Journal of Machine Learning Research 21 (2020) 1-49 Submitted 5/20; Revised 10/20; Published 11/20

Online matrix factorization for Markovian data and applications to Network Dictionary Learning

Hanbaek Lyu HLYU@MATH.UCLA.EDU
Department of Mathematics
University of California, Los Angeles, CA 90025, USA

Deanna Needell DEANNA@MATH.UCLA.EDU
Department of Mathematics
University of California, Los Angeles, CA 90025, USA

Laura Balzano GIRASOLE@UMICH.EDU
Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI 48109, USA

Editor: Julien Mairal

Abstract

Online Matrix Factorization (OMF) is a fundamental tool for dictionary learning problems, giving an approximate representation of complex data sets in terms of a reduced number of extracted features. Convergence guarantees for most of the OMF algorithms in the literature assume independence between data matrices, and the case of dependent data streams remains largely unexplored. In this paper, we show that a non-convex generalization of the well-known OMF algorithm for i.i.d. stream of data in (Mairal et al., 2010) converges almost surely to the set of critical points of the expected loss function, even when the data matrices are functions of some underlying Markov chain satisfying a mild mixing condition. This allows one to extract features more efficiently from dependent data streams, as there is no need to subsample the data sequence to approximately satisfy the independence assumption. As the main application, by combining online non-negative matrix factorization and a recent MCMC algorithm for sampling motifs from networks, we propose a novel framework of *Network Dictionary Learning*, which extracts "network dictionary patches" from a given network in an online manner that encodes main features of the network. We demonstrate this technique and its application to network denoising problems on real-world network data.

Keywords: Online matrix factorization, convergence analysis, Markovian data, dictionary learning, non-negative matrix factorization, networks

1. Introduction

In modern data analysis, a central step is to find a low-dimensional representation to better understand, compress, or convey the key phenomena captured in the data. Matrix factorization provides a powerful setting for one to describe data in terms of a linear combination of factors or atoms. In this setting, we have a data matrix $X \in \mathbb{R}^{d\times n}$, and we seek a factorization of X into the product WH for $W \in \mathbb{R}^{d\times r}$ and $H \in \mathbb{R}^{r\times n}$. This problem has gone by many names over the decades, each with different constraints: dictionary learning, factor analysis, topic modeling, component analysis. It has applications in text analysis, im-

age reconstruction, medical imaging, bioinformatics, and many other scientific fields more generally (Sitek et al., 2002; Berry and Browne, 2005; Berry et al., 2007; Chen et al., 2011; Taslaman and Nilsson, 2012; Boutchko et al., 2015; Ren et al., 2018).

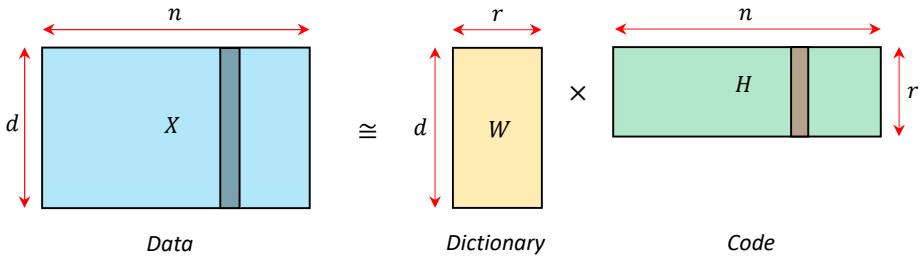
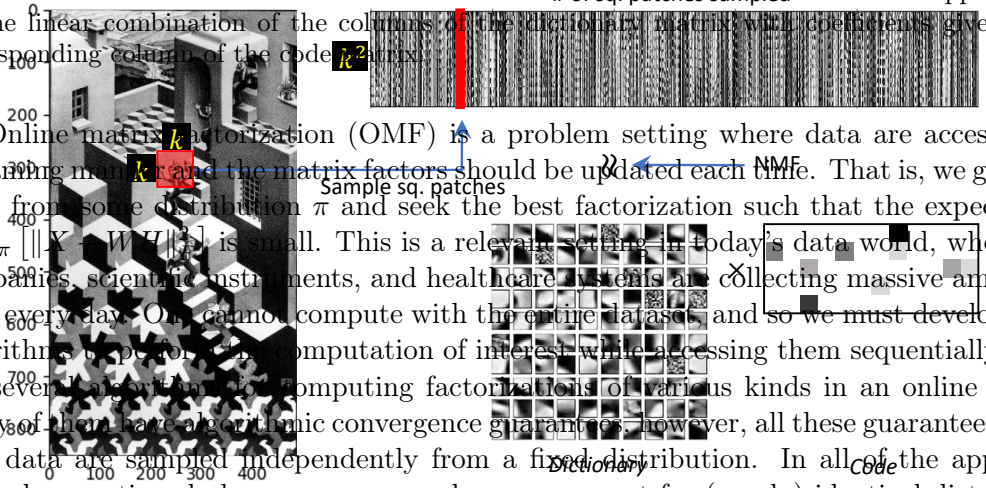
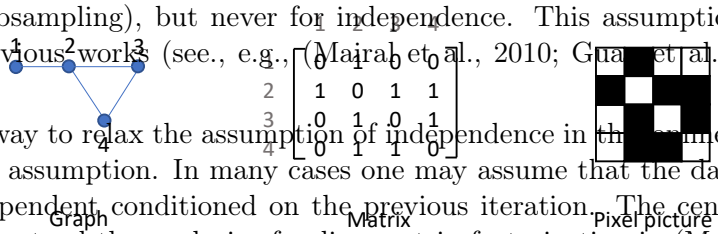


Figure 1: Illustration of matrix factorization. Each column of the data matrix is approximated by the linear combination of the columns of the dictionary matrix with coefficients given by the corresponding column of the code matrix.



Online matrix factorization (OMF) is a problem setting where data are accessed in a streaming manner and the matrix factors should be updated each time. That is, we get draws of X from some distribution π and seek the best factorization such that the expected loss $\mathbb{E}_{X\sim\pi}\left[\|X - WH\|_F^2\right]$ is small. This is a relevant setting in today's data world, where large companies, scientific instruments, and healthcare systems are collecting massive amounts of data every day. One cannot compute with the entire dataset, and so we must develop online algorithms that process one piece of the computation of interest while accessing them sequentially. There are several algorithms for computing factorizations of various kinds in an online context. Many of them have algorithmic convergence guarantees, however, all these guarantees require that data are sampled independently from a fixed distribution. In all of the application examples mentioned above, one may make an argument for (nearly) identical distributions (e.g., using subsampling), but never for independence. This assumption is critical to the analysis of previous works (see., e.g., (Mairal et al., 2010; Gu et al., 2012; Zhao et al., 2017)).



A natural way to relax the assumption of independence in this same context is through the Markovian assumption. In many cases one may assume that the data are not independent, but independent conditioned on the previous iteration. The central contribution of our work is to extend the analysis of online matrix factorization in (Mairal et al., 2010) to the setting where the sequential data form a Markov chain. This is naturally motivated by the fact that the Markov chain Monte Carlo (MCMC) method is one of the most versatile sampling techniques across many disciplines, where one designs a Markov chain exploring the sample space that converges to the target distribution.

As the main application of our result, we propose a novel framework for network data analysis that we call *Network Dictionary Learning* (see Section 6). This allows one to extract "network dictionary atoms" from a given network that capture the most important local subgraph patterns in the network. We also propose a network reconstruction algorithm using the learned network dictionary atoms. A key ingredient is a recent MCMC algorithm for sampling motifs from networks developed by Lyu together with Memoli and Sivakoff (Lyu et al., 2019), which provides a stream of correlated subgraph patterns of the given network. We provide convergence guarantee of our Network Dictionary Learning algorithm

(Corollary 2) as a corollary of our main result, and illustrate our framework through various real-world network data (see Section 6).

1.1 Theoretical contribution

The main result in the present paper, Theorem 1, rigorously establishes convergence of a non-convex generalization (3) of the online matrix factorization scheme in (Mairal et al., 2010; Mairal, 2013b) when the data sequence $(X_t)_{t\in\mathbb{N}}$ is realized as a function $\varphi(Y_t)$ of some underlying Markov chain Y_t (which includes the case that X_t itself forms a Markov chain) with a mild mixing condition. A practical implication of our result is that one can now extract features more efficiently from dependent data streams, as there is no need to subsample the data sequence to approximately satisfy the independence assumption. We illustrate this point through an application to sequences of correlated Ising spin configurations generated by the Gibbs sampler (see Section 5). Our application to the Ising model can easily be generalized to other well-known spin systems such as the cellular Potts model in computational biology (Ouchi et al., 2003; Marée et al., 2007; Szabó and Merks, 2013) and the restricted Boltzmann Machine in machine learning (Nair and Hinton, 2010) (see Section 5).

An important related work is (Mensch et al., 2017), where the authors obtain a perturbative analysis of the original work in (Mairal et al., 2010). In the former, convergence of the OMF algorithm in (Mairal et al., 2010) (see also (3)) has been established when the time-t data matrix X_t conditional on the past data is sampled from an approximate distribution π_t that converges to the true distribution π at an exponential rate (see assumption **(H)** in (Mensch et al., 2017)). While this work provides an important theoretical justification of the use of code approximation in the OMF algorithm in (Mairal et al., 2010) (see H_t in (3)), we emphasize that this result does *not* imply our main result (Theorem 1) in the present work. Indeed, when the data $(X_t)_{t\in\mathbb{N}}$ forms a Markov chain with transition matrix P, we have $\pi_t = P(X_{t-1}, \cdot)$, and this conditional distribution can even be a constant distance away from the stationary distribution π. (For instance, consider the case when X_t alternates between two matrices. Then $\pi = [1/2, 1/2]$ and π_t is either $[1, 0]$ or $[0, 1]$ for all $t \geq 1$.) In fact, this is the main difficulty in analyzing the OMF algorithm (3) for *dependent* data sequences. This is a nontriviality that we address in our current work here.

The proof of our main result (Theorem 1) adopts a number of techniques used in (Mairal et al., 2010; Mairal, 2013b) for the i.i.d. input, but uses a key innovation that handles dependence in the data matrices directly without subsampling, which can potentially be applied to relax independence assumptions for convergence of other online algorithms. The theory of quasi-martingales (Fisk, 1965; Rao, 1969) is a key ingredient in convergence analysis under i.i.d input in (Mairal et al., 2010; Mairal, 2013b) as well as many other related works. Namely, one shows that

$$\sum_{t=0}^{\infty} \left(\mathbb{E}\left[\hat{f}_{t+1}(W_{t+1}) - \hat{f}_t(W_t) \,\middle|\, \mathcal{F}_t \right] \right)^+ < \infty,$$

where $\hat{f}_t$ denotes an associated surrogate loss function, W_t the learned dictionary at time t, and $\mathcal{F}_t$ the filtration of the information up to time t. However, this is not necessarily true without the independence assumption. Our key insight to overcome this issue is that, while the 1-step conditional distribution $P(X_{t-1}, \cdot)$ may be far from the stationary distribution

π, the N-step conditional distribution $P^N(X_{t-N}, \cdot)$ is exponentially close to π under mild conditions. More precisely, we use conditioning on a "distant past" $\mathcal{F}_{t-a_t}$, not on the present $\mathcal{F}_t$, in order to allow the Markov chain to mix close enough to the stationary distribution π for a_t iterations. Then concentration of Markov chains allows us to choose a suitable sequence $1 \leq a_t \leq t$ (see Proposition 9 and Lemma 12), for which we show

$$\sum_{t=0}^{\infty} \mathbb{E}\left[\left(\mathbb{E}\left[\hat{f}_{t+1}(W_{t+1}) - \hat{f}_t(W_t) \,\middle|\, \mathcal{F}_{t-a_t}\right]\right)^+\right] < \infty$$

in the dependent case, from which we derive our main result.

2. Background

In this section, we provide some relevant background and state the main problem and algorithm.

2.1 Topic modeling and matrix factorization

Topic modeling (or *dictionary learning*) aims at extracting important features of a complex dataset so that one can approximately represent the dataset in terms of a reduced number of extracted features (topics) (Blei et al., 2003). Topic models have been shown to efficiently capture latent intrinsic structures of text data in natural language processing tasks (Steyvers and Griffiths, 2007; Blei et al., 2010). One of the advantages of topic modeling based approaches is that the extracted topics are often directly interpretable, as opposed to the arbitrary abstraction of deep neural network based approach.

Matrix factorization is one of the fundamental tools in dictionary learning problems. Given a large data matrix X, can we find some small number of "dictionary vectors" so that we can represent each column of the data matrix as a linear combination of dictionary vectors? More precisely, given a data matrix $X \in \mathbb{R}^{d\times n}$ and sets of admissible factors $\mathcal{C} \subseteq \mathbb{R}^{d\times r}$ and $\mathcal{C}' \subseteq \mathbb{R}^{r\times n}$, we wish to factorize X into the product of $W \in \mathcal{C}$ and $H \in \mathcal{C}'$ by solving the following optimization problem

$$\inf_{W\in\mathcal{C}\subseteq\mathbb{R}^{d\times r},\, H\in\mathcal{C}'\subseteq\mathbb{R}^{r\times n}} \|X - WH\|_F^2, \tag{1}$$

where $\|A\|_F^2 = \sum_{i,j} A_{ij}^2$ denotes the Frobenius norm. Here W is called the *dictionary* and H is the *code* of data X using dictionary W. A solution of such matrix factorization problem is illustrated in Figure 1.

When there are no constraints for the dictionary and code matrices, i.e., $\mathcal{C} = \mathbb{R}^{d\times r}$ and $\mathcal{C}' = \mathbb{R}^{r\times n}$, then the optimization problem (1) is equivalent to *principal component analysis*, which is one of the primary techniques in data compression and dictionary learning. In this case, the optimal dictionary W for X is given by the top r eigenvectors of its covariance matrix, and the corresponding code H is obtained by projecting X onto the subspace generated by these eigenvectors. However, the dictionary vectors found in this way are often hard to interpret. This is in part due to the possible cancellation between them when we take their linear combination, with both positive and negative coefficients.

When the admissible factors are required to be non-negative, the optimization problem (1) is an instance of *Nonnegative matrix factorization* (NMF), which is one of the fundamental tools in dictionary learning problems that provides a parts-based representation of high dimensional data (Lee and Seung, 1999; Lee et al., 2009). Due to the non-negativity constraint, each column of the data matrix is then represented as a non-negative linear combination of dictionary elements (See Figure 1). Hence the dictionaries must be "positive parts" of the columns of the data matrix. When each column consists of a human face image, NMF learns the parts of human face (e.g., eyes, nose, and mouth). This is in contrast to principal component analysis and vector quantization: Due to cancellation between eigenvectors, each "eigenface" does not have to be parts of face (Lee and Seung, 1999).

2.2 Online Matrix Factorization

Many iterative algorithms to find approximate solutions WH to the optimization problem (1), including the well-known Multiplicative Update by Lee and Seung (Lee and Seung, 2001), are based on a block optimization scheme (see (Gillis, 2014) for a survey). Namely, we first compute its representation H_t using the previously learned dictionary W_{t-1}, and then find an improved dictionary W_t (see Figure 2 with setting $X_t \equiv X$).

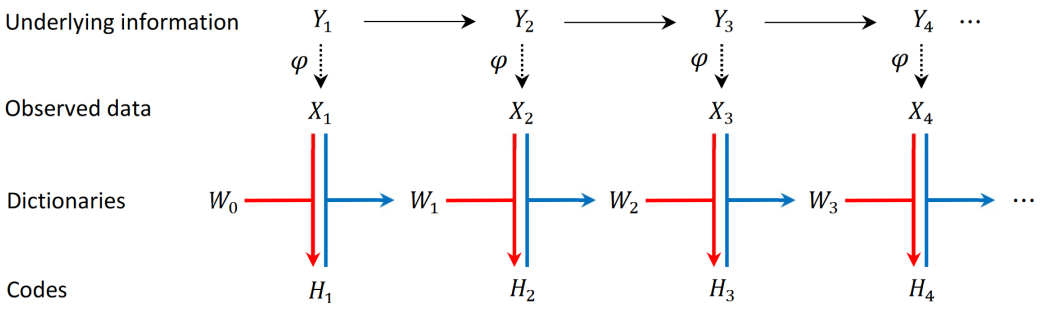


Figure 2: Iterative block scheme of functional OMF. $(Y_t)_{t\in\mathbb{N}}$ is the sequence of underlying information. At each time t, a data matrix X_t of interest is observed from information Y_t via a fixed function φ. Sequences of dictionaries $(W_t)_{t\in\mathbb{N}}$ and codes $(H_t)_{t\in\mathbb{N}}$ are learned from data matrices by a block optimization scheme.

A main example in this setting is when Y_t is a homomorphism $\mathbf{x} : F \to \mathcal{G}$ from a k-node network F with node set $[k] = \{1, 2, \ldots, k\}$ into a n-node network $\mathcal{G}$ with node set V, which is an element of $V^{[k]}$, and X_t is the $k \times k$ matrix that is the adjacency matrix of the k-node subnetwork of $\mathcal{G}$ induced by $\mathbf{x}$. There, one may sample a random homomorphism $\mathbf{x}$ using a Markov chain Monte Carlo (MCMC) algorithm so that $\{Y_t\}$ forms a Markov chain, but the induced $k \times k$ maxtrix sequence X_t may not form a Markov chain (see Section 6).

Despite their popularity in dictionary learning and image processing, one of the drawbacks of these standard iterative algorithms for NMF is that we need to store the data matrix (which is of size $O(dn)$) during the iteration, so they become less practical when there is a memory constraint and yet the size of data matrix is large. Furthermore, in practice only a random sample of the entire dataset is often available, in which case we are not able to apply iterative algorithms that require the entire dataset for each iteration.

The *Online Matrix Factorization* (OMF) problem concerns a similar matrix factorization problem for a *sequence* of input matrices. Here we give a more general and flexible formulation of OMF. Roughly speaking, for each time $t \in \mathbb{N}$, one observes information Y_t, from which a data matrix X_t of interest is extracted as $X_t = \varphi(Y_t)$, for a fixed function φ. We then want to learn sequences of dictionaries $(W_t)_{t\in\mathbb{N}}$ and codes $(H_t)_{t\in\mathbb{N}}$ from the stream $(X_t)_{t\in\mathbb{N}}$ of data matrices.

For a precise formulation, let $(Y_t)_{t\in\mathbb{N}}$ be a discrete-time stochastic process of information taking values in a fixed sample space Ω with unique stationary distribution π. Fix a function $\varphi : \Omega \to \mathbb{R}^{d\times n}$, and define $X_t = \varphi(Y_t)$ for each $t \in \mathbb{N}$. Fix sets of admissible factors $\mathcal{C} \subseteq \mathbb{R}^{d\times r}$ and $\mathcal{C}' \subseteq \mathbb{R}^{r\times n}$ for the dictionaries and codes, respectively. The goal of the functional OMF problem is to construct a sequence $(W_t, H_t)_{t\geq 1}$ of dictionary $W_t \in \mathcal{C} \subseteq \mathbb{R}^{d\times r}$ and codes $H_t \in \mathcal{C}' \subseteq \mathbb{R}^{r\times n}$ such that, almost surely as $t \to \infty$,

$$\|X_t - W_{t-1}H_t\|_F^2 \to \inf_{W\in\mathcal{C},\, H\in\mathcal{C}'} \mathbb{E}_{Y\sim\pi}\left[\|\varphi(Y) - WH\|_F^2\right]. \tag{2}$$

Here and throughout, we write $\mathbb{E}_{Y\sim\pi}$ to denote the expected value with respect to the random variable Y that has the distribution described by π. Thus, we ask that the sequence of dictionary and code pairs provides a factorization error that converges to the best case average error. Since (2) is a non-convex optimization problem, it is reasonable to expect that W_t converges only to a locally optimal solution in general. Convergence guarantees to global optimum is a subject of future work.

We also mention recent related work in the online dictionary learning setting with the following modeling assumption: The time-t data is given by $X_t = W^*H_t$ for some unknown but fixed dictionary W^*, and the code matrix H_t is sampled independently from a distribution concentrated around an unknown code matrix H^*. Under some suitable additional assumptions, a convergence guarantee both for the dictionaries $W_t \to W^*$ and codes $H_t \to H^*$ has been obtained by Rambhatla et al. (2019).

2.3 Algorithm for online matrix factorization

In the literature of OMF, one of the crucial assumptions is that the sequence of data matrices $(X_t)_{t\in\mathbb{N}}$ are drawn independently from a common distribution π (see., e.g., (Mairal et al., 2010; Guan et al., 2012; Zhao et al., 2016)). In this paper, we analyze convergence properties of the following scheme of OMF:

$$\textit{Upon arrival of } X_t\text{:} \qquad \begin{cases} H_t = \arg\min_{H\in\mathcal{C}'\subseteq\mathbb{R}^{r\times n}} \|X_t - W_{t-1}H\|_F^2 + \lambda\|H\|_1 \\ A_t = (1-w_t)A_{t-1} + w_t H_t H_t^T \\ B_t = (1-w_t)B_{t-1} + w_t H_t X_t^T \\ W_t = \arg\min_{W\in\mathcal{C}\subseteq\mathbb{R}^{d\times r}} \left(\mathrm{tr}(WA_tW^T) - 2\mathrm{tr}(WB_t)\right) \\ \qquad \text{s.t. } \mathrm{tr}((B_t^T - WA_t)(W_{t-1} - W)^T) \leq 0 \end{cases}, \tag{3}$$

where $(w_t)_{t\geq 1}$ is a prescribed sequence of weights, and A_0 and B_0 are zero matrices of size $r \times r$ and $r \times d$, respectively. Note that the L_2-loss function is augmented with the ℓ_1-regularization term $\lambda\|H\|_1$ with regularization parameter $\lambda \geq 0$, which forces the code H_t to be sparse. See Appendix B for a more detailed algorithm implementing (3).

In the above scheme, the auxiliary matrices $A_t \in \mathbb{R}^{r\times r}$ and $B_t \in \mathbb{R}^{r\times d}$ effectively aggregate the history of data matrices $X_1, \ldots, X_t$ and their best codes $H_1, \ldots, H_t$. The previous dictionary W_{t-1} is updated to W_t, which minimizes a quadratic loss function $\mathrm{tr}(WA_tW^T) - 2\mathrm{tr}(WB_t)$ in the (not necessarily convex) constraint set $\mathcal{C} \subseteq \mathbb{R}^{d\times r}$ subject to the additional ellipsoidal constraint $\mathrm{tr}((B_t^T - WA_t)(W_{t-1} - W)^T) \leq 0$. This extra condition means that W_t should lie inside an ellipsoid with an axis between the previous iterate W_{t-1} and the global minimum $A_t^{-1}B_t^T$ of the unconstrained quadratic function (see Figure 4 for illustration). We present an algorithm that solves this quadratic problem when $\mathcal{C}$ is the disjoint union of convex sets (see Algorithm 3).

When $\mathcal{C}$ is convex and $w_t = 1/t$ for all $t \geq 1$, the ellipsoidal condition for W_t becomes redundant and (3) reduces to the classical algorithm of OMF in the celebrated work by Mairal et al. (Mairal et al., 2010). Assuming that X_t's are independently drawn from the stationary distribution π, with additional mild assumption, the authors of Mairal et al. (2010) proved that the sequence $(W_t)_{t\in\mathbb{N}}$ converges to a critical point of the expected loss function in (2) augmented with the ℓ_1-regularization term $\lambda\|H\|_1$. Later, Mairal generalized a similar convergence result under independence assumption for a broader class of non-convex objective functions (Mairal, 2013b).

2.4 Preliminary example of dictionary learning from dependent data samples from images

Here we give a preliminary example of dictionary learning from dependent data samples from images. One of the well-known applications of NMF is for learning dictionary patches from images and image reconstruction. For a standard NMF application, we first choose an appropriate patch size $k \geq 1$ and extract all $k \times k$ image patches from a given image. In terms of matrices, this is to consider the set of all $(k \times k)$ submatrices of the image with consecutive rows and columns. If there are N such image patches, we are forming $(k^2 \times N)$ patch matrix to which we apply NMF to extract dictionary patches. It is reasonable to believe that there are some fundamental features in the space of all image patches since nearby pixels in the image are likely to be spatially correlated.

A computationally more efficient way of learning dictionary patches, especially from large images, is to use online NMF algorithms to minibatches of patches sampled randomly from the image. It is easy to sample $k\times k$ patches independently and uniformly from a given image, hence we can generate i.i.d. minibatches of a fixed number of patches. On the other hand, we can also generate a dependent sample of patches by using simple symmetric random walk on the image, meaning that the next patch is chosen from the four single-pixel shifts of the previous one with equal probability (using periodic boundary condition). A toy example for this application of online NMF for these two different sampling methods is shown in Figure 3. However, we report that the set of learned dictionaries for the random walk sampling method does appear more localized when for less iterations, which is reasonable since then the random walk may only explore a restricted portion of the entire image. See Section 5 for more details about applications on dictionary learning from MCMC trajectories.

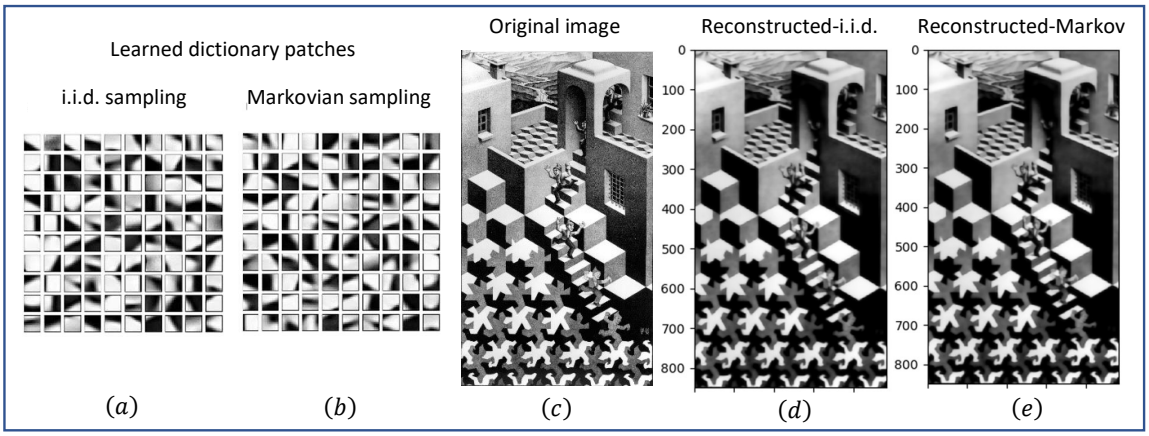
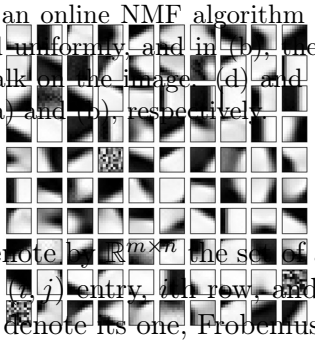


Figure 3: Learning 100 dictionary patches of size 10 from M.C. Escher's *Cycle* (1938) shown in (c) by online NMF and two different sampling methods. Randomly sampled minibatches of 1000 patches of size 10 are fed into an online NMF algorithm for 500 iterations, where in (a), patches are sampled independently and uniformly, and in (b), the top left corner of the patches performs a simple symmetric random walk on the image. (d) and (e) show reconstructed images using the learned dictionary patches in (a) and (b), respectively.



2.5 Notation

Fix integers $m, n \geq 1$. We denote by $\mathbb{R}^{m\times n}$ the set of all $m \times n$ matrices of real entries. For any matrix A, we denote its (i, j) entry, ith row, and jth column by A_{ij}, $[A]_{i\bullet}$, and $[A]_{\bullet j}$. For each $A = (A_{ij}) \in \mathbb{R}^{m\times n}$, denote its one, Frobenius and operator norms by $\|A\|_1$, $\|A\|_F$, and $\|A\|_{\mathrm{op}}$, respectively, where

$$\|A\|_1 = \sum_{ij} |a_{ij}|, \quad \|A\|_F^2 = \sum_{ij} a_{ij}^2, \quad \|A\|_{\mathrm{op}} = \inf\{c > 0 : \|Ax\|_F \leq c\|x\|_F \text{ for all } x \in \mathbb{R}^n\}.$$

For any subset $\mathcal{A} \subset \mathbb{R}^{m\times n}$ and $X \in \mathbb{R}^{n\times m}$, denote

$$R(\mathcal{A}) = \sup_{X\in\mathcal{A}} \|X\|_F, \qquad d_F(X, \mathcal{A}) = \inf_{Y\in\mathcal{A}} \|X - Y\|_F. \tag{4}$$

For any continuous functional $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ and a subset $A \subseteq \mathbb{R}^N$, we denote

$$\arg\min_{x\in A} f = \left\{ x \in A \,\middle|\, f(x) = \inf_{y\in A} f(y) \right\}.$$

When $\arg\min_{x\in A} f$ is a singleton $\{x^*\}$, we identify $\arg\min_{x\in A} f$ as x^*.

For any event A, we let $\mathbf{1}_A$ denote the indicator function of A, where $\mathbf{1}_A(\omega) = 1$ if $\omega \in A$ and 0 otherwise. We also denote $\mathbf{1}_A = \mathbf{1}(A)$ when convenient. For each $x \in \mathbb{R}$, denote $x^+ = \max(0, x)$ and $x^- = \max(0, -x)$. Note that $x = x^+ - x^-$ for all $x \in \mathbb{R}$ and the functions $x \mapsto x^\pm$ are convex.

Let $\mathbb{N} = \{0, 1, 2, \dots\}$ denote the set of nonnegative integers. For each integer $n \geq 1$, denote $[n] = \{1, 2, \dots, n\}$. A *simple graph* $G = ([n], A_G)$ is a pair of its *node set* $[n]$ and its

adjacency matrix A_G, where A_G is a symmetric 0-1 matrix with zero diagonal entries. We say nodes i and j are *adjacent* in G if $A_G(i,j) = 1$.

3. Preliminary discussions

3.1 Markov chains on countable state space

We note that from here on, Markov chains will be denoted as Y_t and we reserve X_t to denote the data. We first give a brief summary on Markov chains. (see, e.g., (Levin and Peres, 2017)). Fix a countable set Ω. A function $P : \Omega^2 \to [0, \infty)$ is called a *Markov transition matrix* if every row of P sums to 1. A sequence of Ω-valued random variables $(Y_t)_{t\in\mathbb{N}}$ is called a *Markov chain* with transition matrix P if for all $y_0, y_1, \ldots, y_n \in \Omega$,

$$\mathbb{P}(Y_n = y_n \,|\, Y_{n-1} = y_{n-1}, \ldots, Y_0 = y_0) = \mathbb{P}(Y_n = y_n \,|\, Y_{n-1} = y_{n-1}) = P(Y_{n-1}, y_n). \quad (5)$$

We say that a probability distribution π on Ω is a *stationary distribution* for the chain $(Y_t)_{t\in\mathbb{N}}$ if $\pi = \pi P$, that is,

$$\pi(x) = \sum_{y\in\Omega} \pi(y)P(y, x).$$

We say the chain $(Y_t)_{t\in\mathbb{N}}$ is *irreducible* if for any two states $x, y \in \Omega$ there exists an integer $t \in \mathbb{N}$ such that $P^t(x, y) > 0$. For each state $x \in \Omega$, let $\mathcal{T}(x) = \{t \geq 1 \,|\, P^t(x, x) > 0\}$ be the set of times when it is possible for the chain to return to starting state x. We define the *period* of x by the greatest common divisor of $\mathcal{T}(x)$. We say the chain Y_t is *aperiodic* if all states have period 1. Furthermore, the chain is said to be *positive recurrent* if there exists a state $x \in \Omega$ such that the expected return time of the chain to x started from x is finite. Then an irreducible and aperiodic Markov chain has a unique stationary distribution if and only if it is positive recurrent (Levin and Peres, 2017, Thm 21.21).

Given two probability distributions μ and ν on Ω, we define their *total variation distance* by

$$\|\mu - \nu\|_{TV} = \sup_{A\subseteq\Omega} |\mu(A) - \nu(A)|.$$

If a Markov chain $(Y_t)_{t\in\mathbb{N}}$ with transition matrix P starts at $y_0 \in \Omega$, then by (5), the distribution of Y_t is given by $P^t(y_0, \cdot)$. If the chain is irreducible and aperiodic with stationary distribution π, then the convergence theorem (see, e.g., (Levin and Peres, 2017, Thm 21.14)) asserts that the distribution of Y_t converges to π in total variation distance: As $t \to \infty$,

$$\sup_{y_0\in\Omega} \|P^t(y_0, \cdot) - \pi\|_{TV} \to 0. \quad (6)$$

See (Meyn and Tweedie, 2012, Thm 13.3.3) for a similar convergence result for the general state space chains. When Ω is finite, then the above convergence is exponential in t (see., e.g., (Levin and Peres, 2017, Thm 4.9))). Namely, there exists constants $\xi \in (0, 1)$ and $C > 0$ such that for all $t \in \mathbb{N}$,

$$\max_{y_0\in\Omega} \|P^t(y_0, \cdot) - \pi\|_{TV} \leq C\xi^t. \quad (7)$$

Markov chain mixing refers to the fact that, when the above convergence theorems hold, then one can approximate the distribution of Y_t by the stationary distribution π.

3.2 Empirical Risk Minimization for OMF

Define the following quadratic loss function of the dictionary $W \in \mathbb{R}^{d\times r}$ with respect to data $X \in \mathbb{R}^{d\times n}$

$$\ell(X, W) = \inf_{H\in\mathcal{C}'\subseteq\mathbb{R}^{r\times n}} \|X - WH\|_F^2 + \lambda\|H\|_1, \tag{8}$$

where $\mathcal{C}'$ denotes the set of admissible codes and $\lambda > 0$ is a fixed ℓ_1-regularization parameter. For each $W \in \mathcal{C}$ define its *expected loss* by

$$f(W) = \mathbb{E}_{Y\sim\pi}[\ell(\varphi(Y), W)]. \tag{9}$$

Suppose arbitrary sequences of data matrices $(X_t)_{t\in\mathbb{N}}$ is given. Fix a non-increasing sequence of weights $(w_t)_{t\in\mathbb{N}}$ in $(0, 1)$. Define the *(weighted) empirical loss* $f_t(W)$ recursively as

$$f_t(W) = (1 - w_t)f_{t-1}(W) + w_t\ell(X_t, W), \qquad t \geq 1, W \in \mathcal{C}, \tag{10}$$

where we take $f_0 \equiv 0$. Note that when we take "balanced weights" $w_t = 1/t$ for all $t \geq 1$, then the weighted empirical loss function takes the usual form $f_t(W) = t^{-1}\sum_{s=1}^{t}\ell(X_s, W)$, where all losses are counted evenly. For $w_t \gg 1/t$ (e.g., $w_t = t^{-3/4}$), we take the recent losses more importantly than the past ones.

Suppose the sequence of data matrices $(X_t)_{t\geq1}$ itself is an irreducible Markov chain on Ω with unique stationary distribution π. Note that for the balanced weights $w_t = 1/t$, by the Markov chain ergodic theorem (see, e.g., (Durrett, 2010, Thm 6.2.1, Ex. 6.2.4) or (Meyn and Tweedie, 2012, Thm. 17.1.7)), for each dictionary W, the empirical loss converges almost surely to the expected loss:

$$\lim_{t\to\infty} f_t(W) = f(W) \quad \text{a.s.}$$

In fact, this almost sure convergence holds for the weighted case uniformly in W varying in compact $\mathcal{C}$ (see Lemma 11). This observation and the block optimization scheme in Subsection 2.2 suggests the following scheme for our functional OMF proglem:

$$\text{Upon arrival of } X_t\text{:} \qquad \begin{cases} H_t = \arg\min_{H\in\mathcal{C}'}\|X_t - W_{t-1}H\|_F^2 + \lambda\|H\|_1 \\ W_t = \arg\min_{W\in\mathcal{C}} f_t(W). \end{cases} \tag{11}$$

Finding H_t in (11) can be done using a number of known algorithms (e.g., LARS (Efron et al., 2004), LASSO (Tibshirani, 1996), and feature-sign search (Lee et al., 2007)) in this formulation. However, there are some important issues in solving the optimization problem for W_t in (11). Note that minimizing empirical loss to find W_t as above is an example of *empirical risk minimization* (ERM), which is a classical problem in statistical learning theory (Vapnik, 1992). For i.i.d., data points, recent advances guarantees that solutions of ERM even for a class of non-convex loss functions converges to the set of local minima of the expected loss function (Mei et al., 2018). However, such convergence guarantee is not known for dependent data points, and there some important computational shortcomings in the above ERM for our OMF problem. Namely, in order to compute the empirical loss $f_t(W)$, we may have to store the entire history of data matrices $X_1, \ldots, X_t$, and we need to solve t instances of optimization problem (8) for each summand of $f_t(W)$. Both of these are a significant requirement for memory and computation. These issues are addressed in the OMF scheme (3), as we discuss in the following subsection.

3.3 Asymptotic solution minimizing surrogate loss function

The idea behind the OMF scheme (3) is to solve the following approximate problem

$$\textit{Upon arrival of } X_t: \qquad \begin{cases} H_t = \arg\min_{H\in\mathcal{C}'} \|X_t - W_{t-1}H\|_F^2 + \lambda\|H\|_1 \\ W_t = \arg\min_{W\in\mathcal{C}} \hat{f}_t(W) \end{cases} \tag{12}$$

with a given initial dictionary $W_0 \in \mathcal{C}$, where $\hat{f}_t(W)$ is an upper bounding surrogate for $f_t(W)$ defined recursively by

$$\hat{f}_t(W) = (1-w_t)\hat{f}_{t-1}(W) + w_t\left(\|X_t - WH_t\|_F^2 + \lambda\|H_t\|_1\right)$$

with $\hat{f}_0 \equiv 0$. Namely, we recycle the previously found codes $H_1, \ldots, H_t$ and use them as approximate solutions of the sub-problem (8). Hence, there is only a single optimization for W_t in the relaxed problem (12).

It seems that this might still require storing the entire history $X_1, X_2, \ldots, X_t$ of data matrices up to time t. But in fact we only need to store two summary matrices $A_t \in \mathbb{R}^{r\times r}$ and $B_t \in \mathbb{R}^{r\times d}$. Indeed, (12) is equivalent to the optimization problem (3) stated in the introduction. To see this, note that

$$\begin{aligned} \|X - WH\|_F^2 &= \operatorname{tr}\left((X-WH)(X-WH)^T\right) \\ &= \operatorname{tr}(WHH^TW^T) - 2\operatorname{tr}(WHX^T) + \operatorname{tr}(XX^T). \end{aligned}$$

Hence if we let A_t and B_t be recursively defined as in (3), then we can write

$$\hat{f}_t(W) = \operatorname{tr}(WA_tW^T) - 2\operatorname{tr}(WB_t) + r_t, \tag{13}$$

where r_t does not depend on W. This explains the quadratic objective function for W_t in (3).

4. Statement of main results

4.1 Setup and assumptions

Fix integers $d, n, r \geq 1$ and a constant $\lambda > 0$. Here we list all technical assumptions required for our convergence results to hold.

(A1). *The observed data matrices X_t are given by $X_t = \varphi(Y_t)$, where Y_t are drawn from a countable sample space Ω (hence Ω is measurable with respect to the counting measure), and $\varphi : \Omega \to \mathbb{R}^{d\times n}$ is a bounded function.*

(A2). *Dictionaries W_t are constrained to the subset $\mathcal{C} \subseteq \mathbb{R}^{d\times r}$, which is the disjoint union of compact and convex sets $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m$ in $\mathbb{R}^{d\times r}$.*

(M1). *The sequence of information $(Y_t)_{t\in\mathbb{N}}$ is an irreducible, aperiodic, and positive recurrent Markov with state space Ω. We let P and π denote the transition matrix and unique stationary distribution of the chain $(Y_t)_{t\in\mathbb{N}}$, respectively.*

(M2). *There exists a sequence* $(a_t)_{t\in\mathbb{N}}$ *of non-decreasing integers such that*

$$0 \leq a_t < t, \quad \sum_{t=1}^{\infty} w_{t-a_t}^2 \sqrt{t} < \infty, \quad \sum_{t=1}^{\infty} w_t^2 a_t < \infty, \quad \sum_{t=1}^{\infty} w_t \sup_{\mathbf{y}\in\Omega} \|P^{a_t+1}(\mathbf{y},\cdot) - \pi\|_{TV} < \infty.$$

(C1). *The loss and expected loss functions* ℓ *and* f *defined in* (8) *and* (9) *are continuously differentiable and have Lipschitz gradient.*

(C2). *The eigenvalues of the positive semidefinite matrix* A_t *defined in* (3) *are at least some constant* $\kappa_1 > 0$ *for all sufficiently large* $t \in \mathbb{N}$.

It is standard to assume compact support for data matrices as well as dictionaries, which we do for as well in (A1) and (A2). We remark that our analysis and main results still hold in the general state space case, but this requires a more technical notion of the positive Harris chains irreducibility assumption in order to use the functional central limit theorem for general state space Markov chains (Meyn and Tweedie, 2012, Thm. 17.4.4). We restrict our attention to the countable state space Markov chains in this paper.

The motivation behind assumptions (A1) and (M1) is the following. If the sample space Ω as well as the desired distribution π are complicated, then one may use a Markov chain Monte Carlo (MCMC) algorithm to sample information according to π, which will then be processed to form a meaningful data matrix. For the MCMC sampling, one designs a Markov chain on Ω that has π as a stationary distribution, and then show that the chain is irreducible, aperiodic, and positive recurrent. Then by the general Markov chain theory we have summarized in Subsection 3.1, π is the unique stationary distribution of the chain.

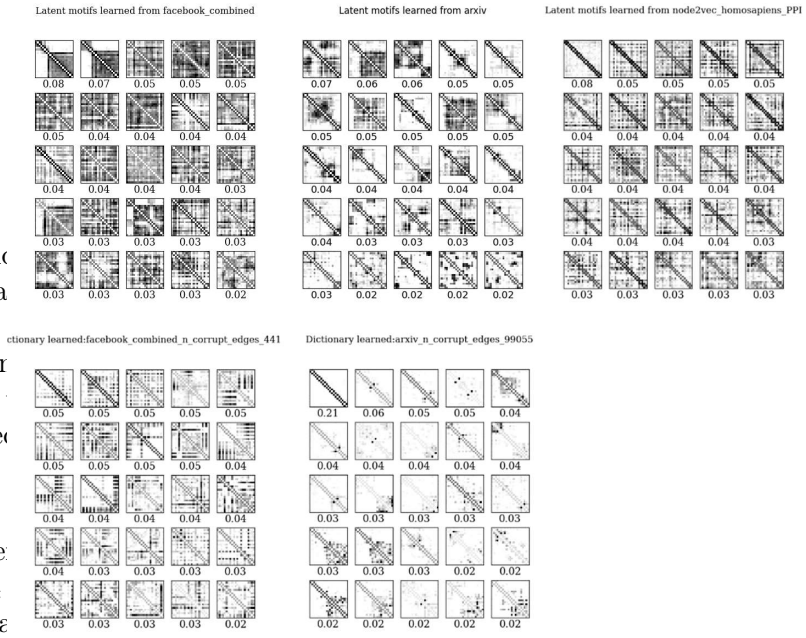
On the other hand, (M2) is a very weak assumption on the rate of convergence of the Markov chain $(Y_t)_{t\in\mathbb{N}}$ to its stationary distribution π. Note that (M2) follows from

(M2)'. *There exist constants* $\beta \in (3/4, 1]$ *and* $\gamma > 2(1-\beta)$ *such that*

$$w_t = O(t^{-\beta}), \qquad \sup_{\mathbf{y}\in\Omega} \|P^t(\mathbf{y},\cdot) - \pi\|_{TV} = O(t^{-\gamma}).$$

Indeed, it is easy to verify that (M2)' with $a_t = \lfloor \sqrt{t} \rfloor$ implies (M2). Furthermore, the mixing condition in (M2)' is automatically satisfied when Ω is finite, which in fact covers many practical situations. Indeed, assuming (M1) and that Ω is finite, the convergence theorem (7) provides an exponential rate of convergence of the empirical distribution of the chain to π, in particular implying the polynomial rate of convergence in (M2)'.

Next, we comment on (A2). Our analysis holds as long as we can solve the quadratic minimization problem for W_t under the constraint $\mathcal{C}$ intersected with the ellipsoid $\mathcal{E}_t := \mathrm{tr}((B_t^T - WA_t)(W_{t-1} - W)^T) \leq 0$ with Hessian matrix A_t (see (3)). A particular instance of interest is when $\mathcal{C}$ is the disjoint union of convex constraint sets $\mathcal{C}_i$ as in (A2). By definition A_t is positive semidefinite, so each $\mathcal{C}_i \cap \mathcal{E}_t$ is convex. Hence under (A2), we can solve W_t in (3) by solving the convex sub-problems on each $\mathcal{C}_i \cap \mathcal{E}_t$ (see Algorithm 3 for details, and also Figure 4 right). This setting will be particularly useful for dictionary learning for multi-cluster data set, where it is desirable to find a dictionary that lies in one of multiple convex hulls of representative elements in each cluster (Peng et al., 2019). In the special case when $\mathcal{C}$ is convex, the additional ellipsoidal condition becomes redundant (see Proposition 6 (iii)



anc ption (A2) reduce to the
sta

he additional ellipsoidal
cor ase. A crucial ingredient
in is the following so-called
'se

$\geq 1.$ (14)

He $2\mathrm{tr}(WB_t)$, where A_t and
B_t it should be guaranteed
tha of the dictionary matrix
$W_t - W_{t-1}$. However, when $\mathcal{C}$ is non-convex, it is possible that one may pay a large change from W_{t-1} to W_t and still do not gain any meaningful improvement in minimizing g_t (see Figure 4, middle). The ellipsoidal constraint $\mathcal{E}_t$ ensures the second-order growth property (14) when we optimize g_t on $\mathcal{C} \cap \mathcal{E}_t$ (see Proposition 6 (i)).

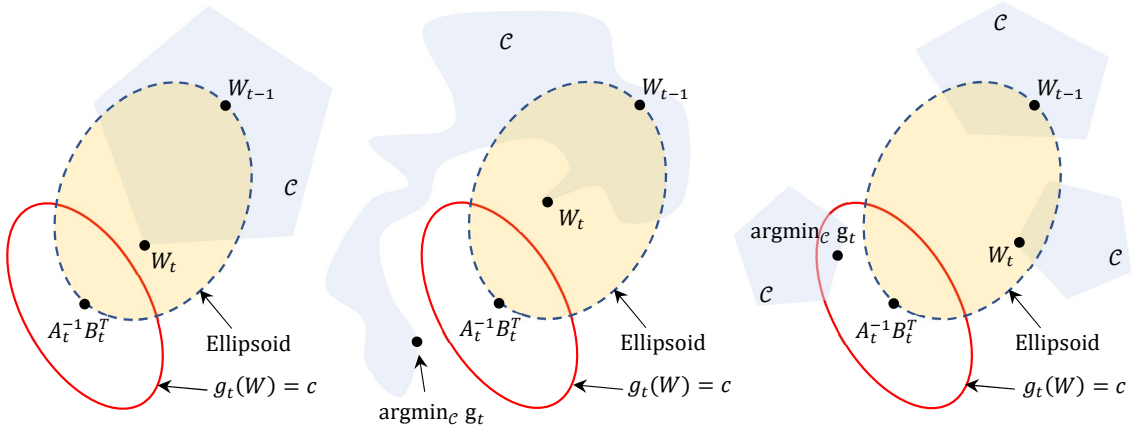


Figure 4: Illustration of the constraint quadratic minimization problem for the dictionary update step in the main algorithm (3). There, the updated dictionary matrix W_t is found by minimizing the quadratic function $g_t(W) = \mathrm{tr}(WA_tW^T) - 2\mathrm{tr}(WB_t)$ (global minimum at $A_t^{-1}B_t^T$), where A_t and B_t are recursively computed by the algorithm (3), over the constraint set $\mathcal{C}$ intersected with the ellipsoid $\mathcal{E}_t := \mathrm{tr}((B_t^T - WA_t)(W_{t-1} - W)^T) \leq 0$. (Left) If $\mathcal{C}$ is convex, the additional ellipsoidal constraint becomes redundant. (Middle) If $\mathcal{C}$ is non-convex, the second-order growth condition (14) may be violated by the minimizer of g_t in $\mathcal{C}$ but is still guaranteed with the ellipsoidal condition. (Right) When $\mathcal{C}$ is the disjoint union of convex sets, then the constraint quadratic problem on $\mathcal{C} \cap \mathcal{E}_t$ can be exactly solved.

Our main result, Theorem 1, guarantees that both the empirical and surrogate loss processes $(f_t(W_t))$ and $(\hat{f}_t(W_t))$ converge almost surely under the assumptions (A1)-(A2) and (M1)-(M2). The assumptions (C1)-(C2), which are also used in (Mairal et al., 2010), are sufficient to ensure that the limit point is a critical point of the expected loss function f defined in (9).

We remark that (C1) follows from the following alternative condition (see (Mairal et al., 2010, Prop. 1)):

(C1)'. *For each $X \in \Omega$ and $W \in \mathcal{C}$, the sparse coding problem in* (8) *has a unique solution.*

In order to enforce (C1)', we may use the elastic net penalization by Zou and Hastie (Zou and Hastie, 2005). Namely, we may replace the first equation in (3) by

$$H_t = \underset{H \in \mathcal{C}' \subseteq \mathbb{R}^{r \times n}}{\arg\min} \|X_t - W_{t-1}H\|_F^2 + \lambda \|H\|_1 + \frac{\kappa_2}{2}\|H\|_F^2 \tag{15}$$

for some fixed constant $\kappa_2 > 0$. See the discussion in (Mairal et al., 2010, Subsection 4.1) for more details.

On the other hand, (C2) guarantees that the eigenvalues of A_t produced by (3) are lower bounded by the constant $\kappa_1 > 0$. It follows that A_t is invertible and $\hat{f}_t$ is strictly convex with Hessian $2A_t$. This is crucial in deriving Proposition 7, which is later used in the proof of Theorem 1. Note that (C2) can be enforced by replacing the last equation in (3) with

$$\begin{aligned} W_t = \underset{W \in \mathcal{C} \subseteq \mathbb{R}^{d \times r}}{\arg\min} & \left(\operatorname{tr}\left(W(A_t + \kappa_1 I)W^T\right) - 2\operatorname{tr}(WB_t)\right) \\ \text{s.t. } & \operatorname{tr}((B_t^T - WA_t)(W_{t-1} - W)^T) \leq 0. \end{aligned} \tag{16}$$

The same analysis for the algorithm (3) that we will develop in the later sections will apply for the modified version with (15) and (16), for which (C1)-(C2) are trivially satisfied.

4.2 Statement of main results

Our main result in this paper, which is stated below in Theorem 1, asserts that under the OMF scheme (3), the induced stochastic processes $(f_t(W_t))_{t\in\mathbb{N}}$ and $(\hat{f}_t(W_t))_{t\in\mathbb{N}}$ converge as $t \to \infty$ in expectation. Furthermore, the sequence $(W_t)_{t\in\mathbb{N}}$ of learned dictionaries converge to the set of critical points of the expected loss function f.

Theorem 1. *Suppose (A1)-(A2) and (M1)-(M2). Let $(W_t, H_t)_{t\geq 1}$ be a solution to the optimization problem* (3)*. Further assume that $\sum_{t=1}^{\infty} w_t = \infty$. Then the following hold.*

(i) $\lim_{t\to\infty} \mathbb{E}[f_t(W_t)] = \lim_{t\to\infty} \mathbb{E}[\hat{f}_t(W_t)] < \infty$.

(ii) $f_t(W_t) - \hat{f}_t(W_t) \to 0$ *and* $f(W_t) - \hat{f}_t(W_t) \to 0$ *as* $t \to \infty$ *almost surely.*

(iii) *Further assume (C1)-(C2). Then almost surely,*

$$\lim_{t\to\infty} \|\nabla_W f(W_t) - 2(W_tA_t - B_t)\|_F = 0.$$

Furthermore, the distance between W_t and the set of all local extrema of f in $\mathcal{C}$ converges to zero almost surely.

The second part of Theorem 1 (ii) is a new result based on the first part and a uniform convergence result in Lemma 11. This implies that for large t, the true objective function $f(W_t)$, which requires averaging over random data matrices sampled from the stationary distribution π, can be approximated by the easily computed surrogate objective $\hat{f}_t(W_t)$. If we further assume that the global minimizer of the quadratic function $g(W) = \operatorname{tr}(WA_tW^T) - 2\operatorname{tr}(WB_t)$ is in the interior of the constraint set $\mathcal{C}$, then $\nabla_W g(W_t) = 2(W_tA_t - B_t) \equiv 0$, so

Theorem 1 (iii) yields $\|\nabla_W f(W_t)\|_F \to 0$ almost surely as $t \to \infty$. We also remark that in the special case of convex constraint set $\mathcal{C}$ for dictionaries, i.i.d. data matrices $(X_t)_{t\in\mathbb{N}}$, and balanced weights $w_t = 1/t$, our results above recover the classical results in (Mairal et al., 2010, Prop. 2 and 3). For general weighting scheme and objective functions, similar results were obtained (Mairal, 2013b) using similar proof techniques.

As discussed in Subsection 1.1, the core of our proof of Theorem 1 is to use conditioning on distant past in order to allow the Markov chain to mix close enough to the stationary distribution π. This allows us to control the difference between the new and the average losses by concentration of Markov chains (see Proposition 9 and Lemma 12), overcoming the limitation of the quasi-martingale based approach typically used for i.i.d. input (Mairal et al., 2010; Mairal, 2013a,b). A practical implication of the theoretical result is that one can now extract features more efficiently from the same dependent data streams, as there is no need to subsample the data sequence to approximately satisfy the independence assumption.

It is worth comparing our approach that directly handles Markovian dependence to the popular approach using subsampling. Namely, if we keep only one Markov chain sample in every τ iterations (subsampling epoch) then the remaining samples are asymptotically independent provided the epoch τ is long enough compared to the mixing time of the Markov chain. A similar line of approach was used in (Yang et al., 2019) for a relevant but different problem of factorizing the unknown transition matrix of a Markov chain by observing its trajectory. However, there are a number of shortcomings in the approach based on subsampling. First, consecutive samples obtained after subsampling are nearly independent, but never completely independent. Hence subsampling dependent sequences does not rigorously verify independence assumption. Second, subsampling makes use of only a small portion of the already obtained samples, which may be not be of the most efficient use of given data samples. Our approach directly handles Markovian dependence in data samples and hence does not suffer from these shortcomings. See Section 5 for a numerical verification of this claim.

5. Application I: Learning features from MCMC trajectories

In this section, we demonstrate learning features from a single MCMC trajectory of dependent samples in the case of two-dimensional *Ising model*, which was first introduced by Lenz in 1920 for modeling ferromagnetism (Lenz, 1920) and has been one of the most well-known spin systems in the physics literature. Our analysis for the Ising model could easily be generalized to the other well-known spin systems such as the Potts model (Wu, 1982), the cellular Potts model (Ouchi et al., 2003; Marée et al., 2007; Szabó and Merks, 2013), or the restricted Boltzmann Machine (Nair and Hinton, 2010).

5.1 Spin systems and the Ising model

Consider a general system of binary spins. Namely, let $G = (V, E)$ be a simple graph with vertex set V and edge set E. Imagine each vertex (site) of G can take either of the two states (spins) $+1$ or -1. A complete assignment of spins for each site in G is given by a *spin configuration*, which we denote by a map $\mathbf{x} : V \to \{-1, 1\}$. Let $\Omega = \{-1, 1\}^V$ denote the set of all spin configurations. In order to introduce a probability measure on Ω, fix a

function $H(\cdot;\theta):\Omega\to\mathbb{R}$ parameterized by θ, which is called a *Hamiltonian* of the system. For each choice of parameter θ, we define a probability distribution π_θ on the set Ω of all spin configurations by

$$\pi_\theta(\mathbf{x}) = \frac{1}{Z_\theta}\exp\left(-H(\mathbf{x};\theta)\right), \tag{17}$$

where the partition function Z_θ is defined by $Z_\theta = \sum_{\mathbf{x}\in\Omega}\exp\left(-H(\mathbf{x};\theta)\right)$. The induced probability measure $\mathbb{P}_\theta$ on $\{-1,1\}^V$ is called a *Gibbs measure*.

The Ising model is defined by the following Hamiltonian

$$H(\mathbf{x};T,h) = \frac{1}{T}\left(-\sum_{\{u,v\}\in E}\mathbf{x}(u)\mathbf{x}(v) - \sum_{v\in V}h(v)\mathbf{x}(v)\right),$$

where $\mathbf{x}$ is the spin configuration, the parameter T is called the *temperature*, and $h:V\to\mathbb{R}$ the *external field*. In this paper we will only consider the case of zero external field. Note that, with respect to the corresponding Gibbs measure, a given spin configuration $\mathbf{x}$ has higher probability if the adjacent spins tend to agree, and this effect of adjacent spin agreement is emphasized (resp., diminished) for low (resp., high) temperature T. Different choice of the spin space Ω and the Hamiltonian H lead to other well-known spin systems such as the Potts model, cellular Potts Model, and the restricted Bolzman Machine (Nair and Hinton, 2010) (see the references given before).

5.2 Gibbs sampler for the Ising model

One of the most extensively studied Ising models is when the underlying graph G is the two-dimensional square lattice (see (McCoy and Wu, 2014) for a survey). It is well known that in this case the Ising model exhibits a sharp phase transition at the critical temperature $T = T_c = 2/\log(1+\sqrt{2}) \approx 2.2691$. Namely, if $T < T_c$ (subcritical phase), then there tends to be large clusters of $+1$'s and -1 spins; if $T > T_c$ (supercritical phase), then the spin clusters are very small and fragmented; at $T = T_c$ (criticality), the cluster sizes are distributed as a power law.

In order to sample a random spin configuration $\mathbf{x}\in\Omega$, we use the following MCMC called the *Gibbs sampler*. Namely, let the underlying graph $G=(V,E)$ be a finite $N\times N$ square lattice. We evolve a given spin configuration $\mathbf{x}_t : V\to\{-1,1\}$ at iteration t as follows:

(i) Choose a site $v\in V$ uniformly at random;

(ii) Let $\mathbf{x}^+$ and $\mathbf{x}^-$ be the spin configurations obtained from $\mathbf{x}_t$ by setting the spin of v to be 1 and -1, respectively. Then

$$p(\mathbf{x}_t,\,\mathbf{x}^+) = \frac{\pi(\mathbf{x}^+)}{\pi(\mathbf{x}^+)+\pi(\mathbf{x}^-)}, \qquad p(\mathbf{x}_t,\,\mathbf{x}^-) = \frac{\pi(\mathbf{x}^-)}{\pi(\mathbf{x}^+)+\pi(\mathbf{x}^-)}.$$

Note that $p(\mathbf{x}_{t+1},\,\mathbf{x}^+) = (1+\exp(2T^{-1}\sum_{u\sim v}\mathbf{x}_t(u)))^{-1}$, where the sum in the exponential is over all neighbors u of v. Iterating the above transition rule generates a Markov chain trajectory $(\mathbf{x}_t)_{t\in\mathbb{N}}$ of Ising spin configurations, and it is well known that it is irreducible, aperiodic, and has the Boltzmann distribution π_T (defined in (17)) as its unique stationary distribution.

5.3 Learning features from Ising spin configurations

Suppose we want to learn features from a random element X in a sample space Ω with distribution π. When the sample space is complicated, it is often not easy to directly sample a random element from it according to the prescribed distribution π. While Markov chain Monte Carlo (MCMC) provides a fundamental sampling technique that uses Markov chains to sample a random element (see, e.g., (Levin and Peres, 2017)), it inherently generates a dependent sequence of samples. In order to satisfy the independence assumption in most online learning algorithms, either one may generate each sample by a separate MCMC trajectory, or uses subsampling to reduce the dependence in a given MCMC trajectory. Both of these approaches suffer from inefficient use of already obtained data sample and never guarantee perfect independence. However, as our main result (Theorem 1) guarantees almost sure convergence of the dictionaries under Markov dependence, we may use arbitrary (or none) subsampling epoch to optimize the learning for a given MCMC trajectory. We will show that learning features from dependent sequence of data yields qualitatively different outcome, and also significantly improves efficiency of the learning process.

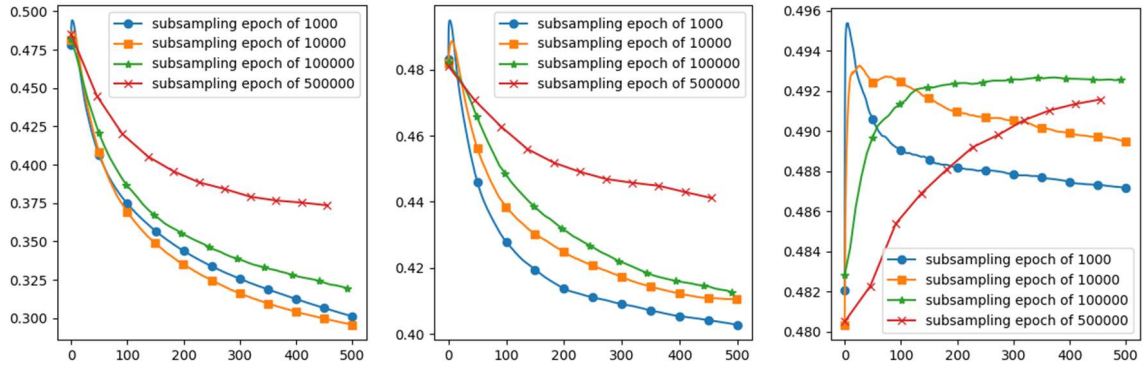


Figure 5: Plot of surrogate losses (normalized by 4×10^4) vs. MCMC iterations (unit 10^4) for subsampling epochs of 1000, 10000, 100000, and 500000 for temperatures $T = 0.5$ (left), $T = 2.26$ (middle), and $T = 5$ (right), respectively.

We first describe the setting of our simulation of online NMF algorithm on the Ising model. We consider a Gibbs sampler trajectory $(\mathbf{x}_t)_{t\in\mathbb{N}}$ of the Ising spin configurations on the 200×200 square lattice at three temperatures $T = 0.5$, 2.26, and 5. Initially $\mathbf{x}_0$ is sampled so that each site takes $+1$ or -1 spins independently with equal probability, and we run the Gibbs sampler for $5 \cdot 10^6$ iterations. We use four different subsampling epochs $\tau = 1000, 10000, 100000$, and 500000 for online dictionary learning. Namely, every τ iterations, we obtain a coarsened MCMC trajectory, which is represented as a $200 \times 200 \times (5 \cdot 10^6/\tau)$ array A whose kth array $A[:, :, k]$ corresponds to the spin configuration $X_k := \mathbf{x}_{\tau k}$. Then $(X_k)_{k\geq 0}$ also defines an irreducible and aperiodic Markov chain on Ω with the same stationary distribution π_T. For each X_k, which is a 200×200 matrix of entries from $\{-1, 1\}$, we sample 1000 patches of size 20×20. After flattening each patch into a column vector, we obtain a 400×1000 matrix, which we denote by $\texttt{Patch}_{20}(X_k)$. We apply the online NMF scheme to the Markovian sequence $(\texttt{Patch}_{20}(X_k))_{k\geq 0}$ of data matrices to extract 100 dictionary patches.

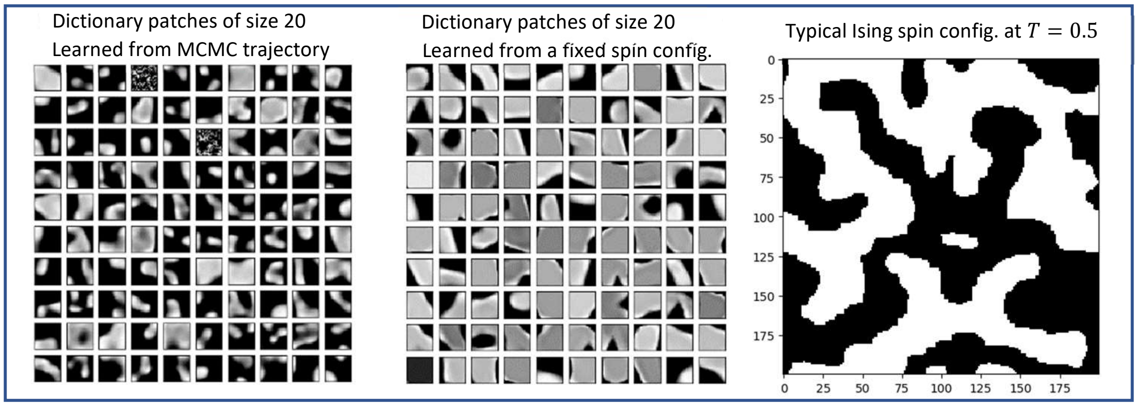
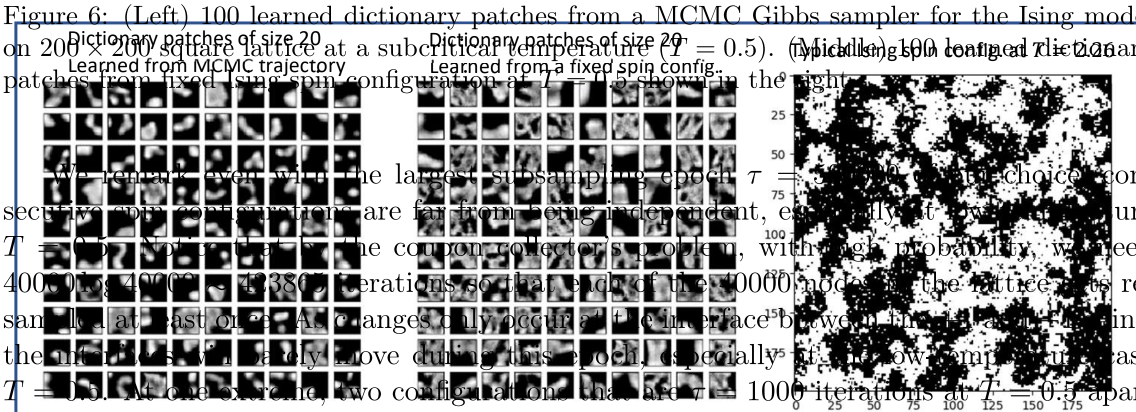
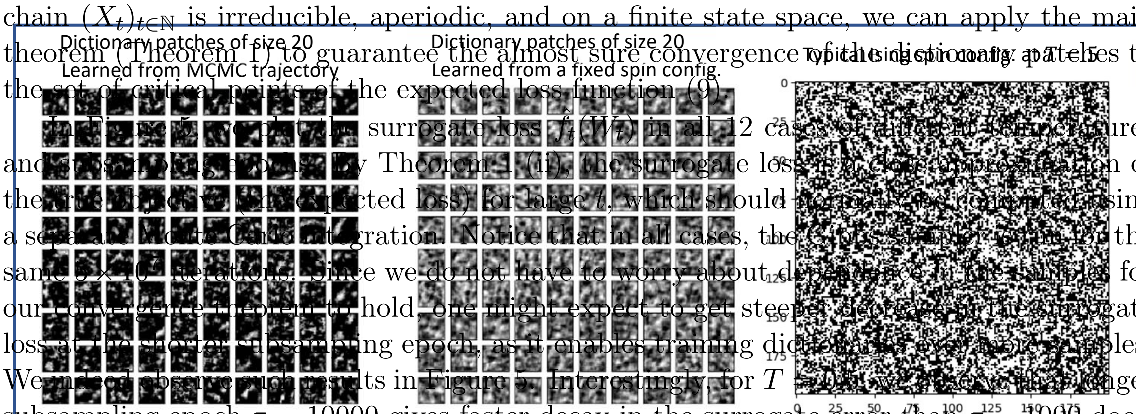


Figure 6: (Left) 100 learned dictionary patches from a MCMC Gibbs sampler for the Ising model on 200×200 square lattice at a subcritical temperature ($T = 0.5$). (Middle) 100 learned dictionary patches from fixed Ising spin configuration at $T = 0.5$ shown in the right.



We remark even with the largest subsampling epoch $\tau = 50000$ of our choice, consecutive spin configurations are far from being independent, especially at low temperature $T = 0.5$. Notice that by the coupon collector's problem, with high probability, we need $40000 \log 40000 \approx 423865$ iterations so that each of the 40000 nodes in the lattice gets resampled at least once. As changes only occur at the interface between the $+$ and $-$ domains, the interfaces will barely move during this epoch, especially at the low temperature case $T = 0.5$. At one extreme, two configurations that are $\tau = 1000$ iterations at $T = 0.5$ apart will look almost identical, hence with significant correlation. However, in all cases, as the chain $(X_t)_{t\in\mathbb{N}}$ is irreducible, aperiodic, and on a finite state space, we can apply the main theorem (Theorem 1) to guarantee the almost sure convergence of the dictionary patches to the set of critical points of the expected loss function (9).



In Figure 5, we plot the surrogate loss $\hat{f}_t(W_t)$ in all 12 cases of different temperatures and subsampling epochs. By Theorem 1 (ii), the surrogate loss is a close approximation of the true objective (the expected loss) for large t, which should normally be computed using a separate Monte Carlo integration. Notice that in all cases, the Gibbs sampler is run for the same 5×10^7 iterations. Since we do not have to worry about dependence in the samples for our convergence theorem to hold, one might expect to get steeper decrease in the surrogate loss at the shorter subsampling epoch, as it enables training dictionaries over more samples. We indeed observe such results in Figure 5. Interestingly, for $T = 0.5$, we observe that longer subsampling epoch $\tau = 10000$ gives faster decay in the surrogate error than $\tau = 1000$ does. This can be explained as an overfitting issue. Namely, since two configurations that are 1000 iterations apart are barely different, training dictionaries too frequently may overfit them to specific configurations, while the objective is to learn from the average configuration. The dictionaries learned from each of the 12 simulations are shown in Figure 14.

In Figures 6, 11, and 12, we compare 100 learned dictionary elements directly from the MCMC trajectory $(X_k)_{0\le k\le 500}$ with subsampling epoch $\tau = 10000$ as well as from a fixed spin configuration for the three temperatures $T = 0.5$ (subcritical), $T = 2.26$ (near critical), and $T = 5$ (supercritical). In all three figures, we see qualitative differences between the two sets of dictionary elements, which can be explained as follows. Since spin configurations

gradually change in the MCMC trajectory $(X_k)_{0\leq k\leq 500}$, the dictionary elements learned from the trajectory should not be overfitted to a particular configuration as the one in the middle of each figure does, but should capture features common to a number of spin configurations at the corresponding temperature.

6. Application II: Network dictionary learning by online NMF and motif sampling

In this section, we propose a novel framework for network data analysis that we call *Network Dictionary Learning*, which enables one to extract "network dictionary patches" from a given network to see its fundamental features and to reconstruct the network using the learned dictionaries. Network Dictionary Learning is based on two building blocks: 1) Online NMF on Markovian data, which is the main subject in this paper, and 2) a recent MCMC algorithm for sampling motifs from networks in (Lyu et al., 2019). More details on Network Dictionary Learning and applications to social networks will be given in an upcoming paper (Lyu et al., 2020).

6.1 Extracting patches from a network by motif sampling

We formally define a *network* as a pair $\mathcal{G} = (V, A)$ of node set V and a *weight matrix* $A : V^2 \to [0, \infty)$ describing interaction strengths between the nodes. In this formulation, we do not distinguish between multi-edges and weighted edges in networks. A given graph $G = (V, E)$ determines a unique network $\mathcal{G} = (V, A_G)$ with A_G the adjacency matrix of $\mathcal{G}$. We call a network $\mathcal{G} = (V, A)$ *simple* if A is symmetric, binary (i.e., $A(x, y) \in \{0, 1\}$), and without self-edges (i.e., $A(x, x) = 0$). We identify a simple graph $G = (V, E)$ with adjacency matrix A_G with the simple network $\mathcal{G} = (V, A_G)$.

For networks, we can think of a $(k \times k)$ patch as a sub-network induced onto a subset of k nodes. As we imposed to select k consecutive rows and columns to get patches from images, we need to impose a reasonable condition on the subset of nodes so that the selected nodes are strongly associated. For instance, if the given network is sparse, selecting three nodes uniformly at random would rarely induce any meaningful sub-network. Selecting such a subset of k nodes from networks can be addressed by the *motif sampling* technique introduced in (Lyu et al., 2019). Namely, for a fixed "template graph" (motif) F of k nodes, we would like to sample k nodes from a given network $\mathcal{G}$ so that the induced sub-network always contains a copy of F. This guarantees that we are always sampling some meaningful portion of the network, where the prescribed graph F serves as a backbone. More precisely, fix an integer $k \geq 1$ and a matrix $A_F : [k]^2 \to [0, \infty)$, where $[k] = \{1, 2, \ldots, k\}$. The corresponding network $F = ([k], A_F)$ is called a *motif*. The particular motif of our interest is the *k-chain*, where $A_F = \mathbf{1}(\{(1, 2), (2, 3), \ldots, (k - 1, k)\})$. The k-chain motif corresponds to a directed path with node set $[k]$.

Based on these ideas, we propose the following preliminary version of Network Dictionary Learning for simple graphs.

Network Dictionary Learning (NDL): Static version for simple graphs

(i) Given a simple graph $\mathcal{G} = (V, A)$ and a motif $F = ([k], A_F)$, let $\texttt{Hom}(F, \mathcal{G})$ denote the set of all *homomorphisms* $F \to \mathcal{G}$:

$$\texttt{Hom}(F, \mathcal{G}) = \left\{ \mathbf{x} : [k] \to [n] \;\middle|\; \prod_{1 \le i,j \le k} A(\mathbf{x}(i), \mathbf{x}(j))^{A_F(i,j)} = 1 \right\}.$$

Compute $\texttt{Hom}(F, \mathcal{G})$ and write $\texttt{Hom}(F, \mathcal{G}) = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$.

(ii) For each homomorphism $\mathbf{x} : F \to \mathcal{G}$, associate a $(k \times k)$ matrix $A_{\mathbf{x}}$ by

$$A_{\mathbf{x}}(a, b) = A(\mathbf{x}(a), \mathbf{x}(b)) \qquad 1 \le a, b \le k. \tag{18}$$

Let X denote the $(k^2 \times N)$ matrix whose ith column is the k^2-dimensional vector obtained by vectorizing $A_{\mathbf{x}_i}$ (using the lexicographic ordering).

(iii) Factorize $X \approx WH$ using NMF. Reshaping the columns of the dictionary matrix W into $k \times k$-squares gives the learned network dictionary elements.

6.2 Motif sampling from networks and MCMC sampler

There are two main issues in the preliminary Network Dictionary Learning scheme for simple graphs we described in the previous subsection. First, computing the full set $\texttt{Hom}(F, \mathcal{G})$ of homomorphisms $F \to \mathcal{G}$[1] is computationally expensive with $O(n^k)$ complexity. Second, in the case of the general network with edge and node weights, some homomophisms could be more important in capturing features of the network than others. In order to overcome the second difficulty, we introduce a probability distribution for the homomorphisms for the general case that takes into account the weight information of the network. To handle the first issue, we use a MCMC algorithm to sample from such a probability measure and apply online NMF to sequentially learn network dictionary patches.

For a given motif $F = ([k], A_F)$ and a n-node network $\mathcal{G} = (V, A)$, we introduce the following probability distribution $\pi_{F \to \mathcal{G}}$ on the set $V^{[k]}$ of all vertex maps $\mathbf{x} : [k] \to V$ by

$$\pi_{F \to \mathcal{G}}(\mathbf{x}) = \frac{1}{\texttt{Z}} \left(\prod_{1 \le i,j \le k} A(\mathbf{x}(i), \mathbf{x}(j))^{A_F(i,j)} \right), \tag{19}$$

where $\texttt{Z}$ is the normalizing constant called the *homomorphism density* of F in $\mathcal{G}$. We call the random vertex map $\mathbf{x} : [k] \to V$ distributed as $\pi_{F \to \mathcal{G}}$ the *random homomorphism* of F into $\mathcal{G}$. Note that $\pi_{F \to \mathcal{G}}$ becomes the uniform distribution on the set of all homomorphisms $F \to \mathcal{G}$ when both A and A_F are binary matrices.

In order to sample a random homomorphism $F \to \mathcal{G}$, we use the MCMC algorithms introduced in (Lyu et al., 2019) called the *Glauber chain* (see Algorithm MG) and the *Pivot chain* (see Algorithm MP). The Glauber chain an exact analogue of the Gibbs sampler for

1. When $\mathcal{G}$ is a complete graph K_q with q nodes, computing homomorphisms $F \to K_q$ equals to computing all proper q-colorings of F.

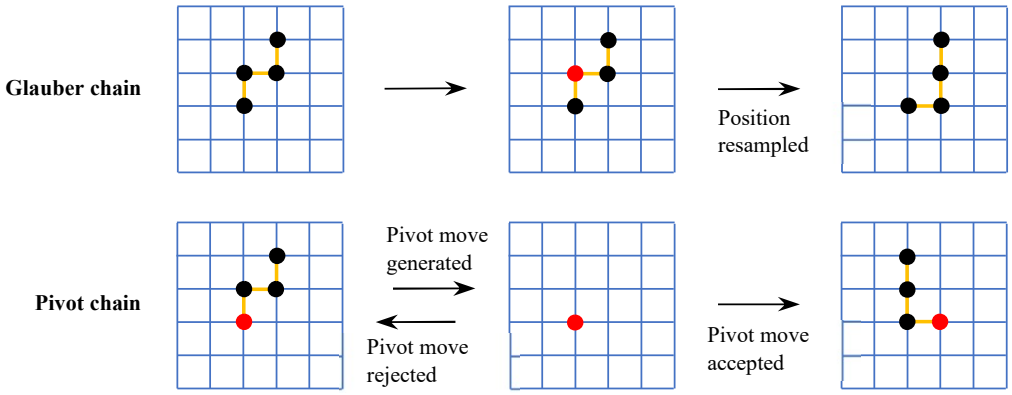


Figure 7: Single iterations of the Glauber chain (first row) and Pivot chain (second row) sampling of homomorphisms $\mathbf{x}_t : F \to \mathcal{G}$, where $\mathcal{G}$ is the (6×6) grid and $F = ([4], \mathbf{1}_{\{(1,2),(2,3),(3,4)\}})$ is the 4-chain motif. For the Glauber chain, a node $v \in [4]$ (in red) in the motif is sampled uniformly at random and its position $\mathbf{x}_t(v)$ is resampled to $\mathbf{x}_{t+1}(v)$. For the Pivot chain, a random walk move $\mathbf{x}_t(1) \to \mathbf{x}_{t+1}(1)$ of the pivot (in red) is generated, which is accepted or rejected according to the acceptance probability computed by the Metropolis-Hastings algorithm. If it is accepted, positions of the subsequent nodes $\mathbf{x}_{t+1}(2), \mathbf{x}_{t+1}(3), \mathbf{x}_{t+1}(4)$ are sampled successively; otherwise, we take $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t$.

the Ising model we discussed in Subsection 5.2. Namely, for the update $\mathbf{x}_t \mapsto \mathbf{x}_{t+1}$, we pick one node $v \in [k]$ of F uniformly at random, and resample the time-t position $\mathbf{x}_t(v) \in V$ of node i in the network $\mathcal{G}$ from the correct conditional distribution. (See the first row of Figure 7 for an illustration.)

The Pivot chain is a combination of a random walk on networks and the Metropolis-Hastings algorithm. There, node 1 in the motif is designated as the 'pivot'. For each update $\mathbf{x}_t \mapsto \mathbf{x}_{t+1}$, we first generate a random walk move $\mathbf{x}_t(1) \to \mathbf{x}_{t+1}(1)$ of the pivot (e.g., when the network is simple, then $\mathbf{x}_{t+1}(1)$ is a uniformly chosen neighbor of $\mathbf{x}_t(1)$), which is then accepted with a suitable acceptance probability (see (34)) according to the Metropolis-Hastings algorithm (see, e.g., (Levin and Peres, 2017, Sec. 3.2)). If rejected, we take $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t$; otherwise, each $\mathbf{x}_{t+1}(i) \in V$ for $i = 2, 3, \ldots, \kappa$ is sampled successively from the appropriate conditional distribution (see (35)) so that the stationary distribution such that the resulting Markov chain is the desired distribution $\pi_{F \to \mathcal{G}}$ in (19) as its unique stationary distribution. (See the second row of Figure 7 for an illustration.)

In Algorithm MP, we provide a variant of the original Pivot chain in (Lyu et al., 2019) that uses an approximate computation of the correct acceptance probability (34) with the boolean variable `AcceptProb` = `Approximate`. This is to reduce the computational cost of computing the exact acceptance probability, which could be costly when the motif F has large number of nodes. (See (Lyu et al., 2020) for a more detailed discussion.)

6.3 Algorithms for Network Dictionary Learning and Reconstruction

In Algorithm NDL, we give the algorithm for Network Dictionary Learning that combines online NMF and MCMC motif sampling with the ideas that we described in (6.1). Below we give a high-level description of Algorithm NDL.

(The detailed algorithm is given in the appendix)

Network Dictionary Learning (NDL): Online version for general networks

Given a simple graph $\mathcal{G} = (V, A)$ and a motif $F = ([k], A_F)$, do the following for $t = 1, 2, \ldots, T$:

(i) Generate N homomorphisms $\mathbf{x}_s$ for $N(t-1) \leq s \leq Nt$ using a MCMC motif sampling algorithm

(ii) Compute N $(k \times k)$ matrices $A_{\mathbf{x}_s}$ by (18). Let X_t be the $(k^2 \times N)$ matrix whose jth column is the vectorization of $A_{\mathbf{x}_\ell}$ with $\ell = N(t-1) + j$.

(iii) Update the previous dictionary matrix $W_{t-1} \in \mathbb{R}_{\geq 0}^{k^2 \times N}$ to W_t with respect to the new data matrix X_t using Online NMF.

At each iteration $t = 1, 2, \ldots, T$, a chosen motif sampling algorithm generates a sequence of N homomorphisms $\mathbf{x}_s : F \to \mathcal{G}$ and corresponding $(k \times k)$ matrices $A_{\mathbf{x}_s}$, which are summarized as the $(k^2 \times N)$ data matrix X_t. The online NMF algorithm (Algorithm 1) then learns a nonnegative factor matrix W_t of size $(k^2 \times r)$ by improving the previous factor matrix W_{t-1} with respect to the new data matrix X_t. Note that during this entire process, the algorithm only needs to hold two auxiliary matrices P_t and Q_t of fixed sizes $(r \times r)$ and $(r \times k^2)$, respectively, but not the previous data matrices $X_1, \ldots, X_{t-1}$. Hence NDL is efficient in memory and scalable in the network size. Moreover, NDL is applicable for temporally changing networks due to its online nature.

Next, in Algorithm NR, we provide an algorithm that reconstructs a given network using a network dictionary learned by Algorithm NDL. The idea behind our network reconstruction algorithm is the following. Similarly as in image reconstruction, network reconstruction is done by first sampling random k-node subgraphs that contain the corresponding motifs, whose adjacency matrices are approximated by the learned dictionary atoms. Then these reconstructions are patched together with a suitable averaging. However, unlike image reconstruction where we can easily access any desired $(k \times k)$ square patch, for network reconstruction, we cannot directly sample a random k-node subgraph that contains a fixed motif. For this, we use the Markov chain $(\mathbf{x}_t)_{t \in \mathbb{N}}$ of homomorphisms $F \to \mathcal{G}$ as we do in network dictionary learning. For each $t \in \mathbb{N}$, we reconstruct the $k \times k$ patch of $\mathcal{G}$ corresponding to the current homomorphism $\mathbf{x}_t$ using the learned dictionaries. We keep track of the overlap count for each entry $A(a, b)$ that we have reconstructed up to time t, and take the average of all the proposed values of each entry $A(a, b)$ up to time t.

As a corollary of our main result (Theorem 1) and (Lyu et al., 2019, Thm 5.7) for the convergence of the Glauber and Pivot chains for homomorphisms, we obtain the following convergence guarantee of Algorithm NDL for Network Dictionary Learning.

Corollary 2. *Let $F = ([k], A_F)$ be the k-chain motif and let $\mathcal{G} = (V, A)$ be a network that satisfies the following properties:*

(i) *Random walk on $\mathcal{G}$ is irreducible and aperiodic;*

(ii) $\mathcal{G}$ *is bidirectional, that is,* $A(a,b) > 0$ *implies* $A(b,a) > 0$.

(iii) *For all* $t \in \mathbb{N}$*, there exists a unique solution for* H_t *in* (36).

(iv) *For all* $t \in \mathbb{N}$*, the eigenvalues of the positive semidefinite matrix* A_t *that is defined in* (36) *are at least as large as some constant* $\kappa_1 > 0$.

Then Algorithm NDL with `MCMC` $\in \{$`Glauber`, `Pivot`$\}$ *for Network Dictionary Learning converges almost surely to the set of local optima of the associated expected loss function.*

Proof Let $(X_t)_{t\in\mathbb{N}}$ be the sequence of $(k^2 \times N)$ matrices of "minibatches of subgraph patterns" X_t defined in Algorithm NDL. Since Algorithm NDL can be viewed as the OMF algorithm (3) applied to the dependent sequence $(X_t)_{t\in\mathbb{N}}$, it suffices to verify the assumptions (A1), (M1), and (M2)' according to Theorem 1.

We first observe that the matrices $X_t \in \mathbb{R}_{\geq 0}^{k^2\times N}$ computed in line 13 of Algorithm NDL do not necessarily form a Markov chain, as the forward evolution of the Markov chain $\mathbf{x}_t$ depends not only on the induced $(k \times k)$ matrix $A_{\mathbf{x}_t}$, but also on the actual homomorphisms $\mathbf{x}_t$. However, note that the 'augmented' sequence $\overline{X}_t := (X_t, \mathbf{x}_{Nt})$ forms a Markov chain. Indeed, the distribution of X_{t+1} given X_t depends only on $\mathbf{x}_{Nt}$ and A, since this determines the distribution of the homomorphisms $(\mathbf{x}_s)_{Nt<s<N(t+1)}$, which in turn determine the $k^2 \times N$ matrix X_{t+1}.

Under the assumptions (i) and (ii), (Lyu et al., 2019, Thm 5.7 and 5.8) shows that the sequence $(\mathbf{x}_t)_{t\in\mathbb{N}}$ of homomorphisms $F \to \mathcal{G}$ is a finite state Markov chain that is irreducible and aperiodic with unique stationary distribution $\pi_{F\to\mathcal{G}}$ (see (19)). This easily implies the N-tuple of homomorphisms $(\mathbf{x}_s)_{N(t-1)<s<Nt}$ also form a finite-state, irreducible, and aperiodic chain with a unique stationary distribution. Consequently, the Markov chain $(\overline{X}_t)_{t\in\mathbb{N}}$ that we defined above is also a finite-state, irreducible, and aperiodic chain with a unique stationary distribution. In this setting, one can regard Algorithm NDL as the Online NMF algorithm in (3) for the input sequence $X_t = \varphi(\overline{X}_t)$, where $\varphi : (X, \mathbf{x}) \mapsto X$ is the projection on its first coordinate. Because $\overline{X}_t$ takes only finitely-many values, the range of φ is bounded. This verifies all hypotheses of Theorem 1, so the assertion follows. ■

Remark 3. A similar convergence result of Algorithm NDL with `MCMC` = `ApproxPivot` also holds. The precise statement and its proof will be given in (Lyu et al., 2020).

6.4 Applications of Algorithm NDL to real-world networks

In this subsection, we apply Algorithms NDL to the following real-world networks:

1. `SNAP Facebook` (`Facebook`) (Leskovec and Mcauley, 2012; Grover and Leskovec, 2016): This network has 4,039 nodes and 88,234 edges. This network is a Facebook network that has been used as a benchmark example for edge inference.

2. `arXiv ASTRO-PH` (`arXiv`) (Leskovec and Krevl, June 2014; Grover and Leskovec, 2016): This network has 18,722 nodes and 198,110 edges. It is a collaboration network between authors of preprints that were posted in the arXiv in astrophysics. Nodes represent scientists and edges indicate coauthorship relationships.

3. Homo sapiens PPI (H. sapiens) (Oughtred et al., 2019; Grover and Leskovec, 2016): This network has 19,706 nodes and 390,633 edges. The nodes represent proteins in the organism of Homo sapiens, and edges represent physical interactions between these proteins.

Let $G = (V, E)$ be any simple graph and let $F = ([6], A_F)$ be the 6-chain motif. Fix a homomorhism $\mathbf{x} : F \to G$. Then the corresponding (6×6) matrix $A_{\mathbf{x}}$ (defined in (18)) is of the following form:

$$A_{\mathbf{x}} = \begin{bmatrix} 0 & 1 & * & * & * & * \\ 1 & 0 & 1 & * & * & * \\ * & 1 & 0 & 1 & * & * \\ * & * & 1 & 0 & 1 & * \\ * & * & * & 1 & 0 & 1 \\ * & * & * & * & 1 & 0 \end{bmatrix}, \tag{20}$$

where entries marked as $*$ may be 0 or 1 (not necessarily the same values). Notice that the 1's in the diagonal line above the main diagonal correspond to the entries $A(\mathbf{x}(i), \mathbf{x}(i+1))$, $1 \leq i < 6$, that are required to be 1 since $A_F(i, i+1) = 1$ and $\mathbf{x} : F \to G$ is a homomorphism (recall that A is binary since G is a simple graph). The same observation holds for any k-chain motifs for any $k \geq 2$. Hence the more interesting information is captured by the entries off of the two diagonal lines.

In Figure 8, we show $r = 25$ network dictionary elements learned from each of the above networks using Algorithm NDL with the following parameters: $F = ([21], A_F)$ the 21-chain motif, $T = 100$, $N = 100$, and $\lambda = 1$. In Figure 8, such entries in the the learned dictionary elements reveal distinctive structures of the networks. Namely, most dictionary elements for Facebook show 'communities' (blocks of pixels) and 'hub nodes' (diagonal entries with the corresponding row and column in black); arXiv show a very few hub nodes, but do exhibit clusters (groups of black pixels), which is reasonable since scientists tend to collaborate often as a team and it is less likely that there is an overly popular scientist (whereas popular users in Facebook networks are natural); In the Homo sapiens PPI, it seems that proteins there hardly form large clusters of mutual interaction.

Note that in Figure 8, we also show the "dominance score" for each dictionary element, which has the meaning of its 'usage' in approximating a sampled $(k \times k)$ matrix $A_{\mathbf{x}}$ from G. It is computed by normalizing the square root of the diagonal entries of the $r \times r$ aggregate matrix P_t in (36). Roughly speaking, the ith diagonal entry of P_t is approximately the average of the L_2 norm of the ith row of the code matrix H where $X_t \approx WH$ for X_t the $k^2 \times N$ matrix of 'subgraph patterns' and W the $k^2 \times r$ dictionary matrix. For instance, in both arXiv and H. sapiens, the 'most dominant' dictionary elements (with dominance score over 0.15) have mostly zeros outside of the two diagonal lines, indicating that these two networks are sparse.

6.5 Applications of Algorithm NR for network denoising problems

In this subsection, we apply Algorithm NR to solve network denoising problems. Namely, if we are given a simple graph $G_{\text{true}} = (V, E)$, we may either add some 'false edges' or delete some true edges (hence creating 'false nonedges') randomly and create a 'corrupted

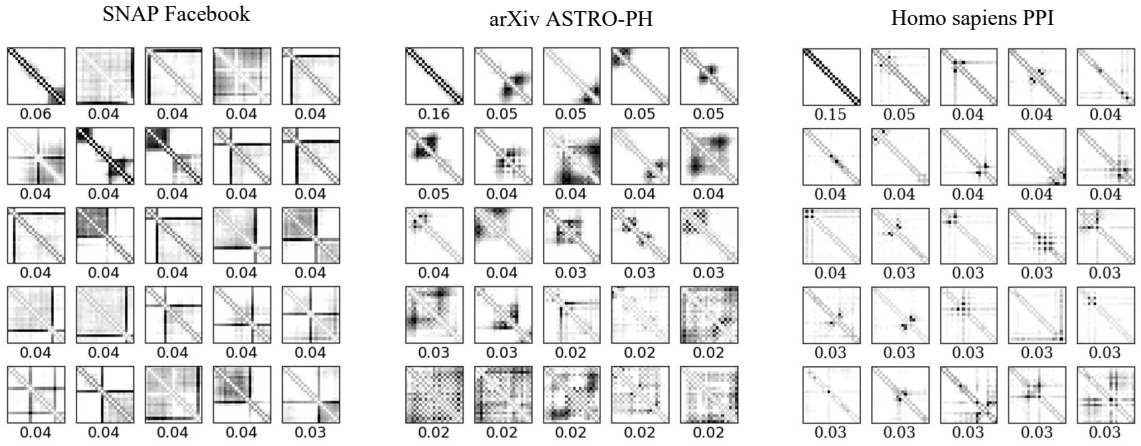


Figure 8: $r = 25$ learned dictionary elements from networks FACEBOOK, ARXIV, and H. SAPIENS with 21-chain motif F. Black=1 and white = 0 with gray scale. The values below the dictionary elements indicate their "dominance score", which is computed by the diagonal entries of the $r \times r$ aggregate matrix P_t in (36).

version' $G_{\text{obs}} = (V, E')$ of the original graph G_{true}. The problem is to recover G_{true} when we are only given with the corrupted observed network G_{obs}. This problem is also known as 'network denoising' (Correia et al., 2019) or 'edge inference' (or prediction) (Liben-Nowell and Kleinberg, 2007; Lü and Zhou, 2011; Menon and Elkan, 2011; Kovács et al., 2019) in the cases of adding or deleting edges, respectively. Here we refer these problems collectively as 'network denoising' with 'additive noise' or 'subtractive noise', correspondingly. Each setting can be regarded as a binary classification problem. Namely, for the additive noise case, it is equal to the binary classification of the edges set E' in the corrupted graph G_{obs} into true (positive) and false (negative) edges; for the subtractive noise case, we are to classify the set of all nonedges in G_{obs} into true (nonedges in G_{true}) and false (deleted edges of G_{true}).

To experiment with these problems, we use the three real-world networks: FACEBOOK, ARXIV, and H. SAPIENS. Given a network $G_{\text{true}} = (V, E)$, we first generate four corrupted networks G_{obs} as follows. In the subtractive noise case, we create a smaller connected network by removing a uniformly chosen random subset that consists of 50% of the edges from our network. In the additive noise case, we create a corrupted network by adding edges between node pairs independently with a fixed probability so that the new network has 50% new edges.

In order to solve the network denoising problems, we first apply Algorithm NDL with 21-chain motifs with $r = 25$ columns in the dictionary matrix to learn a network dictionary for each of these four networks, and we use each dictionary to reconstruct the network from which it was learned using Algorithm NR with parameters $T = 200,000$, $\lambda = 0$, and `MCMC = ApproxPivot`. The reconstruction algorithms output a weighted network $G_{\text{recons}} = (V, A_{\text{recons}})$. For denoising additive noise, we classify each edge in the corrupted network as 'positive' if its weight in A_{recons} is strictly *larger* than some threshold θ. For denoising subtractive noise, we classify each nonedge in the corrupted network as 'positive' if its weight in A_{recons} is strictly *less* than some threshold θ (see Remark 4 for why we use the opposite

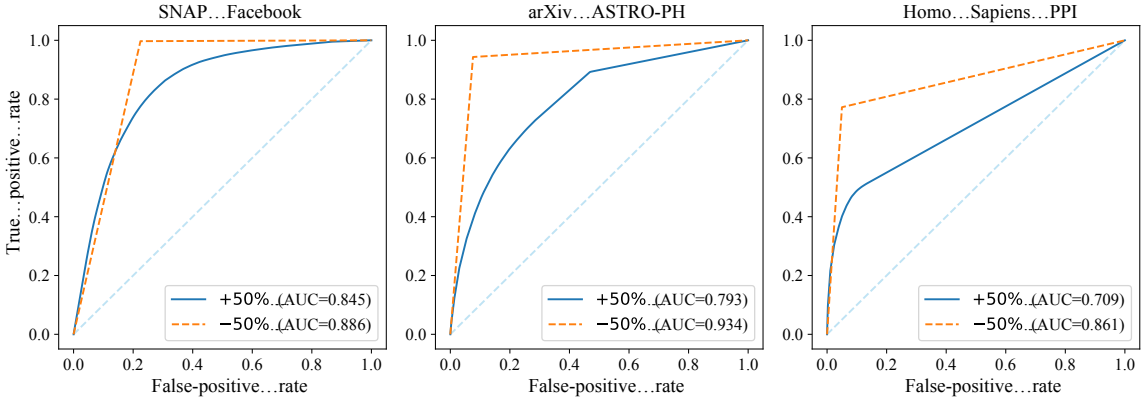


Figure 9: Application of the NDL and NDR algorithms to network denoising with additive and subtractive noise on Facebook and PPI networks. We first use NDL to learn a network dictionary from a corrupted network and then reconstruct the networks using NDR to assign a confidence value to each potential edge. We then use these confidence values to infer membership of potential edges in the uncorrupted network. Importantly, we never use information from the original networks. For each network, we indicate the receiver-operating characteristic (ROC) curves and corresponding area-under-the-curve (AUC) scores for network denoising with additive noise using the labels +50%, and we indicate the ROC curves and corresponding AUC scores for network denoising with subtractive noise using the labels −50%.

directionality of classification for subtractive noise). By varying θ we construct a receiver operating characteristic (ROC) curve that consists of points whose horizontal and vertical coordinates are the false-positive rate and true-positive rate, respectively. For instance, if $\theta = 0$ in the additive (resp., subtractive) noise case, almost all edges (resp., nonedges) will be classified as 'positive' (resp., 'negative'), so it will correspond to the corner $(1, 1)$ (resp., $(0, 0)$) in the ROC curve.

Algorithm	SNAP Facebook	Homo sapiens PPI	arXiv ASTRO-PH
Spectral Clustering	0.619	0.492	0.574
DeepWalk	0.968	0.744	**0.934**
LINE	0.949	0.725	0.890
node2vec	**0.968**	0.772	**0.934**
NDL+NDR (our method)	0.907	**0.861**	**0.934**

Table 10: Comparison of AUC scores for network denoising with additive (+50%) and subtractive (−50%) noises. The results in the first four rows are obtained from Grover and Leskovec (2016).

In Figure 9, we show the ROC curves and corresponding area-under-the-curve (AUC) scores for our network-denoising experiments with subtractive and additive noise for all three networks. For example, if one adds 50% of false edges to the Facebook so that 88,234 edges are true and 44,117 edges are false, then our method achieves AUC of 0.845, and is able to detect over 78% (34,411) of the false edges while misclassifying 20% (17,647) of the true edges (see Figure 9 left). In Table 10, we also compare the performance of our method

against some popular supervised algorithms based on network embedding, such as node2vec (Grover and Leskovec, 2016), DeepWalk (Perozzi et al., 2014), and LINE (Tang et al., 2015) for the task of denoising 50% subtractive noise for SNAP Facebook, H. Sapiens, and arXiv. It is important to note that, unlike these methods, our algorithm for network denoising is *unsupervised* in the sense that it never requires any information from the original network G. Nonetheless, our algorithm shows comparable performance and in two cases the best results among all methods considered here.

Remark 4. The reason that we used the opposite directionality of classification for subtractive noise, that is, a nonedge (p, q) in G_{obs} is classified 'positive' if $A_{\text{recons}}(p, q) < \theta$, is that we often obtain 'flipped' ROC curves using the standard classification scheme

$$A_{\text{recons}}(p, q) > \theta \quad \Longrightarrow \quad (p, q) \text{ is classified as positive} \tag{21}$$

for denoising subtractive noise. While a complete understanding of this phenomenon is yet to be made, we remark here why the standard classification (21) may not work in our favor for subtractive noise. The issue is related with sparsity of real-world networks, which does not arise in image denoising.

First recall that every sampled $k \times k$ matrix $A_{\mathbf{x}}$ is conditioned to have 1's on its first super- and sub-diagonals (see (20)), corresponding to the observed edges of G_{obs} in the image of the homomorhpism $\mathbf{x}$. Hence for subtractive noise, false non-edges (deleted edges in G_{true}) always appear as 0's outside the two super- and sub-diagonal lines of $A_{\mathbf{x}}$. Now if G_{obs} is sparse (as most real-world networks are), then there are very few positive entries in $A_{\mathbf{x}}$ other than the first super- and sub-diagonal entries. Hence if we approximate such $A_{\mathbf{x}}$ using network dictionary atoms, it is more likely to overfit to reconstruct the observed edges (1's in the first super- and sub-diagonal entries), which will result in small reconstructed weights for the other entries of $A_{\mathbf{x}}$, including the ones corresponding to false non-edges.

In Lyu et al. (2020), a modified version of NDR (Algorithm NR) will be introduced that addresses this issue, so that we can use the classification scheme (21) uniformly both for additive and subtractive noise.

7. Proof of Theorem 1

In this section, we provide the proof of our main result, Theorem 1.

7.1 Preliminary bounds

In this subsection, we derive some key inequalities and preliminary bounds toward the proof of Theorem 1. Note that proofs are relagated to the appendix.

Proposition 5. *Let $(W_t, H_t)_{t\geq 1}$ be a solution to the optimization problem* (3)*. Then for each $t \in \mathbb{N}$, the following hold almost surely:*

(i) $\hat{f}_{t+1}(W_{t+1}) - \hat{f}_t(W_t) \leq w_{t+1}\left(\ell(X_{t+1}, W_t) - f_t(W_t)\right) = f_{t+1}(W_t) - f_t(W_t)$.

(ii) $0 \leq w_{t+1}\left(\hat{f}_t(W_t) - f_t(W_t)\right) \leq w_{t+1}\left(\ell(X_{t+1}, W_t) - f_t(W_t)\right) + \hat{f}_t(W_t) - \hat{f}_{t+1}(W_{t+1})$.

Proof See Appendix A. ■

Next, we show that if the data are drawn from compact sets, then the set of all possible codes also form a compact set. This also implies boundedness of the matrices $A_t \in \mathbb{R}^{r\times r}$ and $B_t \in \mathbb{R}^{r\times n}$, which aggregate sufficient statistics up to time t (defined in (3)).

The following proposition provides a second-order growth property of the quadratic function for W_t in the OMF algorithm (3) when the set $\mathcal{C}$ of constraints for the dictionaries is general and not necessarily convex. This is well-known for the convex case (see, e.g., (Mairal, 2013a, Lem. B.5)).

Proposition 6. *Fix symmetric and positive definite $A \in \mathbb{R}^{r\times r}$, arbitrary $B \in \mathbb{R}^{r\times d}$. Denote $g(W) = \mathrm{tr}(WAW^T) - 2\mathrm{tr}(WB)$ for each $W \in \mathbb{R}^{d\times r}$. Then the following hold:*

(i) *Let $W_1, W_2 \in \mathbb{R}^{d\times r}$ be such that $\mathrm{tr}((B^T - W_2A)(W_1 - W_2)^T) \le 0$. Then*

$$g(W_1) - g(W_2) \ge \mathrm{tr}((W_1 - W_2)A(W_1 - W_2)^T) \ge 0. \tag{22}$$

(ii) *Fix $W_1, W_2 \in \mathbb{R}^{d\times r}$ and suppose the function $g(\lambda W_2 + (1-\lambda)W_1)$ is monotone decreasing in $\lambda \in [0,1]$. Then $\mathrm{tr}((B^T - W_2A)(W_1 - W_2)^T) \le 0$.*

(iii) *Let $\mathcal{C} \subseteq \mathbb{R}^{d\times r}$ be convex, $W_{t-1} \in \mathcal{C}$ arbitrary, and $W_t = \arg\min_{W\in\mathcal{C}} g(W)$. Then*

$$g(W_{t-1}) - g(W_t) \ge \mathrm{tr}((W_t - W_{t-1})A(W_t - W_{t-1})^T) \ge 0.$$

Proof See Appendix A. ■

An important consequence of the above second-order growth condition is an upper bound on the change of learned dictionaries, which is also known as "iterate stability" (Mairal, 2013b, Lem B.8)

Proposition 7. *Let $(W_t, H_t)_{t\ge1}$ be a solution to the OMF scheme (3). Assume (A1)-(A2) and (C2) for (3). Then there exist some constant $c > 0$ such that almost surely for all $t \in \mathbb{N}$,*

$$\|W_{t+1} - W_t\|_F \le c w_{t+1}.$$

Proof See Appendix A. ■

Remark 8. Proposition 7 with triangle inequality shows that $\|W_m - W_n\|_F \le \sum_{j=n+1}^{m} w_j$. Hence if $\sum_{j=1}^{\infty} w_j < \infty$, then W_t converges in the compact set $\mathcal{C}$ for arbitrary input data sequence $(X_t)_{t\in\mathbb{N}}$ in a bounded set.

7.2 Convergence of the empirical and surrogate loss

We prove Theorem 1 in this subsection. According to Proposition 5, it is crucial to bound the quantity $\ell(X_{t+1}, W_t) - f_t(W_t)$. When Y_t's are i.i.d., we can condition on the information $\mathcal{F}_t$ up to time t so that

$$\mathbb{E}\left[\ell(X_{t+1}, W_t) - f_t(W_t) \,\middle|\, \mathcal{F}_t\right] = f(W_t) - f_t(W_t).$$

Note that for each fixed $W \in \mathcal{C}$, $f_t(W) \to f(W)$ almost surely as $t \to \infty$ by the strong law of large numbers. To handle time dependence of W_t, one can instead look that the convergence of the supremum $\|f_t - f\|_\infty$ over the compact set $\mathcal{C}$, which is provided by the classical Glivenko-Cantelli theorem. This is the approach taken in (Mairal et al., 2010; Mairal, 2013b) for i.i.d. input.

However, the same approach breaks down when $(Y_t)_{t\in\mathbb{N}}$ is a Markov chain. This is because, conditional on $\mathcal{F}_t$, the distribution of Y_{t+1} is not necessarily the stationary distribution π. Our key innovation to overcome this difficulty is to condition much early on – at time $t-N$ for some suitable $N = N(t)$. Then the Markov chain runs $N+1$ steps up to time $t+1$, so if N is large enough for the chain to mix, then the distribution of Y_{t+1} conditional on $\mathcal{F}_{t-N}$ is close to the stationary distribution π. The error of approximating the stationary distribution by the $N+1$ step distribution is controlled using total variation distance and mixing bound.

Proposition 9. *Suppose (A1)-(A2) and (M1). Fix $W \in \mathcal{C}$. Then for each $t \in \mathbb{N}$ and $0 \le N < t$, conditional on the information $\mathcal{F}_{t-N}$ up to time $t-N$,*

$$\left(\mathbb{E}\left[\ell(X_{t+1}, W) - f_t(W) \,\middle|\, \mathcal{F}_{t-N}\right]\right)^+ \le |f(W) - f_{t-N}(W)| + N w_t f_{t-N}(W) + 2\|\ell(\cdot, W)\|_\infty \sup_{\mathbf{y}\in\Omega} \|P^{N+1}(\mathbf{y}, \cdot) - \pi\|_{TV}.$$

Proof Recall that for each $s \ge 0$, $\mathcal{F}_s$ denotes the σ-algebra generated by the history of underlying Markov chain $Y_0, Y_1, \ldots, Y_s$. Fix $\mathbf{y} \in \Omega$ and suppose $Y_{t-N} = \mathbf{y}$. Then by the Markov property, the distribution of Y_{t+1} conditional on $\mathcal{F}_{t-N}$ equals $P^{N+1}(\mathbf{y}, \cdot)$, where P denotes the transition kernel of the chain $(Y_t)_{t\in\mathbb{N}}$. Using the fact that $2\|\mu - \nu\|_{TV} = \sum_x |\mu(x) - \nu(x)|$ (see (Levin and Peres, 2017, Prop. 4.2)) and recalling $X_t = \varphi(Y_t)$ by (A1), it follows that

$$\begin{aligned}
\mathbb{E}\left[\ell(X_{t+1}, W) \,\middle|\, \mathcal{F}_{t-N}\right] &= \sum_{\mathbf{y}'\in\Omega} \ell(\varphi(\mathbf{y}'), W)\, P^{N+1}(\mathbf{y}, \mathbf{y}') \\
&= \sum_{\mathbf{y}'\in\Omega} \ell(\varphi(\mathbf{y}'), W)\, \pi(\mathbf{y}') + \sum_{\mathbf{y}'\in\Omega} \ell(\varphi(\mathbf{y}'), W)(P^{N+1}(\mathbf{y}, \mathbf{y}') - \pi(\mathbf{y}')) \\
&\le \sum_{\mathbf{y}'\in\Omega} \ell(\varphi(\mathbf{y}'), W)\, \pi(\mathbf{y}') + 2\|\ell(\cdot, W)\|_\infty \|P^{N+1}(\mathbf{y}, \cdot) - \pi\|_{TV} \\
&= f(W) + 2\|\ell(\cdot, W)\|_\infty \|P^{N+1}(\mathbf{y}, \cdot) - \pi\|_{TV}.
\end{aligned}$$

Also, observe that

$$\mathbb{E}\left[-f_t(W)\,\middle|\,\mathcal{F}_{t-N}\right] = -f_{t-N}(W)\prod_{k=t-N+1}^{t}(1-w_k) - \mathbb{E}\left[\sum_{k=t-N+1}^{t}\ell(X_k,W)w_k\prod_{j=k+1}^{t}(1-w_k)\,\middle|\,\mathcal{F}_{t-N}\right]$$

$$\leq -f_{t-N}(W) + f_{t-N}(W)\left(1-\prod_{k=t-N+1}^{t}(1-w_k)\right) \leq -f_{t-N}(W) + Nw_t f_{t-N}(W),$$

where we have used the fact that $\ell \geq 0$ and $w_k \in (0,1)$ is non-increasing in k. Then combining the two bounds and a triangle inequality give the assertion. ■

Next, we provide some probabilistic lemmas.

Lemma 10. *Under the assumptions (A1)-(A2) and (M1),*

$$\mathbb{E}\left[\sup_{W\in\mathcal{C}}\sqrt{t}\left|f(W) - \frac{1}{t}\sum_{s=1}^{t}\ell(X_s,W)\right|\right] = O(1).$$

Furthermore, $\sup_{W\in\mathcal{C}}\left|f(W) - \frac{1}{t}\sum_{s=1}^{t}\ell(X_s,W)\right| \to 0$ *almost surely as* $t\to\infty$.

Proof Let $\mathcal{F}$ denote the collection of functions $\ell(\varphi(\cdot),W):\Omega\to[0,\infty)$ indexed by $W\in\mathcal{C}$, which are bounded and measurable under (A1). The underlying Markov chain $(Y_t)_{t\in\mathbb{N}}$ has countable state space Ω and is positive recurrent under (M1). Then the second part of the statement is a direct consequence of the uniform SLLN for Markov chains (Levental, 1988, Thm. 5.8). For the first part, note that by the uniform functional CLT for Markov chains (Levental, 1988, Thm 5.9), the empirical process $\{g_t(W)\,|\,W\in\mathcal{C}\}$, $g_t(W) := \sqrt{t}(f(W) - t^{-1}\sum_{s=1}^{t}\ell(\varphi(Y_s),W))$ converges weakly to a centered Gaussian process $(\mathbf{X}_W)_{W\in\mathcal{C}}$ indexed by $\mathcal{C}$ (or equivalently, by $\mathcal{F}$), whose sample paths are bounded and uniformly continuous in the space $\ell^\infty(\mathcal{F})$ of bounded functions $\mathcal{F}\to\mathbb{R}$. Moreover, from the theory of Gaussian processes (see, e.g., (Dudley, 2010; Talagrand, 1987)) it is well known that for some universal constant $K>0$,

$$\mathbb{E}\left[\sup_{W\in\mathcal{C}}\mathbf{X}_W\right] \leq K\int_0^\infty\sqrt{\log N(\varepsilon)}\,\mathrm{d}\varepsilon,$$

where $N(\varepsilon)$ denotes the minimum number of ε-balls needed to cover the parameter space $\mathcal{C}$. Since $\mathcal{C}$ is compact by (A2), the right hand side is finite. By the weak convergence of the empirical process, it follows that the expectation in the assertion is uniformly bounded in t. This shows the assertion. ■

While the uniform convergence results in Lemma 10 applies to empirical loss functions of balanced weights (e.g., $w_t = 1/t$ for all $t\geq 1$), we may need a similar uniform convergence results for the general weights. The following lemma is due to Mairal (Mairal, 2013b, Lem B.7), which originally extended the uniform convergence result to weighted empirical loss functions with respect to i.i.d. input data. An identical argument gives the corresponding result in our Markovian case, but we provide it here for the sake of completeness.

Lemma 11. *Under the assumptions (A1)-(A2) and (M1), there exists a constant $C > 0$ such that*

$$\mathbb{E}\left[\sup_{W\in\mathcal{C}} |f(W) - f_t(W)|\right] \leq C w_t \sqrt{t}$$

for all $t \geq 1$. Furthermore, if $\sum_{t=1}^{\infty} w_t^2 \sqrt{t} < \infty$, then $\sup_{W\in\mathcal{C}} |f(W) - f_t(W)| \to 0$ almost surely as $t \to \infty$.

Proof See Appendix A. ■

Next, we use the concentration bound in Lemma 11 together with the mixing condition (M2) to show that the surrogate loss process $(\hat{f}_t(W_t))_{t\in\mathbb{N}}$ has the bounded positive expected variation.

Lemma 12. *Let $(W_t, H_t)_{t\geq 1}$ be a solution to the optimization problem (3). Suppose (A1)-(A2) and (M1) hold.*

(i) *Let $(a_t)_{t\in\mathbb{N}}$ be a sequence of non-decreasing non-negative integers such that $a_t \in o(t)$. Then there exists absolute constants $C_1, C_2, C_3 > 0$ such that for all sufficiently large $t \in \mathbb{N}$,*

$$\mathbb{E}\left[\left(\mathbb{E}\left[w_{t+1}\big(\ell(X_{t+1}, W_t) - f_t(W_t)\big) \,\middle|\, \mathcal{F}_{t-a_t}\right]\right)^+\right] \leq C_1 w_{t-a_t}^2 \sqrt{t} + C_2 w_t^2 a_t + C_3 w_t \sup_{\mathbf{x}\in\Omega} \|P^{a_t+1}(\mathbf{x}, \cdot) - \pi\|_{TV}.$$

(ii) *Further assume that (M2) holds. Then we have*

$$\sum_{t=0}^{\infty} \left(\mathbb{E}\left[\hat{f}_{t+1}(W_{t+1}) - \hat{f}_t(W_t)\right]\right)^+ \leq \sum_{t=0}^{\infty} w_{t+1} \left(\mathbb{E}\left[(\ell(X_{t+1}, W_t) - f_t(W_t))\right]\right)^+ < \infty.$$

Proof Recall that $(Y_t, W_t) \in \Omega \times \mathcal{C}$ for all $t \in \mathbb{N}$ and both Ω and $\mathcal{C}$ are compact. Since $\varphi : \Omega \to \mathbb{R}^{d\times n}$ is bounded, we have

$$L := \sup_{Y\in\Omega, W\in\mathcal{C}} \ell(\varphi(Y), W) < \infty.$$

Denote

$$\Delta_t := \sup_{\mathbf{y}\in\Omega} \|P^t(\mathbf{y}, \cdot) - \pi\|_{TV}.$$

Note that $\|f_s\|_\infty \leq L$ for any $s \geq 0$. Hence according to Propositions 9, we have

$$\left|\mathbb{E}\left[w_{t+1}\left(\ell(X_{t+1}, W_t) - f_t(W_t)\right) \,\middle|\, \mathcal{F}_{t-a_t}\right]\right| \leq w_t \left(\|f - f_{t-a_t}\|_\infty\right) + 2L w_t^2 a_t + L w_t \Delta_t. \quad (23)$$

Since $a_t = o(t)$, we have $a_t \leq t/2$ for all sufficiently large $t \in \mathbb{N}$. Then by Lemma 11, there exists a constant $C_1 > 0$ such that for all sufficiently large $t \in \mathbb{N}$,

$$\mathbb{E}\left[\|f - f_{t-a_t}\|_\infty\right] \leq C_1 w_{t-a_t}\sqrt{t-a_t}.$$

Noting that w_s is non-increasing in s, this gives

$$\mathbb{E}\left[w_t\|f - f_{t-a_t}\|_\infty\right] \leq C_1 w_{t-a_t}^2\sqrt{t}$$

for all sufficiently large $t \geq 1$. Hence taking expectation on both sides of (23) with respect to the information from time $t - a_t$ to t yields the first assertion.

Now we show the second assertion. The first inequality in the assertion follows by Proposition 5 (i). To show that the last expression is finite, denote $Z_t = w_{t+1}^{-1}(\ell(X_{t+1}, W) - f_t(W))$. Note that by the first assertion and (M2), we have

$$\sum_{t=1}^{\infty}\mathbb{E}\left[\left(\mathbb{E}\left[Z_t \,\middle|\, \mathcal{F}_{t-a_t}\right]\right)^+\right] < \infty.$$

Then by iterated expectation and Jensen's inequality, it follows that

$$\sum_{t=1}^{\infty}(\mathbb{E}[Z_t])^+ = \sum_{t=1}^{\infty}\left(\mathbb{E}\left[\mathbb{E}\left[Z_t \,\middle|\, \mathcal{F}_{t-a_t}\right]\right]\right)^+ \leq \sum_{t=1}^{\infty}\mathbb{E}\left[\left(\mathbb{E}\left[Z_t \,\middle|\, \mathcal{F}_{t-a_t}\right]\right)^+\right] < \infty.$$

This completes the proof of (ii). ■

Lemma 13. *Let $(W_t, H_t)_{t\geq 1}$ be a solution to the optimization problem* (3)*. Suppose (A1)-(A2) and (M1)-(M2) hold. Then the following hold.*

(i) $\mathbb{E}[\hat{f}_t(W_t)]$ *converges as* $t \to \infty$.

(ii) $$\mathbb{E}\left[\sum_{t=0}^{\infty} w_{t+1}\left(\hat{f}_t(W_t) - f_t(W_t)\right)\right] = \sum_{t=0}^{\infty} w_{t+1}\left(\mathbb{E}[\hat{f}_t(W_t)] - \mathbb{E}[f_t(W_t)]\right) < \infty.$$

(iii) $\sum_{t=0}^{\infty} w_{t+1}\left(\hat{f}_t(W_t) - f_t(W_t)\right) < \infty$ *almost surely.*

Proof In order to show that $\mathbb{E}[\hat{f}_t(W_t)]$ converges as $t \to \infty$, since $f_t(W_t)$ is bounded uniformly in t, it suffices to show that the sequence $(\mathbb{E}[\hat{f}_t(W_t)])_{t\in\mathbb{N}}$ has a unique limit point. To this end, observe that for any $x, y \in \mathbb{R}$, $(x+y)^+ \leq x^+ + y^+$. Note that, for each $m, n \geq 1$ with $m > n$,

$$\begin{aligned}\left(\mathbb{E}[\hat{f}_m(W_m)] - \mathbb{E}[\hat{f}_n(W_n)]\right)^+ &\leq \sum_{k=n}^{m-1}\left(\mathbb{E}[\hat{f}_{k+1}(W_{k+1})] - \mathbb{E}[f_k(W_k)]\right)^+ \\ &\leq \sum_{k=n}^{\infty}\left(\mathbb{E}[\hat{f}_{k+1}(W_{k+1})] - \mathbb{E}[\hat{f}_k(W_k)]\right)^+.\end{aligned}$$

The last expression converges to zero as $n \to \infty$ by Lemma 12 (ii). This implies that the sequence $(\mathbb{E}[\hat{f}_t(W_t)])_{t\in\mathbb{N}}$ has a unique limit point, as desired.

The first equality follows from Fubini's theorem by noting that $\hat{f}_t - f_t \geq 0$. On the other hand, by using Proposition 5 (ii),

$$\sum_{t=0}^{\infty} w_{t+1} \left(\mathbb{E}[\hat{f}_t(W_t)] - \mathbb{E}[f_t(W_t)] \right) \leq \sum_{t=0}^{\infty} w_{t+1} \left(\mathbb{E}[\ell(X_{t+1}, W_t)] - f_t(W_t)] \right)^+ - \sum_{t=0}^{\infty} \left(\mathbb{E}[\hat{f}_{t+1}(W_{t+1})] - \mathbb{E}[\hat{f}_t(W_t)] \right).$$

The first sum on the right hand side is finite by Lemma 12 (ii), and the second sum is also finite since we have just shown that $\mathbb{E}[\hat{f}_t(W_t)]$ converges as $t \to \infty$. This shows (ii). Lastly, recall that non-negative random variable of finite expectation must be finite almost surely. Hence (iii) follows directly from (ii). ■

Now we prove the first main result in this paper, Theorem 1.

Proof [Proof of Theorem 1] Suppose (A1)-(A2) and (M1)-(M2) hold. We first show (ii). Recall Lemma 13 (iii). Both $\hat{f}_t$ and f_t are uniformly bounded and Lipschitz by Proposition 15. Hence writing $h_t = \hat{f}_t - f_t$, using Proposition 7, there exists a constant $C > 0$ such that for all $t \geq 1$,

$$\begin{aligned} |h_{t+1}(W_{t+1}) - h_t(W_t)| &\leq |h_{t+1}(W_{t+1}) - h_{t+1}(W_t)| + |h_{t+1}(W_t) - h_t(W_t)| \\ &\leq C\|W_{t+1} - W_t\|_F + \left| \left(\hat{f}_{t+1}(W_t) - \hat{f}_t(W_t) \right) - (f_{t+1}(W_t) - f_t(W_t)) \right| \\ &= C\|W_{t+1} - W_t\|_F + w_{t+1}|\hat{f}_t(W_t) - f_t(W_t)| = O(w_{t+1}). \end{aligned}$$

Thus, according to Proposition 16, it follows from Lemma 13 (ii) that

$$\lim_{t\to\infty} \left(\hat{f}_t(W_t) - f_t(W_t) \right) = 0 \qquad \text{a.s.}$$

Moreover, for all $t \geq 1$, triangle inequality gives

$$|f(W_t) - \hat{f}_t(W_t)| \leq \left(\sup_{W\in\mathcal{C}} |f(W) - f_t(W)| \right) + |f_t(W_t) - \hat{f}_t(W_t)|.$$

The right hand side converges to zero almost surely as $t \to \infty$ by what we have just shown above and Lemma 11. This shows (ii).

Next, we show (i). Recall that $\mathbb{E}[\hat{f}_t(W_t)]$ converges by Lemma 13. The Jensen's inequality and the bounds imply

$$|\mathbb{E}[h_{t+1}(W_{t+1})] - \mathbb{E}[h_t(W_t)]| \leq \mathbb{E}\left[|h_{t+1}(W_{t+1}) - h_t(W_t)|\right] = O(w_{t+1}).$$

Since $\mathbb{E}[\hat{f}_t(W_t)] \geq \mathbb{E}[f_t(W_t)]$, Lemma 13 (i)-(ii) and Lemma 16 give

$$\lim_{t\to\infty} \mathbb{E}[f_t(W_t)] = \lim_{t\to\infty} \mathbb{E}[\hat{f}_t(W_t)] + \lim_{t\to\infty} \left(\mathbb{E}[f_t(W_t)] - \mathbb{E}[\hat{f}_t(W_t)] \right) = \lim_{t\to\infty} \mathbb{E}[\hat{f}_t(W_t)] \in (1, \infty).$$

This shows (i).

Lastly, we show (iii). Denote $g_t(W) = \mathrm{tr}(WA_tW^T) - 2\mathrm{tr}(WB_t)$ and $\hat{f}_t(W) = g_t(W) + r_t$ (see (13)). Note that $\nabla_W g_t = 2(WA_t - B_t)$. We will first show

$$\limsup_{t\to\infty} \|\nabla_W f(W_t) - \nabla_W g_t(W_t)\|_F = 0. \tag{24}$$

First choose a subsequence $(t_k)_{k\geq 0}$ such that $\|\nabla_W f(W_{t_k}) - \nabla_W g_{t_k}(W_{t_k})\|_F$ converges. Recall that the sequence $(W_t, A_t, B_t, r_t)_{t\in\mathbb{N}}$ is bounded by Proposition 14 and (A1)-(A2). Hence we may choose a further subsequence of $(t_k)_{k\geq 0}$, which we will denote by $(s_k)_{k\in\mathbb{N}}$, so that $(W_{s_k}, A_{s_k}, B_{s_k}, r_{s_k})$ converges to some $(W_\infty, A_\infty, B_\infty, r_\infty)$ in $\mathbb{R}^{d\times r} \times \mathbb{R}^{r\times r} \times \mathbb{R}^{r\times n} \times \mathbb{R}$ a.s. as $k \to \infty$. Define a function

$$\hat{f}(W) = \mathrm{tr}(WA_\infty W^T) - 2\mathrm{tr}(WB_\infty) + r_\infty.$$

Then we write

$$\|\nabla_W f(W_{s_k}) - \nabla_W g_{s_k}(W_{s_k})\|_F \leq \|\nabla_W f(W_{s_k}) - \nabla_W f(W_\infty)\|_F + \|\nabla_W f(W_\infty) - \nabla_W \hat{f}(W_\infty)\|_F$$
$$+ \|\nabla_W \hat{f}(W_\infty) - \nabla_W g_{s_k}(W_{s_k})\|_F.$$

By the choice of $(s_k)_{k\in\mathbb{N}}$, the first term in the right hand side vanishes as $k \to \infty$. For the second term, note that $\hat{f}_t \geq f_t$ for all $t \in \mathbb{N}$ and over all $\mathcal{C}$. Hence, for each $W \in \mathcal{C}$, almost surely,

$$\hat{f}(W) = \lim_{k\to\infty} \hat{f}_{s_k}(W) \geq \lim_{k\to\infty} f_{s_k}(W) = f(W),$$

where the last equality follows from Markov chain ergodic theorem (see, e.g., (Durrett, 2010, Thm 6.2.1, Ex. 6.2.4) or (Meyn and Tweedie, 2012, Thm. 17.1.7)). Moreover, by part (i), we know that

$$\hat{f}(W_\infty) = \lim_{k\to\infty} \hat{f}_{s_k}(W_{s_k}) = f(W_\infty) \in (0, \infty)$$

almost surely. Hence by using a Taylor expansion and the fact that $\nabla_W f$ is Lipschitz (see (C1)), it follows that

$$\nabla_W f(W_\infty) = \nabla_W \hat{f}(W_\infty).$$

For the last term, note that

$$\|\nabla_W \hat{f}(W_\infty) - \nabla_W g_{s_k}(W_{s_k})\|_F = \|2W(A_\infty - A_{s_k}) - 2(B_\infty^T - B_{s_k})\|_F \to 0,$$

as $A_{s_k} \to A_\infty$ and $B_{s_k} \to B_\infty$ by the choice of $(s_k)_{k\geq 0}$.

$$\limsup_{k\to\infty} \|\nabla_W f(W_{s_k}) - \nabla_W g_{s_k}(W_{s_k})\|_F = 0.$$

Since $(s_k)_{k\in\mathbb{N}}$ is a further subsequence of $(t_k)_{k\geq 0}$ and since $\|\nabla_W f(W_{t_k}) - \nabla_W g_t(W_{t_k})\|_F$ converges along $(t_k)_{k\geq 0}$, the same also holds for $(t_k)_{k\geq 0}$. This shows (24).

To conclude that W_∞ is a local extremum of f, it is enough to show that $\nabla_W f(W_\infty)$ is in the normal cone of $\mathcal{C}$ at W_∞. Choose a subsequence $(t_k)_{k\geq 0}$ such that $(A_{t_k}, B_{t_k}, W_{t_k})$ converges to $(A_\infty, B_\infty, W_\infty)$. According to Assumption (A2), there exists some convex part $\mathcal{C}_j$ of $\mathcal{C}$ such that for all sufficiently large $k \geq 1$, $W_{t_k} \in \mathcal{C}_j$. Recall that W_t is the minimizer of the quadratic function $g_t(W) = \mathrm{tr}(WA_tW^T) - 2\mathrm{tr}(WB_t)$ in $\mathcal{C} \cap \mathcal{E}_t$, where $\mathcal{E}_t$ denotes the ellipsoid $\mathrm{tr}((B_t^T - WA_t)(W_{t-1} - W)) \leq 0$. By Proposition 6 (iii), minimizing g_t over $\mathcal{C}_j \cap \mathcal{E}_t$ is equivalent to minimizing g_t over $\mathcal{C}_j$. Hence W_{t_k} is also the minimizer of g_{t_k} over $\mathcal{C}_j$ for all sufficiently large k. Since $\mathcal{C}$ is a disjoint union of convex parts (see Assumption (A2)), this verifies that for all sufficiently large k, $\nabla_W g_{t_k}(W_{t_k})$ is in the normal cone of the constraint set $\mathcal{C}$ at W_{t_k} (see., e.g., (Boyd and Vandenberghe, 2004)). Then (24) and continuity of the gradients of f and g_t verifies that $\nabla_W f(W_\infty)$ is in the normal cone of $\mathcal{C}$ at W_∞, as desired. ■

Acknowledgement

HL is partially supported by NSF Grant DMS-2010035 and is grateful for helpful discussions with Yacoub Kureh and Joshua Vendrow for network denoising applications of network dictionary learning. DN is grateful to and was partially supported by NSF CAREER DMS #1348721 and NSF BIGDATA #1740325. LB was supported by the Institute for Advanced Study Charles Simonyi Endowment, ARO YIP award W911NF1910027, NSF CAREER award CCF-1845076, and AFOSR YIP award FA9550-19-1-0026.

References

Michael W. Berry and Murray Browne. Email surveillance using non-negative matrix factorization. *Computational & Mathematical Organization Theory*, 11(3):249–264, 2005.

Michael W. Berry, Murray Browne, Amy N. Langville, V. Paul Pauca, and Robert J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational statistics & data analysis*, 52(1):155–173, 2007.

David Blei, Lawrence Carin, and David Dunson. Probabilistic topic models: A focus on graphical model design and applications to document and image analysis. *IEEE signal processing magazine*, 27(6):55, 2010.

David M. Blei, Andrew Y. Ng, and Michael I Jordan. Latent dirichllocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

Rostyslav Boutchko, Debasis Mitra, Suzanne L. Baker, William J. Jagust, and Grant T Gullberg. Clustering-initiated factor analysis application for tissue classification in dynamic brain positron emission tomography. *Journal of Cerebral Blood Flow & Metabolism*, 35(7):1104–1111, 2015.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Yang Chen, Xiao Wang, Cong Shi, Eng Keong Lua, Xiaoming Fu, Beixing Deng, and Xing Li. Phoenix: A weight-based network coordinate system using matrix factorization. *IEEE Transactions on Network and Service Management*, 8(4):334–347, 2011.

Fernanda B. Correia, Edgar D. Coelho, José L. Oliveira, and Joel P. Arrais. Handling noise in protein interaction networks. *BioMed Research International*, 2019.

Richard M. Dudley. Sample functions of the gaussian process. In *Selected Works of RM Dudley*, pages 187–224. Springer, 2010.

Rick Durrett. *Probability: Theory and Examples*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, UK, fourth edition, 2010.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

Donald L. Fisk. Quasi-martingales. *Transactions of the American Mathematical Society*, 120(3): 369–389, 1965.

Nicolas Gillis. The why and how of nonnegative matrix factorization. *Regularization, Optimization, Kernels, and Support Vector Machines*, 12(257), 2014.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864, 2016.

Naiyang Guan, Dacheng Tao, Zhigang Luo, and Bo Yuan. Online nonnegative matrix factorization with robust stochastic approximation. *IEEE Transactions on Neural Networks and Learning Systems*, 23(7):1087–1099, 2012.

Katja Kovács, István A .and Luck, Kerstin Spirohn, Yang Wang, Carl Pollis, Sadie Schlabach, Wenting Bian, Dae-Kyum Kim, Nishka Kishore, and Tong Hao. Network-based prediction of protein interactions. *Nature Communications*, 10(1):1240, 2019.

Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999.

Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, pages 556–562, 2001.

Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*, pages 801–808, 2007.

Hyekyoung Lee, Jiho Yoo, and Seungjin Choi. Semi-supervised nonnegative matrix factorization. *IEEE Signal Processing Letters*, 17(1):4–7, 2009.

Wilhelm Lenz. Beitršge zum verstšndnis der magnetischen eigenschaften in festen kšrpern. *Physikalische Z*, 21:613–615, 1920.

Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. *http://snap.stanford.edu/data*, June 2014.

Jure Leskovec and Julian J. Mcauley. Learning to discover social circles in ego networks. In *Advances in Neural Information Processing Systems*, pages 539–547, 2012.

Shlomo Levental. Uniform limit theorems for harris recurrent markov chains. *Probability theory and related fields*, 80(1):101–118, 1988.

David A. Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for Information Science and Technology*, 58(7):1019–1031, 2007.

Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A*, 390(6): 1150–1170, 2011.

Eyal Lubetzky and Allan Sly. Critical ising on the square lattice mixes in polynomial time. *Communications in Mathematical Physics*, 313(3):815–836, 2012.

Hanbaek Lyu, Facundo Memoli, and David Sivakoff. Sampling random graph homomorphisms and applications to network data analysis. *arXiv:1910.09483*, 2019.

Hanbaek Lyu, Yacoub Kureh, Joshua Vendrow, and Mason Porter. Learning low-rank latent mesoscale structures in networks. *In preparation*, 2020.

Julien Mairal. Optimization with first-order surrogate functions. In *International Conference on Machine Learning (ICML)*, pages 783–791, 2013a.

Julien Mairal. Stochastic majorization-minimization algorithms for large-scale optimization. In *Advances in Neural Information Processing Systems*, pages 2283–2291, 2013b.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, 2010.

Athanasius FM Marée, Verônica A Grieneisen, and Paulien Hogeweg. The cellular potts model and biophysical properties of cells, tissues and morphogenesis. In *Single-cell-based models in biology and medicine*, pages 107–136. Springer, 2007.

Barry M. McCoy and Tai Tsun Wu. *The two-dimensional Ising model*. Courier Corporation, 2014.

Song Mei, Yu Bai, and Andrea Montanari. The landscape of empirical risk for nonconvex losses. *The Annals of Statistics*, 46(6A):2747–2774, 2018.

Aditya Krishna Menon and Charles Elkan. Link prediction via matrix factorization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 437–452. Springer, 2011.

Arthur Mensch, Julien Mairal, Bertrand Thirion, and Gaël Varoquaux. Stochastic subsampling for factorizing huge matrices. *IEEE Transactions on Signal Processing*, 66(1):113–128, 2017.

Sean P. Meyn and Richard L. Tweedie. *Markov Chains and Stochastic Stability*. Springer-Verlag, Heidelberg, Germany, 2012.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning (ICML)*, pages 807–814, 2010.

Noriyuki Bob Ouchi, James A. Glazier, Jean-Paul Rieu, Arpita Upadhyaya, and Yasuji Sawada. Improving the realism of the cellular potts model in simulations of biological cells. *Physica A: Statistical Mechanics and its Applications*, 329(3-4):451–458, 2003.

Rose Oughtred, Chris Stark, Bobby-Joe Breitkreutz, Jennifer Rust, Lorrie Boucher, Christie Chang, Nadine Kolas, Lara O'Donnell, Genie Leung, and Rochelle McAdam. The biogrid interaction database: 2019 update. *Nucleic acids research*, 47(D1):D529–D541, 2019.

Jianhao Peng, Olgica Milenkovic, and Abhishek Agarwal. Online convex matrix factorization with representative regions. In *Advances in Neural Information Processing Systems*, pages 13242–13252, 2019.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 701–710, 2014.

Sirisha Rambhatla, Xingguo Li, and Jarvis Haupt. Noodl: Provable online dictionary learning and sparse coding. *International Conference on Learning Representations (ICLR)*, 2019.

K Murali Rao. Quasi-martingales. *Mathematica Scandinavica*, 24(1):79–92, 1969.

Bin Ren, Laurent Pueyo, Guangtun Ben Zhu, John Debes, and Gaspard Duchêne. Non-negative matrix factorization: robust extraction of extended structures. *The Astrophysical Journal*, 852 (2):104, 2018.

Arkadiusz Sitek, Grant T. Gullberg, and Ronald H. Huesman. Correction for ambiguous solutions in factor analysis using a penalized least squares objective. *IEEE transactions on medical imaging*, 21(3):216–225, 2002.

Mark Steyvers and Tom Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

András Szabó and Roeland MH Merks. Cellular potts modeling of tumor growth, tumor invasion, and tumor evolution. *Frontiers in oncology*, 3:87, 2013.

Michel Talagrand. Regularity of gaussian processes. *Acta mathematica*, 159:99–149, 1987.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077, 2015.

Leo Taslaman and Björn Nilsson. A framework for regularized non-negative matrix factorization, with application to the analysis of gene expression data. *PloS One*, 7(11):e46331, 2012.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

Vladimir Vapnik. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pages 831–838, 1992.

Fa-Yueh Wu. The potts model. *Reviews of modern physics*, 54(1):235, 1982.

Lin F. Yang, Vladimir Braverman, Tuo Zhao, and Mengdi Wang. Online factorization and partition of complex networks from random walks. *Uncertainty in Artificial Intelligence*, 2019.

Renbo Zhao, Vincent YF Tan, and Huan Xu. Online nonnegative matrix factorization with general divergences. *arXiv preprint arXiv:1608.00075*, 2016.

Renbo Zhao, Vincent Tan, and Huan Xu. Online nonnegative matrix factorization with general divergences. In *Artificial Intelligence and Statistics*, pages 37–45, 2017.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Appendix A. Auxiliary proofs and lemmas

In this appendix, we provide some auxiliary proofs and statements that we use in the proof of Theorem 1. Among the results we provide here, the proof of Proposition 7 is original and the rest are due originally due to (Mairal et al., 2010; Mairal, 2013b).

Proof [Proof of Proposition 5] To see the equality in (i), note that

$$\begin{aligned} f_{t+1}(W_t) - f_t(W_t) &= (1 - w_{t+1}) f_t(W_t) + w_{t+1}\ell(X_{t+1}, W_t) - f_t(W_t) \\ &= w_{t+1}(\ell(X_{t+1}, W_t) - f_t(W_t)). \end{aligned}$$

Next, we have

$$\hat{f}_{t+1}(W_t) = (1 - w_{t+1})\hat{f}_t(W_t) + w_{t+1}\ell(X_{t+1}, W_t)$$

for all $t \in \mathbb{N}$, so it follows that

$$\begin{aligned} &\hat{f}_{t+1}(W_{t+1}) - \hat{f}_t(W_t) \\ &\quad = \hat{f}_{t+1}(W_{t+1}) - \hat{f}_{t+1}(W_t) + \hat{f}_{t+1}(W_t) - \hat{f}_t(W_t) \\ &\quad = \hat{f}_{t+1}(W_{t+1}) - \hat{f}_{t+1}(W_t) + (1 - w_{t+1})\hat{f}_t(W_t) + w_{t+1}\ell(X_{t+1}, W_t) - \hat{f}_t(W_t)) \\ &\quad = \hat{f}_{t+1}(W_{t+1}) - \hat{f}_{t+1}(W_t) + w_{t+1}(\ell(X_{t+1}, W_t) - f_t(W_t)) + w_{t+1}(f_t(W_t) - \hat{f}_t(W_t)). \end{aligned}$$

Then the inequalities in both (i) and (ii) follows by noting that $\hat{f}_{t+1}(W_{t+1}) \leq \hat{f}_{t+1}(W_t)$ and $f_t \leq \hat{f}_t$. ■

For each $X \in \mathbb{R}^{d\times n}$, $W \in \mathbb{R}^{d\times r}$, and $H \in \mathbb{R}^{r\times n}$, denote $\ell(X, W, H) = \|X - WH\|_F^2 + \lambda\|H\|_1$ and define

$$H^{\mathrm{opt}}(X, W) = \operatorname*{arg\,min}_{H\in\mathcal{C}'\subseteq\mathbb{R}^{r\times n}} \ell(X, W, H). \tag{25}$$

Note that under (C2), there exists a unique minimizer of the function in the right hand side, with which we identify as $H^{\mathrm{opt}}(X, W)$.

Proposition 14. *Assume (A1) and let $R = R(\varphi(\Omega)) < \infty$ be as defined in (4). Then the following hold:*

(i) *For all $X \in \Omega$ and $W \in \mathcal{C}$,*

$$\|H^{\mathrm{opt}}(X, W)\|_F^2 \leq \lambda^{-2}R^4.$$

(ii) *For any sequence $(X_t)_{t\geq 1} \subseteq \Omega$ and $(W_t)_{t\geq 1} \subseteq \mathcal{C}$, define A_t and B_t recursively as in (3). Then for all $t \geq 1$, we have*

$$\|A_t\|_F \leq \lambda^{-2}R^4, \qquad \|B_t\|_F \leq \lambda^{-1}R^3.$$

Proof From (25), we have

$$\lambda\|H^{\mathrm{opt}}(X, W)\|_1 \leq \inf_{H\in\mathcal{C}'\subseteq\mathbb{R}^{r\times n}} \left(\|X - WH\|_F^2 + \lambda\|H\|_1\right) \leq \|X\|_F^2 \leq R^2.$$

Note that $\|H\|_F^2 \leq \|H\|_1^2$ for any H. This yields (i). To get (ii), we observe $\|XY\|_F \leq \|X\|_F\|Y\|_F$ from the Cauchy-Schwarz inequality. Then (ii) follows immediately from (i) and triangle inequality. ■

Next, we show the Lipschitz continuity of the loss function $\ell(\cdot, \cdot)$. Since Ω and $\mathcal{C}$ are both compact, this also implies that $\hat{f}_t$ and f_t are Lipschitz for all $t \in \mathbb{N}$.

Proposition 15. *Suppose (A1) and (A2) hold, and let* $M = 2R(\varphi(\Omega)) + 2R(\mathcal{C})R(\varphi(\Omega))^2/\lambda$. *Then for each* $X_1, X_2 \in \Omega$ *and* $W_1, W_2 \in \mathcal{C}$,

$$|\ell(X_1, W_1) - \ell(X_2, W_2)| \leq M \left(\|X_1 - X_2\|_F + \lambda^{-1} R(\varphi(\Omega)) \|W_1 - W_2\|_F \right).$$

Proof Fix $X \in \Omega \subseteq \mathbb{R}^{d\times n}$ and $W_1, W_2 \in \mathcal{C}$. Denote $H^* = H^{\mathrm{opt}}(X_2, W_2)$ and $H_* = H^{\mathrm{opt}}(X_1, W_1)$. According to Proposition 14, the Frobenius norm of H^* and H_* are uniformly bounded by $R(\varphi(\Omega))^2/\lambda$. Note that for any $A, B \in \Omega$, the triangle inequality implies

$$\begin{aligned} \|a\|_F^2 - \|b\|_F^2 &= \left(\|a\|_F - \|b\|_F \right) \left(\|a\|_F + \|b\|_F \right) \\ &\leq \|a - b\|_F \left(\|a\|_F + \|b\|_F \right). \end{aligned}$$

Also, the Cauchy-Schwartz inequality, (A1)-(A2), and Proposition 14 imply

$$\begin{aligned} \|X_1 - W_1 H^*\|_F + \|X_2 - W H^*\|_F &\leq \|X_1\|_F + \|W_1 H^*\|_F + \|X_2\|_F + \|W_2 H^*\|_F \\ &\leq \|X_1\|_F + \|W_1\|_F \|H^*\|_F + \|X_2\|_F \|H*\|_F \\ &\leq 2R(\varphi(\Omega)) + 2R(\mathcal{C})R(\varphi(\Omega))^2/\lambda. \end{aligned}$$

Denoting $M = 2R(\varphi(\Omega)) + 2R(\mathcal{C})R(\varphi(\Omega))^2/\lambda$, we have

$$\begin{aligned} |\ell(X_1, W_1) - \ell(X_2, W_2)| &\leq \left| \left(\|X_1 - W_1 H^*\|_F^2 + \lambda \|H^*\|_1 \right) - \left(\|X_2 - W_2 H^*\|_F^2 + \lambda \|H^*\|_1 \right) \right| \\ &\leq M \| (X_1 - X_2) + (W_2 - W_1) H^* \|_F \\ &\leq M \left(\|X_1 - X_2\|_F + \|W_1 - W_2\|_F \cdot \|H^*\|_F \right) \\ &\leq M \left(\|X_1 - X_2\|_F + \lambda^{-1} R(\varphi(\Omega))^2 \|W_1 - W_2\|_F \right). \end{aligned}$$

This shows the assertion. ■

Proof [Proof of Proposition 6] First note that (iii) follows easily from (i) and (ii). Namely, since g is strictly convex and $\mathcal{C}$ is convex, $W_t = \arg\min_{W\in\mathcal{C}} g(W)$ is uniquely defined and $g(\lambda W_t + (1 - \lambda)W_{t-1})$ is monotone decreasing in $\lambda \in [0, 1]$. Hence (iii) follows from (i) and (ii).

In order to show (i) and (ii), we first derive a general second order Taylor expansion for the function g. Let $W, W', \bar{W} \in \mathbb{R}^{d\times r}$ be arbitrary. Then a simple calculation gives that

$$g(W) - g(\bar{W}) = \mathrm{tr}((W - \bar{W})A(W - \bar{W})^T) + 2\mathrm{tr}((W - \bar{W})(A\bar{W}^T - B)),$$

and also a similar expression for $g(W')$. Writing $\Delta W = W' - W$, we get

$$\begin{aligned} g(W) - g(W') &= \mathrm{tr}((W - \bar{W})A(W - \bar{W})^T) - \mathrm{tr}((W' - \bar{W})A(W' - \bar{W})^T) \\ &\quad + 2\mathrm{tr}((W - \bar{W})(A\bar{W}^T - B)) - 2\mathrm{tr}((W' - \bar{W})(A\bar{W}^T - B)) \\ &= -2\mathrm{tr}((W' - \bar{W})A(\Delta W)^T) + \mathrm{tr}(\Delta W A(\Delta W)^T) \qquad (26) \\ &\quad + 2\mathrm{tr}((W - \bar{W})(A\bar{W}^T - B)) + 2\mathrm{tr}((\bar{W} - W')(A\bar{W}^T - B)). \qquad (27) \end{aligned}$$

To show (i), let $W_1, W_2 \in \mathbb{R}^{d\times r}$ be such that $\mathrm{tr}((B^T - W_2 A)(W_1 - W_2)^T) \leq 0$. By taking $W = W_1$, $W' = W_2$, and $\bar{W} = (A^{-1}B)^T$ in the above expansion, we get

$$\begin{aligned} g(W_1) - g(W_2) &= 2\mathrm{tr}((\bar{W} - W_2)A(\Delta W)^T) + \mathrm{tr}(\Delta W A(\Delta W)^T) \\ &= \mathrm{tr}((B^T - W_2 A)(W_2 - W_1)^T) + \mathrm{tr}(\Delta W A(\Delta W)^T) \geq \mathrm{tr}(\Delta W A(\Delta W)^T). \end{aligned}$$

This shows (22), as desired.

To show (ii), fix $W_1, W_2 \in \mathbb{R}^{d\times r}$ and $W^{(\lambda)} = \lambda W_2 + (1-\lambda)W_1$ for $\lambda \in [0,1]$. Then by taking $W = \bar{W} = W^{(\lambda)}$ and $W' = W_2$ in (26)-(27), we get

$$g(W^{(\lambda)}) - g(W_2) = -\mathrm{tr}(\Delta W A(\Delta W)^T) + 2\mathrm{tr}((W_\lambda - W_2)(AW_\lambda^T - B)).$$

Suppose $g(W^{(\lambda)})$ is monotone decreasing in $\lambda \in [0,1]$. In particular, $g(W^{(\lambda)}) \geq g(W_2)$. Since A is positive definite, the above equation gives

$$2\mathrm{tr}((W_\lambda - W_2)(AW_\lambda^T - B)) \leq -\mathrm{tr}(\Delta W A(\Delta W)^T) \leq 0.$$

This yields, for all $\lambda \in [0,1)$,

$$\mathrm{tr}((W_1 - W_2)(AW_\lambda^T - B)) \leq 0.$$

By letting $\lambda \nearrow 1$, this gives the desired inequality

$$\mathrm{tr}((B^T - W_\lambda A)(W_1 - W_2)^T) = \mathrm{tr}((W_1 - W_2)(AW_2^T - B)) \leq 0.$$

■

Proof [Proof of Proposition 7] The argument is almost the same as that of (Mairal et al., 2010, Lem.1). The only difference is that we use Proposition 6 for a second order growth property for non-convex constraint set $\mathcal{C}$ for the quadratic optimization problem for W_t with additional constraint, as in (3).

Let A_t and B_t be as in (3). Denote $\hat{g}_{t+1}(W) = \mathrm{tr}(WA_{t+1}W^T) - 2\mathrm{tr}(WB_t)$ and $\hat{h}_{t+1} := \hat{g}_t - \hat{g}_{t+1}$. We first claim that there exists a constant $c > 0$ such that

$$|\hat{h}_{t+1}(W) - \hat{h}_{t+1}(W')| \leq c w_{t+1}\|W - W'\|_F \tag{28}$$

for all $W, W' \in \mathcal{C}$ and $t \in \mathbb{N}$. To see this, we first write

$$\hat{h}_{t+1}(W) = \mathrm{tr}(W(A_t - A_{t+1})W^T) - 2\mathrm{tr}(W(B_t - B_{t+1})).$$

The Cauchy-Schwartz inequality yields $\mathrm{tr}(A^TB) = \sum_{i,j} A_{ij}B_{ij} \leq \|A\|_F\|B\|_F$, so we have

$$\begin{aligned}\|\hat{h}_{t+1}(W) - \hat{h}_{t+1}(W')\|_F &\leq \left|\mathrm{tr}\left((W - W')(A_t - A_{t+1})W^T\right)\right| + \left|\mathrm{tr}\left(W'(A_t - A_{t+1})(W - W')^T\right)\right| \\ &\quad + 2\left|\mathrm{tr}(W - W')(B_t - B_{t+1})\right| \\ &\leq 2\left(R(\mathcal{C})\|A_t - A_{t+1}\|_F + \|B_t - B_{t+1}\|_F\right) \cdot \|W - W'\|_F,\end{aligned}$$

where $R(\mathcal{C}) = \sup_{W\in\mathcal{C}}\|W\|_F < \infty$ by (A2). Note that $\|H_t - H^{\mathrm{opt}}(X_t, W_{t-1})\|_F = O((\log t)^{-2})$ implies that there exists a constant $c_2 > 0$ such that $\|H_t\|_F \leq \|H^{\mathrm{opt}}(X_t, W_{t-1})\|_F + c_2$ for all $t \in \mathbb{N}$. Hence by Proposition 14, it follows that $\|A_t\|_F$ and $\|B_t\|_F$ are uniformly bounded in t. Thus there exists a constant $C > 0$ such that for all $t \in \mathbb{N}$,

$$\|A_t - A_{t+1}\|_F = w_{t+1}\|A_t - H_{t+1}H_{t+1}^T\|_F \leq C w_{t+1},$$

and similarly

$$\|B_t - B_{t+1}\|_F \leq C' w_{t+1}$$

for some other constant $C' > 0$. Hence the claim (28) follows.

To finish the proof, according to the assumptions (A2) and (C2), we first apply Proposition 6 (i) to deduce the following second order growth condition for all $t \in \mathbb{N}$:

$$\hat{g}_{t+1}(W_t) - \hat{g}_{t+1}(W_{t+1}) \geq \kappa_1\|W_t - W_{t+1}\|_F^2 \geq 0. \tag{29}$$

Using the inequalities $\hat{g}_{t+1}(W_{t+1}) \le \hat{g}_{t+1}(W_t)$ and $\hat{g}_t(W_t) \le \hat{g}_t(W_{t+1})$ given by (29), we deduce

$$\begin{aligned}
0 &\le \hat{g}_{t+1}(W_t) - \hat{g}_{t+1}(W_{t+1}) \\
&= \hat{g}_{t+1}(W_t) - \hat{g}_t(W_t) + [\hat{g}_t(W_t) - \hat{g}_t(W_{t+1})] + \hat{g}_t(W_{t+1}) - \hat{g}_{t+1}(W_{t+1}) \\
&\le \hat{g}_{t+1}(W_t) - \hat{g}_t(W_t) + \hat{g}_t(W_{t+1}) - \hat{g}_{t+1}(W_{t+1}) = \hat{h}_{t+1}(W_{t+1}) - \hat{h}_{t+1}(W_t),
\end{aligned}$$

Hence by (29) and the claim (28), we get

$$\kappa_1 \|W_t - W_{t+1}\|_F^2 \le \hat{h}_{t+1}(W_{t+1}) - \hat{h}_{t+1}(W_t) \le c w_{t+1} \|W_t - W_{t+1}\|_F.$$

This shows the assertion. ■

Proof [Proof of Lemma 11] Fix $t \in \mathbb{N}$. Recall the weighted empirical loss $f_t(W)$ defined recursively using the weights $(w_s)_{s\ge 0}$ in (10). For each $0 \le s \le t$, denote

$$w_s^t = w_s \prod_{j=s}^{t} (1 - w_j). \tag{30}$$

Then for each $t \in \mathbb{N}$, we can write

$$f_t(W) = \sum_{s=1}^{t} \ell(X_s, W) w_s^t$$

Moreover, note that $w_1^t, \dots, w_t^t > 0$ and $w_1^t + \cdots + w_t^t = 1$. Define $F_i(W) = (t-i+1)^{-1} \sum_{j=1}^{t} \ell(X_i, W)$ for each $1 \le i \le t$. By Lemma 10, there exists a constant $c_1 > 0$ such that

$$\mathbb{E}\left[\sup_{W \in \mathcal{C}} |F_i(W) - f(W)|\right] \le \frac{c_1}{\sqrt{t-i+1}} \tag{31}$$

for all $1 \le i \le t$. Noting that $(w_1^t, \dots, w_t^t)$ is a probability distribution on $\{1, \dots, t\}$, a simple calculation shows the following important identity

$$f_t - f = \sum_{i=1}^{t} (w_i^t - w_{i-1}^t)(t-i+1)(F_i - f),$$

with the convention of $w_0^t = 0$. Now by triangle inequality (31),

$$\begin{aligned}
\mathbb{E}\left[\sup_{W \in \mathcal{C}} |f_t(W) - f(W)|\right] &\le \mathbb{E}\left[\sum_{i=1}^{t} (w_i^t - w_{i-1}^t)(t-i+1) \sup_{W \in \mathcal{C}} |F_i(W) - f(W)|\right] \\
&= \sum_{i=1}^{t} (w_i^t - w_{i-1}^t)(t-i+1) \mathbb{E}\left[\sup_{W \in \mathcal{C}} |F_i(W) - f(W)|\right] \\
&\le \sum_{i=1}^{t} (w_i^t - w_{i-1}^t) c_1 \sqrt{t-i+1} \\
&\le c_1 \sqrt{t} \sum_{i=1}^{t} (w_i^t - w_{i-1}^t) = c_1 \sqrt{t} w_t^t.
\end{aligned}$$

Noting that $w_t^t = w_t$ in (30), this shows the first part of assertion. We can show the second part by using Lemma 16, following the argument in the proof of (Mairal, 2013b, Lem. B7). See the reference

for more details. ■

The following deterministic statement on converging sequences is due to Mairal et al. (Mairal et al., 2010).

Lemma 16. *Let $(a_n)_{n\in\mathbb{N}}$ and $(b_n)_{n\in\mathbb{N}}$ non-negative real sequences such that*

$$\sum_{n=0}^{\infty} a_n = \infty, \qquad \sum_{n=0}^{\infty} a_n b_n < \infty, \qquad |b_{n+1} - b_n| = O(a_n).$$

Then $\lim_{n\to\infty} b_n = 0$.

Proof See (Mairal, 2013b, Lem. A.5). ■

Appendix B. Algorithm for the generalized online NMF scheme

In this section, we state an algorithm for the generalized online NMF scheme (3). We denote by $\Pi_S : \mathbb{R}^{p\times q} \to S \subseteq \mathbb{R}^{p\times q}$ the projection onto S. The main algorithm, Algorithm 1 below, is a direct implementation of (3). In Algorithm 3, $A \sqcup B$ denotes the disjoint union for sets A and B.

Algorithm 1 Online NMF for Markovian data

Variables:
- $X_t \in \Omega \subseteq \mathbb{Q}^{d\times n}$: data matrix at time $t \in \mathbb{N}$
- $W_{t-1} \in \mathcal{C} \subseteq \mathbb{R}^{d\times r}$: learned dictionary at time t
- $(A_{t-1}, B_{t-1}) \in \mathbb{R}^{r\times r} \times \mathbb{R}^{r\times d}$: aggregate sufficient statistic up to time t
- $\lambda, \kappa_1, K > 0$: parameters
- $\mathcal{C}' \subseteq \mathbb{R}^{r\times n}$: constraint set of codes

Upon arrival of $\mathbf{X}_t$:
- Compute H_t using Algorithm 2.
- $A_t \leftarrow t^{-1}((t-1)A_{t-1} + H_t H_t^T)$, $B_t \leftarrow t^{-1}((t-1)B_{t-1} + H_t X_t^T)$.
- Compute W_t using Algorithm 3, with W_{t-1} as a warm restart, so that

$$W_t = \underset{W\in\mathcal{C}\subseteq\mathbb{R}^{d\times r}}{\arg\min} \left(\mathrm{tr}(WA_tW^T) - 2\mathrm{tr}(WB_t)\right)$$
$$\text{s.t. } \mathrm{tr}((B_t^T - WA_t)(W_{t-1} - W)^T) \leq 0$$

Algorithm 2 Sparse coding

Variables:
- $X_t \in \Omega \subseteq \mathbb{Q}^{d\times n}$: data matrix at time $t \in \mathbb{N}$
- $W_{t-1} \in \mathcal{C} \subseteq \mathbb{R}^{d\times r}$: learned dictionary at time t
- $\lambda > 0$: sparsity regularizer
- $\mathcal{C}' \subseteq \mathbb{R}^{r\times n}$: constraint set of codes

Repeat until convergence:
- **Do** $J \leftarrow$ All ones matrix in $\in \mathbb{R}^{r\times n}$;

$$H_t \leftarrow \Pi_{\mathcal{C}'}\left(H_t - \frac{1}{\mathrm{tr}(W_{t-1}^T W_{t-1})}(W_{t-1}^T W_{t-1} H_t - W_{t-1}^T X_t + \lambda J)\right) \tag{32}$$

Return H_t

Algorithm 3 Dictionary update

Variables:
  $W_{t-1} \in \mathcal{C} = \mathcal{C}_1 \sqcup \cdots \sqcup \mathcal{C}_m \subseteq \mathbb{R}^{d\times r}$: learned dictionary at time t
  $(A_t, B_t) \in \mathbb{R}^{r\times r} \times \mathbb{R}^{r\times d}$: aggregate sufficient statistic up to time t
  $\lambda, \kappa_1 > 0$: parameters
Do $\mathcal{E}_t \leftarrow \{W \in \mathbb{R}^{r\times d} \,|\, \mathrm{tr}((B_t^T - W_t A_t)(W_{t-1} - W_t)^T) \le 0\}$
For $i = 1, 2, \ldots, m$**:**
  Do $W_t^{(i)} \leftarrow W_{t-1}$ **and** $W_t \leftarrow W_{t-1}$
  If $\mathcal{C}_i \cap \mathcal{E}_t \neq \emptyset$ **Repeat until convergence:**
    For $j = 1$ **to** r**:**

$$[W_t]_{\bullet j} \leftarrow \Pi_{\mathcal{C}_i \cap \mathcal{E}_t}\left([W_{t-1}]_{\bullet j} - \frac{1}{[A_t]_{jj}+1}(W_{t-1}[A_t]_{\bullet j} - [B_t^T]_{\bullet j})\right) \tag{33}$$

  Do $W_t^{(i)} \leftarrow W_t$
Return $W_t = \arg\min_{W \in \{W_t^{(1)}, \ldots, W_t^{(m)}\}} \mathrm{tr}(W A_t W^T) - 2\mathrm{tr}(W B_t)$

When $\mathcal{C}$ is convex, Algorithm 3 reduces to the dictionary update algorithm in (Mairal et al., 2010), as the ellipsoidal condition in (3) becomes redundant (see Proposition 6 (iii)). We also remark that the specific coordinate descent algorithms (32) and (33) can be replaced by any other standard algorithms such as LARS.

Appendix C. Algorithms for Network Dictionary Learning and Reconstruction

In this section, we provide algorithms for network dictionary learning (NDL) (Algorithm NDL) and network reconstruction (Algorithm NR) as well as MCMC motif sampling algorithms (Algorithms MG, MP) that sample a random homomorphism $\mathbf{x} : F \to \mathcal{G}$ from a motif $F = ([k], A_F)$ into a network $\mathcal{G} = (V, A)$.

Algorithm MG . Glauber Chain Update

Input: Network $\mathcal{G} = (V, A)$, k-chain motif $F = ([k], A_F)$, and homomorphism $\mathbf{x} : F \to \mathcal{G}$
Do: Sample $v \in [k]$ uniformly at random
  Sample $\ell \in V$ at random from the distribution p given by

$$p(w) = \frac{1}{Z}\left(\prod_{u\in[k]} A(\mathbf{x}(u), w)^{A_F(u,v)}\right)\left(\prod_{u\in[k]} A(w, \mathbf{x}(u))^{A_F(v,u)}\right), \qquad w \in V$$

  where $Z = \sum_{c\in V}\left(\prod_{u\in[k]} A(\mathbf{x}(u), c)^{A_F(u,v)}\right)\left(\prod_{u\in[k]} A(c, \mathbf{x}(u))^{A_F(v,u)}\right)$ is the normalization constant.
  Define a new homomorphism $\mathbf{x}' : F \to \mathcal{G}$ by $\mathbf{x}'(w) = \ell$ if $w = v$ and $\mathbf{x}'(w) = \mathbf{x}(w)$ otherwise
Output: Homomorphism $\mathbf{x}' : F \to \mathcal{G}$

Algorithm MP . Pivot Chain Update

Input: Symmetric network $\mathcal{G} = (V, A)$, motif $F = ([k], A_F)$, and homomorphism $\mathbf{x} : F \to \mathcal{G}$

Parameters: $\texttt{AcceptProb} \in \{\texttt{Exact}, \texttt{Approximate}\}$

Do: $\mathbf{x}' \leftarrow \mathbf{x}$

  If $\sum_{c \in V} A(\mathbf{x}(1), c) = 0$: **Terminate**

  Else:

    Sample $\ell \in V$ at random from the distribution p_1 given by

$$p_1(w) = \frac{A(\mathbf{x}(1), w)}{\sum_{c \in V} A(\mathbf{x}(1), c)}, \qquad w \in V$$

    Compute the acceptance probability $\lambda \in [0, 1]$ by

$$\lambda \leftarrow \begin{cases} \left[\frac{\sum_{c \in V} A^{k-1}(\ell, c)}{\sum_{c \in V} A^{k-1}(\mathbf{x}(1), c)} \frac{\sum_{c \in V} A(c, \mathbf{x}(1))}{\sum_{c \in V} A(\mathbf{x}(1), c)} \wedge 1 \right] & \text{If } \texttt{AcceptProb} = \texttt{Exact} \\ \left[\frac{\sum_{c \in V} A(c, \mathbf{x}(1))}{\sum_{c \in V} A(\mathbf{x}(1), c)} \wedge 1 \right] & \text{If } \texttt{AcceptProb} = \texttt{Approximate} \end{cases} \tag{34}$$

    Sample $U \in [0, 1]$ uniformly at random, independently of everything else

    $\ell \leftarrow \mathbf{x}(1)$ if $U > \lambda$ and $\mathbf{x}'(1) \leftarrow \ell$.

    For $i = 2, 3, \ldots, k$:

      Sample $\mathbf{x}'(i) \in V$ from the distribution p_i given by

$$p_i(w) = \frac{A(\mathbf{x}(i-1), w)}{\sum_{c \in V} A(\mathbf{x}(i-1), c)}, \qquad w \in V \tag{35}$$

Output: Homomorphism $\mathbf{x}' : F \to \mathcal{G}$

Algorithm A3 . Rejection Sampling of Homomorphisms

Input: Network $\mathcal{G} = (V, A)$, motif $F = ([k], A_F)$

Requirement: There exists at least one homomorphism $F \to \mathcal{G}$

Repeat: Sample $\mathbf{x} = [\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(k)] \in V^{[k]}$ so that $\mathbf{x}(i)$'s are independent and identically distributed

  If $\prod_{1 \le i,j \le k} A(\mathbf{x}(i), \mathbf{x}(j))^{A_F(i,j)} > 0$:

    Return $\mathbf{x} : F \to \mathcal{G}$ and **Terminate**

Output: A homomorphism $\mathbf{x} : F \to \mathcal{G}$

Algorithm NDL . Network Dictionary Learning (NDL)

Input: Network $\mathcal{G} = (V, A)$

Parameters: $F = ([k], A_F)$ (motif) , $T \in \mathbb{N}$ (# iterations) , $N \in \mathbb{N}$ (# homomorphisms per iteration) , $r \in \mathbb{N}$ (# latent motifs) , $\lambda \geq 0$ (ℓ_1-regularizer)

Options: `MCMC` $\in \{$`Pivot`, `PivotApprox`, `Glauber`$\}$

Requirement: There exists at least one homomorphism $F \to \mathcal{G}$

Initialization:

Sample a homomorphism $\mathbf{x} : F \to \mathcal{G}$ by the rejection sampling in Algorithm A3

$W = (k^2 \times r)$ matrix of independent entries that we sample uniformly from $[0, 1]$

$P_0 =$ zero matrix of size $r \times r$; $Q_0 =$ zero matrix of size $r \times k^2$

For $t = 1, 2, \ldots, T$**:**

MCMC update and minibatch extraction:

Successively generate N homomorphisms $\mathbf{x}_{N(t-1)+1}, \mathbf{x}_{N(t-1)+2}, \ldots, \mathbf{x}_{Nt}$ by applying

$$\begin{aligned} \text{Algorithm MP with } \texttt{AcceptProb} = \texttt{Exact} \quad & \text{if } \texttt{MCMC} = \texttt{Pivot} \\ \text{Algorithm MP with } \texttt{AcceptProb} = \texttt{Approximate} \quad & \text{if } \texttt{MCMC} = \texttt{PivotApprox} \\ \text{Algorithm MG with } \texttt{AcceptProb} = \texttt{Glauber} \quad & \text{if } \texttt{MCMC} = \texttt{Glauber} \end{aligned}$$

For $N(t-1) < s \leq Nt$:

$A_{\mathbf{x}_t} \leftarrow k \times k$ matrix defined by $A_{\mathbf{x}_t}(a, b) = A(\mathbf{x}_t(a), \mathbf{x}_t(b))$ for $1 \leq a, b \leq k$

$X_t \leftarrow k^2 \times N$ matrix whose jth column is $\texttt{vec}(A_{\mathbf{x}_\ell})$ with $\ell = N(t-1) + j$
($\texttt{vec}(\cdot)$ denotes the vectorization operator in lexicographic ordering of entries)

Single iteration of Online Nonnegative Matrix Factorization:

$$\begin{cases} H_t \leftarrow \arg\min_{H \in \mathbb{R}^{r \times N}_{\geq 0}} \|X_t - W_{t-1} H\|_F^2 + \lambda \|H\|_1 & \text{(using Algorithm 2)} \\ P_t \leftarrow (1 - t^{-1}) P_{t-1} + t^{-1} H_t H_t^T \\ Q_t \leftarrow (1 - t^{-1}) Q_{t-1} + t^{-1} H_t X_t^T \\ W_t \leftarrow \arg\min_{W \in \mathcal{C}^{\text{dict}} \subseteq \mathbb{R}^{k^2 \times r}_{\geq 0}} \big(\text{tr}(W P_t W^T) - 2\text{tr}(W Q_t) \big) & \text{(using Algorithm 3),} \end{cases} \tag{36}$$

where $\mathcal{C}^{\text{dict}} := \{W \in \mathbb{R}^{k^2 \times r}_{\geq 0} \,|\, \text{columns of } W \text{ have Frobenius norm at most } 1\}$

Output: Network dictionary $W_T \in \mathcal{C}^{\text{dict}} \subseteq \mathbb{R}^{k^2 \times r}_{\geq 0}$

Algorithm NR . Network Reconstruction (NR)

Input: Network $\mathcal{G} = (V, A)$, and network dictionary $W \in \mathbb{R}_{\geq 0}^{k^2 \times r}$

Parameters: $F = ([k], A_F)$ (motif) , $T \in \mathbb{N}$ (# iterations) , $\lambda \geq 0$ (ℓ_1-regularizer)

Options: `MCMC` $\in \{$`Pivot`, `PivotApprox`, `Glauber`$\}$

Requirement: There exists at least one homomorphism $F \to \mathcal{G}$

Initialization:

A_{recons} , $A_{\text{count}} : V^2 \to \{0\}$ (zero matrices)

Sample a homomorphism $\mathbf{x}_0 : F \to \mathcal{G}$ by the rejection sampling in Algorithm A3

For $t = 1, 2, \ldots, T$**:**

MCMC update and mesoscale patch extraction:

$\mathbf{x}_t \leftarrow$ Updated homomorphism obtained by applying

$$\begin{aligned} \text{Algorithm MP with } \texttt{AcceptProb} = \texttt{Exact} \quad & \text{if } \texttt{MCMC} = \texttt{Pivot} \\ \text{Algorithm MP with } \texttt{AcceptProb} = \texttt{Approximate} \quad & \text{if } \texttt{MCMC} = \texttt{PivotApprox} \\ \text{Algorithm MG with } \texttt{AcceptProb} = \texttt{Glauber} \quad & \text{if } \texttt{MCMC} = \texttt{Glauber} \end{aligned}$$

$A_{\mathbf{x}_t} \leftarrow k \times k$ matrix defined by $A_{\mathbf{x}_t}(a, b) = A(\mathbf{x}_t(a), \mathbf{x}_t(b))$ for $1 \leq a, b \leq k$

$X_t \leftarrow k^2 \times 1$ matrix obtained by vectorizing $A_{\mathbf{x}_t}$

Local reconstruction:

$\widetilde{X}_t \leftarrow X_t$ and $\widetilde{W} \leftarrow W$

$H_t \leftarrow \underset{H \in \mathbb{R}_{\geq 0}^{r \times 1}}{\arg\min} ||\widetilde{X}_t - \widetilde{W}H||_F^2 + \lambda \|H\|_1$ and $\hat{X}_t \leftarrow \widetilde{W}H_t$

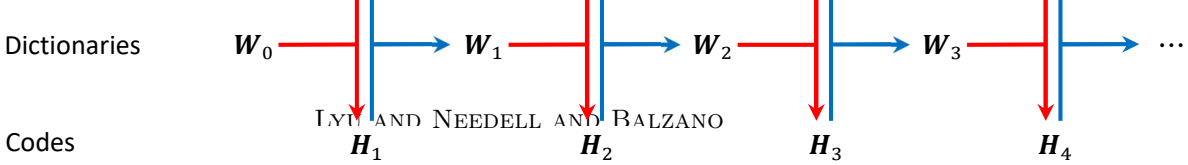
$\hat{A}_{\mathbf{x}_t;W} \leftarrow k \times k$ matrix obtained by reshaping the $k^2 \times 1$ matrix $\hat{X}_t$

Update global reconstruction:

For $a, b \in \{1, \ldots, k\}$:

$$\begin{aligned} A_{\text{count}}(\mathbf{x}_t(a), \mathbf{x}_t(b)) &\leftarrow A_{\text{count}}(\mathbf{x}_t(a), \mathbf{x}_t(b)) + 1 \\ j &\leftarrow A_{\text{count}}(\mathbf{x}_t(a), \mathbf{x}_t(b)) \\ A_{\text{recons}}(\mathbf{x}_t(a), \mathbf{x}_t(b)) &\leftarrow (1 - j^{-1}) A_{\text{recons}}(\mathbf{x}_t(a), \mathbf{x}_t(b)) + j^{-1} \hat{A}_{\mathbf{x}_t;W}(\mathbf{x}_t(a), \mathbf{x}_t(b)) \end{aligned}$$

Output: Reconstructed network $\mathcal{G}_{\text{recons}} = (V, A_{\text{recons}})$



Appendix D. Additional figures

In th... 5. As s... ch noise... ng spins beco... patches that ... ct of the Gibb... ubetzky and ... arantees that ... arkovian depe... equence. This ... (patch size, ... model, effective dictionary learning may not be possible at all, which is also suggested in Figure 5 right.

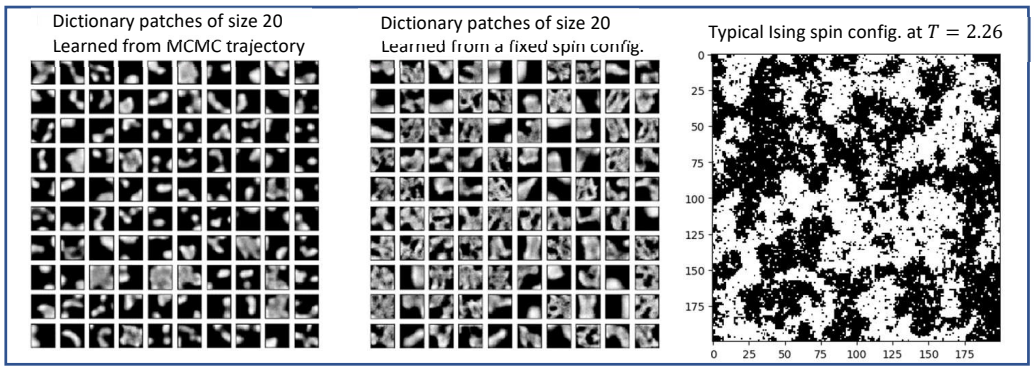


Figure 11: (Left) 100 learned dictionary patches from a MCMC Gibbs sampler for the Ising model on 200×200 square lattice at near critical temperature ($T = 2.26$). (Middle) 100 learned dictionary patches from fixed Ising spin configuration at $T = 0.5$ shown in the right.

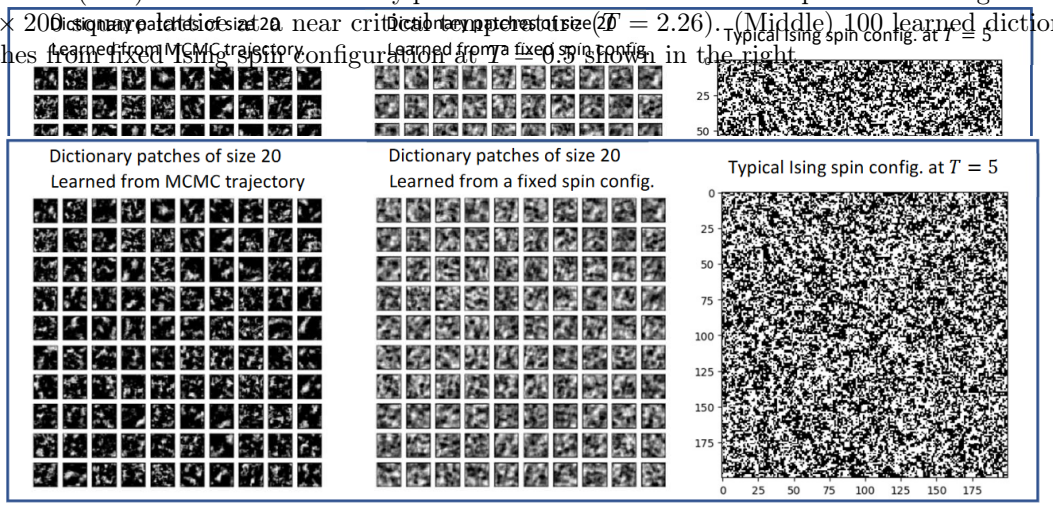


Figure 12: (Left) 100 learned dictionary patches from a MCMC Gibbs sampler for the Ising model on 200×200 square lattice at a supercritical temperature ($T = 5$). (Middle) 100 learned dictionary patches from fixed Ising spin configuration at $T = 0.5$ shown in the right.

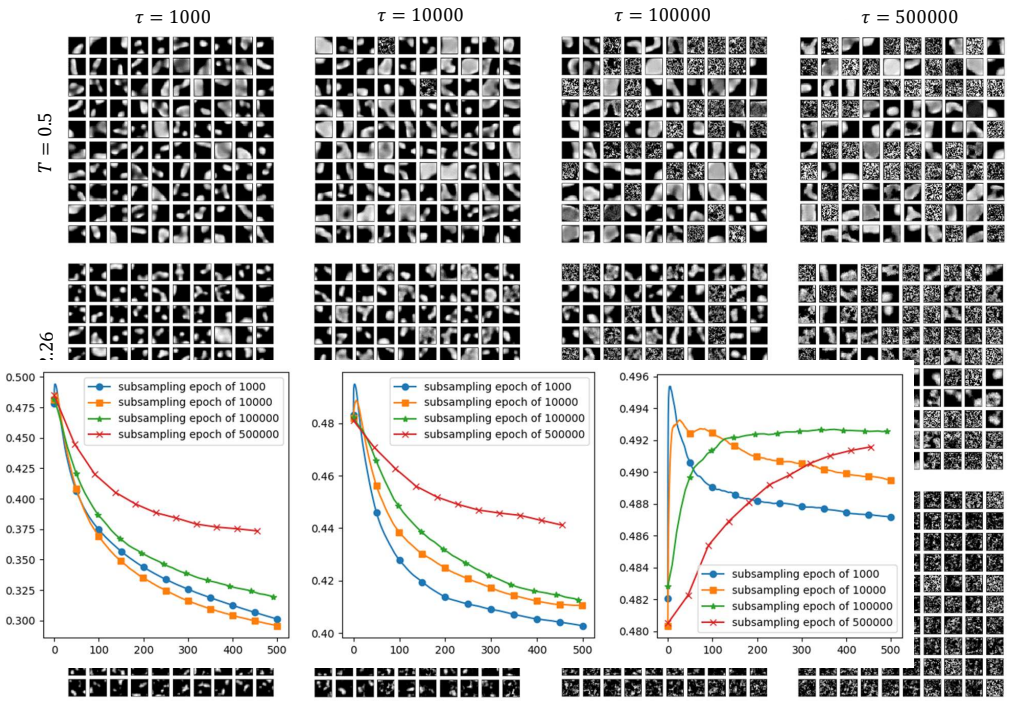


Figure 13: Plot of (normalized) surrogate errors vs. MCMC iterations (unit 10^4) for subsampling epochs of 1000, 10000, 100000, and 500000 for temperatures $T = 0.5$ (left), $T = 2.26$ (middle), and $T = 5$ (right), respectively.

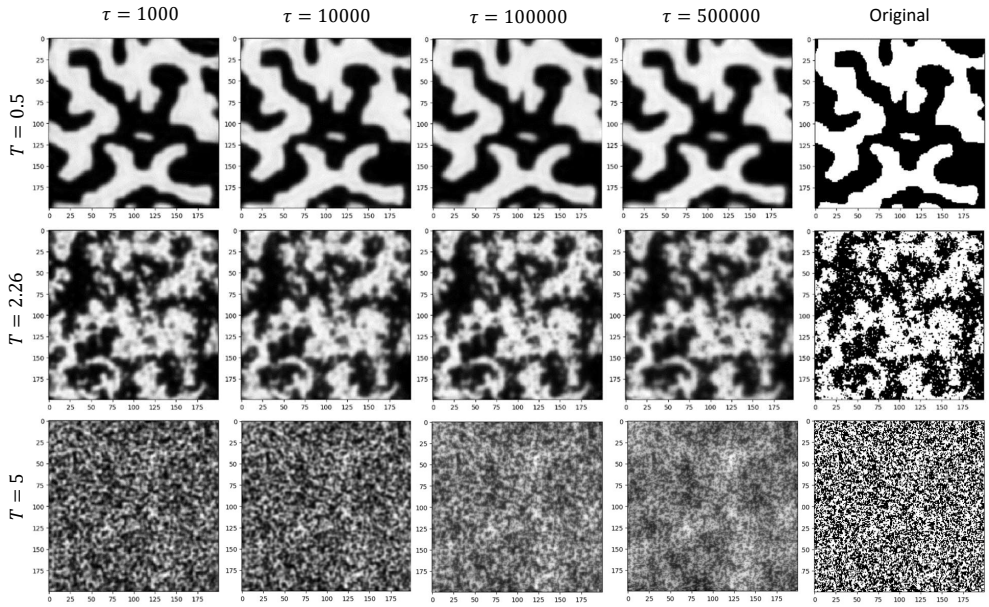


Figure 14: Reconstruction of fixed Ising spin configurations at temperatures $T = 0.5, 2.26,$ and 5 (rightmost column) using the learned dictionaries at different subsampling epochs $\tau = 1000, 10000, 100000,$ and 500000 shown in Figure 14. Since the dictionaries are learned from the entire MCMC trajectories, reconstruction error of a fixed configuration does not change drastically in the subsampling epoch.