then $AXB = KWZ^\mathsf{T}$ and

$$\mathrm{vec}(KWZ^\mathsf{T}) = ((Z^\mathsf{T})^\mathsf{T} \otimes K)\mathrm{vec}(W) = (Z \otimes K)\mathrm{vec}(W).$$

For (2), apply Lemma A.2 with $A = K$, $X = W$, and $B = I_r$, noting $KW = KWI_r$ and $I_r^\mathsf{T} = I_r$.

Lemma A.4 (Adjoint action of a Kronecker factor). Let $K \in \mathbb{R}^{n\times n}$, $Z \in \mathbb{R}^{M\times r}$, and $U \in \mathbb{R}^{n\times M}$. Then

$$(Z \otimes K)^\mathsf{T}\mathrm{vec}(U) = \mathrm{vec}(K^\mathsf{T}UZ).$$

In particular, if K is symmetric, then $(Z \otimes K)^\mathsf{T}\mathrm{vec}(U) = \mathrm{vec}(KUZ)$.

Proof. Since $(Z \otimes K)^\mathsf{T} = Z^\mathsf{T} \otimes K^\mathsf{T}$, Lemma A.2 with $A = K^\mathsf{T}$, $X = U$, and $B = Z$ yields

$$(Z^\mathsf{T} \otimes K^\mathsf{T})\mathrm{vec}(U) = \mathrm{vec}(K^\mathsf{T}UZ),$$

which is exactly the claim.

A.2 Khatri–Rao products

Definition A.5 (Khatri–Rao product). Let $A^{(1)} \in \mathbb{R}^{n_1 \times r}$ and $A^{(2)} \in \mathbb{R}^{n_2 \times r}$. Their Khatri–Rao product is the matrix

$$A^{(2)} \odot A^{(1)} \in \mathbb{R}^{(n_1 n_2) \times r}$$

defined column-wise by

$$(A^{(2)} \odot A^{(1)})_{:,s} := A^{(2)}_{:,s} \otimes A^{(1)}_{:,s}, \qquad s \in \{1, \dots, r\}.$$

More generally, for matrices ${A^{(m)}}_{m=1}^{p}$ with $A^{(m)} \in \mathbb{R}^{n_m \times r}$, we define

$$A^{(p)} \odot \cdots \odot A^{(1)} \in \mathbb{R}^{(\prod_{m=1}^{p} n_m) \times r}$$

by

$$(A^{(p)} \odot \cdots \odot A^{(1)})_{:,s} := A^{(p)}_{:,s} \otimes \cdots \otimes A^{(1)}_{:,s}.$$

Lemma A.6 (Entrywise row formula for Khatri–Rao). Let $Z := A^{(p)} \odot \cdots \odot A^{(1)}$ where $A^{(m)} \in \mathbb{R}^{n_m \times r}$. Let $s \in \{1, \dots, r\}$. For any multi-index $(i_1, \dots, i_p)$ with $i_m \in \{1, \dots, n_m\}$, the entry of the Kronecker product column satisfies

$$(A^{(p)}_{:,s} \otimes \cdots \otimes A^{(1)}_{:,s})_{(i_1, \dots, i_p)} = \prod_{m=1}^{p} (A^{(m)})_{i_m, s}.$$

Consequently, the row of Z corresponding to the multi-index $(i_1, \dots, i_p)$ has coordinates

$$Z_{(i_1, \dots, i_p), s} = \prod_{m=1}^{p} (A^{(m)})_{i_m, s}.$$

Proof. For $p = 2$, the Kronecker product satisfies

$$(a \otimes b)_{(i_1, i_2)} = a_{i_2} b_{i_1},$$

under the standard column-major indexing of $\mathbb{R}^{n_1 n_2}$ (the precise linearization rule is irrelevant for the value at a fixed multi-index). Setting $a = A^{(2)}_{:,s}$ and $b = A^{(1)}_{:,s}$ yields the product $(A^{(2)})_{i_2, s} (A^{(1)})_{i_1, s}$.

For general p, apply the same argument inductively, using associativity of the Kronecker product and the fact that each additional Kronecker factor multiplies the corresponding entry. This yields the product formula in (equation).

Proposition A.7 (Gram matrix identity for Khatri–Rao products). Let $Z := A^{(p)} \odot \cdots \odot A^{(1)}$ where $A^{(m)} \in \mathbb{R}^{n_m \times r}$. Then

$$Z^{\mathsf{T}} Z = \left(A^{(p)\mathsf{T}} A^{(p)}\right) \circ \cdots \circ \left(A^{(1)\mathsf{T}} A^{(1)}\right),$$

where $\circ$ denotes the Hadamard (entrywise) product.

Proof. Fix $s, t \in \{1, \dots, r\}$. By Definition A.5, the s-th column of Z is the Kronecker product

$$Z_{:,s} = A^{(p)}_{:,s} \otimes \cdots \otimes A^{(1)}_{:,s}.$$

Using the standard identity

$$(a_p \otimes \cdots \otimes a_1)^{\mathsf{T}} (b_p \otimes \cdots \otimes b_1) = \prod_{m=1}^{p} (a_m^{\mathsf{T}} b_m),$$

we obtain

$$(Z^{\mathsf{T}} Z)_{st} = Z_{:,s}^{\mathsf{T}} Z_{:,t} = \prod_{m=1}^{p} \left(A^{(m)}_{:,s}\right)^{\mathsf{T}} \left(A^{(m)}_{:,t}\right) = \prod_{m=1}^{p} \left(A^{(m)\mathsf{T}} A^{(m)}\right)_{st}.$$

This is exactly the Hadamard product statement (equation).

A.3 Simple matrix-product identities used by the matrix-free routines

Lemma A.8 (Entrywise inner-product form of a matrix product). Let $Y \in \mathbb{R}^{n \times r}$ and $Z \in \mathbb{R}^{M \times r}$. Then for every $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, M\}$,

$$(YZ^{\mathsf{T}})_{ij} = \langle Y_{i,:}, Z_{j,:} \rangle,$$

where $\langle a, b \rangle := \sum_{s=1}^{r} a_s b_s$ is the standard Euclidean inner product on $\mathbb{R}^r$.

Proof. By the definition of matrix multiplication,

$$(YZ^{\mathsf{T}})_{ij} = \sum_{s=1}^{r} Y_{is} (Z^{\mathsf{T}})_{sj} = \sum_{s=1}^{r} Y_{is} Z_{js} = \langle Y_{i,:}, Z_{j,:} \rangle.$$

Lemma A.9 (Gather–scatter identity for projected products). Let $\Omega \subset \{1, \dots, n\} \times \{1, \dots, M\}$ and let P_Ω be the projection in Definition 2.1 of the main text. For any $Y \in \mathbb{R}^{n \times r}$ and $Z \in \mathbb{R}^{M \times r}$, define $C \in \mathbb{R}^{n \times r}$ by

$$C_{i,:} := \sum_{j:(i,j)\in\Omega} (YZ^{\mathsf{T}})_{ij} Z_{j,:}, \qquad i \in \{1, \dots, n\},$$

with the convention that the sum is zero if no $(i, j) \in \Omega$ exists for that i. Then

$$C = P_\Omega(YZ^{\mathsf{T}})Z.$$

Proof. For each fixed i, the i-th row of $P_\Omega(YZ^{\mathsf{T}})$ coincides with the i-th row of YZ^{T} on columns j such that $(i, j) \in \Omega$ and is zero elsewhere. Therefore,

$$\left(P_\Omega(YZ^{\mathsf{T}})Z\right)_{i,:} = \sum_{j=1}^{M} \left(P_\Omega(YZ^{\mathsf{T}})\right)_{ij} Z_{j,:} = \sum_{j:(i,j)\in\Omega} (YZ^{\mathsf{T}})_{ij} Z_{j,:} = C_{i,:}.$$

Since this holds row-by-row, the matrix identity follows.

Appendix B. Observation Operator: From the Selection Matrix SS^{T} to the Projection P_Ω

This appendix formalizes the observation operator used in the normal equation and proves its equivalence to the sparse-entry projection P_Ω after reshaping into the mode-k unfolding matrix. The statements here justify the replacement of the selection-matrix expression SS^{T} by a projection that can be implemented by iterating over an observation list.

B.1 Indexing conventions for unfolding and vectorization

We work with matrices of shape $n \times M$, where $N := nM$. Throughout, vectorization is column-major.

Definition B.1 (Linear index for column-major vectorization). For $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, M\}$, define the linear index

$$\ell(i, j) := i + (j - 1)n \in \{1, \dots, N\}.$$

Lemma B.2 (Entries and vectorization). Let $X \in \mathbb{R}^{n\times M}$. Then, under column-major vectorization,

$$\left(\operatorname{vec}(X)\right)_{\ell(i,j)} = X_{ij}, \qquad \forall, i \in \{1, \dots, n\},\ j \in \{1, \dots, M\}.$$

Proof. By definition of column-major vectorization, the $(j-1)$-th block of length n in $\operatorname{vec}(X)$ equals the j-th column $X_{:,j}$. Within that block, the i-th entry equals X_{ij}. Since $\ell(i, j) = i + (j - 1)n$, the claim follows.

B.2 Selection matrices for sparse observations

Let $\Omega \subset \{1, \dots, n\} \times \{1, \dots, M\}$ be a set of observed unfolding locations with $|\Omega| = q$.

Assumption B.3 (No duplicate observation locations). We assume Ω is a set (not a multiset), i.e., each location (i, j) appears at most once. If duplicates occur in raw data, they can be aggregated (e.g., by summing or averaging values at identical locations) before constructing Ω.

Fix an enumeration

$$\Omega = \{(i^{(e)}, j^{(e)})\}_{e=1}^{q}.$$

Definition B.4 (Selection matrix associated with Ω). Let $e_\alpha \in \mathbb{R}^N$ denote the α-th standard basis vector. Define the selection matrix $S_\Omega \in \mathbb{R}^{N \times q}$ by

$$S_\Omega := \begin{bmatrix} e_{\ell(i^{(1)}, j^{(1)})} & e_{\ell(i^{(2)}, j^{(2)})} & \cdots & e_{\ell(i^{(q)}, j^{(q)})} \end{bmatrix}.$$

Lemma B.5 (Extraction of observed entries). For any $X \in \mathbb{R}^{n \times M}$,

$$S_\Omega^\mathsf{T} \mathrm{vec}(X) = \begin{bmatrix} X_{i^{(1)} j^{(1)}} \\ X_{i^{(2)} j^{(2)}} \\ \vdots \\ X_{i^{(q)} j^{(q)}} \end{bmatrix} \in \mathbb{R}^q.$$

Proof. The e-th component of $S_\Omega^\mathsf{T} \mathrm{vec}(X)$ equals

$$\left(S_\Omega^\mathsf{T} \mathrm{vec}(X)\right)_e = e_{\ell(i^{(e)}, j^{(e)})}^\mathsf{T} \mathrm{vec}(X) = \left(\mathrm{vec}(X)\right)_{\ell(i^{(e)}, j^{(e)})}.$$

By Lemma B.2, this equals $X_{i^{(e)} j^{(e)}}$.

Lemma B.6 (Masking as a projection in vectorized form). For any $X \in \mathbb{R}^{n \times M}$ and any $\alpha \in \{1, \dots, N\}$,

$$\left(S_\Omega S_\Omega^\mathsf{T} \mathrm{vec}(X)\right)_\alpha = \begin{cases} \left(\mathrm{vec}(X)\right)_\alpha, & \alpha \in \{\ell(i, j) : (i, j) \in \Omega\}, \\ 0, & \text{otherwise.} \end{cases}$$

Proof. By Definition B.4,

$$S_\Omega S_\Omega^\mathsf{T} = \sum_{e=1}^{q} e_{\ell(i^{(e)}, j^{(e)})} e_{\ell(i^{(e)}, j^{(e)})}^\mathsf{T}.$$

Therefore,

$$S_\Omega S_\Omega^\mathsf{T}\mathrm{vec}(X) = \sum_{e=1}^{q} e_{\ell(i^{(e)},j^{(e)})}, \big(e_{\ell(i^{(e)},j^{(e)})}^\mathsf{T}\mathrm{vec}(X)\big).$$

The α-th component equals

$$\big(S_\Omega S_\Omega^\mathsf{T}\mathrm{vec}(X)\big)_\alpha = \sum_{e=1}^{q} \mathbf{1}_{\{\alpha=\ell(i^{(e)},j^{(e)})\}}\big(\mathrm{vec}(X)\big)_{\ell(i^{(e)},j^{(e)})}.$$

Under Assumption B.3 the indices $\ell(i^{(e)}, j^{(e)})$ are distinct, so at most one term in the sum is nonzero. This yields exactly the stated masking rule.

B.3 Equivalence to the sparse-entry projection P_Ω

We now connect $S_\Omega S_\Omega^\mathsf{T}$ to the matrix-level projection P_Ω.

Definition B.7 (Sparse-entry projection on matrices). Define $P_\Omega : \mathbb{R}^{n\times M} \to \mathbb{R}^{n\times M}$ by

$$(P_\Omega(X))_{ij} = \begin{cases} X_{ij}, & (i,j) \in \Omega, \\ 0, & \text{otherwise.} \end{cases}$$

Proposition B.8 (Intertwining identity: $S_\Omega S_\Omega^\mathsf{T}$ equals P_Ω under vectorization). For every $X \in \mathbb{R}^{n\times M}$,

$$S_\Omega S_\Omega^\mathsf{T}\mathrm{vec}(X) = \mathrm{vec}(P_\Omega(X)).$$

Equivalently, if mat : $\mathbb{R}^N \to \mathbb{R}^{n\times M}$ denotes the inverse of vec (column-major reshape), then

$$\mathrm{mat}\left(S_\Omega S_\Omega^\mathsf{T}\mathrm{vec}(X)\right) = P_\Omega(X).$$

Proof. Fix $(i, j) \in \{1, \dots, n\} \times \{1, \dots, M\}$ and let $\alpha = \ell(i, j)$. By Lemma B.6,

$$\big(S_\Omega S_\Omega^\mathsf{T}\mathrm{vec}(X)\big)_{\ell(i,j)} = \begin{cases} \big(\mathrm{vec}(X)\big)_{\ell(i,j)}, & (i,j) \in \Omega, \\ 0, & (i,j) \notin \Omega. \end{cases}$$

By Lemma B.2, $\big(\mathrm{vec}(X)\big)_{\ell(i,j)} = X_{ij}$, hence

$$\big(S_\Omega S_\Omega^\mathsf{T}\mathrm{vec}(X)\big)_{\ell(i,j)} = \begin{cases} X_{ij}, & (i,j) \in \Omega, \\ 0, & (i,j) \notin \Omega. \end{cases}$$

The right-hand side is exactly $(P_\Omega(X))_{ij}$ by Definition B.7, and applying vec to $P_\Omega(X)$ places $(P_\Omega(X))_{ij}$ at index $\ell(i,j)$. Therefore the vectors coincide entrywise, proving the claim.

Corollary B.9 (Projection properties). Under Assumption B.3, the matrix $S_\Omega S_\Omega^\top \in \mathbb{R}^{N\times N}$ is an orthogonal projector, i.e.,

$$(S_\Omega S_\Omega^\top)^\top = S_\Omega S_\Omega^\top, \qquad (S_\Omega S_\Omega^\top)^2 = S_\Omega S_\Omega^\top.$$

Moreover, P_Ω is an orthogonal projection on $\mathbb{R}^{n\times M}$ with respect to the Frobenius inner product, i.e.,

$$\langle P_\Omega(X), Y\rangle_F = \langle X, P_\Omega(Y)\rangle_F, \qquad P_\Omega(P_\Omega(X)) = P_\Omega(X)$$

and the intertwining identity (equation) holds.

Proof. Symmetry of $S_\Omega S_\Omega^\top$ is immediate. For idempotence, note that Assumption B.3 implies

$$S_\Omega^\top S_\Omega = I_q,$$

hence

$$(S_\Omega S_\Omega^\top)^2 = S_\Omega(S_\Omega^\top S_\Omega)S_\Omega^\top = S_\Omega I_q S_\Omega^\top = S_\Omega S_\Omega^\top.$$

For P_Ω, both self-adjointness and idempotence follow directly from the entrywise definition (equation). The final statement is Proposition B.8.

B.4 Compatibility with tensor index orderings (optional but clarifying)

The preceding discussion is expressed in the vectorization order of the mode-k unfolding. In practice, observations are often recorded as tensor multi-indices, and one may also choose a different tensor vectorization order. The next lemma clarifies how selection matrices transform across equivalent orderings.

Lemma B.10 (Selection matrices under permutations). Let $\Pi \in \mathbb{R}^{N\times N}$ be a permutation matrix. Define a new vectorization $\mathrm{vec}_\Pi(X) := \Pi\mathrm{vec}(X)$. Let S_Ω be a selection matrix in the original vec ordering, and define $S_\Omega^\Pi := \Pi S_\Omega$. Then for every $X \in \mathbb{R}^{n\times M}$,

$$S_\Omega^\Pi (S_\Omega^\Pi)^\top \mathrm{vec}_\Pi(X) = \Pi(S_\Omega S_\Omega^\top)\mathrm{vec}(X),$$

and $S_\Omega^\Pi (S_\Omega^\Pi)^\top$ is the selection mask representing the same physical observation set Ω under the permuted vectorization.

Proof. Using $\mathrm{vec}_\Pi(X) = \Pi\mathrm{vec}(X)$ and $S_\Omega^\Pi = \Pi S_\Omega$,

$$S_\Omega^\Pi (S_\Omega^\Pi)^\mathsf{T} \mathrm{vec}_\Pi(X) = (\Pi S_\Omega)(\Pi S_\Omega)^\mathsf{T}(\Pi \mathrm{vec}(X)) = \Pi S_\Omega S_\Omega^\mathsf{T} \Pi^\mathsf{T} \Pi \mathrm{vec}(X) = \Pi(S_\Omega S_\Omega^\mathsf{T})\mathrm{vec}(X),$$

since $\Pi^\mathsf{T}\Pi = I$.

Remark B.11 (How this lemma is used in the main development). In the main text, the Kronecker expression $(Z \otimes K)\mathrm{vec}(W)$ naturally produces $\mathrm{vec}(KWZ^\mathsf{T})$ in the column-major ordering of the $n \times M$ unfolding. Thus, the selection matrix S appearing in the normal equation can always be understood (without loss of generality) as the selection matrix associated with Ω in that same ordering, possibly after an implicit permutation as in Lemma B.10.

Appendix C. Deriving the Normal Equations for the Kernelized ALS Subproblem

This appendix derives the normal equation used in the mode-k ALS update under sparse observations and a kernel parametrization $A_k = KW$. The result is the $nr \times nr$ linear system stated in the main text (equation (equation)).

Throughout, we adopt the vectorization and Kronecker conventions of Appendix A and the observation-operator conventions of Appendix B.

C.1 The regularized least-squares subproblem

We consider a fixed mode index $k \in \{1, \dots, d\}$ and hold the factor matrices $A_{m_{m \neq k}}$ fixed. Let $Z \in \mathbb{R}^{M \times r}$ denote the corresponding Khatri–Rao product, where $M = \prod_{m \neq k} n_m$.

We represent the mode-k factor in a kernel form.

Definition C.1 (Kernelized ALS subproblem with sparse observations). Let $T \in \mathbb{R}^{n \times M}$ denote the mode-k unfolding of the data tensor with missing entries set to zero, and let $\Omega \subset \{1, \dots, n\} \times \{1, \dots, M\}$ be the set of observed unfolding locations. Let $S \in \mathbb{R}^{N \times q}$ be the corresponding selection matrix, with $N = nM$ and $q = |\Omega|$, as in Appendix B.

Given a symmetric positive semidefinite kernel matrix $K \in \mathbb{R}^{n \times n}$ and $\lambda > 0$, the mode-k update is posed as the following regularized least-squares problem in $W \in \mathbb{R}^{n \times r}$:

$$\min_{W \in \mathbb{R}^{n \times r}} \ \frac{1}{2} \left| S^\mathsf{T} \left(\mathrm{vec}(T) - (Z \otimes K)\mathrm{vec}(W)\right)\right|_2^2 \ + \ \frac{\lambda}{2}\mathrm{vec}(W)^\mathsf{T}(I_r \otimes K)\mathrm{vec}(W).$$

This objective corresponds to fitting the model $A_k Z^\top$ to the observed entries only, with $A_k = KW$ and an RKHS-style quadratic penalty on W.

Proposition C.2 (Equivalent matrix form using the projection P_Ω). Problem (equation) is equivalent to

$$\min_{W\in\mathbb{R}^{n\times r}} \frac{1}{2}\left|P_\Omega\left(T - KWZ^\top\right)\right|_F^2 \;+\; \frac{\lambda}{2}\mathrm{trace}(W^\top K W).$$

Proof. Using Corollary A.3(1), we have

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

Thus the residual in (equation) can be written as

$$\mathrm{vec}(T) - (Z \otimes K)\mathrm{vec}(W) = \mathrm{vec}\left(T - KWZ^\top\right).$$

Applying $S^\top$ selects the observed entries, and by Proposition B.8 we have the intertwining identity

$$SS^\top \mathrm{vec}(X) = \mathrm{vec}(P_\Omega(X)) \qquad (\forall X \in \mathbb{R}^{n\times M}).$$

Hence,

$$\left|S^\top \mathrm{vec}\left(T - KWZ^\top\right)\right|_2^2 = \left|P_\Omega\left(T - KWZ^\top\right)\right|_F^2,$$

where we use that the Euclidean norm of the selected-entry vector equals the Frobenius norm of the projected matrix.

For the regularizer, note that $\mathrm{trace}(W^\top KW)$ equals the sum over columns of W of $w_s^\top K w_s$, and also satisfies the vectorization identity

$$\mathrm{trace}(W^\top KW) = \mathrm{vec}(W)^\top (I_r \otimes K)\mathrm{vec}(W),$$

which follows from Corollary A.3(2) together with $\langle X, Y\rangle_F = \mathrm{vec}(X)^\top \mathrm{vec}(Y)$. Combining the above identities yields (equation).

C.2 First-order optimality and the normal equation

We now derive the normal equation for the minimizer of (equation).

Proposition C.3 (Normal equations in vectorized form). Let

$$x := \mathrm{vec}(W) \in \mathbb{R}^{nr}, \qquad A := Z \otimes K \in \mathbb{R}^{N\times(nr)}, \qquad b := \mathrm{vec}(T) \in \mathbb{R}^N.$$

Then the objective in (equation) can be written as

$$\phi(x) := \frac{1}{2}|S^\top(b - Ax)|_2^2 + \frac{\lambda}{2}x^\top(I_r \otimes K)x.$$

Any minimizer $x_\star$ satisfies the linear system

$$\Big(A^\top SS^\top A + \lambda(I_r \otimes K)\Big)x_\star = A^\top SS^\top b.$$

Proof. Expand the data-misfit term:

$$\frac{1}{2}|S^\top(b - Ax)|_2^2 = \frac{1}{2}(b - Ax)^\top SS^\top(b - Ax).$$

Using that $SS^\top$ is symmetric (Appendix B), we obtain

$$\frac{1}{2}(b - Ax)^\top SS^\top(b - Ax) = \frac{1}{2}b^\top SS^\top b - \ x^\top A^\top SS^\top b + \ \frac{1}{2}x^\top A^\top SS^\top Ax.$$

Therefore, ϕ is a convex quadratic function of x, and its gradient is

$$\nabla\phi(x) = A^\top SS^\top(Ax - b) + \lambda(I_r \otimes K)x.$$

At any minimizer $x_\star$, the first-order optimality condition $\nabla\phi(x_\star) = 0$ holds, yielding

$$A^\top SS^\top Ax_\star + \lambda(I_r \otimes K)x_\star = A^\top SS^\top b,$$

which is exactly (equation).

C.3 Right-hand side simplification and identification with $B = TZ$

In the main text we define $B := TZ \in \mathbb{R}^{n\times r}$ (MTTKRP). We now show that the right-hand side of (equation) can be written as $(I_r \otimes K)\mathrm{vec}(B)$ under the conventions of the problem setup.

Assumption C.4 (Zero-filled unfolding). The matrix $T \in \mathbb{R}^{n\times M}$ is the mode-k unfolding with missing entries set to zero, so that

$$P_\Omega(T) = T.$$

This is the standard convention when writing sparse tensor observations into a dense unfolding with implicit zeros.

Proposition C.5 (Right-hand side equals $(I_r \otimes K)\mathrm{vec}(B)$). Let $B := TZ$. Suppose Assumption C.4 holds and K is symmetric. Then

$$A^\mathsf{T} S S^\mathsf{T} b = (I_r \otimes K)\mathrm{vec}(B).$$

Proof. By Proposition B.8 and Assumption C.4,

$$S S^\mathsf{T} \mathrm{vec}(T) = \mathrm{vec}(P_\Omega(T)) = \mathrm{vec}(T).$$

Therefore,

$$A^\mathsf{T} S S^\mathsf{T} b = (Z \otimes K)^\mathsf{T} \mathrm{vec}(T).$$

Applying Lemma A.4 with $U = T$ yields

$$(Z \otimes K)^\mathsf{T} \mathrm{vec}(T) = \mathrm{vec}(K^\mathsf{T} T Z).$$

If K is symmetric, $K^\mathsf{T} = K$, hence

$$(Z \otimes K)^\mathsf{T} \mathrm{vec}(T) = \mathrm{vec}(KTZ) = \mathrm{vec}(KB).$$

Finally, Corollary A.3(2) gives $\mathrm{vec}(KB) = (I_r \otimes K)\mathrm{vec}(B)$, proving (equation).

Corollary C.6 (Recovery of the target linear system). Under the conditions of Proposition C.5, the normal equations (equation) become

$$\Big((Z \otimes K)^\mathsf{T} S S^\mathsf{T} (Z \otimes K) + \lambda(I_r \otimes K)\Big)\mathrm{vec}(W) = (I_r \otimes K)\mathrm{vec}(B),$$

which matches equation (equation) in the main text.

Proof. Substitute $A = Z \otimes K$ and $b = \mathrm{vec}(T)$ into Proposition C.3 and apply Proposition C.5.

Remark C.7 (If the unfolding is not explicitly zero-filled). If one starts from a “true” dense unfolding T_{full} but only has access to its entries on Ω, then one can define the zero-filled unfolding by

$$T := P_\Omega(T_{\mathrm{full}}),$$

so that $\mathrm{vec}(T) = S S^\mathsf{T} \mathrm{vec}(T_{\mathrm{full}})$ by Proposition B.8. In this case, the above derivation applies verbatim after replacing T_{full} by T.

Appendix D. Full Proof of the Matrix-free Operator Equivalence

This appendix provides a complete proof of Proposition 4.2 in the main text, namely the equivalence between the $nr \times nr$ Kronecker normal equation and the matrix-free operator formulation on $\mathbb{R}^{n\times r}$. The proof uses the identities established in Appendix A and the observation-operator equivalence established in Appendix B.

D.1 Setup

Let $n \in \mathbb{N}$, $M \in \mathbb{N}$, and $N := nM$. Let $r \in \mathbb{N}$. Let $K \in \mathbb{R}^{n\times n}$ be a symmetric matrix (in the main text, a kernel matrix), and let $Z \in \mathbb{R}^{M\times r}$ be the Khatri–Rao product associated with the fixed factor matrices for all modes except mode k.

Let $\Omega \subset \{1, \dots, n\} \times \{1, \dots, M\}$ be the set of observed unfolding locations with $|\Omega| = q$. Let $P_\Omega : \mathbb{R}^{n\times M} \to \mathbb{R}^{n\times M}$ be the entrywise projection defined in Definition 2.1 (main text), and let $S \in \mathbb{R}^{N\times q}$ be a selection matrix corresponding to Ω as in Appendix B (Definition B.4). By Proposition B.8, we have

$$SS^\mathsf{T}\mathrm{vec}(X) = \mathrm{vec}(P_\Omega(X)), \qquad \forall X \in \mathbb{R}^{n\times M}.$$

We will consider an unknown matrix $W \in \mathbb{R}^{n\times r}$ and define the matrix-free operator $\mathcal{A} : \mathbb{R}^{n\times r} \to \mathbb{R}^{n\times r}$ by

$$\mathcal{A}(W) := K\Big(P_\Omega(KWZ^\mathsf{T})Z\Big) + \lambda KW,$$

where $\lambda > 0$.

D.2 Equivalence between the Kronecker normal equation and the operator equation

Proposition D.1 (Full proof of Proposition 4.2: operator equivalence). Let

$$A := (Z \otimes K)^\mathsf{T} SS^\mathsf{T}(Z \otimes K) + \lambda(I_r \otimes K) \in \mathbb{R}^{(nr)\times(nr)}.$$

Assume K is symmetric. Then for every $W \in \mathbb{R}^{n\times r}$,

$$\mathrm{vec}(\mathcal{A}(W)) = A\mathrm{vec}(W).$$

Moreover, for every $B \in \mathbb{R}^{n\times r}$,

$$(I_r \otimes K)\mathrm{vec}(B) = \mathrm{vec}(KB).$$

Proof. Fix $W \in \mathbb{R}^{n\times r}$ and define

$$Y := KWZ^{\mathsf{T}} \in \mathbb{R}^{n\times M}.$$

Step 1 (forward Kronecker action). By Corollary A.3(1),

$$(Z \otimes K)\text{vec}(W) = \text{vec}(KWZ^{\mathsf{T}}) = \text{vec}(Y).$$

Step 2 (observation mask). Applying SS^{T} and using the intertwining identity (equation),

$$SS^{\mathsf{T}}(Z \otimes K)\text{vec}(W) = SS^{\mathsf{T}}\text{vec}(Y) = \text{vec}(P_\Omega(Y)).$$

Step 3 (adjoint Kronecker action). Applying $(Z \otimes K)^{\mathsf{T}}$ and using Lemma A.4, we obtain

$$(Z \otimes K)^{\mathsf{T}}\text{vec}(P_\Omega(Y)) = \text{vec}(K^{\mathsf{T}}P_\Omega(Y)Z).$$

Since K is symmetric, $K^{\mathsf{T}} = K$, hence

$$(Z \otimes K)^{\mathsf{T}}SS^{\mathsf{T}}(Z \otimes K)\text{vec}(W) = \text{vec}\Big(KP_\Omega(Y)Z\Big).$$

Recalling $Y = KWZ^{\mathsf{T}}$, this becomes

$$(Z \otimes K)^{\mathsf{T}}SS^{\mathsf{T}}(Z \otimes K)\text{vec}(W) = \text{vec}\Big(KP_\Omega(KWZ^{\mathsf{T}})Z\Big).$$

Step 4 (regularization term). By Corollary A.3(2),

$$(I_r \otimes K)\text{vec}(W) = \text{vec}(KW),$$

and therefore

$$\lambda(I_r \otimes K)\text{vec}(W) = \text{vec}(\lambda KW).$$

Step 5 (sum and identification with $\mathcal{A}$). Summing the contributions from Steps 3 and 4 yields

$$A\text{vec}(W) = \text{vec}\Big(KP_\Omega(KWZ^{\mathsf{T}})Z\Big) + \text{vec}(\lambda KW) = \text{vec}\Big(KP_\Omega(KWZ^{\mathsf{T}})Z + \lambda KW\Big).$$

By the definition of $\mathcal{A}$, the right-hand side equals $\text{vec}(\mathcal{A}(W))$, proving (equation).

Finally, (equation) is exactly Corollary A.3(2) applied to B in place of W.

Corollary D.2 (Equivalence of linear systems). Let $\mathscr{F} \in \mathbb{R}^{n\times r}$ be defined by $\mathscr{F} := KB$ for some $B \in \mathbb{R}^{n\times r}$. Under the assumptions of Proposition D.1, the following are equivalent:

1. $W \in \mathbb{R}^{n\times r}$ satisfies the operator equation

$$\mathcal{A}(W) = \mathscr{F}.$$

2. $x := \mathrm{vec}(W) \in \mathbb{R}^{nr}$ satisfies the Kronecker normal equation

$$\Big((Z \otimes K)^\mathsf{T} SS^\mathsf{T}(Z \otimes K) + \lambda(I_r \otimes K)\Big)x = (I_r \otimes K)\mathrm{vec}(B).$$

Proof. Apply vec to $\mathcal{A}(W) = \mathscr{F}$ and use Proposition D.1 to replace $\mathrm{vec}(\mathcal{A}(W))$ by the left-hand side matrix action and $\mathrm{vec}(\mathscr{F}) = \mathrm{vec}(KB)$ by $(I_r \otimes K)\mathrm{vec}(B)$. Conversely, reshape any x satisfying (equation) into W and reverse the same identities.

Remark D.3 (On the choice of S and indexing conventions). The proof assumes that S corresponds to the observation set Ω under the column-major vectorization of the $n \times M$ unfolding (Appendix B). If a different tensor vectorization ordering is used in the original statement, one may incorporate a permutation matrix as in Lemma B.10 to convert to this convention without changing the underlying observation set.

Appendix E. Correctness and Complexity of Algorithm 5.3 (ApplyA)

This appendix gives a complete proof of the **ApplyA** part of Proposition 5.5 in the main text. Concretely, we show that Algorithm 5.3 computes the matrix-free operator

$$\mathcal{A}(W) = K\Big(P_\Omega(KWZ^\mathsf{T})Z\Big) + \lambda KW$$

defined in equation (equation), without explicitly forming $Z \in \mathbb{R}^{M\times r}$, the product $KWZ^\mathsf{T} \in \mathbb{R}^{n\times M}$, or any $nr \times nr$ matrix. We also provide a detailed arithmetic cost and memory analysis.

We use the conventions of Appendix A (vectorization, Kronecker/Khatri–Rao identities) and Appendix B (observation projection). In particular, we assume the observation set contains no duplicates (Assumption B.3).

E.1 Preliminaries

Let $W \in \mathbb{R}^{n \times r}$ be arbitrary. Let $K \in \mathbb{R}^{n \times n}$ be a symmetric kernel matrix, and let

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

be the Khatri–Rao product of the fixed factor matrices for all modes except mode k.

Let the observation list be

$$\mathcal{O} = \{(i_1^{(e)}, \dots, i_d^{(e)}, t^{(e)})\}_{e=1}^{q},$$

and let $\Omega = \{(i^{(e)}, j^{(e)})\}_{e=1}^{q} \subset \{1, \dots, n\} \times \{1, \dots, M\}$ denote the corresponding set of observed locations in the mode-k unfolding, where $i^{(e)} = i_k^{(e)}$ and $j^{(e)}$ is the unfolding column index associated with the multi-index $(i_m^{(e)})_{m \neq k}$.

Algorithm 5.3 uses the on-the-fly row construction of Algorithm 5.2, which maps $(i_1^{(e)}, \dots, i_d^{(e)})$ to a vector $z^{(e)} \in \mathbb{R}^r$.

Lemma E.1 (On-the-fly Khatri–Rao row equals a row of Z). For each observation $e \in \{1, \dots, q\}$, let $j^{(e)}$ be the unfolding column index associated with the multi-index $(i_m^{(e)})_{m \neq k}$. Let $z^{(e)} \in \mathbb{R}^r$ be produced by Algorithm 5.2, i.e.,

$$z_s^{(e)} := \prod_{m \neq k} (A_m)_{i_m^{(e)}, s}, \qquad s \in \{1, \dots, r\}.$$

Then

$$z^{(e)} = Z_{j^{(e)},:}.$$

Proof. This is exactly the entrywise row formula for the Khatri–Rao product (Lemma A.6), applied to the collection $A_{m_{m \neq k}}$.

E.2 Correctness of Algorithm 5.3 (ApplyA)

We now prove that Algorithm 5.3 returns $\mathcal{A}(W)$.

Algorithm 5.3 first computes

$$Y := KW \in \mathbb{R}^{n \times r},$$

then accumulates a matrix $C \in \mathbb{R}^{n \times r}$ using the observation loop, and finally returns

$$KC + \lambda Y.$$

To identify C with $P_\Omega(KWZ^\mathsf{T})Z$, we proceed in two steps: (i) show that the scalar computed for each observation equals the corresponding entry of KWZ^T, and (ii) show that the scatter accumulation realizes the matrix product with the projection P_Ω.

Lemma E.2 (Observed predicted value as an inner product). Let $Y \in \mathbb{R}^{n\times r}$ and $Z \in \mathbb{R}^{M\times r}$. For any $(i, j) \in \{1, \ldots, n\} \times \{1, \ldots, M\}$,

$$(YZ^\mathsf{T})_{ij} = \langle Y_{i,:}, Z_{j,:}\rangle,$$

where $\langle a, b\rangle := \sum_{s=1}^{r} a_s b_s$ is the Euclidean inner product on $\mathbb{R}^r$.

Proof. This is Lemma A.8.

Lemma E.3 (Value computed in the observation loop). In Algorithm 5.3, for each observation e the scalar

$$u^{(e)} := \langle Y_{i^{(e)},:}, z^{(e)}\rangle$$

satisfies

$$u^{(e)} = (KWZ^\mathsf{T})_{i^{(e)}j^{(e)}}.$$

Proof. By definition, $Y = KW$. By Lemma E.1, $z^{(e)} = Z_{j^{(e)},:}$. Applying Lemma E.2 with $(i, j) = (i^{(e)}, j^{(e)})$ yields

$$\langle Y_{i^{(e)},:}, Z_{j^{(e)},:}\rangle = (YZ^\mathsf{T})_{i^{(e)}j^{(e)}}.$$

Substituting $Y = KW$ gives $u^{(e)} = (KWZ^\mathsf{T})_{i^{(e)}j^{(e)}}$, proving (equation).

Lemma E.4 (Scatter accumulation equals $P_\Omega(YZ^\mathsf{T})Z$). Let $Y \in \mathbb{R}^{n\times r}$ and $Z \in \mathbb{R}^{M\times r}$, and assume Ω contains no duplicates (Assumption B.3). Define $C \in \mathbb{R}^{n\times r}$ by initializing $C = 0$ and, for each $(i^{(e)}, j^{(e)}) \in \Omega$, performing the update

$$C_{i^{(e)},:} \;\leftarrow\; C_{i^{(e)},:} + (YZ^\mathsf{T})_{i^{(e)}j^{(e)}} Z_{j^{(e)},:}.$$

Then

$$C = P_\Omega(YZ^\mathsf{T})Z.$$

Proof. This is exactly the gather–scatter identity of Lemma A.9, with the observation set Ω. The "no duplicates" assumption ensures each observed location contributes once; otherwise the same statement holds for a multiset version of P_Ω with multiplicities.

Proposition E.5 (Correctness of Algorithm 5.3). For every $W \in \mathbb{R}^{n\times r}$, Algorithm 5.3 returns $\mathcal{A}(W)$ as defined in equation (equation).

Proof. Let $W \in \mathbb{R}^{n\times r}$ be arbitrary and set $Y := KW$. Consider the observation loop in Algorithm 5.3.

For each observation e, Lemma E.3 shows that the computed scalar $u^{(e)}$ equals the entry $(KWZ^\mathsf{T})_{i^{(e)}j^{(e)}}$. The update

$$C_{i^{(e)},:} \leftarrow C_{i^{(e)},:} + u^{(e)}z^{(e)}$$

therefore coincides with

$$C_{i^{(e)},:} \leftarrow C_{i^{(e)},:} + (KWZ^\mathsf{T})_{i^{(e)}j^{(e)}}Z_{j^{(e)},:},$$

using Lemma E.1 to replace $z^{(e)}$ by $Z_{j^{(e)},:}$. Hence, after the loop terminates, Lemma E.4 (with $Y = KW$) yields

$$C = P_\Omega(KWZ^\mathsf{T})Z.$$

The return statement of Algorithm 5.3 is `KernelMul(C) +  Y`, which equals

$$KC + \lambda(KW) = K\Big(P_\Omega(KWZ^\mathsf{T})Z\Big) + \lambda KW = \mathcal{A}(W).$$

This is the desired identity.

E.3 Arithmetic cost and storage requirements

We next quantify the computational cost of Algorithm 5.3. We retain the notation $\mathsf{Cost}_K(n,r)$ for the cost of a kernel multiplication $X \mapsto KX$ (Definition 5.1 in the main text).

Proposition E.6 (Arithmetic cost of ApplyA). Assume Algorithm 5.3 is implemented as written, using Algorithm 5.2 to compute each $z^{(e)}$ on the fly. Then the total arithmetic cost of ApplyA is

$$\mathsf{Cost}(\text{ApplyA}) = 2\mathsf{Cost}_K(n,r) \;+\; O(qdr),$$

and no intermediate object of size $n \times M$, $M \times r$, or $nr \times nr$ is formed.

Proof. Algorithm 5.3 performs:

1. One kernel multiplication to compute $Y = KW$, cost $\mathsf{Cost}_K(n, r)$.
2. One pass over q observations. For each observation:
   - Computing $z^{(e)}$ via Algorithm 5.2 requires $(d-1)$ elementwise multiplications of r-vectors, hence $O((d-1)r) = O(dr)$ operations.
   - Computing the inner product $u^{(e)} = \langle Y_{i^{(e)},:}, z^{(e)} \rangle$ costs $O(r)$ operations.
   - Updating the row $C_{i^{(e)},:} \leftarrow C_{i^{(e)},:} + u^{(e)} z^{(e)}$ costs $O(r)$ operations. Thus each observation costs $O(dr)$ operations, and the loop costs $O(qdr)$.
3. One kernel multiplication to compute KC, cost $\mathsf{Cost}_K(n, r)$.
4. A scaled addition λY and matrix addition $KC + \lambda Y$, cost $O(nr)$, which is dominated by $O(qdr)$ under the regime $n < q$ and $r \geq 1$.

Summing these contributions yields $2\mathsf{Cost}_K(n, r) + O(qdr)$.

Finally, the algorithm stores $W, Y, C \in \mathbb{R}^{n \times r}$ and processes each $z^{(e)} \in \mathbb{R}^r$ transiently; it does not form $Z \in \mathbb{R}^{M \times r}$, $KWZ^\mathsf{T} \in \mathbb{R}^{n \times M}$, or any $nr \times nr$ matrix.

Remark E.7 (Storage complexity). Algorithm 5.3 requires storage for the matrices $W, Y, C \in \mathbb{R}^{n \times r}$, the fixed factor matrices $A_{m_{m \neq k}}$, and the observation list $\mathcal{O}$. In particular, the working memory for the ApplyA routine is $O(nr)$ beyond the storage of the fixed inputs. The transient vector $z^{(e)} \in \mathbb{R}^r$ can be reused across iterations and does not scale with q.

Appendix F. Right-hand Side Construction (Algorithm 5.4 BuildRHS)

This appendix gives a complete proof of the **BuildRHS** part of Proposition 5.5 in the main text. Specifically, we show that Algorithm 5.4 constructs the right-hand side

$$\mathcal{F} = KB$$

where $B = TZ$ (MTTKRP in mode k), using only the observation list and on-the-fly Khatri–Rao row generation. We also provide a detailed arithmetic cost and memory analysis, and we explicitly verify that no computation of order N is required.

We use the conventions of Appendix A (vectorization identities) and Appendix B (observation projection and selection matrices). We assume the observation set has no duplicate locations (Assumption B.3).

F.1 Setup and target quantity

Let $T \in \mathbb{R}^{n \times M}$ denote the mode-k unfolding of the data tensor with missing entries treated as zero. Let $\Omega \subset \{1, \dots, n\} \times \{1, \dots, M\}$ be the set of observed unfolding locations with $|\Omega| = q$, enumerated as $\Omega = \{(i^{(e)}, j^{(e)})\}_{e=1}^{q}$.

Let $Z \in \mathbb{R}^{M \times r}$ be the Khatri–Rao product of the fixed factor matrices for all modes except mode k, and let $K \in \mathbb{R}^{n \times n}$ be the (symmetric) kernel matrix.

Definition F.1 (Target right-hand side). Define

$$B := TZ \in \mathbb{R}^{n \times r}, \qquad \mathcal{F} := KB \in \mathbb{R}^{n \times r}.$$

The right-hand side of the vectorized normal equation is then

$$\mathrm{vec}(\mathcal{F}) = (I_r \otimes K)\mathrm{vec}(B),$$

by Corollary A.3(2).

Assumption F.2 (Zero-filled unfolding and observation values). The observation list $\mathcal{O} = \{(i_1^{(e)}, \dots, i_d^{(e)}, t^{(e)})\}_{e=1}^{q}$ encodes T in the following way: for each e, the observed value satisfies

$$t^{(e)} = T_{i^{(e)} j^{(e)}},$$

and for all unobserved locations $(i, j) \notin \Omega$ we have $T_{ij} = 0$.

This is equivalent to $P_\Omega(T) = T$ (cf. Assumption C.4).

F.2 Correctness of Algorithm 5.4 (BuildRHS)

Algorithm 5.4 constructs a matrix $\widehat{B} \in \mathbb{R}^{n \times r}$ by initializing $\widehat{B} = 0$ and performing, for each observation e, the update

$$\widehat{B}_{i^{(e)},:} \;\leftarrow\; \widehat{B}_{i^{(e)},:} + t^{(e)} z^{(e)},$$

where $z^{(e)} \in \mathbb{R}^r$ is computed on the fly from the observation multi-index $(i_1^{(e)}, \dots, i_d^{(e)})$ as in Algorithm 5.2.

The key fact is that $z^{(e)}$ equals the corresponding row of Z.

Lemma F.3 (On-the-fly row identity). For each observation e, the vector $z^{(e)}$ computed by Algorithm 5.2 satisfies

$$z^{(e)} = Z_{j^{(e)},:}.$$

Proof. This is Lemma E.1, which is a direct consequence of the Khatri–Rao row formula (Lemma A.6).

Proposition F.4 (Correctness of BuildRHS). Under Assumption F.2 and Assumption B.3 (no duplicate observation locations), Algorithm 5.4 returns $\mathcal{F} = KB$ with $B = TZ$.

Proof. We first show that the intermediate matrix $\widehat{B}$ constructed by the observation loop equals $B = TZ$.

Fix a row index $i \in \{1, \dots, n\}$. By the definition of matrix multiplication, the i-th row of B is

$$B_{i,:} = (TZ)_{i,:} = \sum_{j=1}^{M} T_{ij} Z_{j,:}.$$

By Assumption F.2, $T_{ij} = 0$ whenever $(i, j) \notin \Omega$, hence the sum reduces to

$$B_{i,:} = \sum_{j:(i,j)\in\Omega} T_{ij} Z_{j,:}.$$

Now consider the row $\widehat{B}_{i,:}$ produced by the observation loop. Each observation e such that $i^{(e)} = i$ contributes the update

$$\widehat{B}_{i,:} \;\leftarrow\; \widehat{B}_{i,:} + t^{(e)} z^{(e)}.$$

By Assumption F.2, $t^{(e)} = T_{i^{(e)} j^{(e)}} = T_{ij^{(e)}}$, and by Lemma F.3, $z^{(e)} = Z_{j^{(e)},:}$. Therefore each such update adds

$$T_{ij^{(e)}} Z_{j^{(e)},:}$$

to $\widehat{B}_{i,:}$. Because Ω contains no duplicates (Assumption B.3), each $(i, j) \in \Omega$ appears exactly once in the enumeration, so after processing all observations we have

$$\widehat{B}_{i,:} = \sum_{j:(i,j)\in\Omega} T_{ij} Z_{j,:}.$$

Comparing with (equation) shows $\widehat{B}_{i,:} = B_{i,:}$ for every i, hence $\widehat{B} = B$.

Finally, Algorithm 5.4 returns `KernelMul(B)`, which equals $K\widehat{B} = KB = \mathcal{F}$ by Definition F.1.

F.3 Arithmetic cost and storage requirements

We now quantify the complexity of Algorithm 5.4. As in the main text, let $\mathsf{Cost}_K(n,r)$ denote the cost of applying the kernel multiplication $X \mapsto KX$ to an $n \times r$ matrix.

Proposition F.5 (Arithmetic cost of BuildRHS). Assume Algorithm 5.4 computes each $z^{(e)}$ via Algorithm 5.2 (on-the-fly Khatri–Rao row generation). Then the total arithmetic cost of BuildRHS is

$$\mathsf{Cost}(\text{BuildRHS}) = \mathsf{Cost}_K(n,r) + O(qdr),$$

and no intermediate object of size $n \times M$, $M \times r$, or $nr \times nr$ is formed.

Proof. Algorithm 5.4 consists of:

1. An initialization $B \leftarrow 0_{n\times r}$, cost $O(nr)$.
2. A loop over q observations. For each observation:
   - Computing $z^{(e)}$ via Algorithm 5.2 performs $(d-1)$ elementwise multiplications of r-vectors, hence costs $O((d-1)r) = O(dr)$.
   - Updating a single row by $B_{i^{(e)},:} \leftarrow B_{i^{(e)},:} + t^{(e)}z^{(e)}$ costs $O(r)$. Thus, each observation costs $O(dr)$ and the loop costs $O(qdr)$.
3. One kernel multiplication $\mathcal{F} \leftarrow KB$, cost $\mathsf{Cost}_K(n,r)$.
4. No other step dominates these costs.

Therefore, the total cost is $\mathsf{Cost}_K(n,r) + O(qdr)$. The only matrices formed are $B \in \mathbb{R}^{n\times r}$ and the output $\mathcal{F} \in \mathbb{R}^{n\times r}$; the vectors $z^{(e)} \in \mathbb{R}^r$ are transient. No $n \times M$ unfolding, full $Z \in \mathbb{R}^{M\times r}$, or $nr \times nr$ system matrix is constructed.

Remark F.6 (Duplicates and aggregation). If the raw observation list contains multiple entries with the same unfolding location (i,j), then Algorithm 5.4 implicitly aggregates them by summation at the level of $\widehat{B}$. This corresponds to replacing T by a zero-filled unfolding whose observed entries are the sums of all measurements at each location. If a different aggregation rule is desired (e.g., averaging), one should preprocess the observation list so that Assumption B.3 holds.

Remark F.7 (Explicit avoidance of $O(N)$ work). Algorithm 5.4 performs exactly one pass over the observation list (length q) and one kernel multiplication on an $n \times r$ matrix. In particular, it does not require constructing or iterating over the full index set of size $N = nM$, and thus satisfies the requirement of avoiding $O(N)$ computation in the regime $q \ll N$.

Appendix G. Symmetry and Positive Definiteness: Conditions for PCG

This appendix provides the complete justification for the symmetry and positive definiteness assumptions required to apply the (preconditioned) conjugate gradient method to the normal equation in the main text (equation (equation)). We also state explicitly what fails when the kernel matrix is only positive semidefinite and how to restore the conditions needed by PCG.

Throughout, we use the conventions of Appendix A (vectorization, Kronecker identities) and Appendix B (selection matrices and the projection P_Ω). We denote by $\langle X, Y\rangle_F := \operatorname{trace}(X^\intercal Y)$ the Frobenius inner product, so that $\langle X, Y\rangle_F = \operatorname{vec}(X)^\intercal \operatorname{vec}(Y)$.

G.1 Basic properties of the selection mask

Definition G.1 (Selection operator and mask). Let $N \in \mathbb{N}$ and let $S \in \mathbb{R}^{N\times q}$ be a selection matrix whose columns are standard basis vectors in $\mathbb{R}^N$. Define the mask matrix

$$P := SS^\intercal \in \mathbb{R}^{N\times N}.$$

Lemma G.2 (Symmetry and positive semidefiniteness of $SS^\intercal$). The matrix $P = SS^\intercal$ is symmetric and positive semidefinite. Moreover, if the selected indices contain no duplicates (equivalently, if $S^\intercal S = I_q$), then P is an orthogonal projector, i.e., $P^2 = P$.

Proof. Symmetry is immediate:

$$P^\intercal = (SS^\intercal)^\intercal = SS^\intercal = P.$$

For any $y \in \mathbb{R}^N$,

$$y^\intercal P y = y^\intercal SS^\intercal y = (S^\intercal y)^\intercal (S^\intercal y) = |S^\intercal y|_2^2 \geq 0,$$

so P is positive semidefinite. If additionally $S^\intercal S = I_q$, then

$$P^2 = SS^\intercal SS^\intercal = S(S^\intercal S)S^\intercal = SI_qS^\intercal = P,$$

so P is an orthogonal projector.

G.2 Symmetry and positive semidefiniteness of the data term

Lemma G.3 (Sandwich form $M^\mathsf{T} P M$ is symmetric positive semidefinite). Let $P \in \mathbb{R}^{N\times N}$ be symmetric positive semidefinite and let $M \in \mathbb{R}^{N\times m}$ be arbitrary. Then the matrix

$$H := M^\mathsf{T} P M \in \mathbb{R}^{m\times m}$$

is symmetric positive semidefinite. In particular, for all $x \in \mathbb{R}^m$,

$$x^\mathsf{T} H x = (Mx)^\mathsf{T} P (Mx) \geq 0.$$

Proof. Symmetry follows from $P^\mathsf{T} = P$:

$$H^\mathsf{T} = (M^\mathsf{T} P M)^\mathsf{T} = M^\mathsf{T} P^\mathsf{T} M = M^\mathsf{T} P M = H.$$

For positive semidefiniteness, fix $x \in \mathbb{R}^m$ and set $y := Mx \in \mathbb{R}^N$. Then

$$x^\mathsf{T} H x = x^\mathsf{T} M^\mathsf{T} P M x = y^\mathsf{T} P y \geq 0$$

since P is positive semidefinite.

Applying Lemma G.3 with $P = SS^\mathsf{T}$ and $M = Z \otimes K$ shows that the data-fitting term in equation (equation) is always symmetric positive semidefinite, independent of definiteness of K (as long as K is symmetric).

G.3 Positive definiteness of the regularization term

The PCG method requires a symmetric positive definite (SPD) coefficient matrix. The regularization term $\lambda(I_r \otimes K)$ is the source of strict positive definiteness when K is SPD.

Lemma G.4 (Kronecker product preserves SPD). Let $K \in \mathbb{R}^{n\times n}$ be symmetric. Then K is SPD if and only if $I_r \otimes K$ is SPD. Equivalently, for $x \in \mathbb{R}^{nr}$, $x \neq 0$ implies

$$x^\mathsf{T}(I_r \otimes K)x > 0$$

if and only if K is SPD.

Proof. First assume K is SPD. Let $x \in \mathbb{R}^{nr}$ be nonzero and reshape it into $W \in \mathbb{R}^{n\times r}$ so that $x = \mathrm{vec}(W)$. By Corollary A.3(2),

$$(I_r \otimes K)\mathrm{vec}(W) = \mathrm{vec}(KW).$$

Therefore,

$$x^\mathsf{T}(I_r \otimes K)x = \mathrm{vec}(W)^\mathsf{T}\mathrm{vec}(KW) = \langle W, KW \rangle_F = \mathrm{trace}(W^\mathsf{T}KW).$$

Writing $W = [w_1, \dots, w_r]$ in columns, we have

$$\mathrm{trace}(W^\mathsf{T}KW) = \sum_{s=1}^{r} w_s^\mathsf{T}Kw_s.$$

Since $x \neq 0$, at least one column w_s is nonzero, and since K is SPD, $w_s^\mathsf{T}Kw_s > 0$ for that column. Hence the sum is strictly positive, proving that $I_r \otimes K$ is SPD.

Conversely, assume $I_r \otimes K$ is SPD. Take any nonzero $v \in \mathbb{R}^n$ and define $x := e_1 \otimes v \in \mathbb{R}^{nr}$, where $e_1 \in \mathbb{R}^r$ is the first standard basis vector. Then $x \neq 0$ and

$$x^\mathsf{T}(I_r \otimes K)x = (e_1^\mathsf{T}I_re_1)(v^\mathsf{T}Kv) = v^\mathsf{T}Kv.$$

Since $I_r \otimes K$ is SPD, the left-hand side is > 0, hence $v^\mathsf{T}Kv > 0$ for all $v \neq 0$, which means K is SPD.

G.4 SPD of the full coefficient matrix

We now prove the main SPD statement used to justify PCG in the main text.

Proposition G.5 (SPD of the Kronecker normal equation coefficient). Let $K \in \mathbb{R}^{n\times n}$ be symmetric positive definite and let $\lambda > 0$. Let $Z \in \mathbb{R}^{M\times r}$ be arbitrary, and let $S \in \mathbb{R}^{N\times q}$ be a selection matrix with $N = nM$. Define

$$A := (Z \otimes K)^\mathsf{T}SS^\mathsf{T}(Z \otimes K) + \lambda(I_r \otimes K).$$

Then A is symmetric positive definite.

Proof. Symmetry: by Lemma G.2, SS^T is symmetric; hence by Lemma G.3, $(Z \otimes K)^\mathsf{T}SS^\mathsf{T}(Z \otimes K)$ is symmetric. Also, $I_r \otimes K$ is symmetric since K is symmetric. Therefore A is symmetric.

Positive definiteness: fix $x \in \mathbb{R}^{nr}$ with $x \neq 0$. Let $M := Z \otimes K$ and $P := SS^\mathsf{T}$. Then

$$x^\mathsf{T}Ax = x^\mathsf{T}M^\mathsf{T}PMx + \lambda\, x^\mathsf{T}(I_r \otimes K)x.$$

By Lemma G.3, the first term is nonnegative. By Lemma G.4, $I_r \otimes K$ is SPD because K is SPD, hence the second term is strictly positive:

$$\lambda\, x^\mathsf{T}(I_r \otimes K)x > 0.$$

Therefore $x^{\mathsf{T}}Ax > 0$ for all $x \neq 0$, which proves that A is SPD.

G.5 SPD of the matrix-free operator

The coefficient matrix A corresponds to the matrix-free operator $\mathcal{A}$ of Definition 4.1 (main text) via vectorization (Appendix D). We now state the corresponding operator-level SPD statement and provide a useful energy identity.

Proposition G.6 (Self-adjointness and coercivity of $\mathcal{A}$). Assume K is symmetric positive definite and $\lambda > 0$. Let $\mathcal{A} : \mathbb{R}^{n\times r} \to \mathbb{R}^{n\times r}$ be defined by

$$\mathcal{A}(W) := K\Big(P_\Omega(KWZ^{\mathsf{T}})Z\Big) + \lambda KW.$$

Then $\mathcal{A}$ is self-adjoint and positive definite with respect to the Frobenius inner product, i.e.,

$$\langle U, \mathcal{A}(V)\rangle_F = \langle \mathcal{A}(U), V\rangle_F, \qquad \langle W, \mathcal{A}(W)\rangle_F > 0 \;\text{ for all } W \neq 0.$$

Moreover, the quadratic form admits the representation

$$\langle W, \mathcal{A}(W)\rangle_F = \left|S^{\mathsf{T}}(Z\otimes K)\mathrm{vec}(W)\right|_2^2 + \lambda \mathrm{trace}(W^{\mathsf{T}}KW),$$

and equivalently

$$\langle W, \mathcal{A}(W)\rangle_F = \left|P_\Omega(KWZ^{\mathsf{T}})\right|_F^2 + \lambda \mathrm{trace}(W^{\mathsf{T}}KW).$$

Proof. Let $x := \mathrm{vec}(W) \in \mathbb{R}^{nr}$ and let A be the coefficient matrix of equation (equation). By Proposition D.1 (Appendix D),

$$\mathrm{vec}(\mathcal{A}(W)) = A\mathrm{vec}(W) = Ax.$$

Therefore,

$$\langle W, \mathcal{A}(W)\rangle_F = \mathrm{vec}(W)^{\mathsf{T}}\mathrm{vec}(\mathcal{A}(W)) = x^{\mathsf{T}}Ax.$$

To obtain the identity (equation), expand $x^{\mathsf{T}}Ax$ using $A = M^{\mathsf{T}}PM + \lambda(I_r \otimes K)$ with $M := Z\otimes K$ and $P := SS^{\mathsf{T}}$:

$$x^{\mathsf{T}}Ax = x^{\mathsf{T}}M^{\mathsf{T}}PMx + \lambda\, x^{\mathsf{T}}(I_r\otimes K)x.$$

For the first term,

$$x^{\mathsf{T}}M^{\mathsf{T}}PMx = (Mx)^{\mathsf{T}}P(Mx) = |S^{\mathsf{T}}Mx|_2^2,$$

where we used $P = SS^\mathsf{T}$ and the identity $y^\mathsf{T} SS^\mathsf{T} y = |S^\mathsf{T} y|_2^2$ (Lemma G.2). For the second term, Lemma G.4 implies

$$x^\mathsf{T}(I_r \otimes K)x = \operatorname{trace}(W^\mathsf{T} KW).$$

Combining yields (equation).

To obtain (equation), note that by Corollary A.3(1),

$$Mx = (Z \otimes K)\operatorname{vec}(W) = \operatorname{vec}(KWZ^\mathsf{T}).$$

Also, $S^\mathsf{T}\operatorname{vec}(X)$ extracts the observed entries of X (Lemma B.5), hence

$$|S^\mathsf{T}\operatorname{vec}(KWZ^\mathsf{T})|_2^2 = |P_\Omega(KWZ^\mathsf{T})|_F^2$$

by Proposition B.8. This gives (equation).

Self-adjointness and positive definiteness follow from the fact that A is symmetric positive definite (Proposition G.5) and the identification $\langle U, \mathcal{A}(V)\rangle_F = \operatorname{vec}(U)^\mathsf{T} A \operatorname{vec}(V)$.

G.6 The semidefinite-kernel case: why strict SPD can fail

The main text adopts an SPD assumption on K (or an SPD perturbation $K+\varepsilon I$) to guarantee the applicability of PCG. Here we formalize why strict SPD may fail if K is only positive semidefinite.

Lemma G.7 (If K is singular, the coefficient matrix can be singular). Assume $K \succeq 0$ is symmetric but not SPD, so $\ker(K) \neq 0$. Let $\lambda > 0$, and define

$$A := (Z \otimes K)^\mathsf{T} SS^\mathsf{T}(Z \otimes K) + \lambda(I_r \otimes K).$$

Then $\ker(I_r \otimes K) \subseteq \ker(A)$. In particular, A is not SPD.

Proof. Let $x \in \ker(I_r \otimes K)$ and set $W \in \mathbb{R}^{n\times r}$ such that $x = \operatorname{vec}(W)$. The condition $x \in \ker(I_r \otimes K)$ is equivalent (by Corollary A.3(2)) to

$$0 = (I_r \otimes K)\operatorname{vec}(W) = \operatorname{vec}(KW),$$

hence $KW = 0$.

By Corollary A.3(1),

$$(Z \otimes K)\operatorname{vec}(W) = \operatorname{vec}(KWZ^\mathsf{T}) = \operatorname{vec}(0) = 0.$$

Therefore,

$$(Z \otimes K)^{\mathsf{T}} S S^{\mathsf{T}} (Z \otimes K) x = (Z \otimes K)^{\mathsf{T}} S S^{\mathsf{T}} 0 = 0,$$

and also $\lambda(I_r \otimes K)x = 0$ by assumption. Summing, we obtain $Ax = 0$, i.e., $x \in \ker(A)$. Since $\ker(I_r \otimes K)$ contains nonzero vectors when K is singular, A has a nontrivial nullspace and cannot be SPD.

Remark G.8 (Restoring PCG applicability). Lemma G.7 shows that strict SPD of the coefficient matrix generally fails when K is only positive semidefinite, because directions in $\ker(K)$ are "invisible" to both the data term and the regularization term (both contain K). Two standard remedies are:

1. **Ridge stabilization:** replace K by $K + \varepsilon I$ with $\varepsilon > 0$, making K SPD and hence A SPD by Proposition G.5.
2. **Restricted-space formulation:** solve the system on $\mathrm{Range}(K)$ (or factor K as $K = LL^{\mathsf{T}}$ on its range and reparameterize), which removes the null directions explicitly.

In this manuscript we adopt the first remedy (SPD kernel) as it yields a straightforward, fully matrix-free PCG implementation.