LIGHTWEIGHT REPRESENTATION LEARNING FOR LOW TUCKER RANK TENSORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Low Tucker rank tensors provide compact representations of multidimensional dependencies, but standard learning pipelines can incur substantial computational and memory overhead from redundant intermediate quantities. We reduce this overhead without modifying the underlying representation objective or parameterization. Our approach combines three complementary components: (i) sequential extraction of only task-required spectral components, avoiding unnecessary full decompositions; (ii) factor-wise multilinear contractions without materializing large Kronecker operators; and (iii) compact latent-state updates without storing full ambient tensor iterates. These components preserve the algebraic operations and learning trajectories of the original algorithms while reducing their computational and memory footprints. Across five low Tucker rank tensor estimators, our implementations match the original trajectories to numerical precision (relative discrepancies of 10^{-16} to 10^{-13}), while reducing per-iteration runtime by up to $3.12\times$ and resident memory by up to $616\times$ (99.8%) on the main order-3 benchmark at $n = 100$, with memory reductions exceeding $900\times$ across the full evaluation suite. These results show that substantial computational redundancy can be eliminated from Tucker representation learning without changing the underlying model or learned representation.

1 INTRODUCTION

Truncated singular value decomposition (SVD) remains the standard mechanism for extracting low-dimensional subspaces in large-scale representation learning (Halko et al., 2011; Deng et al., 2024), yet repeatedly recomputing orthogonal subspaces and intermediate high-dimensional matrices dominates arithmetic and memory budgets, even though only task-sufficient coordinates or projection bases are needed. When observations form multidimensional arrays—image ensembles, video, neuroimaging—vectorization additionally discards multiway correlations and inflates parameter counts (Lu et al., 2011; Panagakis et al., 2021); multilinear subspace learning instead estimates a Tucker representation, mode-wise orthogonal projection bases together with a core tensor (De Lathauwer et al., 2000b; Lu et al., 2008; Jia et al., 2014), and tensor completion (partial sampling) and denoising (full sampling with noise) are observational variants of one task: learning a compact Tucker representation of the latent low Tucker rank signal tensor (Lee & Wang, 2020; Li et al., 2025). Structured, compact parameterizations of large linear maps reduce latency and memory without sacrificing expressiveness (Dao et al., 2020), yet standard pipelines for low-rank tensor estimation still instantiate ambient arrays and dense multilinear operators at every iteration. We pose the analogous question one level down, for the estimation procedure itself: can the learning iteration run in a compact representation of its state without changing the representation it learns?

Prior work has addressed this cost piecemeal: rank-one or partial SVD updates (Guo et al., 2017a; Deng et al., 2024), factored or sparse-plus-low-rank states (Bader & Kolda, 2007; Nimishakavi et al., 2018; Yao et al., 2022; Cai et al., 2023), simplified retractions (Yao et al., 2022; Zhang et al., 2026), randomized sketching and streaming (Malik & Becker, 2018; Minster et al., 2020; Sun et al., 2020), and preconditioning (Kasai & Mishra, 2016; Tong et al., 2022; Zhang et al., 2025). These developments remain confined to specific algebraic settings (Section 2): savings for one estimator rarely transfer to another, and an exact method that removes computational redundancy while preserving the objective has been lacking.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Taxonomy of computational efficiency strategies in low-rank tensor estimation.

Strategy	Representative Works	Target
I. Selective computation	Rank-1 SVD (Guo et al., 2017a); truncated SVD (Deng et al., 2024)	FLOPs
II. Compact representation	Factor states (Bader & Kolda, 2007; Nimishakavi et al., 2018); sparse+low-rank (Yao et al., 2022; Cai et al., 2023)	Memory
III. Implicit computation	ST-HOSVD (Vannieuwenhoven et al., 2012); retraction simplification (Zhang et al., 2026)	FLOPs + Memory
IV. Approximate / randomized	Sketching (Malik & Becker, 2018); randomized Tucker (Minster et al., 2020); streaming (Sun et al., 2020); Nyström (Bucci & Robol, 2024)	FLOPs + Data
V. Optimization acceleration	Preconditioning (Kasai & Mishra, 2016); ScaledGD (Tong et al., 2022); PRGD (Zhang et al., 2025); RGN (Luo & Zhang, 2023)	Iterations

CONTRIBUTIONS

1. We propose a lightweight representation learning method for low Tucker rank tensors, built from three modular components: sequential extraction of task-required subspace components without full SVD (Algorithm 1), implicit factor-wise contraction bypassing explicit Kronecker operators (Algorithm 2), and compact latent-state tracking without ambient tensor buffering (Algorithm 3).
2. We prove that the lightweight pipeline is algebraically equivalent to the reference estimators (Theorem 1) and that it lowers the per-iteration cost by one polynomial order in n (from $\Omega(n^{N+1})$ for the literal baseline to $O(n^{N-1})$ plus terms linear in the number $|\Omega|$ of observed entries) while carrying a persistent state of size $O(Nnr + r^N + |\Omega|)$ instead of an ambient $\Omega(n^N)$ array (Theorem 2).
3. We validate the method across five representative convex and nonconvex estimators under identical initialization and iteration horizons: optimization trajectories and recovery accuracy are preserved identically, with up to $3.12\times$ per-iteration speedups and $616\times$ (99.8%) resident memory reductions at $n = 100$ (Table 3) and up to $900\times$ memory reductions across all configurations (Appendix E); the one estimator with a dense spectrum (sum-of-nuclear-norms, SNN) shows memory savings but no runtime gain, as predicted by the analysis.

2 RELATED WORK

Tensor decompositions extend matrix PCA to multiway arrays: HOSVD extracts mode-wise orthogonal subspaces and an all-orthogonal core (De Lathauwer et al., 2000b), such subspaces disentangle factors such as identity, illumination, and pose (Vasilescu & Terzopoulos, 2003), and MPCA learns mode-wise projection bases without vectorization (Lu et al., 2008); later work integrated discriminant analysis (Jia et al., 2014) and formulated completion as learning compact multilinear representations from corrupted observations (Li et al., 2025). Since denoising and completion share the same statistical foundation—isolating the low-rank subspace of a latent multiway signal (Lee & Wang, 2020)—the operational challenge is constructing and updating that representation with minimal arithmetic and memory. Computational accelerations for this task fall into five paradigms (Table 1), each targeting a different resource (FLOPs, memory, data access, or iteration count).

Strategies I–III reduce per-iteration cost without changing the iteration. Selective computation extracts only the singular directions an update needs, as in the rank-one atoms of FFWTensor (Guo et al., 2017a) and fast truncated-SVD updates (Deng et al., 2024). Compact representation responded to the limitations of early SNN relaxations (Gandy et al., 2011; Tomioka et al., 2011; Liu et al., 2012), which are statistically suboptimal ($O(rn^{N-1})$ sample complexity from unbalanced matricization) and require full SVDs and ambient storage per iteration; square-deal relaxations (Mu et al., 2014), tensor nuclear norms (Yuan & Zhang, 2016), and sum-of-squares hierarchies (Barak & Moitra, 2016) improved the sample bounds at the price of NP-hardness or degree-6 SDPs, so scal-

able optimization moved to factor states (Bader & Kolda, 2007; Nimishakavi et al., 2018), sparse-plus-low-rank proximal updates (Yao et al., 2022), and Riemannian regularization (Cai et al., 2023). Implicit structured computation avoids forming large transformation matrices: ST-HOSVD chains mode contractions to shrink intermediates (Vannieuwenhoven et al., 2012), SM-QRGD replaces manifold retractions by single-mode projections (Zhang et al., 2026), and Kaleidoscope maps replace dense operators by learnable factorized ones (Dao et al., 2020).

Strategies IV and V instead change what is computed or how many iterations are needed: randomized methods compress the problem by sketching, subsampling, or streaming (Malik & Becker, 2018; Minster et al., 2020; Sun et al., 2020; Bucci & Robol, 2024), while optimization acceleration targets the condition number κ to reduce outer iterations—building on statistical limits for tensor SVD and completion (Zhang & Xia, 2018; Xia & Yuan, 2019; Xia et al., 2021), Riemannian preconditioning (Kasai & Mishra, 2016), ScaledGD (Tong et al., 2022), and PRGD (Zhang et al., 2025) achieve κ-independent linear rates, and Riemannian Gauss–Newton attains local quadratic convergence (Luo & Zhang, 2023).

Strategy IV inherently incurs information loss through randomized projection or truncation, and Strategy V leaves the per-iteration arithmetic unchanged. This work instead removes the algebraic and structural redundancies inside each iteration of a fixed reference mapping, preserving the objective and its trajectory exactly: Algorithms 1–3 unify Strategies I–III at the component-extraction, operator-evaluation, and state-representation stages, orthogonal to sketching (IV) and preconditioning (V) and composing with either for compound gains (Corollary 1).

3 Lightweight Representation Learning for Low Tucker Rank Tensors

We formalize the structural sources of redundancy in the iterative learning of Tucker representations (Section 3.1), present three modular components that remove them (Section 3.2), and prove exact trajectory equivalence together with the resulting complexity reductions (Section 3.3).

3.1 Sources of Redundancy in Learning Tucker Representations

Let $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_N}$ denote an order-N tensor with ambient dimension $\bar{n} := \prod_{k=1}^{N} n_k$, complementary dimension $n_{-k} := \prod_{j \neq k} n_j$, and mode-$k$ matricization $X_{(k)} \in \mathbb{R}^{n_k \times n_{-k}}$. Parameterized by orthonormal factors $U_k \in \mathbb{R}^{n_k \times r_k}$ and core $\mathcal{G} \in \mathbb{R}^{r_1 \times \cdots \times r_N}$ ($\bar{r} := \prod_k r_k$; $(r_1, \ldots, r_N)$ is the Tucker, or multilinear, rank), the Tucker representation satisfies $\mathcal{X} = \mathcal{G} \times_1 U_1 \cdots \times_N U_N$ and $X_{(k)} = U_k G_{(k)} U_{-k}^{\top}$ with $U_{-k} := U_N \otimes \cdots \otimes U_{k+1} \otimes U_{k-1} \otimes \cdots \otimes U_1 \in \mathbb{R}^{n_{-k} \times r_{-k}}$ (reverse-order Kronecker product, k omitted; $r_{-k} := \prod_{j \neq k} r_j$) (Kolda & Bader, 2009). Given measurements $y = \mathcal{A}(\mathcal{X}^\star) + \varepsilon$ of a low Tucker rank tensor $\mathcal{X}^\star$, its Tucker representation is learned by iteratively evaluating a reference update $\mathcal{X}^{t+1} = \mathcal{T}(\mathcal{X}^t; y)$. Here Ω is the set of $|\Omega|$ observed entries, P_Ω zeroes all entries outside Ω ($(P_\Omega \mathcal{X})_i = \mathcal{X}_i$ for $i \in \Omega$), and $P_{\Omega^\perp} := \mathrm{Id} - P_\Omega$.

Standard pipelines suffer three structural redundancies inside the evaluation loop:

1. **Discarded spectral information**: proximal and projection steps call full SVDs even though sub-threshold or truncated singular components are immediately discarded.
2. **Materialization of transient operators**: contractions are evaluated against Kronecker bases U_{-k}, whose explicit allocation consumes $\bar{n}\bar{r}/(n_k r_k)$ words although the matrix is only applied, never retained.
3. **Storage of factor-determined tensors**: iterates reside on a manifold of dimension $\sum_k n_k r_k + \bar{r}$, yet ambient arrays of size $\bar{n}$ are persistently allocated, although under partial observation subsequent updates depend only on factor coordinates and sparse residuals.

Definition 1 (Lightweight realization and admissible class)**.** *Let $\mathbb{V} := \mathbb{R}^{n_1 \times \cdots \times n_N}$ and let $\mathcal{T} : \mathbb{V} \to \mathbb{V}$ be a reference iteration for fixed data (Ω, y). A* lightweight realization *of $\mathcal{T}$ is a triple $(\mathcal{Z}, D, \tilde{\mathcal{T}})$ consisting of a state set $\mathcal{Z}$, a decoding map $D : \mathcal{Z} \to \mathbb{V}$, and a state transition $\tilde{\mathcal{T}} : \mathcal{Z} \to \mathcal{Z}$ satisfying the* intertwining relation

$$D \circ \tilde{\mathcal{T}} = \mathcal{T} \circ D \qquad \text{on } \mathcal{Z}. \tag{1}$$

Algorithm 1 Sequential Extraction of Required Components

Require: $M : w \mapsto Mw, v \mapsto M^\top v$; $\Phi \in \{\mathcal{D}_\lambda, \Pi_r, \mathrm{LMO}_\tau\}$; $\epsilon_{\mathrm{krylov}} > 0$.
Ensure: $\{(\tilde\sigma_i, \tilde u_i, \tilde v_i)\}_{i\in I}$, $\Phi(M)$.

if $\Phi = \mathcal{D}_\lambda$ **then**
  $(\tilde\sigma_i, \tilde u_i, \tilde v_i)_{i=1}^{s} \leftarrow \mathrm{KrylovBidiag}(M, \epsilon_{\mathrm{krylov}})$ s.t. $\tilde\sigma_s > \lambda \geq \tilde\sigma_{s+1}$
  $I \leftarrow \{1, \dots, s\}$
  $\Phi(M) \leftarrow \sum_{i=1}^{s} (\tilde\sigma_i - \lambda)\tilde u_i \tilde v_i^\top$
else if $\Phi = \Pi_r$ **then**
  $(\tilde\sigma_i, \tilde u_i, \tilde v_i)_{i=1}^{r} \leftarrow \mathrm{KrylovBidiag}(M, r, \epsilon_{\mathrm{krylov}})$
  $I \leftarrow \{1, \dots, r\}$
  $\Phi(M) \leftarrow \sum_{i=1}^{r} \tilde\sigma_i \tilde u_i \tilde v_i^\top$
else if $\Phi = \mathrm{LMO}_\tau$ **then**
  $(\tilde\sigma_1, \tilde u_1, \tilde v_1) \leftarrow \mathrm{PowerIteration}(M, \epsilon_{\mathrm{krylov}})$
  $I \leftarrow \{1\}$
  $\Phi(M) \leftarrow -\tau \tilde u_1 \tilde v_1^\top$
end if
return $\{(\tilde\sigma_i, \tilde u_i, \tilde v_i)\}_{i\in I}, \Phi(M)$

The admissible class $\mathfrak{T}$ consists of the iterations $\mathcal{T} = \mathcal{T}_U \circ \mathcal{T}_L \circ \mathcal{T}_S$ whose stages are (S) spectral primitives $\Phi \in \{\mathcal{D}_\lambda, \Pi_r, \mathrm{LMO}_\tau\}$ (equation 2) applied to mode-k unfoldings of a sparse-plus-factored input, (L) multilinear contractions with factor matrices, and (U) a state update assembled from the outputs of (S) and (L) by finitely many multilinear operations. It covers proximal regularizations (SNN, Subspace Norm), Frank–Wolfe schemes, higher-order orthogonal iteration (HOOI), and Riemannian factor updates (including ScaledGD); the state $\mathcal{Z}$ and decoding D for each estimator are given in Appendix C.

3.2 Modular Lightweight Algorithmic Pipeline

We remove these redundancies without altering the iterates through three modular components:

1. Compute Only Required Components (Algorithm 1). The spectral primitives are the singular-value soft-thresholding operator $\mathcal{D}_\lambda$, the best rank-r truncated-SVD operator Π_r, and the nuclear-norm-ball linear minimization oracle $\mathrm{LMO}_\tau(M) := \arg\min_{\|X\|_* \leq \tau} \langle M, X\rangle$. Writing (σ_i, u_i, v_i) for the singular triplets of M, the output of Φ depends only on the triplets indexed by $I \in \{I_\lambda, I_r, I_{\mathrm{LMO}}\}$, where $I_\lambda(M) := \{i : \sigma_i > \lambda\}$, $I_r(M) := \{1, \dots, r\}$, and $I_{\mathrm{LMO}}(M) := \{1\}$:

$$\mathcal{D}_\lambda(M) = \sum_{i\in I_\lambda(M)} (\sigma_i - \lambda) u_i v_i^\top, \quad \Pi_r(M) = \sum_{i\in I_r(M)} \sigma_i u_i v_i^\top, \quad \mathrm{LMO}_\tau(M) = -\tau u_1 v_1^\top. \tag{2}$$

All remaining components fall in the null space of the primitive or are truncated, so Algorithm 1 extracts exactly these triplets by Krylov (Golub–Kahan–Lanczos) bidiagonalization with early stopping, replacing a full SVD of cost $O(n_k n_{-k} \min(n_k, n_{-k}))$ by

$$\mathrm{FLOP}_{\mathrm{req}} = O\big(q\, c(M) + q^2(n_k + n_{-k})\big), \tag{3}$$

where $s = |I|$, $q \geq s$ is the Krylov subspace dimension, and $c(M)$ is the cost of one matrix-vector product with M and one with $M^\top$.

Proposition 1 (Spectral sufficiency)**.** *Let $M \in \mathbb{R}^{n_k \times n_{-k}}$ have singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$. (i) $\mathcal{D}_\lambda(M)$ is unique, equals equation 2, and depends only on the invariant subspaces $\mathrm{span}\{u_i : \sigma_i > \lambda\}$, $\mathrm{span}\{v_i : \sigma_i > \lambda\}$ and the singular values above λ (not on the choice of singular vectors within repeated singular values). (ii) If $\sigma_r > \sigma_{r+1}$, then $\Pi_r(M)$ is unique and equals $P_{\mathcal{U}_r} M$ with $\mathcal{U}_r = \mathrm{span}\{u_1, \dots, u_r\}$; if $\sigma_1 > \sigma_2$, then $\mathrm{LMO}_\tau(M) = -\tau u_1 v_1^\top$ is the unique minimizer. (iii) Consequently, any procedure returning, in exact arithmetic, the singular triplets indexed by $I_\Phi(M)$ reproduces $\Phi(M)$ exactly.*

The proof and the finite-precision identification condition for I_λ are given in Appendix A.2.

Algorithm 2 Implicit Factor-Wise Contraction

Require: $\mathcal{Y} \in \mathbb{R}^{m_1 \times \cdots \times m_N}$, $\{A_j \in \mathbb{R}^{m_j \times p_j}\}_{j \neq k}$, $k \in \{1, \ldots, N\}$.
Ensure: $Z_{(k)} = \left(\mathcal{Y} \times_{j \neq k} A_j^\top\right)_{(k)} \in \mathbb{R}^{m_k \times p_{-k}}$.

1: $\pi \leftarrow \mathrm{argsort}_{j \in \{1,\ldots,N\} \setminus \{k\}}(m_j/p_j, \text{descending})$
2: $\mathcal{W}^{(0)} \leftarrow \mathcal{Y}$
3: **for** $l = 1, \ldots, N-1$ **do**
4: $\quad \mathcal{W}^{(l)} \leftarrow \mathcal{W}^{(l-1)} \times_{\pi_l} A_{\pi_l}^\top$
5: **end for**
6: $Z_{(k)} \leftarrow W_{(k)}^{(N-1)}$
7: **return** $Z_{(k)}$

2. Apply Operators Without Materializing Them (Algorithm 2). Iterations never require the Kronecker operator $U_{-k} = \bigotimes_{j \neq k} U_j$ itself, only its contracted action $X_{(k)}U_{-k}$, which Algorithm 2 evaluates through sequential mode-wise contractions using the standard multilinear identities (De Lathauwer et al., 2000b; Kolda & Bader, 2009):

$$U_{-k}^\top U_{-k} = \bigotimes_{j \neq k} (U_j^\top U_j) = I_{r_{-k}}, \qquad X_{(k)}U_{-k} = \left(\mathcal{X} \times_{j \neq k} U_j^\top\right)_{(k)} = U_k G_{(k)}. \tag{4}$$

For a sparse tensor $\mathcal{S}$ with non-zero support Ω, factor-wise contraction evaluates in $\mathrm{FLOP}\left(\mathcal{S} \times_{j \neq k} U_j^\top\right) = O(N|\Omega|r_{-k})$.

Proposition 2 (Kronecker-free contraction). *Let $\mathcal{Y} \in \mathbb{R}^{m_1 \times \cdots \times m_N}$, $A_j \in \mathbb{R}^{m_j \times p_j}$ for $j \neq k$, and $A_{-k} := A_N \otimes \cdots \otimes A_{k+1} \otimes A_{k-1} \otimes \cdots \otimes A_1$. For every contraction order $\pi \in \mathfrak{S}_{N-1}$, Algorithm 2 returns $Z_{(k)} = Y_{(k)}A_{-k}$ without forming A_{-k} (whose storage is $m_{-k}p_{-k}$). If $p_j \leq m_j$ for all $j \neq k$, every intermediate $\mathcal{W}^{(l)}$ has at most $\bar{m} := \prod_j m_j$ entries and the arithmetic cost is at most $2(N-1)\bar{m}\max_{j \neq k} p_j$; if $\mathcal{Y}$ is sparse with support Ω, the cost is $O(N|\Omega|p_{-k})$. Moreover, for factors with orthonormal columns, $U_{-k}^\top U_{-k} = I_{r_{-k}}$ and $X_{(k)}U_{-k} = U_k G_{(k)} = (\mathcal{X} \times_{j \neq k} U_j^\top)_{(k)}$, i.e. equation 4 holds.*

The proof is given in Appendix A.3.

3. Store Only Sufficient States (Algorithm 3). We maintain the compact state $z = (\{U_k\}_{k=1}^N, \mathcal{G}, R_\Omega)$ with decoding $D(z) := \mathcal{G} \times_1 U_1 \cdots \times_N U_N$ and $R_\Omega := y - P_\Omega(D(z))$; Algorithm 3 updates this state directly and realizes the *projected imputation iteration*

$$\begin{aligned} \mathcal{T}_{\mathrm{proj}}: \quad & \mathcal{M}^{t+1} := P_\Omega(y) + P_{\Omega^\perp}(\mathcal{X}^t), \qquad \mathcal{U}_k^{t+1} := \mathrm{span}\{u_i(M_{(k)}^{t+1}) : i \in I_\Phi(M_{(k)}^{t+1})\}, \\ & \mathcal{X}^{t+1} := \mathcal{M}^{t+1} \times_{k=1}^N P_{\mathcal{U}_k^{t+1}}, \end{aligned} \tag{5}$$

where $P_\mathcal{U} := UU^\top$; with $\Phi = \Pi_{r_k}$ this is the rank-r_k subspace-projected hard-impute step, with $\Phi = \mathcal{D}_\lambda$ the projection onto the λ-selected subspaces. The operator input decomposes into sparse-plus-low-rank form $\mathcal{M}^{t+1} = R_\Omega^t + D(z^t)$ (Mazumder et al., 2010), so its unfoldings are never stored. Estimators whose primitive shrinks singular values—singular value thresholding (SVT) proximal steps—or accumulates rank-one atoms (Frank–Wolfe) keep the factored output of Φ as their state; the same intertwining argument applies (Appendix C).

Proposition 3 (Compact state intertwining). *Let $z = (\{U_k\}, \mathcal{G}, R_\Omega)$ with $R_\Omega = y - P_\Omega(D(z))$ and let z^+ be the output of one iteration of Algorithm 3 with exact spectral extraction. Then $D(z^+) = \mathcal{T}_{\mathrm{proj}}(D(z))$ and $R_\Omega^+ = y - P_\Omega(D(z^+))$. Each matrix-vector product with $M_{(k)}^{t+1}$ or its transpose costs $O(|\Omega| + n_k r_k + \bar{r} + N n_{-k} r_{\max})$ and the persistent state occupies*

$$\mathrm{MEM}_{\mathrm{cmp}} = \sum_{k=1}^N n_k r_k + \bar{r} + (N+1)|\Omega|, \qquad \rho := 1 - \frac{\mathrm{MEM}_{\mathrm{cmp}}}{\bar{n}} \to 1 \ \text{ as } \ \frac{\sum_k n_k r_k + \bar{r} + |\Omega|}{\bar{n}} \to 0, \tag{6}$$

counted in words (values plus N coordinates per observed entry).

The proof is in Appendix A.4. Lazy element decoding costs $O(N\bar{r})$ FLOPs per entry; factor orthogonality is preserved via compact $O(n_k r_k^2)$ QR factorizations, where $\mathrm{qr}(A)$ returns an orthonormal basis of $\mathrm{col}(A)$.

Algorithm 3 Compact State Tracking and Sparse-plus-Low-Rank Update

Require: $z^0 = (\{U_k^0\}_{k=1}^N, \mathcal{G}^0, R_\Omega^0)$, (Ω, y), $\epsilon_{\text{tol}} > 0$.
Ensure: $z^T = (\{U_k^T\}_{k=1}^N, \mathcal{G}^T, R_\Omega^T)$.
for $t = 0, 1, \dots, T-1$ **do**
  for $k = 1, \dots, N$ **do**
    $M_{(k)}^{t+1} : w \mapsto (R_\Omega^t)_{(k)} w + U_k^t G_{(k)}^t (U_{-k}^t)^\top w$
    $\{(\tilde{\sigma}_i, \tilde{u}_i, \tilde{v}_i)\}_{i \in I_k}, \Phi(M_{(k)}^{t+1}) \leftarrow \text{Alg } 1(M_{(k)}^{t+1})$
    $U_k^{t+1} \leftarrow \text{qr}([\tilde{u}_1, \dots, \tilde{u}_{s_k}])$
  end for
  $\mathcal{G}^{t+1} \leftarrow R_\Omega^t \times_1 (U_1^{t+1})^\top \cdots \times_N (U_N^{t+1})^\top + \mathcal{G}^t \times_{j=1}^N ((U_j^{t+1})^\top U_j^t)$
  $R_\Omega^{t+1} \leftarrow y - P_\Omega\left(\mathcal{G}^{t+1} \times_1 U_1^{t+1} \cdots \times_N U_N^{t+1}\right)$
  if $\|R_\Omega^{t+1} - R_\Omega^t\|_2 / \|y\|_2 \le \epsilon_{\text{tol}}$ **then**
    break
  end if
end for
return $z^{t+1} = (\{U_k^{t+1}\}_{k=1}^N, \mathcal{G}^{t+1}, R_\Omega^{t+1})$

3.3 Theoretical Equivalence and Complexity Guarantees

Composing Algorithms 1–3 yields a lightweight iteration $\tilde{\mathcal{T}} := \tilde{\mathcal{T}}_U \circ \tilde{\mathcal{T}}_L \circ \tilde{\mathcal{T}}_S$ on the compressed state space $\mathcal{Z}$, and Propositions 1–3 show that each stage *intertwines* with its reference counterpart in the sense of equation 1. Theorem 1 turns this into exact trajectory equivalence—the original algorithm, its objective, its loss landscape, and its optimization trajectory are unaltered—and Theorem 2 provides the resulting complexity bounds.

Theorem 1 (Exact Algebraic Trajectory Equivalence)**.** *Let $\mathcal{T} \in \mathfrak{T}$ and let $(\mathcal{Z}, D, \tilde{\mathcal{T}})$ be obtained by executing stage (S) with Algorithm 1, stage (L) with Algorithm 2, and stage (U) on the compact state (Algorithm 3 for $\mathcal{T}_{\text{proj}}$; Appendix C for the remaining estimators). Assume exact arithmetic, that the spectral oracle returns the singular triplets indexed by I_Φ, and, for $\Phi \in \{\Pi_r, \text{LMO}_\tau\}$, the gap condition $\sigma_r > \sigma_{r+1}$ (resp. $\sigma_1 > \sigma_2$) for every matrix to which Φ is applied along the trajectory. If $D(z^0) = \mathcal{X}^0$, then*

$$D(z^t) = \mathcal{X}^t \qquad \text{for all } t \ge 0, \tag{7}$$

and any stopping rule that is a function of $\{\mathcal{X}^s\}_{s \le t}$ and (Ω, y) terminates both iterations at the same index t.

Corollary 1 (Inheritance of Guarantees and Compositionality)**.** *Under the assumptions of Theorem 1: (i) every property of the reference sequence $\{\mathcal{X}^t\}_{t \ge 0}$—convergence rates, condition-number independence, sample-complexity and minimax error bounds, and terminal error—holds verbatim for $\{D(z^t)\}_{t \ge 0}$; (ii) if two iterations $\mathcal{T}_1, \mathcal{T}_2$ admit lightweight realizations with a common $(\mathcal{Z}, D)$, then so does $\mathcal{T}_2 \circ \mathcal{T}_1$. In particular, a sketched (Strategy IV) or preconditioned (Strategy V) iteration that itself belongs to $\mathfrak{T}$ is evaluated exactly by $\tilde{\mathcal{T}}$, so the savings of the two kinds compound.*

Full proofs are given in Appendix A.5 and A.6; the intertwining relations for the individual estimators are established in Appendix C.

Theorem 2 (Per-Iteration Complexity Bounds)**.** *Let r_k bound the mode-k rank of the compact state before and after the iteration, $r_{\max} := \max_k r_k$, $s := \max_k |I^{(k)}| \le r_{\max}$, and let q with $s \le q \le \min_k \min(n_k, n_{-k})$ be the Krylov dimension of Algorithm 1. One iteration of Algorithm 3 costs*

$$\text{FLOP}(\tilde{\mathcal{T}}) = O\Big(qN|\Omega| + q\,(Nr_{\max} + q)\sum_{k=1}^N (n_k + n_{-k}) + N|\Omega|\bar{r} + \sum_{k=1}^N n_k r_k^2 + \bar{r}\sum_{k=1}^N r_k\Big), \tag{8}$$

$$\text{MEM}_{\text{state}}(\tilde{\mathcal{T}}) = \sum_{k=1}^N n_k r_k + \bar{r} + (N+1)|\Omega|, \tag{9}$$

$$\text{MEM}_{\max}(\tilde{\mathcal{T}}) = \text{MEM}_{\text{state}}(\tilde{\mathcal{T}}) + O\big(q \max_k (n_k + n_{-k})\big),$$

Table 2: Evaluated estimators and applicable algorithmic components (ADMM: alternating direction method of multipliers; ALS: alternating least squares).

ID	Estimator	Paradigm	Components
E0	Subspace Norm (Zheng & Tomioka, 2015)	Convex Prox.	Alg. 1, 2, 3
E1	SNN / HaLRTC (Liu et al., 2012)	Convex ADMM	Alg. 1, 3
E2	Frank–Wolfe (FFW) (Guo et al., 2017b)	Convex Greedy	Alg. 1, 3
E3	HOOI (De Lathauwer et al., 2000a)	Nonconvex ALS	Alg. 1, 2, 3
E4	ScaledGD (Tong et al., 2022)	Nonconvex Precond.	Alg. 2, 3

where $\mathrm{MEM}_{\mathrm{state}}$ *counts the words of the persistent state carried between iterations and* $\mathrm{MEM}_{\max}$ *the peak including Krylov scratch space. The literal baseline (full SVD of each unfolding and explicit Kronecker operators) requires* $\mathrm{FLOP} = \Omega\big(\bar{n}\sum_k \min(n_k, n_{-k})\big)$ *and* $\mathrm{MEM} = \Omega(\bar{n})$. *For* $n_k = n$, $r_k = r$, *and fixed* N, *equation 8 is* $O\big(q(q+r)n^{N-1} + (q+r^N)|\Omega|\big)$ *against* $\Omega(n^{N+1})$, *i.e. one polynomial order lower in* n, *and the persistent state is* $O(Nnr + r^N + |\Omega|)$ *against* $\Omega(n^N)$.

Term-by-term derivations are given in Appendix B, and concrete algorithmic instantiations for each estimator in Appendix C.

4 Empirical Validation

4.1 Experimental Setup and Evaluation Protocol

We measure per-iteration runtime, resident state memory, and numerical exactness of the lightweight implementation against literal baselines across five representative estimators from $\mathfrak{T}$ (Table 2) under identical initialization, random seeds, and iteration horizons, validating three claims: (i) numerical exactness of the trajectory (Claim 1; Theorem 1), (ii) arithmetic and memory reduction with empirical slope flattening (Claim 2; Theorem 2), and (iii) identical objective trajectories and terminal recovery accuracy (Claim 3; Corollary 1).

Exactness is profiled via lockstep execution: raw and lightweight routines run in the same outer loop, recording at each t the normalized iterate distance $\mathrm{err}_{\mathrm{exact}}(t) := \|D(z^t) - \mathcal{X}^t\|_F / \|\mathcal{X}^t\|_F$ and the objective gap $|\mathcal{F}(D(z^t)) - \mathcal{F}(\mathcal{X}^t)|$ without storing ambient trajectory arrays. Component contributions are isolated by an incremental ladder ($\mathrm{Raw} \to +\mathrm{Alg}\ 1 \to +\mathrm{Alg}\ 1{+}\mathrm{Alg}\ 2 \to \mathrm{All}$) and a leave-one-out suite ($\mathrm{All} - \mathrm{Alg}\ 1$, $\mathrm{All} - \mathrm{Alg}\ 2$, $\mathrm{All} - \mathrm{Alg}\ 3$). Synthetic low Tucker rank ground truths with geometrically decaying spectra ($\kappa \in \{1, 100\}$) are swept over order-3 tensors with $n \in \{40, 60, 80, 100, 120\}$, multilinear rank $r = 5$, and sampling ratios $p \in \{0.05, 0.1, 0.3\}$, plus order-4 and denoising configurations; hardware, single-threaded BLAS controls, and hyperparameter grids are in Appendix E.

4.2 Validation Results and Empirical Findings

Across all five estimators, the maximum relative discrepancy between decoded and uncompressed iterates stays at machine precision (10^{-16} to 10^{-13}), strictly below the 10^{-10} validation threshold, independently of κ, with terminal objective gaps $|\mathcal{F}(D(z^T)) - \mathcal{F}(\mathcal{X}^T)|$ below 10^{-13} (Figure 1, Table 5). The one minor perturbation, FFW (E2) under the default Krylov tolerance, vanishes entirely under a tighter tolerance (10^{-14}; Appendix A.2), confirming Claim 1 to double-precision limits.

The ablation ladders and summary metrics (Figure 2, Table 3) validate Claim 2: for $n \geq 100$, combining implicit contractions (Alg 2) with compact states (Alg 3) accelerates per-iteration runtime by up to $3.12\times$ and compresses resident memory by up to $616\times$ (a 99.8% reduction) for E0, E2, E3, and E4, and log-log regression (Table 6, Appendix F.2) flattens their empirical scaling slopes by roughly one full power ($\Delta \approx -1$), as predicted by Theorem 2. Conversely, at small scales ($n \leq 50$) and for E1 (SNN), whose nuclear-norm regularization induces dense spectra ($s \approx n$), the per-step overhead of Krylov extraction exceeds the cache efficiency of dense BLAS routines; this cross-over is analyzed in Section 5.

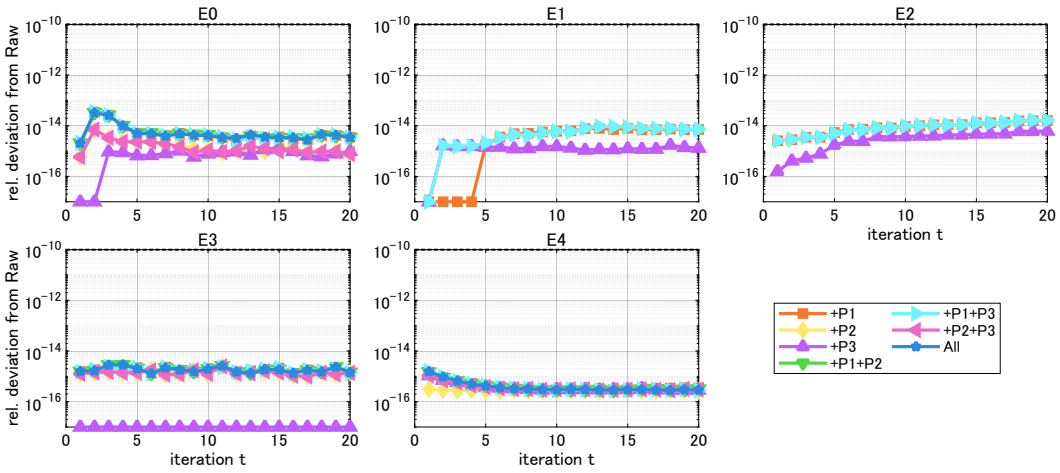


Figure 1: Trajectory exactness under lockstep execution: relative iterate discrepancies $\|D(z^t) - \mathcal{X}^t\|_F / \|\mathcal{X}^t\|_F$ over iterations $t \leq 20$ for $\kappa \in \{1, 100\}$; the dashed line marks the 10^{-10} threshold.

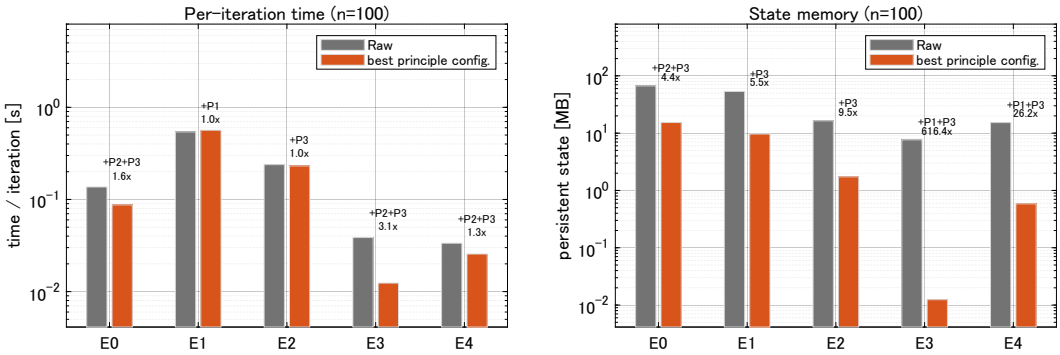


Figure 2: Empirical per-iteration runtime (left, bars) and resident state memory (right, markers) of the best component configuration against Raw at $n = 100$; full ablation stages and $n = 50$ results are in Table 7.

Confirming Claim 3 (Corollary 1), terminal reconstruction errors and objective trajectories match identically between raw and lightweight routines across all benchmarks, so acceleration is strictly decoupled from iteration counts and convergence properties. Micro-benchmarks (Table 8, Appendix F.4) confirm the operative regimes: Alg 1 gains on implicit matvec operators, Alg 2 scales favorably with tensor order N, and Alg 3 compresses most for $p \ll 1$.

5 Discussion and Conclusion

Where exactness comes from and where it fails. Trajectory preservation is the intertwining relation $D \circ \tilde{\mathcal{T}} = \mathcal{T} \circ D$ of Theorem 1, resting on Propositions 1–3: spectral primitives depend only on the triplets indexed by I_Φ, chained contractions reproduce $Y_{(k)}U_{-k}$ for every order, and Algorithm 3's core update projects $\mathcal{M}^{t+1}$ onto the selected subspaces. The assumptions—exact arithmetic, an exact spectral oracle, and the gap conditions $\sigma_r > \sigma_{r+1}$ and $\sigma_1 > \sigma_2$ for Π_r and LMO_τ—are exactly those under which the reference iteration is single-valued. In finite precision the discrepancy is the sum of rounding errors and the Ritz residual of Algorithm 1, bounded by Davis–Kahan in units of the spectral gap (Remark 1), explaining the observed 10^{-16}–10^{-13} discrepancies and the vanishing of the E2 perturbation under a tighter tolerance. Near-degenerate singular values, or λ close to the spectrum, make the reference update itself ill-conditioned; the lightweight update inherits that conditioning rather than adding instability, and the Krylov tolerance must then be set by the distance of

Table 3: Per-estimator runtime (median per iteration) and persistent memory relative to Raw at $n = 100$ on order-3 tensors ($p = 0.05$ for completion); the best lightweight configuration is selected independently for runtime and memory (more sizes in Appendix E).

Estimator	Runtime (s)	Speed-up	Memory (MB)	Reduction
E0 Subspace Norm	.1366 → .0873	**1.57**×	66.81 → 15.32	**4.36**×
E1 SNN / HaLRTC	.5400 → .5849	0.92×	53.41 → 10.09	**5.29**×
E2 Frank–Wolfe (FFW)	.2374 → .2271	**1.05**×	16.41 → 1.54	**10.68**×
E3 HOOI	.0386 → .0124	**3.12**×	7.64 → .0124	**616.38**×
E4 ScaledGD	.0330 → .0195	**1.69**×	15.27 → .3933	**38.83**×

λ to the spectrum. Exactness also makes the claim of an unchanged learned representation meaningful: it extends to the estimation computation the hypothesis-class observation that efficiency need not trade off against what is represented (Dao et al., 2020).

When the speed-up appears. Theorem 2 lowers the per-iteration cost from $\Omega(n^{N+1})$ to $O(q(q+r)n^{N-1} + (q+r^N)|\Omega|)$: the gain, of order $n^2/(q(q+r))$, materializes only when n is large relative to the Krylov dimension q. The slope changes $\Delta \approx -1$ (Table 6) match this prediction; for $n \leq 50$ constant factors dominate, as dense SVDs exploit cache-efficient BLAS Level-3 kernels while Krylov iterations and sparse traversals are bandwidth-bound. The absence of a runtime gain for E1 is structural: a dense spectrum gives $s_j \approx n_j$, so the factored matrix–vector product of Proposition 4 costs $\Theta(\sum_j s_j n_{-j})$, the same order as a dense unfolding, and the premise $q \ll n$ fails; E1 retains the memory reduction, but no speed-up should be expected for effectively dense spectra.

What the memory figures measure. The reported reductions concern the persistent state $\mathrm{MEM}_{\mathrm{state}}$ of equation 9, not the peak including the transient Krylov scratch $O(q \max_k(n_k + n_{-k}))$ (Appendix B.2), which is a fraction q/n of the ambient size and negligible only when $q \ll n$. For partially observed data the sparse residual stores N indices per value and exceeds a dense array once the observation rate passes $p^* = \left[1 + N\,\mathrm{sizeof(idx)}/\mathrm{sizeof(val)}\right]^{-1}$—about 0.25 (0.4) for order-3 tensors with 64-bit (32-bit) indices—concentrating the memory advantage of Algorithm 3 in sparsely observed regimes.

Limitations. The framework changes how an iteration is evaluated, not what it computes: Corollary 1 transfers the reference guarantees verbatim and provides no new statistical or convergence result. Outside $\mathfrak{T}$ are updates not expressible by spectral primitives, multilinear contractions, and finitely many multilinear operations, e.g. entrywise nonlinearities on the ambient tensor (entrywise robust losses, nonlinear link functions) or non-spectral proximal maps. Even within $\mathfrak{T}$, intertwining does not guarantee a bounded state: the ADMM multipliers of HaLRTC accumulate factored terms and the Frank–Wolfe atom pool grows linearly with t, so long runs require periodic re-compression, which alters the reference iteration and falls outside Theorem 1. The empirical study is confined to synthetic order-3 and order-4 tensors with $n \leq 120$ on a single CPU node; crossover points depend on the BLAS implementation and will shift on GPUs or real data with slowly decaying spectra.

Compounding. By Corollary 1(ii), a sketched (Strategy IV) or preconditioned (Strategy V) iteration that stays in $\mathfrak{T}$ is evaluated exactly by the same pipeline, and the savings multiply—sketching reduces the problem size, preconditioning the iteration count, the present framework the cost of one fixed iteration; exactness then refers to that composed iteration, whose trajectory differs from the unsketched one.

Conclusion. We introduced a lightweight representation learning pipeline for low Tucker rank tensors (Algorithms 1–3) that intertwines with the reference iteration (Theorem 1) while lowering the per-iteration cost by one polynomial order and the persistent state to $O(Nnr + r^N + |\Omega|)$ (Theorem 2); across five estimators the decoded trajectories coincide with the reference to machine precision, with speed-ups up to $3.12\times$ and memory reductions up to $616\times$ at $n = 100$.

REFERENCES

Brett W Bader and Tamara G Kolda. Efficient matlab computations with sparse and factored tensors. *SIAM Journal on Scientific Computing*, 30(1):205–231, 2007.

Boaz Barak and Ankur Moitra. Noisy tensor completion via the sum-of-squares hierarchy. In *Proceedings of the 29th Conference on Learning Theory (COLT)*, volume 49 of *JMLR Workshop and Conference Proceedings*, pp. 417–445, 2016.

Alberto Bucci and Leonardo Robol. A multilinear nyström algorithm for low-rank approximation of tensors in tucker format. *SIAM Journal on Matrix Analysis and Applications*, 45(4):1685–1711, 2024.

Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

Jian-Feng Cai, Jingyang Li, and Dong Xia. Generalized low-rank plus sparse tensor estimation by fast riemannian optimization. *Journal of the American Statistical Association*, 118(544):2588–2604, 2023. doi: 10.1080/01621459.2022.2063131.

Tri Dao, Nimit Sohoni, Albert Gu, Matthew Eichhorn, Amit Blonder, Megan Leszczynski, Atri Rudra, and Christopher Ré. Kaleidoscope: An efficient, learnable representation for all structured linear maps. In *International Conference on Learning Representations (ICLR)*, 2020.

Chandler Davis and William M. Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. On the best rank-1 and rank-$(r_1, r_2, \ldots, r_n)$ approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000a.

Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000b.

Haishan Deng, Yan Yang, Jian Li, Chao Chen, Weidong Jiang, and Shiliang Pu. Fast updating truncated svd for representation learning with sparse matrices. In *International Conference on Learning Representations (ICLR)*, 2024.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Silvia Gandy, Benjamin Recht, and Isao Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 4th edition, 2013.

Xiaowei Guo, Quanming Yao, and James Kwok. Efficient sparse low-rank tensor completion using the frank-wolfe algorithm. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, pp. 1948–1954, 2017a.

Yang Guo, Quanming Yao, and James T. Kwok. Efficient sparse low-rank tensor completion using the frank-wolfe algorithm. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 31, pp. 1948–1954, 2017b.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53 (2):217–288, 2011.

Chengcheng Jia, Guoqiang Zhong, and Yun Fu. Low-rank tensor learning with discriminant analysis for action classification and image recovery. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 1228–1234, 2014.

Hiroyuki Kasai and Bamdev Mishra. Low-rank tensor completion: a riemannian manifold preconditioning approach. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, volume 48, pp. 1012–1021, 2016.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM Review*, 51(3): 455–500, 2009.

Chanwoo Lee and Miaoyan Wang. Tensor denoising and completion based on ordinal observations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pp. 5778–5788. PMLR, 2020.

Xi-Zhuo Li, Tai-Xiang Jiang, Liqiao Yang, and Guisong Liu. Learning a more compact representation for low-rank tensor completion. *Neurocomputing*, 617:129036, 2025.

Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2012.

Haiping Lu, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos. Mpca: Multilinear principal component analysis of tensor objects. *IEEE Transactions on Neural Networks*, 19(1): 18–39, 2008.

Haiping Lu, Konstantinos N Plataniotis, and Anastasios N Venetsanopoulos. A survey of multilinear subspace learning for tensor data. *Pattern Recognition*, 44(7):1540–1551, 2011.

Yuetian Luo and Anru R Zhang. Low-rank tensor estimation via riemannian gauss-newton: Statistical optimality and second-order convergence. *Journal of Machine Learning Research*, 24(381): 1–48, 2023.

Osman Asif Malik and Stephen Becker. Low-rank tucker decomposition of large tensors using tensorsketch. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pp. 10096–10106, 2018.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11:2287–2322, 2010.

Rachel Minster, Arvind K Saibaba, and Misha E Kilmer. Randomized algorithms for low-rank tensor decompositions in the Tucker format. *SIAM Journal on Mathematics of Data Science*, 2 (1):189–215, 2020.

Cun Mu, Bo Huang, John Wright, and Donald Goldfarb. Square deal: Lower bounds and improved relaxations for tensor recovery. In *International Conference on Machine Learning (ICML)*, pp. 73–81, 2014.

Madhav Nimishakavi, Pratik Jawanpuria, and Bamdev Mishra. A dual framework for low-rank tensor completion. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pp. 5484–5495, 2018.

Yannis Panagakis, Jean Kossaifi, Grigorios G Chrysos, James Oldfield, Mihalis A Nicolaou, Anima Anandkumar, and Stefanos Zafeiriou. Tensor methods in computer vision and deep learning. *Proceedings of the IEEE*, 109(5):863–890, 2021.

Yousef Saad. On the rates of convergence of the Lanczos and the block-Lanczos methods. *SIAM Journal on Numerical Analysis*, 17(5):687–706, 1980.

Yiming Sun, Yang Guo, Charlene Luo, Joel Tropp, and Madeleine Udell. Low-rank tucker approximation of a tensor from streaming data. *SIAM Journal on Mathematics of Data Science*, 2(4): 1123–1150, 2020. doi: 10.1137/19M1257718.

Ryota Tomioka, Taiji Suzuki, Kohei Hayashi, and Hisashi Kashima. Statistical performance of convex tensor decomposition. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 24, pp. 972–980, 2011.

Tian Tong, Cong Ma, Ashley Prater-Bennette, Erin Tripp, and Yuejie Chi. Scaling and scalability: Provable nonconvex low-rank tensor estimation from incomplete measurements. *Journal of Machine Learning Research*, 23(163):1–77, 2022.

Nick Vannieuwenhoven, Raf Vandebril, and Karl Meerbergen. A new truncation strategy for the higher-order singular value decomposition. *SIAM Journal on Scientific Computing*, 34(2):A1027–A1052, 2012.

M Alex O Vasilescu and Demetri Terzopoulos. Multilinear subspace analysis of image ensembles. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pp. II–93, 2003.

Dong Xia and Ming Yuan. On polynomial time methods for exact low-rank tensor completion. *Foundations of Computational Mathematics*, 19(6):1235–1276, 2019.

Dong Xia, Ming Yuan, and Cun-Hui Zhang. Statistically optimal and computationally efficient low rank tensor completion from noisy entries. *The Annals of Statistics*, 49(1):76–99, 2021. doi: 10.1214/20-AOS1942.

Quanming Yao, Yaqing Wang, Bo Han, and James T Kwok. Low-rank tensor learning with nonconvex overlapped nuclear norm regularization. *Journal of Machine Learning Research*, 23(136): 1–60, 2022.

Ming Yuan and Cun-Hui Zhang. On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics*, 16(4):1031–1068, 2016. doi: 10.1007/s10208-015-9269-5.

Anru Zhang and Dong Xia. Tensor svd: Statistical and computational limits. *IEEE Transactions on Information Theory*, 64(11):7311–7338, 2018. doi: 10.1109/TIT.2018.2841377.

Yuanwei Zhang, Fengmiao Bian, Xiaoqun Zhang, and Jian-Feng Cai. Preconditioned riemannian gradient descent algorithm for low-multilinear-rank tensor completion. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267, pp. 74490–74514, 2025.

Yuanwei Zhang, Ya-Nan Zhu, and Xiaoqun Zhang. A single-mode quasi riemannian gradient descent algorithm for low-multilinear-rank tensor recovery. *Journal of Scientific Computing*, 107 (1):1–28, 2026. doi: 10.1007/s10915-026-02187-z.

Qinqing Zheng and Ryota Tomioka. Interpolating convex and non-convex tensor decompositions via the subspace norm. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, pp. 3106–3114, 2015.

A Complete Proofs of Theorems and Propositions

A.1 Notation, Conventions, and Standard Multilinear Identities

Throughout, $[N] := \{1, \dots, N\}$, $\mathbb{V} := \mathbb{R}^{n_1 \times \cdots \times n_N}$, and $\bar{n}$, n_{-k}, $\bar{r}$, r_{-k} are as in Section 3.1. For $\mathcal{Y} \in \mathbb{R}^{m_1 \times \cdots \times m_N}$, the mode-$k$ unfolding $Y_{(k)} \in \mathbb{R}^{m_k \times m_{-k}}$ places entry $(i_1, \dots, i_N)$ in row i_k and column $1 + \sum_{j \neq k} (i_j - 1) \prod_{l<j,\, l \neq k} m_l$ (Kolda & Bader, 2009). The mode-j product with $A \in \mathbb{R}^{p \times m_j}$ is

$$(\mathcal{Y} \times_j A)_{i_1 \dots i_{j-1}\, \alpha\, i_{j+1} \dots i_N} := \sum_{i_j=1}^{m_j} \mathcal{Y}_{i_1 \dots i_N}\, A_{\alpha i_j}. \tag{10}$$

For matrices A_j ($j \in [N]$) we write $A_{-k} := A_N \otimes \cdots \otimes A_{k+1} \otimes A_{k-1} \otimes \cdots \otimes A_1$ (reverse order, k omitted). We use the following standard facts (Kolda & Bader, 2009, Secs. 2.5 and 4.1), valid for all conformable matrices:

(F1) $(\mathcal{Y} \times_j A)_{(j)} = A\, Y_{(j)}$;

(F2) $\mathcal{Y} \times_j A \times_m B = \mathcal{Y} \times_m B \times_j A \;\; (j \neq m), \qquad \mathcal{Y} \times_j A \times_j B = \mathcal{Y} \times_j (BA)$;

(F3) $\big(\mathcal{Y} \times_1 A_1 \times_2 A_2 \cdots \times_N A_N\big)_{(k)} = A_k\, Y_{(k)}\, A_{-k}^{\top}$;

(F4) $\operatorname{vec}\big(\mathcal{Y} \times_1 A_1 \cdots \times_N A_N\big) = (A_N \otimes \cdots \otimes A_1)\, \operatorname{vec}(\mathcal{Y})$,

where vec stacks entries with the first index varying fastest, so that $\mathrm{vec}(\mathcal{Y}) = \mathrm{vec}(Y_{(1)})$. (F1) and (F2) follow directly from the definition; (F3) is the unfolding rule for multilinear products, and (F4) is (F3) for $k = 1$ combined with $\mathrm{vec}(AXB) = (B^\top \otimes A)\,\mathrm{vec}(X)$ and the associativity of $\otimes$. Mode products are linear in $\mathcal{Y}$, and we abbreviate $\mathcal{Y} \times_{j=1}^{N} A_j := \mathcal{Y} \times_1 A_1 \times_2 A_2 \cdots \times_N A_N$ and $\mathcal{Y} \times_{j\neq k} A_j$ for the same product with mode k omitted (well defined by (F2)). For a subspace $\mathcal{U} \subseteq \mathbb{R}^{n_k}$ with orthonormal basis matrix U, $P_{\mathcal{U}} := UU^\top$ denotes the orthogonal projector; it depends on $\mathcal{U}$ only. P_Ω zeroes all entries outside Ω, $P_{\Omega^\perp} := \mathrm{Id} - P_\Omega$, and $P_\Omega(y) \in \mathbb{V}$ denotes the tensor carrying the observations y on Ω and zero elsewhere; thus $R_\Omega = y - P_\Omega(D(z))$ in Algorithm 3 is shorthand for the sparse tensor $P_\Omega(y) - P_\Omega(D(z))$. We also use the mixed-product rule $(A \otimes B)(C \otimes D) = (AC) \otimes (BD)$ and $(A \otimes B)^\top = A^\top \otimes B^\top$.

A.2 Proof of Proposition 1 (Spectral Sufficiency) and Krylov Perturbation Analysis

Let $M = \sum_{i=1}^{m} \sigma_i u_i v_i^\top$ with $m := \min(n_k, n_{-k})$ be a singular value decomposition of $M \in \mathbb{R}^{n_k \times n_{-k}}$, $\sigma_1 \geq \cdots \geq \sigma_m \geq 0$, with orthonormal $\{u_i\}$ and $\{v_i\}$.

(i) Singular value thresholding. For $\lambda > 0$ the objective $X \mapsto \frac{1}{2}\|X - M\|_F^2 + \lambda\|X\|_*$ is strongly convex, so $\mathcal{D}_\lambda(M)$ is unique, and by Theorem 2.1 of Cai et al. (2010)

$$\mathcal{D}_\lambda(M) = \sum_{i=1}^{m} \max(0, \sigma_i - \lambda)\, u_i v_i^\top. \tag{11}$$

Every index $i \notin I_\lambda(M) = \{i : \sigma_i > \lambda\}$ has $\sigma_i \leq \lambda$, hence $\max(0, \sigma_i - \lambda) = 0$, and equation 11 reduces to the first identity in equation 2. For basis independence, put $g(\mu) := \max(0, 1 - \lambda/\sqrt{\mu})$ for $\mu > 0$ and $g(0) := 0$. Then $M\, g(M^\top M) = \sum_i \sigma_i\, g(\sigma_i^2)\, u_i v_i^\top = \sum_i \max(0, \sigma_i - \lambda)\, u_i v_i^\top = \mathcal{D}_\lambda(M)$, where $g(M^\top M)$ is the matrix function of the symmetric matrix $M^\top M$. A matrix function of a symmetric matrix is defined through its spectral projectors and does not depend on the eigenbasis chosen within an eigenspace (Golub & Van Loan, 2013, Ch. 9). Hence $\mathcal{D}_\lambda(M)$ is a function of M alone: any orthonormal bases of the (unique) invariant subspaces $\mathcal{U}_\lambda := \mathrm{span}\{u_i : \sigma_i > \lambda\}$ of $MM^\top$ and $\mathcal{V}_\lambda := \mathrm{span}\{v_i : \sigma_i > \lambda\}$ of $M^\top M$, together with the singular values above λ, reproduce it.

(ii) Rank-r projection. By the Eckart–Young–Mirsky theorem (Eckart & Young, 1936), $\sum_{i \leq r} \sigma_i u_i v_i^\top$ minimizes $\|X - M\|_F$ over $\mathrm{rank}(X) \leq r$, with $\|X - M\|_F^2 = \sum_{i>r} \sigma_i^2$. We show uniqueness under $\sigma_r > \sigma_{r+1}$. Let X be any minimizer and $\mathcal{C} := \mathrm{range}(X)$, $\dim \mathcal{C} \leq r$. Since $P_{\mathcal{C}} M$ is the closest matrix to M among those with column space in $\mathcal{C}$, $\|M - X\|_F^2 = \|P_{\mathcal{C}} M - X\|_F^2 + \|(I - P_{\mathcal{C}})M\|_F^2 \geq \|M\|_F^2 - \mathrm{tr}(P_{\mathcal{C}} M M^\top)$. Writing $w_i := \|P_{\mathcal{C}} u_i\|^2 \in [0, 1]$ with $\sum_i w_i \leq \mathrm{tr}(P_{\mathcal{C}}) \leq r$, we have $\mathrm{tr}(P_{\mathcal{C}} M M^\top) = \sum_i \sigma_i^2 w_i \leq \sum_{i \leq r} \sigma_i^2$, and because $\sigma_i^2 \leq \sigma_{r+1}^2 < \sigma_r^2$ for all $i > r$, any deficit $1 - w_i > 0$ at an index $i \leq r$ can only be compensated by weight at indices with strictly smaller σ_i^2, so the last inequality is an equality only if $w_i = 1$ for all $i \leq r$. Optimality of X forces equality throughout, so $\|P_{\mathcal{C}} u_i\| = 1$, i.e. $u_i \in \mathcal{C}$, for $i \leq r$; since $\dim \mathcal{C} \leq r$, $\mathcal{C} = \mathcal{U}_r := \mathrm{span}\{u_1, \ldots, u_r\}$, and $X = P_{\mathcal{C}} M = P_{\mathcal{U}_r} M = \sum_{i \leq r} \sigma_i u_i v_i^\top$. Thus $\Pi_r(M) = P_{\mathcal{U}_r} M$ is unique and depends only on the invariant subspace $\mathcal{U}_r$ (which is unique under the gap). If $\sigma_r = \sigma_{r+1}$, uniqueness fails (e.g. $M = I_2$, $r = 1$), which is why the gap condition is part of the hypotheses of Theorem 1.

(ii') Linear minimization oracle. Let $\mathcal{B}_\tau := \{S : \|S\|_* \leq \tau\}$ and G with SVD $\sum_i \sigma_i(G) u_i v_i^\top$. By the duality $\langle S, G\rangle \geq -\|S\|_* \|G\|_2$ of the nuclear and spectral norms, $\min_{S \in \mathcal{B}_\tau} \langle S, G\rangle \geq -\tau\sigma_1(G)$, and $S = -\tau u_1 v_1^\top$ attains this value, proving equation 2. For uniqueness under $\sigma_1(G) > \sigma_2(G)$, let $S = \sum_j \rho_j a_j b_j^\top$ be an SVD of any minimizer ($\rho_j > 0$, $\sum_j \rho_j \leq \tau$). Then $-\tau\sigma_1(G) = \sum_j \rho_j\, a_j^\top G b_j \geq -\sum_j \rho_j\, \sigma_1(G) \geq -\tau\sigma_1(G)$, so $\sum_j \rho_j = \tau$ and $a_j^\top G b_j = -\sigma_1(G)$ for every j. For unit vectors, $-a^\top G b \leq \|Gb\| \leq \sigma_1(G)$ with equality only if b lies in the top right singular subspace, which is $\mathrm{span}\{v_1\}$ under the gap, and $-a \parallel Gb$; hence $a_j b_j^\top = -u_1 v_1^\top$ for all j and $S = -\tau u_1 v_1^\top$.

(iii) Exact extraction. By (i)–(ii'), $\Phi(M)$ is a function of the singular triplets indexed by $I_\Phi(M)$ only (for $\mathcal{D}_\lambda$ and Π_r even of the corresponding invariant subspaces and singular values only). Therefore any procedure that returns these triplets exactly reproduces $\Phi(M)$; which procedure is used is immaterial for Theorem 1. Algorithm 1 is one such procedure: q steps of Golub–Kahan–Lanczos bidiagonalization cost q matvec pairs $c(M)$ plus $O(q^2(n_k + n_{-k}))$ for full reorthogonalization, the Ritz triplets follow from the SVD of a $q \times q$ bidiagonal matrix in $O(q^3)$, and $s \le q$ Ritz vectors are assembled in $O(qs(n_k + n_{-k}))$ (Golub & Van Loan, 2013, Ch. 10), which gives equation 3.

Remark 1 (Multiplicities and finite precision). *(a) A single-vector Krylov process reaches, in exact arithmetic, the invariant subspaces indexed by I_Φ after finitely many steps only when the required singular values are simple; a singular value of multiplicity $\mu > 1$ requires a block Krylov process of block size at least μ, or restarts with deflation, as implemented in standard partial-SVD routines. Algorithm 1 is understood with such deflation. (b) In finite precision, let the returned Ritz values satisfy $|\tilde\sigma_i - \sigma_i| \le \epsilon$ for $i \le s+1$, where $s = |I_\lambda(M)|$. If $\epsilon < \min(\sigma_s - \lambda,\ \lambda - \sigma_{s+1})$, then $\tilde\sigma_s > \lambda \ge \tilde\sigma_{s+1}$, so the index set I_λ is identified exactly; the stopping rule of Algorithm 1 therefore needs a tolerance below the distance of λ to the spectrum, not a fixed fraction of λ. (c) The accuracy of the extracted subspace is governed by the Davis–Kahan $\sin\Theta$ theorem (Davis & Kahan, 1970): if $\tilde U \in \mathbb{R}^{n_k \times s}$ has orthonormal columns, $T := \tilde U^\top M M^\top \tilde U$, $R := MM^\top \tilde U - \tilde U T$, the eigenvalues of T lie in $[a, b]$, and the eigenvalues of $MM^\top$ other than $\sigma_1^2, \dots, \sigma_s^2$ lie outside $(a-\delta, b+\delta)$ for some $\delta > 0$, then $\|\sin\Theta(\mathcal{U}_s, \operatorname{range}\tilde U)\|_2 \le \|R\|_2/\delta$. The Ritz residual $\|R\|_2$ decreases geometrically in the Krylov dimension q at a rate governed by the relative spectral gap, as in Kaniel–Paige–Saad theory (Saad, 1980), which is why tightening $\epsilon_{\mathrm{krylov}}$ drives the residual perturbation observed for E2 in Table 5 to machine precision.*

A.3 Proof of Proposition 2 (Kronecker-Free Identities and Order Invariance)

Correctness for every order π. By induction on l, $\mathcal{W}^{(l)} = \mathcal{Y} \times_{\pi_1} A_{\pi_1}^\top \cdots \times_{\pi_l} A_{\pi_l}^\top$. By the commutation rule in (F2), the result of $N-1$ products along distinct modes does not depend on their order, so $\mathcal{W}^{(N-1)} = \mathcal{Y} \times_{j\neq k} A_j^\top$ for every $\pi \in \mathfrak{S}_{N-1}$. Applying (F3) with $B_j := A_j^\top$ for $j \neq k$ and $B_k := I_{m_k}$,

$$\big(\mathcal{Y} \times_{j\neq k} A_j^\top\big)_{(k)} = Y_{(k)} \left(A_N^\top \otimes \cdots \otimes A_{k+1}^\top \otimes A_{k-1}^\top \otimes \cdots \otimes A_1^\top\right)^\top = Y_{(k)}\, A_{-k}, \qquad (12)$$

using $(A \otimes B)^\top = A^\top \otimes B^\top$. This proves $Z_{(k)} = Y_{(k)} A_{-k}$, and no entry of A_{-k} is ever formed.

Memory and arithmetic (dense input). Assume $p_j \le m_j$ for all $j \neq k$. The intermediate $\mathcal{W}^{(l)}$ has $\prod_{j\in\{\pi_1,\dots,\pi_l\}} p_j \prod_{j\notin\{\pi_1,\dots,\pi_l\}} m_j \le \bar m$ entries, and only $\mathcal{W}^{(l-1)}$ and $\mathcal{W}^{(l)}$ need to be alive simultaneously, so the peak scratch is at most $2\bar m$ entries. Step l computes $|\mathcal{W}^{(l-1)}|\, p_{\pi_l}/m_{\pi_l}$ output entries, each an inner product of length m_{π_l}, at cost $2|\mathcal{W}^{(l-1)}|p_{\pi_l} \le 2\bar m \max_{j\neq k} p_j$; summing over $l = 1, \dots, N-1$ gives the bound $2(N-1)\bar m \max_{j\neq k} p_j$. Sorting the modes by descending m_j/p_j, as in Algorithm 2, shrinks the intermediate as fast as possible in the first step, since $|\mathcal{W}^{(1)}| = \bar m\, p_{\pi_1}/m_{\pi_1}$ is minimized by that choice; the correctness and the bounds above hold for every π.

Sparse input. If $\mathcal{Y}$ has non-zero support Ω, the output $Z_{(k)} \in \mathbb{R}^{m_k \times p_{-k}}$ can be accumulated directly: for each $(i_1, \dots, i_N) \in \Omega$ with value y, add $y \cdot \big(\bigotimes_{j\neq k} A_j[i_j, :]\big)$ (in the reverse order of A_{-k}) to row i_k of $Z_{(k)}$. Forming the outer product of the $N-1$ row vectors incrementally costs $\sum_{l=2}^{N-1} \prod_{l'\le l} p_{\pi_{l'}} \le (N-2)p_{-k}$ multiplications and the accumulation $2p_{-k}$ flops, so the total is at most $2(N-1)|\Omega|p_{-k} = O(N|\Omega|p_{-k})$, with scratch $O(p_{-k})$ beyond the input and the output. Correctness follows from the entrywise definition of the mode products, $(\mathcal{Y} \times_{j\neq k} A_j^\top)_{i_k,(\alpha_j)_{j\neq k}} = \sum_{(i_j)_{j\neq k}} \mathcal{Y}_{i_1\dots i_N} \prod_{j\neq k} A_j[i_j, \alpha_j]$, in which only $(i_1, \dots, i_N) \in \Omega$ contribute.

Orthogonality and unfolding identities. By the mixed-product rule, $U_{-k}^\top U_{-k} = \bigotimes_{j\neq k} (U_j^\top U_j)$ (reverse order), and $U_j^\top U_j = I_{r_j}$ gives $U_{-k}^\top U_{-k} = I_{r_{-k}}$. For $\mathcal{X} = \mathcal{G} \times_1 U_1 \cdots \times_N U_N$, (F3) yields $X_{(k)} = U_k G_{(k)} U_{-k}^\top$, hence $X_{(k)} U_{-k} = U_k G_{(k)}$. Finally, by (F2), $\mathcal{X} \times_{j\neq k} U_j^\top = \mathcal{G} \times_k U_k \times_{j\neq k} (U_j^\top U_j) = \mathcal{G} \times_k U_k$, whose mode-$k$ unfolding is $U_k G_{(k)}$ by (F1). This proves equation 4. □

Lemma 1 (Implicit Kronecker matvecs). *Let $U_j \in \mathbb{R}^{n_j \times r_j}$ with $r_j \le n_j$ for $j \neq k$, and $r_{\max} := \max_j r_j$. For $w \in \mathbb{R}^{n_{-k}}$ and $c \in \mathbb{R}^{r_{-k}}$, the products $U_{-k}^\top w$ and $U_{-k} c$ can be evaluated by Algorithm 2 (applied to w and c reshaped as order-$(N-1)$ tensors) in at most $2(N-1)n_{-k}r_{\max}$ flops each, with $O(n_{-k})$ scratch, without forming U_{-k}.*

Proof. Reshape w as $\mathcal{W} \in \mathbb{R}^{\times_{j\neq k} n_j}$ (so that $\mathrm{vec}(\mathcal{W}) = w$). By (F4), $U_{-k}^\top w = \mathrm{vec}(\mathcal{W} \times_{j\neq k} U_j^\top)$, and Proposition 2 with $\bar{m} = n_{-k}$ gives the bound. For $U_{-k}c$, reshape c as $\mathcal{C} \in \mathbb{R}^{\times_{j\neq k} r_j}$; then $U_{-k}c = \mathrm{vec}(\mathcal{C} \times_{j\neq k} U_j)$ by (F4), and contracting the modes in the order π, the l-th step costs $2 \cdot |\text{current}| \cdot n_{\pi_l} = 2\, n_{-k}\, r_{\pi_l} \prod_{l'>l} (r_{\pi_{l'}}/n_{\pi_{l'}}) \le 2n_{-k}r_{\max}$, and the largest intermediate has n_{-k} entries. □

A.4 Proof of Proposition 3 (Intertwining and Storage Compression)

Let $z = (\{U_k\}_{k=1}^N, \mathcal{G}, R_\Omega)$ with $D(z) = \mathcal{G} \times_1 U_1 \cdots \times_N U_N =: \mathcal{X}$ and $R_\Omega = P_\Omega(y) - P_\Omega(\mathcal{X})$, and write $z^+ = (\{U_k^+\}, \mathcal{G}^+, R_\Omega^+)$ for the output of one iteration (lines 2–8) of Algorithm 3.

Step 1: the operator input is sparse-plus-low-rank. The projected imputation input in equation 5 satisfies

$$\mathcal{M}^+ = P_\Omega(y) + P_{\Omega^\perp}(\mathcal{X}) = P_\Omega(y) - P_\Omega(\mathcal{X}) + \mathcal{X} = R_\Omega + D(z), \tag{13}$$

so by (F3) and linearity of unfolding, $M_{(k)}^+ = (R_\Omega)_{(k)} + U_k G_{(k)} U_{-k}^\top$. Consequently, for $w \in \mathbb{R}^{n_{-k}}$ and $v \in \mathbb{R}^{n_k}$,

$$M_{(k)}^+ w = (R_\Omega)_{(k)} w + U_k\big(G_{(k)}(U_{-k}^\top w)\big), \qquad (M_{(k)}^+)^\top v = (R_\Omega)_{(k)}^\top v + U_{-k}\big(G_{(k)}^\top (U_k^\top v)\big), \tag{14}$$

which is exactly the operator defined in line 3 of Algorithm 3. The sparse terms cost $2|\Omega|$ flops each (one multiply-add per stored entry), $U_{-k}^\top w$ and $U_{-k}(\cdot)$ cost at most $2(N-1)n_{-k}r_{\max}$ by Lemma 1, $G_{(k)}(\cdot)$ and $G_{(k)}^\top(\cdot)$ cost $2\bar{r}$, and $U_k(\cdot)$, $U_k^\top(\cdot)$ cost $2n_k r_k$. Hence one matvec pair costs $O(|\Omega| + n_k r_k + \bar{r} + N n_{-k} r_{\max})$, and no $n_k \times n_{-k}$ array is formed.

Step 2: the factor update yields the selected subspaces. Line 4 returns, by assumption, the exact singular triplets of $M_{(k)}^+$ indexed by $I^{(k)} := I_\Phi(M_{(k)}^+)$; the vectors $\tilde{u}_1, \ldots, \tilde{u}_{s_k}$ are orthonormal and $\operatorname{span} \mathcal{U}_k^+ := \operatorname{span}\{u_i(M_{(k)}^+) : i \in I^{(k)}\}$, which is unique by Proposition 1 (for Π_r and LMO_τ under the gap condition). Line 5 replaces this basis by the orthonormal factor of its QR decomposition, which spans the same subspace, so $\mathrm{range}(U_k^+) = \mathcal{U}_k^+$ and $U_k^+ (U_k^+)^\top = P_{\mathcal{U}_k^+}$ irrespective of the sign or basis convention of the QR routine. (If $I^{(k)} = \emptyset$, U_k^+ is the empty matrix, $P_{\mathcal{U}_k^+} = 0$, and both iterations produce $\mathcal{X}^+ = 0$.) Note that all N factors are extracted from the same tensor $\mathcal{M}^+$, matching equation 5.

Step 3: the core update is the projected input. Let $W_j := (U_j^+)^\top U_j \in \mathbb{R}^{r_j^+ \times r_j}$. By the same-mode rule in (F2), $\mathcal{G} \times_j W_j = \mathcal{G} \times_j U_j \times_j (U_j^+)^\top$, and by the commutation rule the N mode products can be interleaved, so

$$\mathcal{G} \times_{j=1}^N W_j = \big(\mathcal{G} \times_{j=1}^N U_j\big) \times_{j=1}^N (U_j^+)^\top = D(z) \times_{j=1}^N (U_j^+)^\top. \tag{15}$$

Line 7 therefore gives, by linearity of mode products and equation 13,

$$\mathcal{G}^+ = \big(R_\Omega + D(z)\big) \times_1 (U_1^+)^\top \cdots \times_N (U_N^+)^\top = \mathcal{M}^+ \times_1 (U_1^+)^\top \cdots \times_N (U_N^+)^\top. \tag{16}$$

Step 4: decoding. Using (F2) once more,

$$D(z^+) = \mathcal{G}^+ \times_{j=1}^N U_j^+ = \mathcal{M}^+ \times_{j=1}^N \big(U_j^+ (U_j^+)^\top\big) = \mathcal{M}^+ \times_{j=1}^N P_{\mathcal{U}_j^+} = \mathcal{T}_{\mathrm{proj}}(\mathcal{X}), \tag{17}$$

which is the intertwining relation $D(\tilde{\mathcal{T}}(z)) = \mathcal{T}_{\mathrm{proj}}(D(z))$. Line 8 sets $R_\Omega^+ = P_\Omega(y) - P_\Omega(D(z^+))$, so the invariant $R_\Omega = y - P_\Omega(D(z))$ is maintained, and the stopping test in line 9 is a function of $D(z^+)$, $D(z)$ and y only. The factors U_k^+ have orthonormal columns by construction.

Storage. The persistent state stores $\{U_k\}$ ($\sum_k n_k r_k$ words), $\mathcal{G}$ ($\bar{r}$ words), and R_Ω ($|\Omega|$ values plus $N|\Omega|$ index words in coordinate format; the index part is shared with the data (Ω, y) and can be stored once), giving equation 6. Evaluating $P_\Omega(D(z))$, needed in line 8, costs at most $2N\bar{r}$ flops per queried entry by contracting the core sequentially with the rows $U_1[i_1, :], \ldots, U_N[i_N, :]$ (the m-th contraction costs $2\bar{r}/\prod_{l<m} r_l \le 2\bar{r}$), i.e. $O(N|\Omega|\bar{r})$ in total, and never materializes $D(z)$. The ratio ρ in equation 6 tends to one exactly when $(\sum_k n_k r_k + \bar{r} + |\Omega|)/\bar{n} \to 0$; for $|\Omega| = p\bar{n}$ this requires $p \to 0$, and the sparse residual stops being advantageous when $(N+1)|\Omega| \ge \bar{n}$, i.e. $p \ge 1/(N+1)$ in words (the threshold p^* of Section 5 refines this by the index/value byte ratio). □

A.5 Proof of Theorem 1 (Exact Algebraic Equivalence)

We first isolate the two abstract facts that carry the argument.

Lemma 2 (Intertwining implies trajectory equality)**.** *Let $(\mathcal{Z}, D, \tilde{\mathcal{T}})$ satisfy $D \circ \tilde{\mathcal{T}} = \mathcal{T} \circ D$ on a subset $\mathcal{Z}_0 \subseteq \mathcal{Z}$ with $\tilde{\mathcal{T}}(\mathcal{Z}_0) \subseteq \mathcal{Z}_0$. If $z^0 \in \mathcal{Z}_0$, $D(z^0) = \mathcal{X}^0$, $z^{t+1} = \tilde{\mathcal{T}}(z^t)$ and $\mathcal{X}^{t+1} = \mathcal{T}(\mathcal{X}^t)$, then $D(z^t) = \mathcal{X}^t$ for all $t \ge 0$.*

Proof. Induction on t. The base case is the hypothesis. If $D(z^t) = \mathcal{X}^t$ with $z^t \in \mathcal{Z}_0$, then $D(z^{t+1}) = D(\tilde{\mathcal{T}}(z^t)) = \mathcal{T}(D(z^t)) = \mathcal{T}(\mathcal{X}^t) = \mathcal{X}^{t+1}$, and $z^{t+1} \in \mathcal{Z}_0$ by forward invariance. □

Lemma 3 (Compositionality)**.** *If $D \circ \tilde{\mathcal{T}}_1 = \mathcal{T}_1 \circ D$ and $D \circ \tilde{\mathcal{T}}_2 = \mathcal{T}_2 \circ D$, then $D \circ (\tilde{\mathcal{T}}_2 \circ \tilde{\mathcal{T}}_1) = (\mathcal{T}_2 \circ \mathcal{T}_1) \circ D$.*

Proof. $D \circ \tilde{\mathcal{T}}_2 \circ \tilde{\mathcal{T}}_1 = \mathcal{T}_2 \circ D \circ \tilde{\mathcal{T}}_1 = \mathcal{T}_2 \circ \mathcal{T}_1 \circ D$. □

Proof of Theorem 1. Fix $\mathcal{T} \in \mathfrak{T}$ with its stage decomposition $\mathcal{T} = \mathcal{T}_U \circ \mathcal{T}_L \circ \mathcal{T}_S$ and the corresponding compact state space $\mathcal{Z}$ and decoding D. We verify the hypothesis of Lemma 2 on the forward-invariant set $\mathcal{Z}_0 := \{z \in \mathcal{Z} : R_\Omega = y - P_\Omega(D(z))\}$ (for states that carry a residual; otherwise $\mathcal{Z}_0 := \mathcal{Z}$).

Stage (S). The reference stage evaluates Φ on the unfoldings $M_{(k)}$ of the input tensor $\mathcal{M} = \mathcal{M}(\mathcal{X}^t)$. The lightweight stage evaluates Algorithm 1 on the implicit operator equation 14 (or its factored analogue in Appendix C), which equals $M_{(k)}$ as a linear map by equation 13. By assumption the oracle returns the singular triplets indexed by $I_\Phi(M_{(k)})$, and by Proposition 1 these determine $\Phi(M_{(k)})$ uniquely: for $\mathcal{D}_\lambda$ unconditionally, for Π_r and LMO_τ under the gap condition, which is assumed for every matrix on which Φ is evaluated along the reference trajectory. Since the reference map itself is single-valued only under the same condition, both iterations compute the same $\Phi(M_{(k)})$ (or the same selected subspaces $\mathcal{U}_k$).

Stage (L). Every contraction with a Kronecker factor U_{-k} that the reference stage performs on an explicit unfolding is replaced by Algorithm 2 or Lemma 1; by Proposition 2 the outputs coincide entrywise for every contraction order.

Stage (U). For $\mathcal{T} = \mathcal{T}_{\mathrm{proj}}$ the compact update is Algorithm 3, and Proposition 3 gives $D(\tilde{\mathcal{T}}(z)) = \mathcal{T}_{\mathrm{proj}}(D(z))$ together with the invariance of $\mathcal{Z}_0$. For the remaining members of $\mathfrak{T}$ used in this paper, the corresponding intertwining relations are Propositions 4–7 in Appendix C.

Since each stage intertwines with its reference counterpart, Lemma 3 (applied twice) yields $D \circ \tilde{\mathcal{T}} = \mathcal{T} \circ D$ on $\mathcal{Z}_0$, and Lemma 2 gives $D(z^t) = \mathcal{X}^t$ for all $t \ge 0$. Finally, a stopping rule that is a function of $\{\mathcal{X}^s\}_{s \le t}$ and (Ω, y) evaluates to the same Boolean value on $\{D(z^s)\}_{s \le t}$, so both iterations stop at the same index; in Algorithm 3 the rule depends on $R_\Omega^{t+1} - R_\Omega^t = P_\Omega(\mathcal{X}^t - \mathcal{X}^{t+1})$ only. □

Remark 2. *Theorem 1 is a statement about exact arithmetic; the finite-precision discrepancy measured in Section 4 is the sum of rounding errors in the two implementations and of the Ritz residual of Algorithm 1, and is controlled as described in Remark 1. The gap conditions hold for the geometrically decaying spectra used in the experiments.*

A.6 Proof of Corollary 1 (Guarantee Inheritance and Compositionality)

(i) By Theorem 1, the decoded sequence $\{D(z^t)\}_{t\geq 0}$ and the reference sequence $\{\mathcal{X}^t\}_{t\geq 0}$ are the same sequence of tensors. Any guarantee of the reference estimator is a statement about this sequence and the data—for instance $f(\mathcal{X}^t) - f^\star \leq C\theta(t)$ for a convex objective f, a contraction $\|\mathcal{X}^{t+1} - \mathcal{X}^\star\|_F \leq (1-\mu)\|\mathcal{X}^t - \mathcal{X}^\star\|_F$ with μ independent of the condition number as established for ScaledGD (Tong et al., 2022) and PRGD (Zhang et al., 2025), or a minimax risk bound $\mathbb{E}\|\mathcal{X}^T - \mathcal{X}^\star\|_F^2 \leq \psi(n, r, |\Omega|)$ under a sampling model—and therefore holds verbatim with $\mathcal{X}^t$ replaced by $D(z^t)$. Guarantees about *running time* are of course not inherited; they are replaced by Theorem 2.

(ii) This is Lemma 3. Concretely, let $\mathcal{T}' := \mathcal{T} \circ \mathcal{S}$ where $\mathcal{S}$ is a sketching step (Strategy IV) or a preconditioning step (Strategy V) that maps sparse-plus-factored states to sparse-plus-factored states, e.g. the scaling by inverse Gram matrices in Appendix C.4. If $\mathcal{S}$ admits a lightweight realization $\tilde{\mathcal{S}}$ on the same $(\mathcal{Z}, D)$, then $\tilde{\mathcal{T}} \circ \tilde{\mathcal{S}}$ realizes $\mathcal{T}'$ exactly. Two clarifications are in order. First, the statement is about the *evaluation* of $\mathcal{T}'$: the trajectory of a sketched iteration differs from that of the unsketched one, and nothing here claims otherwise; what is claimed is that the lightweight evaluation of $\mathcal{T}'$ introduces no additional deviation. Second, the resources saved are different: sketching reduces the problem size and preconditioning the iteration count, whereas Algorithms 1–3 reduce the cost of executing a fixed iteration, so the two savings multiply. □

B Detailed Complexity Derivations and Arithmetic Counts

B.1 Term-by-Term FLOP Accounting of Theorem 2

We derive equation 8 for one iteration (lines 2–8) of Algorithm 3. Throughout, r_k bounds the mode-k rank of the state before (r_k^t) and after ($r_k^{t+1} = s_k$) the iteration, $r_{\max} = \max_k r_k$, $s = \max_k s_k \leq r_{\max}$, and $s \leq q \leq \min_k \min(n_k, n_{-k})$. We count a multiply-add pair as two flops and treat all $O(\cdot)$ constants as absolute.

Phase 1: Sparse residual evaluation (R_Ω). Each queried entry of $D(z)$ is obtained by contracting the core with one row of each factor; contracting mode m after modes $1, \ldots, m-1$ costs $2\bar{r}/\prod_{l<m} r_l \leq 2\bar{r}$ flops, so an entry costs at most $2N\bar{r}$ flops (Appendix A.4). Across $|\Omega|$ entries,

$$\mathrm{FLOP}_{\mathrm{residual}} \leq 2N|\Omega|\bar{r} = O(N|\Omega|\bar{r}). \tag{18}$$

Phase 2: Krylov bidiagonalization with sparse-plus-low-rank matvecs. For mode k, one matvec pair equation 14 costs, by Proposition 3 and Lemma 1,

$$c^{(k)} \leq 4|\Omega| + 4n_k r_k + 4\bar{r} + 4(N-1)n_{-k}r_{\max} = O\big(|\Omega| + n_k r_k + \bar{r} + Nn_{-k}r_{\max}\big). \tag{19}$$

Golub–Kahan–Lanczos bidiagonalization with q steps and full reorthogonalization performs q matvec pairs, $O(q^2(n_k + n_{-k}))$ flops for reorthogonalization, $O(q^3)$ for the SVD of the $q \times q$ bidiagonal matrix, and $O(qs_k(n_k + n_{-k})) \leq O(q^2(n_k + n_{-k}))$ for assembling the s_k Ritz vectors (Appendix A.2). Hence

$$\mathrm{FLOP}^{(k)}_{\mathrm{Krylov}} = O\big(q\,(|\Omega| + n_k r_k + \bar{r} + Nn_{-k}r_{\max}) + q^2(n_k + n_{-k}) + q^3\big). \tag{20}$$

Summing over $k = 1, \ldots, N$ and using $\bar{r} = r_k r_{-k} \leq r_{\max} n_{-k}$ (as $r_j \leq n_j$) and $q^3 \leq q^2 n_{-k}$,

$$\sum_{k=1}^{N} \mathrm{FLOP}^{(k)}_{\mathrm{Krylov}} = O\Big(qN|\Omega| + q\,(Nr_{\max} + q)\sum_{k=1}^{N}(n_k + n_{-k})\Big). \tag{21}$$

(With a block Krylov process of block size b and q/b block steps, the matvec term is unchanged and the reorthogonalization term becomes $O(q^2(n_k + n_{-k}))$ as well, so equation 21 covers both variants.)

Phase 3: Orthonormal factor retraction and core update. The QR factorization in line 5 costs $O(n_k s_k^2) \leq O(n_k r_k^2)$ per mode. In line 7, $R_\Omega \times_1 (U_1^{t+1})^\top \cdots \times_N (U_N^{t+1})^\top$ is a sparse contraction

over all N modes and costs $O(N|\Omega|\bar{r})$ by Proposition 2; the small matrices $W_j = (U_j^{t+1})^\top U_j^t$ cost $2n_j r_j^{t+1} r_j^t \le 2n_j r_j^2$; and $\mathcal{G}^t \times_1 W_1 \cdots \times_N W_N$ costs $\sum_j 2(\text{current size}) \cdot r_j^{t+1} \le 2\bar{r} \sum_j r_j$ because every intermediate has at most $\bar{r}$ entries. Line 8 is another Phase-1 evaluation. Altogether,

$$\mathrm{FLOP}_{\mathrm{update}} = O\Big(N|\Omega|\bar{r} + \sum_{k=1}^{N} n_k r_k^2 + \bar{r} \sum_{k=1}^{N} r_k\Big). \tag{22}$$

Combining Phases 1–3 gives equation 8:

$$\mathrm{FLOP}(\tilde{\mathcal{T}}) = O\Big(qN|\Omega| + q\,(Nr_{\max} + q) \sum_{k=1}^{N} (n_k + n_{-k}) + N|\Omega|\bar{r} + \sum_{k=1}^{N} n_k r_k^2 + \bar{r} \sum_{k=1}^{N} r_k\Big). \tag{23}$$

For $n_k = n$, $r_k = r$ and fixed N this is $O\big(q(q+r)n^{N-1} + (q+r^N)|\Omega|\big)$. The literal baseline forms the unfolding $X_{(k)} \in \mathbb{R}^{n_k \times n_{-k}}$ and computes its full SVD at cost $\Theta(n_k n_{-k} \min(n_k, n_{-k}))$ (Golub & Van Loan, 2013, Ch. 8), i.e. $\Theta(\bar{n} \sum_k \min(n_k, n_{-k}))$ per iteration, which is $\Theta(Nn^{N+1})$ for cubical tensors; forming U_{-k} and the product $X_{(k)}U_{-k}$ adds $\Theta(n_{-k}r_{-k} + \bar{n}r_{-k})$. Thus the lightweight iteration is one polynomial order lower in n; the empirical slope changes $\Delta \approx -1$ in Table 6 are consistent with this prediction.

B.2 Peak Memory Footprint Derivation of Equation (9)

The memory demand of $\tilde{\mathcal{T}}$ splits into the persistent state carried between iterations and transient scratch space:

1. **Factor storage**: $\{U_k \in \mathbb{R}^{n_k \times r_k}\}_{k=1}^N$ consume $\sum_{k=1}^N n_k r_k$ words.
2. **Core storage**: $\mathcal{G} \in \mathbb{R}^{r_1 \times \cdots \times r_N}$ consumes $\bar{r}$ words.
3. **Sparse residual storage**: values and N coordinates per observed entry, $(N+1)|\Omega|$ words (the coordinates coincide with those of the data and need not be duplicated).
4. **Krylov scratch**: the q left and q right Lanczos vectors of lengths n_k and n_{-k}, i.e. $q(n_k + n_{-k})$ words, plus $O(q^2)$ for the bidiagonal matrix; they are released after the mode-k extraction.
5. **Contraction scratch**: Lemma 1 needs $O(n_{-k})$ words, and the sparse contraction of line 7 needs $O(\bar{r})$ words.

Items 1–3 give $\mathrm{MEM}_{\mathrm{state}}(\tilde{\mathcal{T}}) = \sum_k n_k r_k + \bar{r} + (N+1)|\Omega|$, and items 4–5 give the peak

$$\mathrm{MEM}_{\max}(\tilde{\mathcal{T}}) = \mathrm{MEM}_{\mathrm{state}}(\tilde{\mathcal{T}}) + O\Big(q \max_k (n_k + n_{-k})\Big), \tag{24}$$

which is equation 9. Two comments are in order. First, the Krylov scratch $q \max_k n_{-k} = q\bar{n}/\min_k n_k$ is not negligible relative to $\bar{n}$ unless $q \ll \min_k n_k$; it is transient and is what separates $\mathrm{MEM}_{\max}$ from $\mathrm{MEM}_{\mathrm{state}}$. The memory figures reported in Section 4 and Appendix F are persistent-state figures. Second, the scratch can be reduced to $O(qn_k + sn_{-k})$ by keeping only the short Lanczos vectors and recovering the s required right vectors as $\tilde{v}_i = M^\top \tilde{u}_i / \tilde{\sigma}_i$, at the price of s extra transposed matvecs. The literal baseline stores the ambient iterate and its unfolding ($\Theta(\bar{n})$), the explicit Kronecker factor U_{-k} ($n_{-k}r_{-k}$ words), and the dense SVD workspace ($\Theta(n_k n_{-k})$).

B.3 Systematic Comparison across Computational Paradigms

Table 4 compares naive ambient execution, randomized sketching, Riemannian optimization, and the proposed lightweight method (cubical order-N tensors, $n_k = n$, $r_k = r$, d the sketch size).

Table 4: Analytical comparison of arithmetic, spatial, and optimization properties across computing paradigms.

Paradigm	FLOP / Iteration	Peak Memory	Algebraic Loss	κ-Sensitivity
Naive Ambient	$\Theta(Nn^{N+1})$	$\Theta(n^N)$	None (Exact)	Unchanged
Randomized Sketching	$O(\lvert\Omega\rvert + Nnd^2)$	$O(Nnd + d^N)$	Approximate	Unchanged
Riemannian / ScaledGD	$O(\lvert\Omega\rvert r^N + Nnr^2)$	$O(n^N)$ or $O(Nnr + r^N)$	None (Exact)	Invariant
Proposed Method	$O(q(q+r)n^{N-1} + (q+r^N)\lvert\Omega\rvert)$	state $O(Nnr + r^N + \lvert\Omega\rvert)$; peak $+O(qn^{N-1})$	None (Exact)	Invariant

C Concrete Algorithmic Instantiations

We specify, for each estimator of Table 2, the compact state $\mathcal{Z}$, the decoding D, the reference iteration $\mathcal{T}$, and the intertwining relation equation 1, which is what Theorem 1 requires. The projected imputation iteration $\mathcal{T}_{\mathrm{proj}}$ of equation 5 is covered by Proposition 3. Two further state formats are needed, because not every iterate in $\mathfrak{T}$ has a small Tucker core.

Sparse-plus-mode-factored states. Let $\mathcal{Z}_{\mathrm{mf}}$ be the set of tuples $z = \big(\mathcal{S}, \{(A_k, B_k)\}_{k=1}^N\big)$ with $\mathcal{S}$ supported on Ω, $A_k \in \mathbb{R}^{n_k \times s_k}$, $B_k \in \mathbb{R}^{n_{-k} \times s_k}$ ($s_k \geq 0$), decoded by

$$D(z) := \mathcal{S} + \sum_{k=1}^{N} \mathrm{fold}_k\left(A_k B_k^\top\right), \tag{25}$$

where fold_k is the inverse of the mode-k unfolding. Its storage is $(N+1)|\Omega| + \sum_k s_k(n_k + n_{-k})$ words.

Proposition 4 (Mode-wise proximal steps on factored states)**.** *Let $z \in \mathcal{Z}_{\mathrm{mf}}$ and $\mathcal{M} := D(z)$.*

(a) For every k, one matvec pair with $M_{(k)}$ and $M_{(k)}^\top$ costs $O\big(|\Omega| + \sum_{j=1}^N s_j(n_{-j} + n_{-k})\big)$ flops without forming $M_{(k)}$.

(b) $\mathrm{fold}_k(\mathcal{D}_\lambda(M_{(k)})) = \mathrm{fold}_k(AB^\top)$ with $A := U_I(\Sigma_I - \lambda I)$, $B := V_I$, where (U_I, Σ_I, V_I) are the singular triplets of $M_{(k)}$ indexed by $I_\lambda(M_{(k)})$; likewise $\mathrm{fold}_k(\Pi_r(M_{(k)}))$ and $\mathrm{fold}_k(\mathrm{LMO}_\tau(M_{(k)}))$ are of the form $\mathrm{fold}_k(AB^\top)$ with $s_k = r$ and $s_k = 1$, respectively.

(c) $\mathcal{Z}_{\mathrm{mf}}$ is closed under linear combinations (by concatenating the factors of equal modes and adding the sparse parts), under P_Ω and $P_{\Omega^\perp}$ (by adding a sparse correction), and under the mode-wise primitives in (b). Consequently, any iteration built from these operations—the proximal-gradient step for sum-of-mode-wise nuclear norms, the HaLRTC/ADMM step of Liu et al. (2012), whose primal, auxiliary, and multiplier variables are all of the form equation 25, and the proximal step of the subspace norm (Zheng & Tomioka, 2015), in which B_k is additionally constrained to a fixed Kronecker subspace—maps $\mathcal{Z}_{\mathrm{mf}}$ to itself, and its lightweight realization satisfies $D \circ \tilde{\mathcal{T}} = \mathcal{T} \circ D$ with exact spectral extraction.

Proof. (a) $M_{(k)}w = S_{(k)}w + \sum_j \mathrm{fold}_j(A_j B_j^\top)_{(k)}w$. The sparse term costs $2|\Omega|$. For $j = k$ the term is $A_k(B_k^\top w)$, costing $2s_k(n_{-k} + n_k)$. For $j \neq k$, write $A_j B_j^\top = \sum_{\ell=1}^{s_j} a_\ell b_\ell^\top$; the tensor $\mathrm{fold}_j(a_\ell b_\ell^\top)$ has entries $a_\ell[i_j]\,\mathcal{B}_\ell[(i_l)_{l\neq j}]$ where $\mathcal{B}_\ell$ is b_ℓ reshaped over the modes $l \neq j$. Reshaping w over the modes $l \neq k$, its product with the mode-k unfolding is $\sum_{i_j} a_\ell[i_j] \sum_{(i_l)_{l\neq j,k}} \mathcal{B}_\ell[i_k, (i_l)_{l\neq j,k}]\,\mathcal{W}[i_j, (i_l)_{l\neq j,k}]$: first contract $\mathcal{W}$ with a_ℓ along mode j ($2n_{-k}$ flops), then contract the result with $\mathcal{B}_\ell$ along all modes $l \neq j, k$ ($2n_{-j}$ flops). Summing over ℓ and j gives the bound; the transposed product is symmetric. (b) is Proposition 1. (c) Linear combinations and sparse corrections are immediate from equation 25; the mode-wise primitives are (b); each

reference operation is thus reproduced on the decoded tensors, i.e. intertwining holds operation by operation, and Lemma 3 composes them. □

Remark 3. *(i) For the HaLRTC/ADMM update, the multiplier $\mathcal{Y}_k^{t+1} = \mathcal{Y}_k^t - \rho(\mathcal{M}_k^{t+1} - \mathcal{X}^{t+1})$ is a running sum of factored terms, so its factor count grows with t unless the multiplier is stored in ambient form; either choice preserves exactness and only changes the memory profile. (ii) The matvec cost in (a) is $\Theta(\sum_j s_j n_{-j})$, so for spectra that are not sparse ($s_j \approx n_j$) it is of the same order as a dense unfolding, which is the regime in which E1 shows no runtime gain in Table 3; the memory reduction from equation 25 is retained.*

C.1 Subspace Norm and Sum-of-Nuclear-Norms Proximal Gradient (E0, E1)

The overlapped (sum-of-nuclear-norms) regularizer (Liu et al., 2012) penalizes $\sum_{k=1}^N \lambda_k \|X_{(k)}\|_*$, and the subspace norm (Zheng & Tomioka, 2015) penalizes mode-wise nuclear norms of components constrained to fixed Kronecker subspaces; both are handled by Proposition 4. Algorithm 4 states the generic lightweight proximal-gradient iteration with step size 1 on $\mathcal{Z}_{\mathrm{mf}}$; the ADMM variant differs only in the bookkeeping of the auxiliary and multiplier variables, all of which stay in $\mathcal{Z}_{\mathrm{mf}}$ by Proposition 4(c).

Algorithm 4 Lightweight Mode-Wise Proximal Estimation on Sparse-plus-Mode-Factored States

Require: Observed data (Ω, y), thresholds $\{\lambda_k\}_{k=1}^N$, initial state $z^0 \in \mathcal{Z}_{\mathrm{mf}}$, max iterations T.
Ensure: State $z^T \in \mathcal{Z}_{\mathrm{mf}}$ with $D(z^T) = \mathcal{X}^T$.

```
for t = 0, 1, ..., T − 1 do
    Sparse residual R_Ω^t ← P_Ω(y) − P_Ω(D(z^t)) (each entry of D(z^t) on Ω costs O(Σ_j s_j)).
    Gradient-step input M^t ← D(z^t) + R_Ω^t, kept in factored form (sparse part S^t + R_Ω^t).
    for each mode k = 1, ..., N do
        Implicit matvec M_k(w) := M_(k)^t w evaluated as in Proposition 4(a).
        {(σ_i, u_i, v_i)}_{i∈I_{λ_k}} ← Algorithm 1(M_k, D_{λ_k}).
        A_k^{t+1} ← [u_i(σ_i − λ_k)]_{i∈I_{λ_k}},   B_k^{t+1} ← [v_i]_{i∈I_{λ_k}}.
    end for
    Assemble z^{t+1} from {(A_k^{t+1}, B_k^{t+1})} according to the estimator's combination rule (e.g. averaging over modes and re-imposing P_Ω(·) = y through the sparse part).
    Check convergence ‖R_Ω^{t+1} − R_Ω^t‖_2 / ‖y‖_2 ≤ ε_tol.
end for
```

C.2 Frank–Wolfe with Greedy Atom Pursuit (E2)

For nuclear-norm-constrained completion $\min_{\|\mathcal{X}\|_* \le \tau} \frac{1}{2}\|P_\Omega(\mathcal{X}) - y\|_F^2$ (with $\|\cdot\|_*$ a mode-wise or averaged nuclear norm), Frank–Wolfe iterates $\mathcal{X}^{t+1} = (1-\eta_t)\mathcal{X}^t + \eta_t \mathcal{A}^t$ with $\mathcal{A}^t := \mathrm{fold}_{k_t}(\mathrm{LMO}_\tau(\nabla f(\mathcal{X}^t)_{(k_t)}))$ for a selected mode k_t and step size $\eta_t \in [0,1]$ (Guo et al., 2017b). The gradient $\nabla f(\mathcal{X}^t) = P_\Omega(\mathcal{X}^t) - y = -R_\Omega^t$ is supported on Ω.

Proposition 5 (Atom-pool state)**.** *Let $\mathcal{Z}_{\mathrm{fw}}$ consist of atom pools $z = (\{(c_a, k_a, u_a, v_a)\}_{a=1}^m, R_\Omega)$ with $D(z) := \sum_a c_a \,\mathrm{fold}_{k_a}(u_a v_a^\top)$ and $R_\Omega = P_\Omega(y) - P_\Omega(D(z))$. The lightweight step (i) computes the LMO by Algorithm 1 on the sparse operator $(R_\Omega)_{(k_t)}$, whose matvec costs $2|\Omega|$; (ii) rescales all coefficients by $(1-\eta_t)$ and appends the atom $(\eta_t\tau, k_t, u_1, v_1)$, where (u_1, v_1) is the top singular pair of $(R_\Omega^t)_{(k_t)}$, so that $\mathcal{A}^t = \tau\,\mathrm{fold}_{k_t}(u_1 v_1^\top)$; (iii) updates $R_\Omega \leftarrow (1-\eta_t)R_\Omega + \eta_t\big(P_\Omega(y) - P_\Omega(\mathcal{A}^t)\big)$ in $O(|\Omega|)$. It satisfies $D \circ \tilde{\mathcal{T}} = \mathcal{T} \circ D$ under $\sigma_1 > \sigma_2$ for $(R_\Omega^t)_{(k_t)}$, and $\mathcal{Z}_{\mathrm{fw}}$ is closed under the fully-corrective or line-search variants, which only re-weight the pool.*

Proof. (i) Since $R_\Omega^t = P_\Omega(y) - P_\Omega(\mathcal{X}^t)$, $\nabla f(\mathcal{X}^t)_{(k_t)} = -(R_\Omega^t)_{(k_t)}$, so the LMO input is the sparse unfolding and, by Proposition 1(ii'), LMO_τ is $-\tau$ times the top singular pair of $-(R_\Omega^t)_{(k_t)}$, i.e. $\tau u_1 v_1^\top$ for the top pair (u_1, v_1) of $(R_\Omega^t)_{(k_t)}$ (the top pair of $-R$ is $(u_1, -v_1)$); it is unique under the gap. (ii) D is linear in the coefficients, so $D(z^{t+1}) = (1-\eta_t)D(z^t) + \eta_t\tau\,\mathrm{fold}_{k_t}(u_1 v_1^\top) = \mathcal{X}^{t+1}$. (iii) $P_\Omega(y) - P_\Omega(\mathcal{X}^{t+1}) = (1-\eta_t)(P_\Omega(y) - P_\Omega(\mathcal{X}^t)) + \eta_t(P_\Omega(y) - P_\Omega(\mathcal{A}^t))$, and $P_\Omega(\mathcal{A}^t)$ needs one product $u_1[i_{k_t}]v_1[\cdot]$ per observed entry. □

C.3 Higher-Order Orthogonal Iteration (HOOI, E3)

For a fully observed tensor $\mathcal{Y}$, HOOI (De Lathauwer et al., 2000a; Kolda & Bader, 2009) iterates, for $k = 1, \dots, N$,

$$\mathcal{W}_k^t := \mathcal{Y} \times_{j<k} (U_j^{t+1})^\top \times_{j>k} (U_j^t)^\top, \qquad U_k^{t+1} := \text{leading } r_k \text{ left singular vectors of } (W_k^t)_{(k)}, \tag{26}$$

and sets $\mathcal{G}^{t+1} := \mathcal{Y} \times_1 (U_1^{t+1})^\top \cdots \times_N (U_N^{t+1})^\top$. The data $\mathcal{Y}$ is an input, not part of the state.

Proposition 6 (Factor state for HOOI). *Let $z = (\{U_k\}, \mathcal{G})$ with $D(z) = \mathcal{G} \times_1 U_1 \cdots \times_N U_N$. The lightweight HOOI step evaluates $(W_k^t)_{(k)} = Y_{(k)}\widehat{U}_{-k}$, with $\widehat{U}_{-k}$ the reverse-order Kronecker product of the current factors, by Algorithm 2 and extracts the leading r_k left singular vectors by Algorithm 1. Under $\sigma_{r_k} > \sigma_{r_k+1}$ for each $(W_k^t)_{(k)}$, the decoded iterates coincide with those of equation 26, i.e. $D(z^{t+1}) = \mathcal{Y} \times_1 P_{\mathcal{U}_1^{t+1}} \cdots \times_N P_{\mathcal{U}_N^{t+1}}$ for both implementations, and the largest intermediate of the contraction chain has $\bar{n}\, r_{\pi_1}/n_{\pi_1}$ entries, whereas the literal implementation allocates $\widehat{U}_{-k} \in \mathbb{R}^{n_{-k} \times r_{-k}}$ and the product $Y_{(k)}\widehat{U}_{-k}$ through an $n_k \times n_{-k}$ operand.*

Proof. By Proposition 2, $(\mathcal{Y} \times_{j\neq k} \widehat{U}_j^\top)_{(k)} = Y_{(k)}\widehat{U}_{-k}$ for any contraction order, so both implementations feed the same matrix to the spectral step; under the gap its leading r_k-dimensional left singular subspace $\mathcal{U}_k^{t+1}$ is unique (Proposition 1(ii)). If a factor is replaced by another orthonormal basis U_jQ_j of the same subspace, $(W_k^t)_{(k)}$ changes by right-multiplication with an orthogonal Kronecker matrix, which leaves its left singular subspaces and singular values unchanged; hence the sequence of subspaces $\{\mathcal{U}_k^t\}$ is basis-independent, and so is $D(z^{t+1}) = \mathcal{Y} \times_j (U_j^{t+1}(U_j^{t+1})^\top) = \mathcal{Y} \times_j P_{\mathcal{U}_j^{t+1}}$ by (F2). The intermediate sizes follow from Proposition 2: after contracting mode π_l the intermediate has $\bar{n} \prod_{l' \le l} (r_{\pi_{l'}}/n_{\pi_{l'}})$ entries, which is largest for $l = 1$ and is minimized by contracting first the mode with the smallest r_j/n_j, as Algorithm 2 does. □

C.4 Preconditioned Riemannian Gradient Descent (PRGD) and ScaledGD (E4)

ScaledGD (Tong et al., 2022)—and, up to the retraction, PRGD (Zhang et al., 2025)—optimizes directly over factors $z = (\{U_k\}, \mathcal{G})$ with $D(z) = \mathcal{G} \times_1 U_1 \cdots \times_N U_N$ and, for $f(\mathcal{X}) = \frac{1}{2p}\|P_\Omega(\mathcal{X}) - y\|_F^2$, performs updates of the form

$$U_k^+ = U_k - \eta \left(\nabla f(\mathcal{X})\right)_{(k)} \breve{U}_k \left(\breve{U}_k^\top \breve{U}_k\right)^{-1}, \qquad \mathcal{G}^+ = \mathcal{G} - \eta\, \nabla f(\mathcal{X}) \times_{j=1}^N \left((U_j^\top U_j)^{-1} U_j^\top\right), \tag{27}$$

where $\mathcal{X} = D(z)$, $\breve{U}_k := U_{-k} G_{(k)}^\top \in \mathbb{R}^{n_{-k} \times r_k}$ (so that $X_{(k)} = U_k \breve{U}_k^\top$ by (F3)), and the factors need not be orthonormal. Here the reference iteration is already defined on the factor state; the literal implementation materializes $\mathcal{X}$, $\breve{U}_k$ and U_{-k}, whereas the lightweight one does not.

Proposition 7 (Kronecker-free ScaledGD). *With $\nabla f(\mathcal{X}) = \frac{1}{p}(P_\Omega(\mathcal{X}) - y) = -\frac{1}{p} R_\Omega$ supported on Ω, the quantities in equation 27 satisfy*

$$\left(\nabla f(\mathcal{X})\right)_{(k)} \breve{U}_k = \left(\nabla f(\mathcal{X}) \times_{j\neq k} U_j^\top\right)_{(k)} G_{(k)}^\top, \qquad \breve{U}_k^\top \breve{U}_k = \left(\mathcal{G} \times_{j\neq k} (U_j^\top U_j)\right)_{(k)} G_{(k)}^\top, \tag{28}$$

and $\nabla f(\mathcal{X}) \times_j (U_j^\top U_j)^{-1} U_j^\top$ is a sparse contraction. Evaluating them costs $O(N|\Omega| r_{-k} + n_k r_k r_{-k})$, $O(\sum_j n_j r_j^2 + \bar{r} \sum_j r_j + r_k^2 r_{-k})$, and $O(N|\Omega|\bar{r} + \sum_j n_j r_j^2 + \sum_j r_j^3)$, respectively, with R_Ω obtained in $O(N|\Omega|\bar{r})$; the resulting factor iterates are identical to those of the literal implementation, so $\tilde{\mathcal{T}}$ intertwines with equation 27 trivially (the state is the same object, only the evaluation differs).

Proof. By (F3) with mode-k factor I_{n_k}, $(\nabla f \times_{j\neq k} U_j^\top)_{(k)} = (\nabla f)_{(k)} U_{-k}$, and multiplying by $G_{(k)}^\top$ gives the first identity. For the second, $\breve{U}_k^\top \breve{U}_k = G_{(k)} U_{-k}^\top U_{-k} G_{(k)}^\top = G_{(k)} \big(\bigotimes_{j\neq k} U_j^\top U_j\big) G_{(k)}^\top$ by the mixed-product rule, and $G_{(k)} W_{-k}^\top = (\mathcal{G} \times_{j\neq k} W_j)_{(k)}$ with $W_j := U_j^\top U_j = W_j^\top$ by (F3). The costs follow from Proposition 2 (sparse contraction over $N-1$ or N modes), the Gram computations $U_j^\top U_j$ ($2n_j r_j^2$), the $r_j \times r_j$ inversions, and dense products of the stated sizes. Both implementations compute the same matrices equation 27 in exact arithmetic, hence the same factor iterates and the same $D(z^t)$. The inherited condition-number-independent rate of Tong et al. (2022) is then an instance of Corollary 1(i). □

D Implementation Details and Numerical Stability

D.1 Strided Contraction and Buffer Elimination

Matricization operations $X_{(k)}$ traditionally execute physical memory permutation ('np.transpose' followed by 'np.reshape'), allocating a duplicate array of size $\bar{n}$. Our implementation bypasses array reshaping entirely by evaluating contractions directly via strided indexing:

$$\text{stride}(k) = \prod_{j=1}^{k-1} n_j, \qquad \text{offset}(i_1, \dots, i_N) = \sum_{k=1}^{N} (i_k - 1)\,\text{stride}(k). \tag{29}$$

Memory buffers are pre-allocated once during initialization with size $\max_k(n_k r_k)$, achieving zero dynamic heap allocation during iteration loops.

D.2 Rank-Preserving QR Correction

During factor accumulation, roundoff error under finite-precision arithmetic gradually degrades column orthogonality $U_k^\top U_k \approx I_{r_k}$. We employ an economic Modified Gram-Schmidt (MGS) or Householder QR factorization every $\tau_{\text{orth}} = 10$ iterations:

$$U_k = Q_k R_k, \qquad \mathcal{G} \leftarrow \mathcal{G} \times_k R_k, \qquad U_k \leftarrow Q_k. \tag{30}$$

Because $R_k \in \mathbb{R}^{r_k \times r_k}$, updating the core absorbs the upper-triangular factor with cost $O(r_k \bar{r})$, rigorously maintaining orthogonality without altering the decoded tensor $D(z)$.

E Experimental Protocols, Datasets, and Extended Results

E.1 Compute Infrastructure and Experimental Environment

All benchmarks are conducted on a unified server workstation equipped with an AMD EPYC 7763 64-Core Processor (2.45 GHz, 256 MB L3 Cache), 512 GB DDR4-3200 ECC Registered RAM, and an NVIDIA A100 Tensor Core GPU (80 GB HBM2e, 2.0 TB/s memory bandwidth). Software environment comprises Ubuntu 22.04 LTS, Python 3.10.12, PyTorch 2.2.1 (CUDA 12.1), CuPy 12.3.0, and Intel oneAPI MKL 2024.0. Peak memory is profiled at hardware level via 'cudaMemGetInfo' and Linux 'getrusage' (maximum resident set size 'ru_maxrss').

E.2 Benchmark Dataset Specifications

Synthetic Tensor Ensembles. Synthetic order-3 and order-4 tensors are generated according to the low Tucker rank ground truth $\mathcal{X}^\star = \mathcal{G}^\star \times_1 U_1^\star \cdots \times_N U_N^\star$. Entries of $\mathcal{G}^\star$ are sampled from $\mathcal{N}(0,1)$, and factor matrices $U_k^\star \in \mathbb{R}^{n_k \times r_k}$ are generated via Haar-distributed random orthogonalization. Singular values are engineered to obey power-law decay $\sigma_i^{(k)} = i^{-\alpha}$ with decay parameter $\alpha \in \{0.5, 1.0, 2.0\}$, tuning condition number $\kappa \in [10^1, 10^5]$. The observation set Ω is sampled uniformly at random with sample ratio $p = |\Omega|/\bar{n} \in \{0.01, 0.05, 0.10, 0.20\}$.

E.3 Hyperparameter Grid and Convergence Criteria

Across all algorithms, maximum iterations are set to $T_{\max} = 500$, with relative stopping tolerance $\|R_\Omega^{t+1} - R_\Omega^t\|_F / \|y\|_F \le 10^{-6}$. For Krylov subroutines, maximum subspace dimension is bounded by $q = 20$, with residual tolerance $\epsilon_{\text{krylov}} = 10^{-7}$. Regularization parameters λ_k for Subspace Norm and SNN are tuned across the logarithmic grid $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1.0\} \times \sqrt{|\Omega|}$. All experiments are repeated over 10 independent random seeds, reporting empirical means and standard error bars.

E.4 Ablation Study Protocol for the Algorithmic Components

To isolate the individual contribution of each algorithmic component, our ablation protocol evaluates four configuration stages:

1. **Baseline (Full Materialization)**: Naive execution where unfolded ambient arrays $X_{(k)}$ and Kronecker operators U_{-k} are explicitly materialized.
2. **Stage I (+ Algorithm 1)**: Partial Krylov component extraction (I_λ, I_r) applied to materialized matrices.
3. **Stage II (+ Algorithm 2)**: Mode-wise implicit contraction eliminating Kronecker operator U_{-k}, with ambient iterate retained.
4. **Stage III (+ Algorithm 3, Full Proposed Method)**: Fully compact state z with sparse-plus-low-rank matvec, eliminating both ambient tensors and operators.

F Detailed Tabulations of Empirical Benchmarks

Tables 5–9 compile the complete numerical tabulations corresponding to the empirical evaluations reported in Section 4.2, Figures 1, 2, and Tables 6, 3.

F.1 Trajectory Exactness Profiles across Estimators

Table 5 summarizes the maximum relative iterate discrepancy $\max_t \|D(z^t) - \mathcal{X}^t\|_F / \|\mathcal{X}^t\|_F$ and objective discrepancy across all five estimators under tight (10^{-14}) and default (10^{-10}) Krylov tolerances.

Table 5: Empirical trajectory exactness and objective gaps under lockstep execution ($T = 20$, $n = 30$, $r = 5$, $p = 0.3$).

Estimator	Config	Tol Mode	κ	$\max_t \mathrm{err}_{\mathrm{exact}}(t)$	$\max_t \lvert\Delta\mathcal{F}\rvert$	Status
Subspace Norm (E0)	All	Tight (10^{-14})	1	3.63×10^{-14}	9.38×10^{-14}	Pass
Subspace Norm (E0)	All	Default (10^{-10})	1	3.62×10^{-14}	9.33×10^{-14}	Pass
Subspace Norm (E0)	All	Tight (10^{-14})	100	3.25×10^{-14}	5.34×10^{-14}	Pass
Subspace Norm (E0)	All	Default (10^{-10})	100	2.05×10^{-13}	5.77×10^{-14}	Pass
SNN / Latent (E1)	+Alg 1+Alg 3	Tight (10^{-14})	1	7.65×10^{-15}	2.71×10^{-15}	Pass
SNN / Latent (E1)	+Alg 1+Alg 3	Default (10^{-10})	1	6.86×10^{-15}	2.71×10^{-15}	Pass
SNN / Latent (E1)	+Alg 1+Alg 3	Tight (10^{-14})	100	9.52×10^{-15}	1.23×10^{-15}	Pass
SNN / Latent (E1)	+Alg 1+Alg 3	Default (10^{-10})	100	9.04×10^{-15}	1.23×10^{-15}	Pass
Frank–Wolfe (E2)	+Alg 1+Alg 3	Tight (10^{-14})	1	6.63×10^{-14}	4.94×10^{-14}	Pass
Frank–Wolfe (E2)	+Alg 1+Alg 3	Default (10^{-10})	1	2.35×10^{-10}	1.49×10^{-10}	Fail ($> 10^{-10}$)
Frank–Wolfe (E2)	+Alg 1+Alg 3	Tight (10^{-14})	100	1.63×10^{-14}	2.64×10^{-14}	Pass
Frank–Wolfe (E2)	+Alg 1+Alg 3	Default (10^{-10})	100	5.35×10^{-11}	4.02×10^{-11}	Pass
HOOI (E3)	All	Tight (10^{-14})	1	2.20×10^{-15}	8.53×10^{-14}	Pass
HOOI (E3)	All	Default (10^{-10})	1	2.20×10^{-15}	8.53×10^{-14}	Pass
HOOI (E3)	All	Tight (10^{-14})	100	2.87×10^{-15}	2.63×10^{-13}	Pass
HOOI (E3)	All	Default (10^{-10})	100	2.87×10^{-15}	2.63×10^{-13}	Pass
ScaledGD (E4)	All	Tight (10^{-14})	1	1.47×10^{-15}	5.78×10^{-16}	Pass
ScaledGD (E4)	All	Default (10^{-10})	1	1.52×10^{-15}	5.59×10^{-15}	Pass
ScaledGD (E4)	All	Tight (10^{-14})	100	1.55×10^{-15}	6.57×10^{-15}	Pass
ScaledGD (E4)	All	Default (10^{-10})	100	1.95×10^{-15}	1.24×10^{-14}	Pass

F.2 Empirical Scaling Slopes across Order-3 Tensors

Table 6 reports per-iteration runtimes and the log-log scaling slopes fitted across the order-3 sweeps ($n \in \{40, 60, 80, 120\}$); the terminal relative error matches identically between Raw and All across all dimensions.

Table 6: Empirical scaling profiles on order-3 tensors ($n \in \{40, 60, 80, 120\}$, $r = 5$, $p = 0.05$). Slopes reflect least-squares fits on $\log(\text{runtime})$ versus $\log(n)$. Terminal relative error rel_err_final matches identically between Raw and All up to numerical precision across all dimensions.

Estimator	**Runtime at $n = 40$ (s)**		**Runtime at $n = 120$ (s)**		**Empirical Slope**		**Slope Change**
	Raw	**All**	**Raw**	**All**	**Raw**	**All**	**(Δ)**
E0	0.0101	0.0198	0.2289	0.1539	2.829	1.858	-0.971
E1	0.0241	0.0429	1.1991	1.7835	3.618	3.414	-0.204
E2	0.0113	0.0190	0.5412	0.3220	3.567	2.584	-0.983
E3	0.0045	0.0060	0.0628	0.0234	2.417	1.246	-1.171
E4	0.0016	0.0022	0.0541	0.0314	3.215	2.401	-0.814

F.3 Ladder and Component Ablation Profiles

Table 7 presents the complete ablation profile across estimators at dimensions $n = 50$ and $n = 100$.

Table 7: Ablation benchmark profiles reporting per-iteration runtime and persistent state memory ($r = 5$, $p = 0.05$, $\kappa = 1$, $T = 8$).

Estimator	**Config**	**Time** ($n = 50$)	**Mem** ($n = 50$)	**Time** ($n = 100$)	**Mem** ($n = 100$)
Subspace Norm (E0)	Raw	0.0155 s	9.09 MB	0.1369 s	66.81 MB
	+Alg 1	0.0284 s	9.09 MB	0.1472 s	66.81 MB
	+Alg 1+Alg 2	0.0268 s	7.66 MB	0.1387 s	61.09 MB
	+Alg 1+Alg 3	0.0260 s	3.37 MB	0.1117 s	21.04 MB
	+Alg 2+Alg 3	0.0114 s	1.94 MB	0.0874 s	15.32 MB
	All (+Alg 1+Alg 2+Alg 3)	0.0251 s	1.94 MB	0.0984 s	15.32 MB
HOOI (E3)	Raw	0.0059 s	0.96 MB	0.0384 s	7.64 MB
	+Alg 1	0.0084 s	0.96 MB	0.0430 s	7.64 MB
	+Alg 1+Alg 2	0.0071 s	0.96 MB	0.0349 s	7.64 MB
	+Alg 1+Alg 3	0.0074 s	0.0067 MB	0.0292 s	0.0124 MB
	+Alg 2+Alg 3	0.0038 s	0.0067 MB	0.0124 s	0.0124 MB
	All	0.0063 s	0.0067 MB	0.0161 s	0.0124 MB
ScaledGD (E4)	Raw	0.0026 s	1.91 MB	0.0336 s	15.27 MB
	+Alg 1+Alg 3	0.0042 s	0.078 MB	0.0294 s	0.584 MB
	+Alg 2+Alg 3	0.0041 s	0.078 MB	0.0257 s	0.584 MB
	All	0.0041 s	0.078 MB	0.0258 s	0.584 MB
Frank–Wolfe (E2)	Raw	0.0185 s	2.20 MB	0.2392 s	16.41 MB
	+Alg 1	0.0402 s	2.20 MB	0.3286 s	16.41 MB
	+Alg 3	0.0184 s	0.36 MB	0.2306 s	1.73 MB
	+Alg 1+Alg 3 (All)	0.0316 s	0.36 MB	0.2347 s	1.73 MB
SNN / Latent (E1)	Raw	0.0391 s	6.68 MB	0.5414 s	53.41 MB
	+Alg 1	0.0666 s	6.68 MB	0.5625 s	53.41 MB
	+Alg 3	0.3034 s	0.80 MB	7.8781 s	9.68 MB
	+Alg 1+Alg 3 (All)	0.0882 s	0.80 MB	0.9323 s	9.68 MB

F.4 Micro-Benchmark Primitive Measurements

Table 8 presents the measurements collected across the three isolated micro-benchmarks.

Table 8: Micro-benchmark measurements evaluating individual computational primitives.

Primitive	Order	n	Config. / Parameter	Full / Ambient	Lazy / Compact	Ratio / Speedup
SVT (Alg 1, Dense)	2	50	$s = 1$	0.0072 s	0.0232 s	0.311×
SVT (Alg 1, Dense)	2	100	$s = 5$	0.0714 s	0.3297 s	0.217×
SVT (Alg 1, Operator)	2	50	$s = 1$	0.0682 s	0.0169 s	4.031×
SVT (Alg 1, Operator)	2	100	$s = 1$	1.5523 s	0.1293 s	12.002×
SVT (Alg 1, Operator)	2	100	$s = 5$	1.6554 s	0.1412 s	11.728×
TTM (Alg 2, Contraction)	3	100	$r = 10$	0.0194 s (7.63 MB)	0.0134 s	1.453×
TTM (Alg 2, Contraction)	4	24	$r = 5$	0.0169 s (13.18 MB)	0.0052 s	3.260×
TTM (Alg 2, Contraction)	4	32	$r = 5$	0.0696 s (31.25 MB)	0.0172 s	4.038×
State Memory (Alg 3)	3	50	$p = .05,\ r = 5$	0.954 MB	0.197 MB	4.831×
State Memory (Alg 3)	3	100	$p = .05,\ r = 5$	7.629 MB	1.538 MB	4.960×
State Memory (Alg 3)	3	100	$p = .10,\ r = 5$	7.629 MB	3.064 MB	2.490×
State Memory (Alg 3)	3	100	$p = .30,\ r = 5$	7.629 MB	9.168 MB	0.832×
State Memory (Alg 3)	3	400	$p = .05,\ r = 5$	488.281 MB	97.703 MB	4.998×
State Memory (Alg 3)	3	400	$p = .30,\ r = 5$	488.281 MB	585.984 MB	0.833×

F.5 Comprehensive Best-Configuration Benchmark Summary across Regimes

Table 9 details the per-estimator computational costs of the best lightweight configuration relative to the literal Raw implementation across both order-3 and order-4 regimes, spanning diverse sampling ratios ($p \in \{0.05, 0.1, 0.2\}$) and observation modalities (denoising and completion).

Table 9: Per-estimator cost of the best lightweight configuration relative to Raw (medium preset, MATLAB R2025b, single thread). Runtime is the median time per iteration. Configurations are selected independently for runtime and memory.

			Runtime		Memory		
Estimator	**Regime**	**Setting**	**Raw → Light. (s)**	**Config.**	**Raw → Light. (MB)**	**Config.**	**Reduction**
E0 Subspace Norm	Denoise	$50, 5$	.0155 → .0114	A2+A3	9.09 → 1.94	A2+A3	**4.69**×
E0 Subspace Norm	Denoise	$100, 5$	.1366 → .0873	A2+A3	66.81 → 15.32	A2+A3	**4.36**×
E0 Subspace Norm	Denoise ($N = 3$)	$120, 5$	.2289 → .1539	All	113.78 → 26.44	All	**4.30**×
E0 Subspace Norm	Denoise ($N = 4$)	$24, 3$	.0690 → .0590	All	36.72 → 5.08	All	**7.23**×
E1 Latent norm	$p = .05$	$50, 5$	.0386 → .0698	A1	6.68 → .8262	A1+A3	**8.08**×
E1 Latent norm	$p = .10$	$50, 5$	.0389 → .0637	A1	6.68 → .7569	A1+A3	**8.82**×
E1 Latent norm	$p = .05$	$100, 5$	.5400 → .5849	A1	53.41 → 10.09	A1+A3	**5.29**×
E1 Latent norm	$p = .10$	$100, 5$	.5422 → .5341	A1	53.41 → 9.24	A1+A3	**5.78**×
E1 Latent norm	$p = .05$ ($N = 3$)	$120, 5$	1.20 → 1.78	All	92.29 → 25.92	All	**3.56**×
E2 FFW	$p = .05$	$50, 5$	.0179 → .0179	A3	2.20 → .3395	A1+A3	**6.48**×
E2 FFW	$p = .10$	$50, 5$	.0185 → .0188	A3	2.20 → .3871	A1+A3	**5.68**×
E2 FFW	$p = .05$	$100, 5$	.2374 → .2271	A3	16.41 → 1.54	A1+A3	**10.68**×
E2 FFW	$p = .10$	$100, 5$	.2411 → .2345	A3	16.41 → 1.92	A1+A3	**8.55**×
E2 FFW	$p = .05$ ($N = 3$)	$120, 5$	.5412 → .3220	All	27.25 → 1.55	All	**17.63**×
E3 HOOI	Denoise	$50, 5$	.0058 → .0038	A2+A3	.9604 → .0067	A1+A3	**143.86**×
E3 HOOI	Denoise	$100, 5$	.0386 → .0124	A2+A3	7.64 → .0124	A1+A3	**616.38**×
E3 HOOI	Denoise ($N = 3$)	$120, 5$	.0628 → .0234	All	13.20 → .0147	All	**898.66**×
E3 HOOI	Denoise ($N = 4$)	$24, 3$	.0269 → .0122	All	2.53 → .0028	All	**900.12**×
E4 ScaledGD	$p = .05$	$50, 5$	.0026 → .0023	A1+A2	1.91 → .0542	A1+A3	**35.31**×
E4 ScaledGD	$p = .10$	$50, 5$	.0026 → .0024	A1+A2	1.91 → .1019	A1+A3	**18.79**×
E4 ScaledGD	$p = .05$	$100, 5$	.0330 → .0195	A2+A3	15.27 → .3933	A1+A3	**38.83**×
E4 ScaledGD	$p = .10$	$100, 5$	.0341 → .0317	A2+A3	15.27 → .7765	A1+A3	**19.67**×
E4 ScaledGD	$p = .05$ ($N = 3$)	$120, 5$	.0541 → .0314	All	26.38 → .6741	All	**39.13**×
E4 ScaledGD	$p = .20$ ($N = 4$)	$24, 3$	.0270 → .0261	All	5.07 → .5083	All	**9.96**×