Low Rank Multi-Dictionary Selection at Scale

Boya Ma
University at Albany, State University of New York
Department of Computer Science
Albany, New York, USA
bma@albany.edu

Maxwell McNeil
University at Albany, State University of New York
Department of Computer Science
Albany, New York, USA
mmcneil2@albany.edu

Abram Magner
University at Albany, State University of New York
Department of Computer Science
Albany, New York, USA
amagner@albany.edu

Petko Bogdanov
University at Albany, State University of New York
Department of Computer Science
Albany, New York, USA
pbogdanov@albany.edu

Abstract

The sparse dictionary coding framework represents signals as a linear combination of a few predefined dictionary atoms. It has been employed for images, time series, graph signals and recently for 2-way (or 2D) spatio-temporal data employing jointly temporal and spatial dictionaries. Large and over-complete dictionaries enable high-quality models, but also pose scalability challenges which are exacerbated in multi-dictionary settings. Hence, an important problem that we address in this paper is: *How to scale multi-dictionary coding for large dictionaries and datasets?*

We propose a multi-dictionary atom selection technique for low-rank sparse coding named LRMDS. To enable scalability to large dictionaries and datasets, it progressively selects groups of row-column atom pairs based on their alignment with the data and performs convex relaxation coding via the corresponding sub-dictionaries. We demonstrate both theoretically and experimentally that when the data has a low-rank encoding with a sparse subset of the atoms, LRMDS is able to select them with strong guarantees under mild assumptions. Furthermore, we demonstrate the scalability and quality of LRMDS in both synthetic and real-world datasets and for a range of coding dictionaries. It achieves 3× to 10× speed-up compared to baselines, while obtaining up to two orders of magnitude improvement in representation quality on some of the real world datasets given a fixed target number of atoms.

CCS Concepts

• **Information systems → Data mining**.

Keywords

sparse coding, dictionary selection, low rank methods

ACM Reference Format:
Boya Ma, Maxwell McNeil, Abram Magner, and Petko Bogdanov. 2024. Low Rank Multi-Dictionary Selection at Scale. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3671723

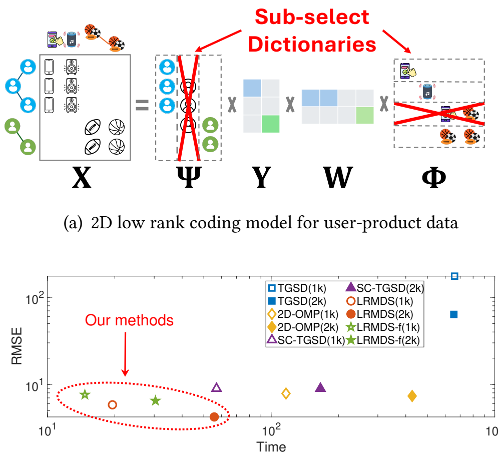
KDD '24, August 25–29, 2024, Barcelona, Spain
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671723

1 Introduction

Sparse coding methods represent data as a linear combination of a predefined basis (called atoms) arranged in a dictionary [26]. Dictionaries are either derived analytically, for example, discrete Fourier transform, Wavelets, Ramanujan periodic basis [28] or learned from data [29]. A key assumption in sparse coding is that real-world signals are sparse (or compressive) and can be represented via a small subset of dictionary atoms. Sparse coding has been widely adopted in signal processing [26], machine learning [18],time series analysis [15, 34], image processing [7], data mining [20, 21] and computer vision [31] among others.

Existing approaches, depending on their sparsity-promoting functions, fall in three main categories [17]: convex relaxation [4], non-convex algorithms [12], and greedy strategies based on matching pursuit [23]. Most existing work focuses on 1D (vector) signals such as time series [28] and graph signals [6]. More recent approaches employ sparse coding for multi-way datasets such as images [8, 10], spatio-temporal data [22] and higher order tensors [19]. These 2D and higher-D methods employ separate dictionaries for the different modes (dimensions) of the data. Convex relaxation 2D approaches are typically efficient in practice, but their runtime significantly increases with the size of dictionaries and datasets and they also require careful tuning of hyper-parameters to precisely control the density of encoding coefficients [22]. Greedy approaches recover a desired number of coefficients (model size), but re-estimate all coding coefficients as new ones are added, and thus do not scale to large model sizes [8]. Fig. 1(a) demonstrates the 2D setting in the context of user-product purchase data by employing a separate user and product graph dictionaries (e.g., graph Fourier dictionaries [27]) based on the corresponding friendship/association graphs.

Some multi-way techniques further model the coding matrix as low-rank [19, 22, 24] which enables improved performance due to sharing of atom-specific patterns within factors. This modeling assumption is demonstrated in the toy example from Fig. 1(a). Given the purchase history of users, represented as one-hot encoded icons, and association graphs (e.g., user-user friendship and product similarity graphs), one can define graph-based user Ψ and product Φ dictionaries which represent natural communities for



(a) 2D low rank coding model for user-product data

(b) RMSE v.s. time in road traffic data

Figure 1: (a) A 2D low rank coding example for user-product preference data. The left Ψ and right Φ dictionaries are derived from user and product association graphs and the goal is to encode the data sparsely via sparse and low-rank coefficient matrices Y, W. Our method LRMDS sub-selects the dictionary atoms on both sides to speed up the coding process. (b) Comparison of competing techniques on a Road traffic dataset. Variants of LRMDS outperform all baselines in both representation quality (RMSE) and running time (best regime in the lower-left corner).

each of the two data dimensions. If purchase behaviors "conform" to user and product communities (e.g., first three users purchase electronics while the remaining two purchase sports products), then the purchase data can be described efficiently via a few sparse factors as demonstrated in the Y, W coding matrices. For example, the user factors in Y require only two coefficients to represent the corresponding user groups, and similarly W represent groups of products via three coefficients.

While (low-rank) 2D sparse coding is advantageous and widely applicable, existing methods do not scale to large dictionaries and data. The 2D-OMP approach [8] composes atoms as outer products of left and right atoms and greedily selects the best aligning pairs for encoding one at a time. Low rank approaches [19, 22] adopt convex relaxation based on alternating directions methods of multipliers (ADMM). Both groups suffer from poor scalability with the size of the employed dictionaries. 2D-OMP considers a quadratic number of atom combinations while low-rank approaches rely on inversion of matrices whose size depend on that of the employed dictionaries. While this challenge of limited scalability has been addressed in the 1D sparse coding scenario via dictionary screening [32] and greedy dictionary selection [9], these methods are not readily applicable to the 2D scenario. Furthermore, as we demonstrate experimentally, naive generalizations of dictionary screening algorithms for the 2D setting result in limited representation accuracy and scalability.

We propose a low-rank multi-dictionary selection and coding approach for 2D data called LRMDS. Our approach is general, scalable and theoretically justified. To scale to large dictionary sizes, it iteratively performs adaptive joint dictionary sub-selection and efficient low-rank coding based on convex optimization. LRMDS iteratively improves the encoding by adding dictionary atoms as needed in rounds. We prove theoretically and demonstrate empirically that if the input data conforms to a low-rank coding model via a sparse subset of atoms, LRMDS is guaranteed to select these atoms in noisy regimes. An experimental snapshot showcasing the advantages of LRMDS's variants is presented in Fig. 1(b) for a real-world sensor network dataset. In terms of representation error (vertical axis) and running time (horizontal axis), variants of our method (LRMDS and LRMDS-f) occupy the lower left corner which is the optimal regime. We experimentally demonstrate similar advantageous behavior on 3 other datasets detailed in the evaluation section. While in this work we focus on evaluating dictionary selection in 2-way (matrix) data, we believe that our framework can be extended to multi-dictionary settings (tensor data) , though we leave such evaluation for future investigation. Our contributions in this paper are as follows:

- **Novelty:** To the best of our knowledge, LRMDS is the first dictionary selection method for multi-dictionary sparse coding.
- **Scalability:** Our approach scales better than alternatives on large real-world datasets and when employing large dictionaries.
- **Accuracy:** LRMDS consistently produces solutions of lower representational error compared to the closest baselines from the literature for a fixed number of coding coefficients (models size).
- **Theoretical guarantees:** We prove that LRMDS's dictionary selection optimally identifies the necessary atoms for low-rank encoding of the input data in the presence of noise.

2 Related work

Sparse coding is widely employed in signal processing [26, 35], image analysis [7] and computer vision [31]. Existing methods can be grouped into three main categories: convex optimization solutions, non-convex techniques, and greedy algorithms [17]. Relaxation techniques impose sparsity on the coding coefficients via L1 regularizers [19, 22], while greedy algorithms select one atom at a time [5, 14, 30]. Most existing methods focus on 1D signals while our focus in this paper is on 2D signals.

2D and multi-way coding methods generalize the one dimensional setting by employing separate dictionaries for each dimension of the data [8, 10, 19, 22, 33]. Some methods in this group place no assumptions on the rank of the encoding matrix [8, 10, 24, 33], while others employ a low-rank model for the encoding matrix [19, 22]. Most related to LRMDS among above the above are 2D-OMP [8] which also utilizes a greedy projection to select atoms, and TGSD [22] as it also enforces that learned coding coefficients are low rank for 2D data. We elaborate further on these similarities in the following section and demonstrate a superior performance of LRMDS over these baselines in the experimental section.

Dictionary screening and selection. Dictionary screening [32] is a suite of methods/bounds for "discarding" dictionary atoms of 0 encoding weights at a given sparsity level with the goal of reducing the running time of the encoding process. These techniques are limited to 1D data (i.e., only one dictionary), and are not immediately extendable to the two-dictionary setting. We include naive extensions to 2D by creating composite (pairwise) atoms as baselines and demonstrate that they do not scale well to large-dictionaries due to the quadratic space of possible composite atoms. There is also work on greedy atom selection for the 1D case [9] and our method can be viewed as a generalization of such techniques to the 2D sparse coding setting.

3 Preliminaries

Before we define our problem of low rank sub-dictionary selection, we introduce necessary preliminaries and notation. The goal of 1D sparse coding is to represent a signal via a single (column) dictionary $\Psi \in \mathcal{R}^{N\times I}$ optimizing the following objective:

$$\min_y f(y) \text{ s.t. } x = \Psi y,$$

where $x \in \mathcal{R}^N$ represents the given signal, $y \in \mathcal{R}^I$ is the learned encoding and $f(y)$ is a sparsity promoting function (often the L_1 norm). A popular greedy strategy to solve the problem, particularly when the dictionary forms an over-complete basis, is the orthogonal matching pursuit (OMP) [23]. The OMP algorithm maintains a residual of the signal r that is not yet represented, and proceeds in greedy steps to identify the dictionary atom best aligned to the residual: $\psi_t = \text{argmax}_{\psi_i}(r^T\psi_i)$, where $i \in [1, I]$, and ψ_i is the i-th atom in Ψ. The selected atom ψ_t at step t is appended to the result set, the signal is re-encoded and the residual re-computed. The process continues until a desired number of atoms are employed, while satisfying the sparsity function $f(y)$.

In this paper we consider the 2D setting involving two dictionaries. The input to our problem is a real valued data matrix $X \in \mathcal{R}^{N\times M}$ which can be represented via two dictionaries: a left (column) dictionary $\Psi \in \mathcal{R}^{N\times I}$ and a right (row) dictionary $\Phi^T \in \mathcal{R}^{J\times M}$, where I is the number of atoms in Ψ and J is the number of atoms in Φ^T. It is important to note that both analytical and data-driven dictionaries can be employed [26]. The 2D problem generalizes that from the 1D case as follows:

$$\min_Z f(Z) \text{ s.t. } X = \Psi Z \Phi^T, \tag{1}$$

where $Z \in \mathcal{R}^{I\times J}$ is an encoding matrix, and $f(Z)$ is the corresponding sparsity promoting function. Intuitively this decomposition facilitates a representation which aligns to dictionaries across both modes (dimensions) instead of just one. An early solution for the problem in Eq. 1 was motivated by decomposing a 2D image via copies of the same dictionary, i.e. $\Psi = \Phi$ [8]. It generalizes OMP to obtain a 2D-OMP algorithm by forming 2D atoms $B_{i,j} = \psi_i^T\phi_j$ as outer products of individual left ψ_i and right ϕ_j atoms, and by selecting 2D atoms based on their alignment with the residual R at every iteration. Importantly, while sparse, this solution might in general result in high-rank encoding matrix Z and as we demonstrate experimentally it does not scale to large spatio-temporal datasets and large dictionaries.

A recent method called TGSD [22] employs 2D sparse coding for general spatio-temporal datasets, and specifically temporal graph signals, where graph and temporal dictionaries are employed as Ψ and Φ respectively. Another major difference from the 2D-OMP solution is that in order to enforce a low-rank solution, TGSD considers a model with two "slim" dictionary-specific encoding matrices $Y \in \mathcal{R}^{I\times r}$ and $W \in \mathcal{R}^{r\times J}$, where the middle dimension r restricts the rank of the encoding $X = \Psi Y W \Phi^T$. The resulting objective is:

$$\underset{Y,W}{\text{argmin}} ||X - \Psi Y W \Phi^T||_F^2 + \lambda_1||Y||_1 + \lambda_2||W||_1, \tag{2}$$

where sparsity via an L_1 norm is enforced for both Y and W. Here, $||M||_F$ denotes the Frobenius norm of the matrix M. The solution adopts an ADMM convex relaxation approach (unlike the greedy solution of 2D-OMP) producing an explainable decomposition model relating non-zero coefficients to periodic behavior and active spatial/graph domains in the data [22]. However, this solution also does not scale to large coding dictionaries and datasets—a challenge we address in our solution.

4 Problem formulation and solutions

4.1 Problem formulation

The existing methods for multi-dictionary sparse coding, TGSD and 2D-OMP, do not scale to large datasets and dictionaries for different reasons. 2D-OMP selects one atom pair from each dictionary at a time, re-encodes the data and proceeds with the data residual. While each step is initially fast, the number of atom pairs grows quadratically with the size of the dictionaries. In addition, when the data has a low-rank representation through a subset of atoms, 2D-OMP is not guaranteed to uncover it due to its formulation employing an unconstrained coding matrix Z of quadratic size in the dictionary atoms. Different from that, TGSD is a low rank model, however, its optimization relies on inverting matrices whose sizes are determined by the dictionaries, hence it also does not scale with the size of the dictionaries. The scalability limitations are further exacerbated by the use of over-complete dictionaries which has been shown to produce accurate and succinct models in various signal processing and machine learning applications.

Our goal is to enable a (i) scalable, (ii) low-rank, (iii) multi-dictionary sparse coding, accommodating large over-complete dictionaries without compromising the quality of the learned model by subs-electing the dictionary atoms. An additional goal is applicability to any 2D signals, including spatio-temporal data, graph signals evolving over time, images and others, by employing appropriate dictionaries for the corresponding data dimensions. Based on the above intuition we formalize our problem as follows:

Problem definition: *Given a 2D signal X, large potentially over-complete dictionaries Ψ and Φ, and a desired rank r, fit a sparse low-rank model $X \approx \Psi_s Y W \Phi_s^T$, employing a subset of the dictionary atoms Ψ_s, Φ_s and coding matrices of Y, W with inner dimension r.*

4.2 LRMDS: iterative atom selection and coding

Both TGSD and 2D-OMP with minor modifications can be adopted for our problem formulation, however, as we discussed earlier they have limited scalability. The key idea behind our approach is to sub-select both dictionaries jointly and fit a low-rank encoding model through the reduced dictionaries. Hence, *a key assumption in LRMDS is that the data can be represented well by a low-rank encoding matrix and by employing a subset of atoms from the left and the right dictionaries.* This setting is illustrated in Fig. 1(a) where only a subset of atoms from Ψ and Φ are necessary to represent the data. There are two main steps to obtain LRMDS's representation: (i) identify an appropriate subset of dictionary atoms which align well with the data and (ii) employ them to perform a low rank dictionary decomposition. We repeatedly perform these steps against the "unexplained" residual of the data after each iteration. As a result, our approach can be considered a combination of a greedy sub-dictionary identification followed by a convex encoding step. Importantly, we demonstrate that the greedy atom selection step recovers the optimal atoms to best encode a dataset with low-rank and sparse encoding in noisy regimes under mild assumptions.

To jointly sub-select atoms from both dictionaries, we consider all pairwise 2D atoms of the form $B_{i,j} = \psi_i \phi_j^T, \forall i \in [1, I], \forall j \in [1, J]$ and the magnitude of the projection of the data on them. Specifically we maintain a residual matrix $R \in \mathcal{R}^{N \times M}$ initialized as the input data X and subsequently capturing the signal not yet represented by LRMDS. The alignment scores of atom pairs are computed as:

$$P_{i,j} = \frac{\langle R, B_{i,j} \rangle}{||B_{i,j}||_F}, \implies P = \hat{\Psi}^T R \hat{\Phi}, \tag{3}$$

where on the left-hand-side $\langle R, B_{i,j} \rangle \triangleq \psi_i^T R \phi_j$ is the alignment of the i-th left atom and the $j - th$ right atom with the residual, and $||B_{i,j}||_F \triangleq \psi_i \phi_j^T$ is a normalization factor based on the Frobenius norm of the atoms' outer product. The right-hand-side is the equivalent to the left for all $P_{i,j}$ when using per-atom normalized dictionaries $\hat{\Psi}$ and $\hat{\Phi}$ (details in the supplement). At each iteration, our method selects the top k total atoms from a combination of left or right dictionary atoms with respect to this alignment P.

Once atoms are selected we calculate a low-rank decomposition of the data via encoding matrices $Y \in \mathcal{R}^{I_s \times r}$ and $W \in \mathcal{R}^{r \times J_s}$ where r represents the rank of the model, while I_s and J_s are the number of atoms selected from Ψ and Φ respectively. It is important to note, that the sparsity of the representation is ensured thanks to the atom sub-selection and the low-rank (via two encoding matrices) model, hence in this step we do not further enforce sparsity on the encoding coefficients as it is typical to convex relaxation approaches. This modeling decision enables scalable direct solutions as opposed to more complicated ADMM optimizers. The encoding problems has the following form:

$$\underset{Y,W}{\operatorname{argmin}} \; ||R - \Psi_s Y W \Phi_s^T||_F^2, \tag{4}$$

where Ψ_s and Φ_s are the subselected dictionaries and R is the data residual originally initialized as X. We propose two alternating optimization schemes for the two variables Y, W leading to two variants of LRMDS (LRMDS and LRMDS-f). Both variants iteratively updated Y and W until convergence, however, LRMDS-f does so faster but at the potential price of accuracy. Detailed explanation and derivations are available in the supplement.

The overall LRMDS algorithm. The steps of the complete algorithm (corresponding to both versions of our method) are listed in Alg. 1. The inputs include the data X, left Ψ and right Φ dictionaries and parameters for the number of atoms to select per iteration k and the rank r of the encoding, i.e., the inner dimension of the two output encoding matrices Y, W. In the initialization steps, we first compute per-atom normalized versions of the dictionaries needed for the alignment scoring (Steps 4-5) and initialize empty sets of atom indices for both dictionaries (Step 6). Dictionary sub-selection takes place in Steps 8-15. In Steps 10-14 we add the top aligned atoms i and j to the set of select atoms I_s and J_s so long as they are not already selected. We repeat until a total of k new atoms are selected. Finally, we sub-select the relevant atoms from their dictionaries in Step 15 to create sub-dictionaries Ψ_s and Φ_s.

Steps 16-30 perform the low-rank coding by estimating Y and W based on the sub-dictionaries Ψ_s and Φ_s. We list the iterative updates of both versions LRMDS (Steps 18-23) and LRMDS-f (Steps 24-29) of our method. For the former, we pre-compute the pseudo inverses of the subselected dictionaries $\Psi^\dagger, \Phi^\dagger$ and iterate between close form updates of Y and W while in the latter we precompute the projection of the data on the pseudo inverses of the subselected dictionaries $\Psi^\dagger X \Phi^\dagger$ and perform simpler updates for the coefficients in Y and W. Further discussion of the difference between the two versions and derivation details for the updates are available in the supplement. Finally, in Step 31 we re-calculate the residual matrix R. We repeat all steps until a fixed number of iterations is reached or if the norm of the residual ($||R||_F$) approaches zero.

Algorithm 1 Low Rank Multi-Dictionary Selection (LRMDS)

```
 1: Input: Data $X$; dictionaries $\Psi$ and $\Phi$; atoms per iteration $k$, decomposition rank $r$
 2: Output: Encoding matrices $Y, W$
 3: Initialize residual $R = X$
 4:  // Compute normalized dictionaries
 5: Compute $\hat{\Psi} : \{||\hat{\psi}_i||_2 = 1, \forall i \leq I\}, \hat{\Phi} : \{||\hat{\phi}_j||_2 = 1, \forall j \leq J\}$
 6: Initialize sets of selected atoms: $I_s = \emptyset, J_s = \emptyset$
 7: repeat
 8:     // Dictionary sub-selection
 9:     Let $P = \hat{\Psi}^T R \hat{\Phi}$                                  ▷ Eq. 3
10:     $cnt = 0$
11:     for $P_{i,j}$ in descending order and while $cnt < k$ do
12:         if $i \notin I_s$ then $I_s = I_s \cup i; cnt = cnt + 1$
13:         if $j \notin J_s$ then $J_s = J_s \cup j; cnt = cnt + 1$
14:     end for
15:     Sub-select dictionaries: $\Psi_s = \Psi(I_s), \Phi_s = \Phi(J_s)$
16:     // Encoding based on $\Psi_s, \Phi_s$
17:     Initialize randomly $Y_{|I_s| \times r}$ and $W_{r \times |J_s|}$
18:     if LRMDS then
19:         Pre-compute $\Psi_s^{(inv)} = \Psi_s^\dagger$ and $\Phi_s^{(inv)} = \Phi_s^\dagger$
20:         repeat
21:             $Y = \Psi_s^{(inv)} X (W \Phi_s)^\dagger$
22:             $W = (\Psi_s Y)^\dagger X \Phi_s^{(inv)}$
23:         until $Y, W$ converge
24:     else if LRMDS-f then
25:         Pre-compute $C = \Psi^\dagger X \Phi^\dagger$
26:         repeat
27:             $Y = C W^\dagger$
28:             $W = Y^\dagger C$
29:         until $Y, W$ converge
30:     end if
31:     $R = X - \Psi_s Y W \Phi_s^T$
32: until $||R||_F$ converges to 0 or after a fixed number of iterations
```

The overall complexity of LRMDS is $O(t(q(N + M)r^2 + MJ_s^2 + NI_s^2))$ for LRMDS or $O(t(q(I_s + J_s)r^2 + MJ_s^2 + NI_s^2))$ when using LRMDS-f where t, q are the total number of iterations of the main loop and Y, W updating. Our framework is designed to flexibly accommodate any dictionaries Φ and Ψ. Discussion of dictionaries employed for modes of different types (e.g., time series, graphs, etc.) can be found in the extended version [16].

4.3 Dictionary subselection theoretical analysis

In this section, we give an accuracy guarantee for the quality of our method's selection of top k atoms in each step. when used to recover a low-rank signal matrix R from a data matrix $\hat{R} = R + Q$ that includes Gaussian noise. Throughout, we assume without loss of generality that $M = o(\sqrt{N})$ as $N \to \infty$ (the argument applies equally well when the roles of M and N are switched). We describe this denoising problem setting and assumptions for our theoretical guarantee below.

Noise model: We will consider the recovery of a low-rank, sparse signal matrix R from a noise-perturbed version $\hat{R} := R + Q$, where Q

Dataset	#Nodes	#Time Steps	Res.	Associated Graph	TGSD		2D-OMP		SC-TGSD		LRMDS		LRMDS-f	
					RMSE	Time	RMSE	Time	RMSE	Time	RMSE	Time	RMSE	Time
Synthetic	1k-4k	1k-8k	-	SBM	0.06	146	0.02	3196	0.03	61.7	**0.009**	31.3	**0.009**	**30.1**
Road [2]	1923	920	1h	Road network	17.8	285	10.1	228	12.1	32	**5.4**	37	5.8	**15**
Twitch [25]	78,389	512	1h	Shared audience	1.38	8,413	1.36	341,294	1.35	5,353	**1.23**	9655	1.26	**4,280**
Wiki [1]	999	792	1h	Co-clicks	15.7	422	9.7	1390	11.7	41	**2.8**	52	3.5	**37**
Covid [13]	3047	678	1d	Spatial k-NN	31969	551	23908	2668	21320	267	**204**	145	228	**88**

Table 1: Statistics of the datasets used for evaluation (left sub-table) and quality and running times for competing techniques (right sub-table). All datasets have a temporal and graph mode with corresponding dictionaries. The temporal resolution of each dataset is specified in column Res while the following column lists the kind of associated graph. RMSE and timing results of all competing methods using the same number of atoms are listed in the remaining columns. The target number of atoms are as follows: Synthetic with ground truth (GT) atom count (200, GW+RS test in Fig. 2); Road, Twitch, and Covid: 40% of total atoms; Twitch: 20% of total atoms.

is a matrix in $\mathcal{R}^{N\times M}$ with independent and identically distributed standard Gaussian entries, appropriately normalized. Specifically, the assumption that $N \gg M$ implies that with high probability, each column of a standard Gaussian matrix has L_2 norm approximately $\sqrt{N}$. Thus, we define the noise component as:

$$Q = \frac{\sigma}{\sqrt{NM}} \cdot \mathcal{N}(0, I_{N\times M}), \tag{5}$$

for some positive standard deviation σ. With this normalization, $\|Q\|_F = \Theta(1)$, and thus has the same order of growth as $\|R\|_F$, with high probability.

Low rank and sparsity assumptions on R**:** We will assume that R has rank r, which implies that R has an expansion of the form

$$R = \Psi Y W \Phi^T = \sum_{i=1}^{I} \sum_{j=1}^{J} p_{i,j} \psi_i \phi_j^T, \tag{6}$$

where $Y \in \mathcal{R}^{I\times r}$ and $W \in \mathcal{R}^{r\times J}$, and where the double-sum atom-based representation is under the assumption that Ψ and Φ^T are not under-complete. Eq. 6 implies the following explicit $p_{i,j}$ form:

$$p_{i,j} = (Y_{i,\cdot})(W_{\cdot,j})^T. \tag{7}$$

Furthermore, we will make the following sparsity assumption: there exist only $s = \Theta(1)$ dictionary coefficients $p_{i,j}$ in the expansion of R that are nonzero, and these are $\Theta(1)$ uniformly in N and M.

Assumption on approximate orthogonality of dictionary atoms: We next formulate an approximate orthogonality condition for the dictionary atoms. To do so, we first recall the definition of the L_∞ operator norm of a matrix M:

$$\|M\|_{op,\infty} := \sup_{\|x\|_\infty = 1} \|Mx\|_\infty. \tag{8}$$

We will assume that the dictionaries $\Psi \in \mathcal{R}^{N\times I}, \Phi^T \in \mathcal{R}^{J\times M}$ are such that $I \in [N, const \cdot N], J \in [M, const \cdot M]$ and that a subset of the atoms for each dictionary constitutes a basis for $\mathcal{R}^N$ and $\mathcal{R}^M$, respectively. We fix an $\alpha \geq 0$, which may depend on N and M. We also define a matrix $\Sigma \in \mathcal{R}^{I\times I}$ collecting the pairwise inner products between dictionary elements in Ψ: namely, $\Sigma_{i,j} := (\psi_i)^T \cdot \psi_j$. We will assume that $\|\Sigma^{1/2}\|_{op,\infty} \leq \alpha = o(1)$. We similarly define $\Gamma \in \mathcal{R}^{J\times J}$ for Φ^T, with the same bound on $\|\Gamma^{1/2}\|_{op,\infty}$. Intuitively, this operator norm upper bound translates to an upper bound on the sum of absolute values of inner products between dictionary atoms. Thus, this constitutes an approximate orthogonality assumption. Such assumptions are common in analyses of orthogonal matching pursuit, under the name of *mutual incoherence* between dictionary atoms. See, e.g., [3]. We will also assume that the columns of Ψ and the rows of Φ^T are normalized in L_2.

Statement of the accuracy guarantee: We finally state our main theoretical result. We denote by $R_{reconst}$ the output of the top-$\hat{k}$ atom selection algorithm with input data matrix $\hat{R}$ and dictionaries Ψ, Φ^T. Specifically, by this we mean that $R_{reconst}$ is the result of first choosing the $\hat{k}$ atoms (outer products of left and right dictionary elements) with the highest alignment scores with $\hat{R}$, then approximating $\hat{R}$ via a linear combination of the chosen atoms obtained by solving (4).

Theorem 4.1 (Accuracy guarantee for top-k atom selection denoising). *Let $N, M, \Psi, \Phi^T, R, Q, \hat{R}, R_{reconst}$ be as outlined above. We then have that if $\hat{k} \geq s$ and $\hat{k} = \Theta(1)$, where s is the sparsity parameter of the the signal matrix R, then $\|R - R_{reconst}\|_F = o(\|R\|_F)$.*

In other words, the relative error in approximating R by $R_{reconst}$ is $o(1)$ as $N \longrightarrow \infty$. That is, when the data consists of a Gaussian-noise perturbation of a low-rank signal matrix that is a sparse linear combination of dictionary atoms, and when the greedy atom selection algorithm chooses sufficiently many atoms, the signal matrix is recovered to within a vanishingly small relative error. We prove Theorem 4.1 in Appendix A. We also extend the result to cover the case where $\hat{k} < s$ but the algorithm is run for sufficiently many iterations.

Our analysis provides a theoretical justification for the top k atom selection strategy as a recovery guarantee for a noise-perturbed low-rank and sparse signal matrix, which forms a subroutine of our method. We also demonstrate empirically that this recovery guarantee holds in Sec. 5.5. The scalability of our technique comes from the fact that we are working with a few atoms as opposed to the complete dictionary and the theoretical analysis shows that this running time reduction affects minimally the model quality since the"true" atoms necessary for encoding the noise-free version of the data are retained.

5 Experimental evaluation

Our experimental design focuses on the running time and the representation quality of competing methods with both variants of LRMDS on synthetic and real-world datasets listed in Tbl. 1. We also empirically confirm our theoretical results. We compare our approaches to state-of-the-art baselines for 2D sparse coding. We measure running time in seconds for execution on a dedicated Intel(R) Xeon(R) Gold 6138 CPU @ 2.00GHz and 251 GB memory server using MATLAB's R2019a 64-bit version. The representation quality is quantified as the root mean squared error (RMSE) between the data and the learned representation. It is important to note, that beyond representation quality, the low-rank sparse 2D coding model offers advantages in a number of downstream tasks as reported by baselines employing this or similar models [8, 22].

We focus our evaluation of scalability and representation quality as speed-up via dictionary sub-selection is the main contribution of our work. An implementation of LRMDS is available at https://www.cs.albany.edu/~petko/lab/code.html.

5.1 Datasets

Synthetic data generation. Our synthetic data is generated based on the low-rank encoding model $\Psi_s Y W \Phi_s^T + \epsilon$, where Ψ_s and Φ_s are small (ground truth) randomly selected subsets of the overall dictionaries and the corresponding coding coefficients for those atoms in Y and W are also randomly sampled with ϵ-mean random noise added to the input. **Real-world datasets.** We employ real-world datasets with temporal and spatial dimensions to evaluate competing techniques employing both time and graph dictionaries. The datasets span a variety of domains: data from content exchange within *Twitch* [25], web traffic data *Wiki* [1], spatio-temporal disease spread over time in the *Covid* [13] dataset, and sensor network data from road traffic *Road* [2]. We provide further details on data generation and real-world evaluation datasets in the supplement.

5.2 Experimental setup.

Baselines. We compare the versions of LRMDS to the two available methods for multi-dictionary coding: (**TGSD** [22] and **2D-OMP** [8]). These methods have already been discussed in detail in Sec. 3. As a brief summary, **TGSD** solves the problem of low-rank encoding within an L_1 sparsity regularized optimization framework. **2D-OMP** selects dictionary atom pairs in a greedy manner and estimates the corresponding coding coefficients one at a time. It produces a solution which is not guaranteed to be low-rank.

Since a key advantage of our method is its sub-selection of large dictionaries, we also seek to understand if extending 1D dictionary screening to the 2D setting results in a scalable 2D approach. To this end, we generalize a 1D dictionary screening approach [32] to work with multiple dictionaries and combine it with TGSD to facilitate a more thorough comparison. The resulting method **SC-TGSD** screens (removes) the worst dictionary atoms from a dictionary by calculating alignment scores between atoms and associated data. To perform the subsequent coding, we employ TGSD with only the sub-selected dictionaries. Intuitively this baseline can be thought as a 2-step combination of screening and TGSD coding. Details of the screening process are available in the extended version [16].

Metrics. We measure the reconstruction error of the learned representation using root mean squared error (RMSE= $\sqrt{\frac{\sum_{i,j}(X_{i,j}-X'_{i,j})^2}{|X|}}$, where X is the original signal, X' is the reconstruction and $|X|$ denotes the number of elements in X) of the learned representation's departure from the input data. We measure the running time in seconds for all competing methods.

Experimental design. The goal of our experiments is to demonstrate the utility of LRMDS in both synthetic and real world datasets. In synthetic data tests, we varying different properties of the data generation process, such as SNR and the dictionary size and type. We seek to quantify the speed up and quality improvement that LRMDS enables compared to baselines. Parameter settings can be found in the supplement.

5.3 Evaluation on synthetic data.

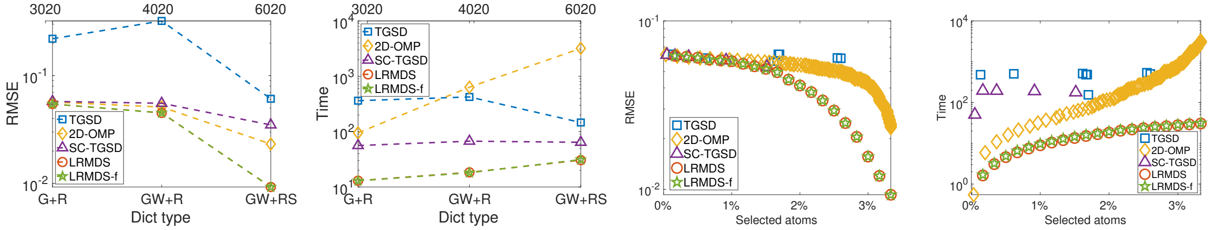
In our synthetic data evaluation we compare the effects of dictionary size and type on the performance of LRMDS and its competitors. Using the ground truth number of atoms as a target, we compare the reconstruction error (RMSE) and running time (secs) for all techniques. We also characterize the effect of varying noise and show these results in the supplement.

Varying dictionary composition and size. As all competitors perform a type of dictionary sub-selection (implicitly in the case of TGSD due to its regularization), a natural question to ask is: *How does the composition and size of the input dictionaries affect the ability of a method to quickly and accurately represent a data matrix?* To answer this question we first utilize a set of composite dictionaries to generate the data input. In this case the left composite dictionary is a stack of a GFT (G) and a graph Wavelet (W) dictionaries and the right composite dictionary is a stack of Ramanujan (R) and Spline (S) temporal dictionaries. We use 50 randomly chosen atoms from each of the four dictionaries (200 in total) to generate the synthetic input data. We then prepare 3 test settings by varying the dictionaries compositions available to the competitors. The model input dictionaries in these 3 settings are as follows (1) $\Psi = [G], \Phi = [R]$ denoted G+R; (2) $\Psi = [G, W], \Phi = [R]$ denoted GW+R; and (3) $\Psi = [G, W], \Phi = [R, S]$ denoted GW+RS.

We first compare the RMSE and running time for all techniques when using a fixed number of dictionary atoms and report results of this experiment in Fig. 2(a), 2(b). The x-axis lists the consecutive dictionary compositions and the total number of atoms is listed on top of the figure. We report RMSE and run time of each method when employing 200 atoms, which is also the number of ground truth (GT) atoms. Note that for SC-TGSD and TGSD, we pick the point for which the employed number of atoms is closest to the GT since they have no parameters to directly control the exact number of atoms in their methods. As larger dictionaries are being used, the RMSE of all methods improves, however LRMDS achieves the best reconstruction quality at all points. This is due to the higher quality atom selection compared to baselines. LRMDS variants are also the fastest to select these atoms as seen in Fig. 2(b).

In Figs. 2(c), 2(d), we further break down the performance of each method when utilizing ground truth composite dictionaries GW+RS. Explicitly, we track the RMSE and run time as a function of the percentage of selected atoms. We additionally show that the representation quality improves as a function of the total run time, and similar results for other choices of dictionary combinations settings (G+R, GW+R) in the extended version [16].

Both variants of LRMDS obtain the most accurate representations among competitors while simultaneously taking the least amount of time. The next best approach, 2D-OMP initially selects and updates its coefficients quickly, closely trailing LRMDS when the representation quality for both is poor. However, with further iterations the representation quality gains by 2D-OMP slow down. This is due to 2D-OMP's restriction to only select one coefficient corresponding to an atom pair per iteration and the need to re-estimate the coefficients for all previously selected pairs. This is highly inefficient when many pairs of a small subset of atoms in the optimal sparse coding are non zero. In such cases, 2D-OMP still



(a) RMSE (b) Time (c) GW+RS: RMSE vs Atom% (d) GW+RS: Time vs Atom%

Figure 2: Comparison of competing techniques on synthetic data. (a), (b): RMSE and running time for varying dictionaries available to each method (listed under the x axis). The total number of (left and right dictionary) atoms is specified at the top of each figure. We stack increasing sets of dictionaries on the left and right, while the ground truth atoms are selected from the full set GW+RS. (c): RMSE as a function of the number of selected atoms when multiple dictionaries are provided. (d): Run time as a function of the number of selected atoms. GW+RS stands for GFT and Graph Haar wavelets stacked together for the graph dimension and RS stands for Ramanujan and Spline dictionaries stacked for the temporal dimension (details of the dictionary definitions are available in the extended version [16]).

adds pairs one at a time, while our approach allows for coding with all combinations of already selected left and right atoms.

5.4 Evaluation on real-word datasets.

We next evaluate all techniques on the real-world datasets and report the RMSE and running time at set percentages of total available atoms selected. Results from all datasets for a fixed percentages of atoms are listed in Tbl. 1. This high-level comparison demonstrates that given a fixed number of target atoms, LRMDS produces the most accurate representations, while LRMDS-f is the most scalable at the cost of slight deteriorating in RMSE compared to LRMDS. Note that LRMDS-f is still the most accurate among baselines from the literature.

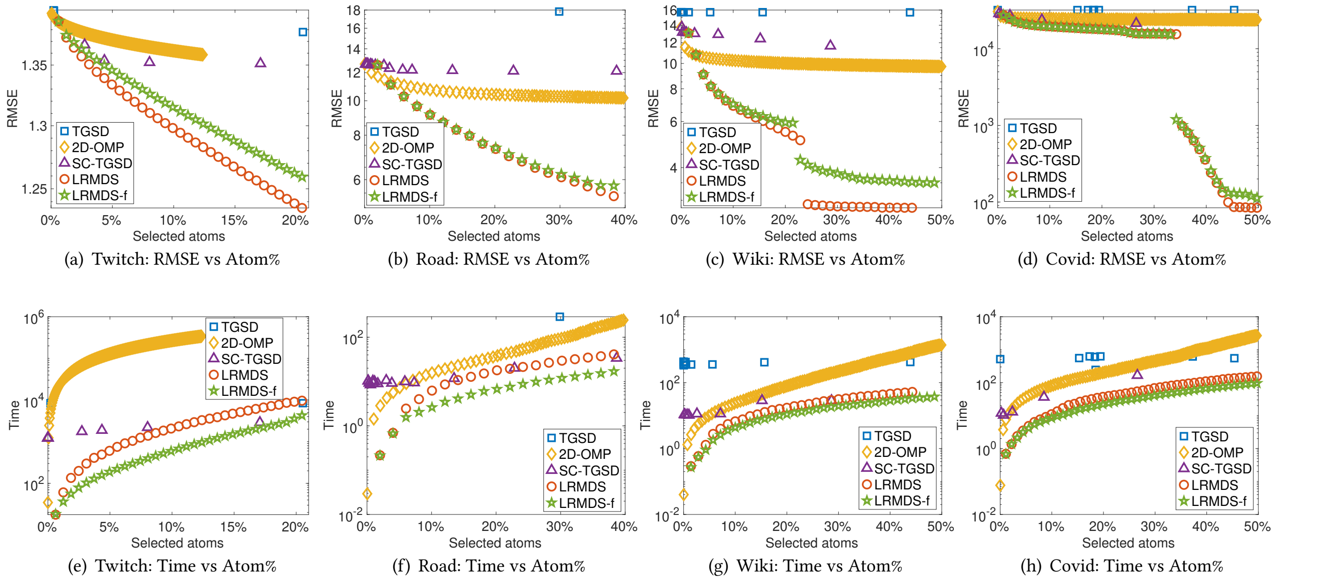
More detailed results on real-world datasets are presented in Fig. 3. We employ a graph Fourier dictionary (GFT) for Ψ and Ramanujan periodic dictionary for Φ for all datasets. The sizes of these dictionaries are listed in the caption of Fig. 3. The detailed analysis also demonstrates that variants of LRMDS dominate based on both accuracy and running time across a wide variety of settings and datasets. We show the representation error as a function of the percentage of selected atoms in Figs 3(a)-3(d) and the run time necessary to obtain said percentages in Figs. 3(e)-3(h). Together these plots show both the quality of representation and the time necessary to obtain it for a varying percentage of selected atoms (additional figures in the extended version [16] explicitly show this relationship). For each dataset, 2D-OMP selects highly representative atoms at first due to its greedy strategy, however, its trend is quickly overtaken by those of our methods as more atoms are allowed for selection. LRMDS matches or outperforms LRMDS-f in terms of representation quality given the same number (percentage) of atoms, however, LRMDS-f selects new atoms faster demonstrating the trade-off between running time and quality between the two. Both of our methods are as fast or faster than all baselines at selecting atoms with the exception of the 2D-OMP in its first several iterations. The only method matching the speed of LRMDS is TGSD-SC (LRMDS-f is always faster). Although fast, TGSD-CS exhibits poor representation quality rendering it not useful in settings in which the representation quality is critical. For the Twitch datasets when a large percentage of atoms are selected, TGSD is able to obtain similar running time to LRMDS but at a far worse representation quality. TGSD is not well suited for dictionary sub-selection as sparsity is only implicitly encouraged through L_1 regularization over all possible coefficients and it has no direct control on which atoms are used. Thus, even a single nonzero coefficient corresponding to an otherwise poorly selected atom may cause the atom to be "selected" by TGSD.

An interesting finding is that there is an abrupt drop in RMSE in Wiki and Covid data for both variants of LRMDS. This indicates that the learned representation is initially missing some crucial atoms that LRMDS is able to eventually detect and incorporate into the selected dictionaries. Competitors omit these crucial atoms in their representations leading to poorer RMSE. This drop also corresponds to a setting where the difference in quality between LRMDS and competitors is most striking. For example, in Fig. 3(d) LRMDS obtains a roughly two orders of magnitude reduction in RMSE when 50% of the available atoms are selected.

5.5 Theoretical guarantees validation (Thm 4.1)

Here, we study the performance of the LRMDS for denoising which serves as empirical validation of Thm. 4.1. Specifically, we demonstrate that LRMDS is able to recover the underlying "clean" signal R from a noisy signal $\hat{R} = R + Q$ (Fig.4(a)). The experimental setup is as follows: N, M, I, J are set to 500, 10, 1000, 20, respectively. The rank r of the signal is set to 3. For each dictionary, the first half of its atoms are almost orthogonal (generated as an orthogonal basis with Gaussian noise added to the atoms at SNR=20), and the atoms in the second half of the dictionary are generated as Gaussian $\mathcal{N}(0, 1)$ random. The sparsity parameter s is set to 10% of the total number of the almost-orthogonal atoms. The GT atoms for the signal matrix R are chosen uniformly at random from the first half of the atoms (almost-orthogonal), and the atom coefficients are selected randomly ($\mathcal{N}(0, 1)$). This constitutes the clean signal matrix R. We also create a pure independent Gaussian noise matrix Q by first calculating the standard deviation σ of R, and set $Q = \mathcal{N}(0, \frac{\sigma}{20})$. Finally, we set $\hat{R} = R + Q$.

To demonstrate LRMDS's ability to denoise input data $\hat{R}$, we run LRMDS on both R and $\hat{R}$ producing two sets of coefficients $(YW)_R$ and $(YW)_{R+Q}$. We then track the RMSE of the reconstruction for both variants against the clean data R (i.e., RMSE($R - \Psi(YW)_R\Phi^T$) and RMSE($R - \Psi(YW)_{R+Q}\Phi^T$)). Results from this analysis are presented in Fig. 4(a). The curves are nearly identical regardless of the input, demonstrating that LRMDS successfully extracts the underlying signal while ignoring the noise. To further investigate this property we compare the three different sets of dictionary coefficients corresponding to R, $\hat{R}$, and Q: $(YW)_R$, $(YW)_{R+Q}$ (as above) and Z_Q. Z_Q contains coefficients computed via 2D-OMP



(a) Twitch: RMSE vs Atom%

(b) Road: RMSE vs Atom%

(c) Wiki: RMSE vs Atom%

(d) Covid: RMSE vs Atom%

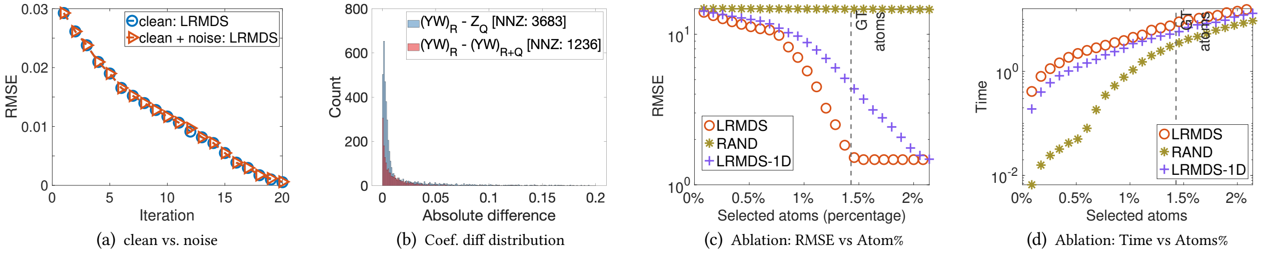
(e) Twitch: Time vs Atom%

(f) Road: Time vs Atom%

(g) Wiki: Time vs Atom%

(h) Covid: Time vs Atom%

Figure 3: Comparison between competitors of representation quality as a function of the percentage of selected atoms Figs.(a)-(d), and runtime as a function of the percentage of selected atoms Figs.(e)-(h). All methods use a GFT for Ψ and a Ramanujan periodic dictionary for Φ. The dimensions of the utilized dictionaries are as follows: Twitch: $\Psi \in \mathcal{R}^{78389\times78389}$, $\Phi \in \mathcal{R}^{512\times2230}$; Wiki: $\Psi \in \mathcal{R}^{999\times999}$,$\Phi \in \mathcal{R}^{792\times6000}$; Road: $\Psi \in \mathcal{R}^{1923\times1923}$,$\Phi \in \mathcal{R}^{720\times3044}$; Covid: $\Psi \in \mathcal{R}^{3047\times3047}$,$\Phi \in \mathcal{R}^{678\times6000}$. Note: 2D-OMP's trace on the Twitch dataset is truncated early as it does not scale (fails to complete in 72 hours) when selecting more than 13% of the atoms.



(a) clean vs. noise

(b) Coef. diff distribution

(c) Ablation: RMSE vs Atom%

(d) Ablation: Time vs Atoms%

Figure 4: (a)-(b): Empirical demonstration of the theoretical guarantee on LRMDS's ability to denoise a signal. (a): "clean: LRMDS" operates on the clean matrix R whereas "clean + noise" operates on the noisy signal $\hat{R} = R + Q$. The RMSE for both methods is measured with respect to the clean data R. (b) The absolute difference between the learned coefficient matrices for the clean data $(YW)_R$, noisy data $(YW)_{R+Q}$, and pure noise Z_Q. (c)(d): Ablation study demonstrating the importance of joint selection of atoms from both dictionaries. We compare LRMDS to variants in which atoms are selected from the left and right dictionaries independently (LRMDS-1D) or randomly (RAND). We measure RMSE (c) and runtime (d) as a function of the percentage of selected atoms.

of the noise matrix. We utilize this instead of LRMDS as due to its low rank constraint it is not capable of well representing an arbitrary noise matrix (as demonstrated above). All methods are run until they converge for their respective inputs. We then calculate the absolute difference in the learned coefficients. Explicitly, $|(YW)_R - (YW)_{R+Q}|$ and $|(YW)_R - (Z)_Q|$ and plot the histograms of the nonzero difference values in Fig. 4(b). While the noise Z_Q and clean data $(YW)_R$ differ significantly (3683 non-zero differences between the two), the fits of the noisy $(YW)_{R+Q}$ and clean $(YW)_R$ data align much better (1236 non-zero differences). The histograms of these differences also indicate that the addition of noise does not significantly impact the coefficients learned by LRMDS.

5.6 Ablation study: Is joint selection critical?

LRMDS uses the projection of the residual onto left-right atom pairs (i.e., $P = \hat{\Psi}^T R \hat{\Phi}$) to select atoms. This opens a natural question on the necessity of this technique: *Can we select atoms from each of the dictionaries independently employing 1D approaches directly on the left and right dictionary? In other words, is joint selection based on the projection we employ critical?* To answer these questions, we implement two variants of LRMDS: i) LRMDS-1D selects atoms from one dictionary at a time via 1D projection, while ii) RAND chooses 2D atoms randomly. We then evaluate their performance on a version of our synthetic dataset with an equal number of ground truth atoms in Ψ and Φ. More details on the implementation and setting for this experiment are available in the extended version [16].

In Fig. 4(c) we plot RMSE of the three variants of our method as a function of the number of selected atoms. LRMDS approaches its optimal fit (smallest RMSE) when using the ground truth number of atoms. LRMDS-1D requires more atoms to achieve the same level of RMSE, demonstrating that the joint atom selection is essential for identifying good representative atoms from both dictionaries. The RAND method (random 2D atom selection) is unlikely to select atoms aligned with the data leading to its poor performance. The running time of LRMDS and LRMDS-1D are similar with the latter having a slight advantage due to its cheaper selection mechanism and residual re-calculation (Fig. 4(d)). For LRMDS, the projection requires multiplication of complexity $O(min(INM + IMJ, INJ + NMJ)$, whereas the projection in LRMDS-1D has a complexity of $O(INM+NMJ)$. Note that $M < J$ in this experiment, explaining the runtime advantage of LRMDS-1D. RAND runs much faster at the beginning as there is no projection to select atoms, however, when more atoms are added this advantage shrinks dramatically. This is because the computational complexity quickly becomes dominated by the coefficient updates which take similar time regardless of which atoms are selected.

Another significant weakness of LRMDS-1D and RAND not highlighted by this experiment is their inability to adaptively select different number of atoms from the right and left dictionaries. The user must specify how many atoms should be selected from each dictionary manually. In contrast, LRMDS can dynamically select the best atoms from either dictionary in a data-driven manner. Thus, this experiment represents an ideal scenario where a user has correctly identified the proportion of atoms need from Ψ and Φ.

6 Conclusion

In this paper we introduced LRMDS, a scalable and accurate method for sparse multi-dictionary coding of 2D datasets. Our approach sub-selects dictionary atoms and employs convex optimization to encode the data using the selected atoms. We provided a theoretical guarantee for the quality of the atom sub-selection for the task of denoising the data. We also demonstrated the quality and scalability of LRMDS on several real-world datasets and by employing multiple analytical dictionaries. It outperformed state-of-the-art 2D sparse coding baselines by up to 1 order of magnitude in terms of running time and up to 2 orders of magnitude in representation quality on some of the real-world datasets. As a future direction, we plan to extend our dictionary selection approach to multi-way data (i.e., tensors) making the core idea applicable to a wider range of problem settings.

7 Acknowledgements

This research was funded by the NSF SC&C grant CMMI-1831547. AM was funded by NSF CCF grants CIF-2212327 and CIF-2338855.

References

[1] [n. d.]. Wikipedia Page Views Statistics http://dumps.wikimedia.org/other/pagecounts-raw/.

[2] Peter Bickel, Chao Chen, Jaimie Kwon, John Rice, and Erik Zwet. 2002. Traffic Flow on a Freeway Network. (01 2002). https://doi.org/10.1007/978-0-387-21579-2_5

[3] T Tony Cai and Lie Wang. 2011. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information theory* 57, 7 (2011), 4680–4688.

[4] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. 2001. Atomic decomposition by basis pursuit. *SIAM review* 43, 1 (2001), 129–159.

[5] Nilson Maciel de Paiva, Elaine Crespo Marques, and Lirida Alves de Barros Naviner. 2017. Sparsity analysis using a mixed approach with greedy and LS algorithms on channel estimation. In *2017 3rd International Conference on Frontiers of Signal Processing (ICFSP)*. IEEE, 91–95.

[6] Xiaowen Dong, Dorina Thanou, Laura Toni, Michael Bronstein, and Pascal Frossard. 2020. Graph signal processing for machine learning: A review and new perspectives. *IEEE Signal processing magazine* 37, 6 (2020), 117–127.

[7] Michael Elad and Michal Aharon. 2006. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing* 15, 12 (2006), 3736–3745.

[8] Yong Fang, JiaJi Wu, and BorMin Huang. 2012. 2D sparse signal recovery via 2D orthogonal matching pursuit. *Science China Information Sciences* 55 (2012), 889–897.

[9] Kaito Fujii and Tasuku Soma. 2018. Fast greedy algorithms for dictionary selection with generalized sparsity constraints. *Advances in Neural Information Processing Systems* 31 (2018).

[10] Aboozar Ghaffari, Massoud Babaie-Zadeh, and Christian Jutten. 2009. Sparse decomposition of two dimensional signals. In *2009 IEEE international conference on acoustics, speech and signal processing*. IEEE, 3157–3160.

[11] Thomas Nall Eden Greville. 1966. Note on the generalized inverse of a matrix product. *Siam Review* 8, 4 (1966), 518–521.

[12] Shihao Ji, Ya Xue, and Lawrence Carin. 2008. Bayesian compressive sensing. *IEEE Transactions on signal processing* 56, 6 (2008), 2346–2356.

[13] Samue Kemp, Jason W Howel, and Peter C Lu. 2020. Bing COVID-19 Tracker. www.bing.com/covid

[14] Jaeseok Lee, Jun Won Choi, and Byonghyo Shim. 2016. Sparse signal recovery via tree search matching pursuit. *Journal of Communications and Networks* 18, 5 (2016), 699–712.

[15] Boya Ma, Maxwell McNeil, and Petko Bogdanov. 2023. GIST: Graph Inference for Structured Time Series. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM, 433–441.

[16] Boya Ma, Maxwell McNeil, Abram Magner, and Petko Bogdanov. 2024. Low Rank Multi-Dictionary Selection at Scale. arXiv:2406.06960 [cs.LG]

[17] Elaine Crespo Marques, Nilson Maciel, Lirida Naviner, Hao Cai, and Jun Yang. 2018. A review of sparse recovery algorithms. *IEEE access* 7 (2018), 1300–1322.

[18] Andreas Maurer, Massi Pontil, and Bernardino Romera-Paredes. 2013. Sparse coding for multitask and transfer learning. In *International conference on machine learning*. PMLR, 343–351.

[19] Maxwell McNeil and Petko Bogdanov. 2023. Multi-dictionary tensor decomposition. In *2023 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1217–1222.

[20] Maxwell McNeil, Boya Ma, and Petko Bogdanov. 2022. SAGA: Signal-aware graph aggregation. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*. SIAM, 136–144.

[21] Maxwell McNeil, Carolina Mattsson, Frank W Takes, and Petko Bogdanov. 2023. CADENCE: Community-Aware Detection of Dynamic Network States. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM, 1–9.

[22] Maxwell J McNeil, Lin Zhang, and Petko Bogdanov. 2021. Temporal Graph Signal Decomposition. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1191–1201.

[23] Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. 1993. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*. IEEE, 40–44.

[24] Wei Qiu, Jianxiong Zhou, and Qiang Fu. 2019. Jointly using low-rank and sparsity priors for sparse inverse synthetic aperture radar imaging. *IEEE Transactions on Image Processing* 29 (2019), 100–115.

[25] Jérémie Rappaz, Julian McAuley, and Karl Aberer. 2021. Recommendation on Live-Streaming Platforms: Dynamic Availability and Repeat Consumption. In *Fifteenth ACM Conference on Recommender Systems*. 390–399.

[26] Ron Rubinstein, Alfred M Bruckstein, and Michael Elad. 2010. Dictionaries for sparse representation modeling. *Proc. IEEE* 98, 6 (2010), 1045–1057.

[27] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. 2013. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine* 30, 3 (2013), 83–98.

[28] Srikanth V. Tenneti and P. P. Vaidyanathan. 2015. Nested Periodic Matrices and Dictionaries: New Signal Representations for Period Estimation. *IEEE Trans. Signal Processing* 63, 14 (2015), 3736–3750. https://doi.org/10.1109/TSP.2015.2434318

[29] Ivana Tošić and Pascal Frossard. 2011. Dictionary learning. *IEEE Signal Processing Magazine* 28, 2 (2011), 27–38.

[30] Jian Wang, Seokbeop Kwon, and Byonghyo Shim. 2012. Generalized orthogonal matching pursuit. *IEEE Transactions on signal processing* 60, 12 (2012), 6202–6216.

[31] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. 2008. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence* 31, 2 (2008), 210–227.

[32] Zhen James Xiang, Yun Wang, and Peter J Ramadge. 2016. Screening tests for lasso problems. *IEEE transactions on pattern analysis and machine intelligence* 39, 5 (2016), 1008–1027.

[33] Dong Zhang, Yongshun Zhang, and Cunqian Feng. 2017. Joint-2D-SL0 Algorithm for Joint Sparse Matrix Reconstruction. *International Journal of Antennas and Propagation* 2017, 1 (2017), 6862852.

[34] Lin Zhang, Wenyu Zhang, Maxwell J McNeil, Nachuan Chengwang, David S Matteson, and Petko Bogdanov. 2021. AURORA: A Unified fRamework fOR Anomaly detection on multivariate time series. *Data Mining and Knowledge Discovery* 35, 5 (2021), 1882–1905.

[35] Zheng Zhang, Yong Xu, Jian Yang, Xuelong Li, and David Zhang. 2015. A survey of sparse representation: algorithms and applications. *IEEE access* 3 (2015), 490–530.

Supplemental Material

We next add supplemental content to aid reproducibility and the understanding of our theoretical analysis including proofs of the theoretical results, method derivation, dataset preparation and parameter tuning.

A Proofs of Theorem 4.1

Here we give details for the proof of Theorem 4.1. Intuitively, the task boils down to showing that the coefficients in any dictionary expansion of the noise matrix Q are uniformly $o(1)$, which allows us to recover the dictionary atoms that contribute to the signal matrix R. As a reminder $Q \in \mathcal{R}^{N\times M}$ with independent and identically distributed standard Gaussian entries.

We start with several lemmas. In essence, the first lemma allows us to focus on upper bounding the inner product of the columns of Q with those of Ψ in order to upper bound the coefficients in any dictionary expansion of Q. Before stating it, we note that $\mathcal{R}^{N\times M}$ is an inner product space with inner product $\langle A, B\rangle_{N,M} := \sum_{i=1}^{N}\sum_{j=1}^{M} A_{i,j}B_{i,j}$. It is a matter of simple algebra to show the following formula for $\langle Q, \psi_i\phi_j^T\rangle_{N,M}$:

$$\langle Q, \psi_i\phi_j^T\rangle_{N,M} = \sum_{k=1}^{M} \phi_{j,k}\cdot\langle Q_{\cdot,k}, \psi_i\rangle_{N,M}. \tag{9}$$

This implies the following upper bound, since the rows of Φ^T are normalized in L_2:

$$|\langle Q, \psi_i\phi_j^T\rangle_{N,M}| \le \sqrt{M}\cdot \max_{k\in[M]} |\langle Q_{\cdot,k}, \psi_i\rangle_{N,M}|, \tag{10}$$

using the fact that for any vector $x \in \mathcal{R}^d$, $\|x\|_1 \le \sqrt{d}\|x\|_2$. In other words, to upper bound the inner product of Q with any dictionary element, it suffices to upper bound the inner product of any column of Q with any element of the left-hand dictionary.

Lemma A.1 (Comparison of inner products with dictionary coefficients). *Under the boundedness assumptions on dictionary atoms, if an $O(1)$-norm vector $Q \in \mathcal{R}^{N\times M}$ has an expansion $Q = \sum_{i=1,j=1}^{I,J} c_{i,j}\cdot\psi_i\phi_j^T$ for $c_{i,j} \in \mathcal{R}$, then we may upper bound $\max_{i\in[I],j\in[J]} |c_{i,j}|$ by upper bounding the inner product $\max_{i,j}\langle Q, \psi_i\phi_j^T\rangle_{N,M}$. Specifically, for all $(i,j) \in [I]\times[J]$,*

$$|\langle Q, \psi_i\phi_j^T\rangle_{N,M} - c_{i,j}| = O(M\alpha^2) = o(1). \tag{11}$$

Proof. We note that by linearity of the inner product,

$$\langle Q, \psi_i\phi_j^T\rangle_{N,M} = c_{i,j} + \sum_{(k,\ell)\neq(i,j)} c_{k,\ell}\langle\psi_k\phi_\ell^T, \psi_i\phi_j^T\rangle_{N,M}. \tag{12}$$

The coefficients $c_{k,\ell}$ are uniformly $O(1)$ by virtue of Q having norm $O(1)$, so this simplifies to

$$\langle Q, \psi_i\phi_j^T\rangle_{N,M} - c_{i,j} = O(1)\cdot\sum_{(k,\ell)\neq(i,j)}\langle\psi_k\phi_\ell^T, \psi_i\phi_j^T\rangle_{N,M} \tag{13}$$

$$= O(1)\sum_{(k,\ell)\neq(i,j)}\langle\psi_k, \psi_i\rangle_{N,M}\langle\phi_\ell^T, \phi_j^T\rangle_{N,M} \tag{14}$$

$$\le O(1)\sum_{(k,\ell)\neq(i,j)}\langle\psi_k, \psi_i\rangle_{N,M} \tag{15}$$

$$\le O(M\alpha^2). \tag{16}$$

□

Lemma A.1 implies that we may upper bound the coefficients of an expansion of a matrix Q using the inner products of Q with dictionary elements. The upper bound (10) allows us to further upper bound $\langle Q, \psi_i\phi_j^T\rangle_{N,M}$, reducing our problem to upper bounding the entries of a multivariate Gaussian random variable. Specifically, if we denote by $K \in \mathcal{R}^{M\times I}$ the matrix $K = Q^T\Psi$, then $K_{i,j} = \langle Q_{\cdot,i}, \psi_j\rangle_{N,M}$. We then have that

$$\max_{i,j}|c_{i,j}| \le \sqrt{M}\max_{i,j}|K_{i,j}| + o(1). \tag{17}$$

The next lemma gives us a tool to upper bound $\max_{i,j}|K_{i,j}|$ by using the fact that the covariance of $K_{i,j}$ and $K_{i,\ell}$, for $\ell \neq j$, is equal to $\langle\psi_j, \psi_\ell\rangle_{N,M}/\sqrt{NM} = \Sigma_{j,\ell}/\sqrt{NM}$. In other words, the vector $\hat{K}$ obtained by appending the columns of K^T into a column vector of dimension $M\cdot I$ has distribution $\mathcal{N}(0,\hat{\Sigma})$, where $\hat{\Sigma} \in \mathcal{R}^{MI\times MI}$ and satisfies $\|\hat{\Sigma}^{1/2}\|_{op,\infty} = \frac{1}{\sqrt{N}}\cdot\|\Sigma^{1/2}\|_{op,\infty}$.

Lemma A.2 (Upper bound on the maximum of correlated Gaussians). *Let $X \sim \mathcal{N}(0,\Sigma)$ be a Gaussian vector in $\mathcal{R}^n$ with covariance matrix $\Sigma \in \mathcal{R}^{n\times n}$. Then we have that*

$$\mathbb{E}[\|X\|_\infty] = O(\|\Sigma^{1/2}\|_{op,\infty}\cdot\sqrt{\log n}). \tag{18}$$

Proof. This is a consequence of a well-known upper bound on the maximum of independent and identically distributed standard normal random variables, along with the fact that, for an isotropic, mean 0 Gaussian vector Z, $\Sigma^{1/2}Z$ has covariance matrix Σ. □

Lemma A.3 (Upper bound on the maximum inner product between a noise vector and a dictionary atom). *Consider the matrix $K = Q^T\cdot\Psi \in \mathcal{R}^{M\times I}$ whose (i,j)th entry is the inner product of the ith column of Q with the jth dictionary element of Ψ. We have that with high probability,*

$$\max_{i,j}|K_{i,j}| = O(\sqrt{\log(MI)}/\sqrt{N}). \tag{19}$$

Proof. We start with Lemma A.2 applied to $\hat{K}$. This yields

$$\mathbb{E}[\|\hat{K}\|_\infty] = O(\|\hat{\Sigma}^{1/2}\|_{op,\infty}\cdot\frac{\sqrt{\log(MI)}}{\sqrt{NM}}) \tag{20}$$

$$= O(\frac{\sqrt{\log(MI)}}{\sqrt{N}}\cdot\|\Sigma^{1/2}\|_{op,\infty}). \tag{21}$$

The proof is finished by applying Markov's inequality. □

A corollary of Lemma A.3 is that the maximum inner product between Q and the dictionary elements is $o(1)$ with high probability. This implies, by Lemma A.1, that the coefficients of Q in any of its dictionary expansions are uniformly $o(1)$. Because of the sparsity assumption on the coefficients of R, the data matrix $\hat{R}$ has s coefficients that are $\Theta(1)$, while the rest are $o(1)$. Thus, provided that the dictionary atom selection procedure selects $\hat{k} \ge s$ atoms with $\hat{k} = \Theta(1)$, those atoms for which R has nonzero coefficients will be among those selected. As a result, the reconstructed matrix $R_{reconst}$ differs from the signal matrix R by only $o(\|R\|_F)$. This completes the proof of Theorem 4.1.

We note that only a minor tweak of the above proof is needed to extend to the case where dictionary selection is iteratively applied for a fixed number $t \geq s/\hat{k}$ of steps, each time to the residual of the previous step. Specifically, in order to formulate this, we need more notation. Suppose, as before, that R is a linear combination of s atoms, each with coefficient uniformly $\Theta(1)$. Suppose that $\hat{k} < s$. Let $R_{\leq \hat{k}}$ denote the truncation of R to its top $\hat{k}$ atoms (i.e., those with the largest coefficients in absolute value), and let $R_{>\hat{k}} := R - R_{\leq \hat{k}}$. That is, $R_{>\hat{k}}$ is the residual of the signal matrix after subtracting $R_{\leq \hat{k}}$. Finally, we define $R_{reconst,\leq \hat{k}}$ to be the output of the algorithm after a single iteration. The proof of our theorem so far showed that $R_{reconst,\leq \hat{k}} - R_{\leq \hat{k}} = o(R_{\leq \hat{k}})$. This implies the following:

$$\begin{aligned} \hat{R} - R_{reconst,\leq \hat{k}} &= (Q + R_{\leq \hat{k}} + R_{>\hat{k}}) - R_{reconst,\leq \hat{k}} && (22) \\ &= (R_{>\hat{k}} + Q) + (R_{\leq \hat{k}} - R_{reconst,\leq \hat{k}}) && (23) \\ &= (R_{>\hat{k}} + Q) + o(R_{\leq \hat{k}}). && (24) \end{aligned}$$

We note that $\hat{R} - R_{reconst,\leq \hat{k}}$ is the residual after applying a single iteration of the top-$\hat{k}$ atom selection algorithm. After at least $t-1$ applications of the algorithm to the residual matrix of the previous step, the final residual matrix consists of fewer than $\hat{k}$ nonzero atoms, plus Gaussian noise. This satisfies the hypotheses of our theorem statement. Since the sparsity parameter s is assumed to be $\Theta(1)$, the total accumulated error over all steps of the algorithm is $o(R_{\leq \hat{k}})$, which is $o(R)$ in the Frobenius norm. Additional lemma-supporting numerical experiments can be found in the extended version [16].

B Reproducibility

B.1 LRMDS Solution Details

LRMDS has two key steps: atom selection and encoding. To perform atom selection we quantify the alignment of each 2D atom with the current residual via projection. We can compute R's projection as follows:

$$P_{i,j} = \frac{\langle R, B_{i,j} \rangle}{||B_{i,j}||_F}, \tag{25}$$

where $\langle R, B_{i,j} \rangle \triangleq \psi_i^T R \phi_j$ is the alignment, and $||B_{i,j}||_F$ is a normalization based on the Frobenius norm of the 2D atom product. Intuitively atoms of good alignment will be advantageous for encoding the data. Instead of utilizing Eq. 25 in the algorithm for LRMDS we perform the functionally equivalent projection via $P = \hat{\Psi}^T R \hat{\Phi}$ employing the normalized dictionaries $\hat{\Psi}$ and $\hat{\Phi}$. Due to the normalization, the denominator from Eq. 25 can be omitted since:

$$||B_{i,j}||_F = ||\hat{\psi}_i||_2 \cdot ||\hat{\phi}_j||_2 = 1, \forall i, j.$$

This in turn allows to us utilize simply matrix multiplication $\hat{\Psi}^T R \hat{\Phi}$ to obtain alignment scores for atoms.

We then select the top k total atoms from a combination of left or right dictionary atoms with respect to this alignment. To illustrate the methodology, suppose $k = 3$, and the top alignments correspond to $P_{2,3}, P_{3,3}$ in descending order. We would then add the atoms ψ_2, ϕ_3, ψ_3 and in that order to our sub-dictionaries we call $\Psi_s \in \mathcal{R}^{N \times I_s}$ and $\Phi_s \in \mathcal{R}^{M \times J_s}$, where I_s, J_s are the number of selected atoms from the left and the right dictionaries. It is important to note that we only add atoms if they don't already exists in our selected sub-dictionary. Importantly this may result in uneven selection from the dictionaries (i.e $I_s \neq J_s$). This is desirable as there may be significantly more complexity in one of X's modes, necessitating more atoms from the corresponding dictionary for good representation. Intuitively, we let the data guide the selection on both sides. Ties between atoms are resolved arbitrarily.

Once we have sub-selected the dictionary via these chosen atoms we need to solve for the encoding coefficients in Y and W by solving the following:

$$\underset{Y,W}{\text{argmin}} \; ||R - \Psi_s Y W \Phi_s^T||_F^2, \tag{26}$$

To achieve this we iteratively alternate through solving for Y and W while the other is fixed. The updates in each case can be derived by taking the gradients with respect to the non-fixed variable, setting them to 0, and solving. This results in the following update rules:

(1) Given W, the update rule for Y is as follows:

$$\begin{aligned} \Psi_s^\dagger \Psi_s Y W \Phi_s^T (W \Phi_s^T)^\dagger &= \Psi_s^\dagger R (W \Phi_s^T)^\dagger \\ Y &= \Psi_s^\dagger R (W \Phi_s^T)^\dagger, \end{aligned} \tag{27}$$

where † denotes the pseudo-inverse of the corresponding matrix.

(2) Given Y, the update rule for W is:

$$\begin{aligned} (\Psi_s Y)^\dagger \Psi_s Y W \Phi_s^T (\Phi_s^\dagger)^T &= (\Psi_s Y)^\dagger R (\Phi_s^\dagger)^T \\ W &= (\Psi_s Y)^\dagger R (\Phi_s^\dagger)^T \end{aligned} \tag{28}$$

Note that at every iteration, the update rules for the two variables require four pseudo-inversions solved via singular value decomposition with per-iteration complexity of $O(min(mn^2, m^2 n))$, where m, n are the size of the target matrix. The dictionary inversions $\Psi_s^\dagger$ and $\Phi_s^\dagger$ can be computed only once per decomposition as they are fixed with respect to Y and W. Thus, the overall complexity of these steps assuming $N > I_s$, and $M > J_s$ is $O(N I_s^2)$ for $\Psi_s^\dagger$ and $O(M J_s^2)$ for $\Phi_s^\dagger$. We need to compute $(\Psi_s Y)^\dagger$ and $(\Phi_s W)^\dagger$ for every iteration, thus assuming q iterations to convergence the total complexity of the coding step is $O(q(N+M)r^2 + M J_s^2 + N I_s^2)$ assuming the selected decomposition rank is lower than the corresponding data dimensions, i.e., $N > r$ and $M > r$.

We can further optimize the run time based on the assumption that the inversions of both products $(W \Phi_s^T)^\dagger$ and $(\Psi_s Y)^\dagger$ involve matrices of full column (left matrix in the product) and row (right matrix) rank. Then we can separate the inversions and open more opportunities for savings by using the following matrix product inversion rule due to [11]:

$$(AB)^\dagger = B^\dagger A^\dagger \tag{29}$$

Updates from Eq. 27 and Eq. 28 can then be rewritten as:

$$Y = \Psi_s^\dagger X (\Phi_s^\dagger)^T W^\dagger \tag{30}$$

$$W = Y^\dagger \Psi_s^\dagger X (\Phi_s^\dagger)^T, \tag{31}$$

where $\Psi_s^\dagger X (\Phi^\dagger)^T$ is a common term that can be pre-computed outside of the iterative updates. This enables us to only need to compute the pseudo-inversion of Y and W within the inner-loop. $Y^\dagger$ and $W^\dagger$ have complexity $O(r^2 I_s)$ and $O(r^2 J_s)$ respectively. Reducing the overall complexity is to $O(q(I_s + J_s)r^2 + M J_s^2 + N I_s^2)$ which is linear with respect to the size of input matrix. We name this faster LRMDS variation method LRMDS-f. Note that when

Method	Parameters	Range	Synthetic	Convergence	Ablation	Twitch	Road	Wiki	Covid
TGSD	λ_1, λ_2	$[10^{-3}, 10^{-2}, \cdots, 10^{3}]$	Vary	Vary	Vary	Vary	Vary	Vary	Vary
2D-OMP	T_0	$3 - 100\%$ #atoms	3.5%	100%	NA	13%	40%	50%	50%
SC-TGSD	$\frac{\lambda_0}{\lambda_{0max}}, \lambda_1, \lambda_2$	$(0.1 : 0.01 : 0.9)$; $[10^{-3}, 10^{-2}, \cdots, 10^{3}]$	Vary	NA	NA	Vary	Vary	Vary	Vary
LRMDS	k	$[5, 6, 10, 100, 500]$	5	10	6	500	100	100	100
LRMDS-f	k	$[5, 6, 10, 100, 500]$	5	10	6	500	100	100	100

Table 2: Parameters for competing methods where λ_1, λ_2 are sparsity parameters for TGSD; T_0 is the targeting number of coefficients for 2D-OMP; $\frac{\lambda_0}{\lambda_{0max}}$ is the regularizer for SC; k is the number of atoms selected per iteration for LRMDS and LRMDS-f. Some methods are not included in the convergence and ablation experiments and the corresponding cells are marked as NA for Not Applicable. Ranges for tested values are listed in the Range column.

the conditions for Eq. 29 are not met, our encoding will be not as accurate in this variant, but we demonstrate experimentally that this alternative solver offers a good runtime-quality trade-off.

B.2 Datasets Generation and Pre-processing

Synthetic data generation: Unless otherwise noted in specific experiments, the variables in our model for synthetic data are set to the following: Ψ_s corresponds to 20 randomly chosen atoms from a GFT dictionary which itself is generated from a Stochastic Block Model (SBM) graph with 3 blocks of equal size and 1000 total nodes and internal and cross-block edge probabilities set to 0.2 and 0.02 respectively. Φ_s contains 20 randomly selected atoms from a Ramanujan periodic dictionary. The entries of Y and W are set to uniformly random numbers between 0 and 1 and the rank r is set to 3. Finally, ϵ is Gaussian white noise with magnitude ensuring an overall $SNR = 10$ for the signal.

Real-world dataset: The original Twitch dataset specifies active viewers over time and the streams that they are viewing. We create a graph among viewers, where an edge between a pair of viewers exists if they viewed the same stream at least 3 times over a period of 512 hours (which is the temporal dimension of the dataset). The largest connected component of this co-viewing graph involve 78,389 viewers. Each entry of the data matrix $X \in \mathcal{R}^{78389 \times 512}$ from Twitch represents the number of minutes in any given hour that the viewer spent viewing streams on the platform. The Wiki dataset captures hourly number of views of Wikipedia articles for a total of 792 hours. We construct a graph among the articles by placing edges between articles with at least 10 pairwise (clicked by the same IPs) click events within a day. Furthermore, we pick a starting node (the Wikipedia article on China) and construct a breadth-first-search (snowball) subgraph of 1000 nodes around it. We removed an article that was not sufficiently active during the observed period resulting in 999 total nodes. The Covid dataset tracks daily confirmed COVID cases for 3047 counties in the US for 678 days. We use a k-nearest neighbor ($k = 5$) spatial graph connecting counties to their closest neighbors. The Road dataset consists of 1923 highway speed sensors in the LA area, we use the hourly average speed for 30 days as our signal matrix, and the graph is based on connected road segments.

B.3 Hyper-parameter Settings

The parameter settings for all competing techniques unless otherwise specified are as follows. We set the rank for low rank decomposition methods (LRMDS, LRMDS-f, TGSD, SC-TGSD, LRMDS-1D, RAND) to be $r = 3$ in synthetic (equal to the ground truth) and $r = 50$ in real-world datasets. For all real-world datasets, we set the number of atoms per iteration for LRMDS and LRMDS-f to be $k = 100$ for all experiments except for Twitch in which $k = 500$ since this dataset is large and the input dictionaries have in total close to $80,000$ atoms. In both synthetic and real world datasets, we vary SC-TGSD's screening parameter $\frac{\lambda_0}{\lambda_{0max}}$ in the range of 0.1 to 0.9 with a step size of 0.01 to create a set of regimes of selected sub-dictionaries and associated RMSE/running time.

Controlling the number of selected atoms exactly for all competitors is not trivial as TGSD and SC-TGSD employ sparsity regularizers (λ_1, λ_2) that do not offer explicit control over the number of used atoms. In order to facilitate direct but fair comparison we use the ground truth number of selected atoms in synthetic datasets as targets for 2D-OMP and LRMDS, and report results for TGSD and SC-TGSD using sparsity levels resulting in atom "selection" closest to (but exceeding) the ground truth number.

In our ablation study we set $k_1, k_2 = 3$ for both LRMDS-1D and RAND, and $k = 6$ for LRMDS. We do this so that the total number of atoms selected by each method is 6 at each iteration to facilitate fair comparison. We re-run RAND 10 times and report its average performance in terms of RMSE and running time.

Complete details on how parameters were searched (i.e., ranges) and set for each dataset can are listed in Tbl. 2.

We also performed a more detailed analysis of the effect of the number of selected atoms (k) and determined that it controls a trade-off between quality and runtime. Given a total target (optimal but unknown) number of atoms k^*, when $k << k^*$ our algorithm requires more iterations to converge (involving multiple sparse coding fits with increasing dictionaries). On the other hand, a larger k similar to k^* will result in fewer iterations, however, the algorithm may require more than k^* atoms to achieve the same RMSE. For example, if two "good" atoms are similar, then the projections on them will also be similar and high-valued and they will both be selected although potentially redundant. We chose a middle-ground k for our experiments.

C Further experimental analysis and discussion

Additional material is available in the extended version [16] including: (i) extra numerical experiments supporting the lemmas; (ii) baseline solution details; (iii) extra experiments and figures, including synthetic data analysis for varying noise level and dictionary sizes and the corresponding representation quality vs runtime; and (iv) discussion of generalization to multi-way data.