# Sampling From Gaussian Markov Random Fields Using Stationary and Non-Stationary Subgraph Perturbations

Ying Liu, *Member, IEEE*, Oliver Kosut, *Member, IEEE*, and Alan S. Willsky, *Life Fellow, IEEE*

***Abstract*—Gaussian Markov random fields (GMRFs) or Gaussian graphical models have been widely used in many applications. Efficiently drawing samples from GMRFs has been an important research problem. In this paper, we introduce the subgraph perturbation sampling algorithm, which makes use of any pre-existing tractable inference algorithm for a subgraph by perturbing this algorithm so as to yield asymptotically exact samples for the intended distribution. We study the stationary version where a single fixed subgraph is used in all iterations, as well as the non-stationary version where tractable subgraphs are adaptively selected. The subgraphs used can have any structure for which efficient inference algorithms exist: for example, tree-structured, low tree-width, or having a small feedback vertex set. We present new theoretical results that give convergence guarantees for both stationary and non-stationary graphical splittings. Our experiments using both simulated models and large-scale real models demonstrate that this subgraph perturbation algorithm efficiently yields accurate samples for many graph topologies.**

***Index Terms*—Feedback vertex set, Gaussian graphical models, Gaussian Markov random fields, graphical splittings.**

## I. Introduction

MARKOV RANDOM FIELDS (MRFs) are graphical models in which the conditional independence structure of a set of random variables is represented by an undirected graph. An important sub-class of MRFs are Gaussian Markov random fields (GMRFs), where the joint distribution is Gaussian. GMRFs have been widely used in computer vision [2], computational biology [3], medical diagnostics [4], and communication systems [5]. GMRFs are particularly important in very large probabilistic networks involving millions of variables [6], [7].

This paper develops efficient algorithms for sampling from large-scale GMRFs. Throughout this paper sampling refers to drawing samples from a given probabilistic distribution specified by model parameters. We distinguish the sampling problem from the inference problem, wherein one computes the mean and variance of each random variable. As a fundamental problem by itself, sampling also has the relative advantage of allowing estimation of *arbitrary* statistics from the random field, rather than only the mean and variance. Moreover, sampling is useful for statistical models in which a GMRF is one of several interacting components. In such a setting, a sampler for the GMRF is an essential piece of any Markov chain Monte-Carlo (MCMC) framework for the entire system. Efficient sampling algorithms have been used to solve inference problems [8], to estimate model parameters [9], and are also used for model determination [10].

Very efficient algorithms for both inference and sampling exist for GMRFs in which the underlying graph is a tree (i.e., it has no cycles). Such models include hierarchical hidden Markov models [11], linear state space models [12], and multi-scale auto-regressive models [13]. For these models exact inference can be computed in linear time using belief propagation (BP) [14] (which generalizes the Kalman filter and the Rauch-Tung-Striebel smoother [12]), and exact samples can be generated using the forward sampling method [14]. However, the modeling capacity of trees is limited. Graphs with cycles can more accurately model real-world phenomena, but exact inference or sampling is often prohibitively costly for large-scale models.

MCMC samplers for general probabilistic models have been widely studied and can generally be applied directly to GMRFs. The most straightforward is the Gibbs sampler, wherein a new sample for each variable is generated by conditioning on the most recent sample of its neighbors [15]. However, the Gibbs sampler can have extremely slow convergence even for trees, making it impractical in large networks. For this reason, many techniques, such as reordering [16], blocking [17], [18], or collapsing [19], have been proposed to improve Gibbs sampling. In particular, the authors of [20] have proposed a blocked Gibbs sampler where each block includes a set of nodes whose induced subgraph does not have cycles; in [8] a Metropolis-Hastings sampler is studied, where a set of "control variables" are adaptively selected.

Manuscript received March 12, 2014; revised August 15, 2014; accepted November 02, 2014. Date of publication November 26, 2014; date of current version December 19, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Gustau Camps-Valls. This work was supported in part by AFOSRunder Grant FA9550-12-1-0287. This paper was presented in part at the International Symposium of Information Theory, Istanbul, Turkey, July 2013.

Y. Liu is with Google Inc., Cambridge Office, Cambridge, MA 02142 USA (e-mail: yliuy@google.com).

O. Kosut is with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: okosut@asu.edu).

A. S. Willsky is with the Department of Electrical Engineering and Computer Science and the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: willsky@mit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2014.2375134

There are also sampling algorithms for GMRFs that make explicit use of the joint Gaussianity. Since inference in a GMRF is equivalent to solving a linear system, sampling algorithms are often closely related to direct or iterative linear solvers. One approach is using the Cholesky decomposition to generate exact samples. If a sparse Cholesky decomposition is provided directly from the problem formulation, then generating samples using that decomposition is the preferred approach. Similarly, in [21] the problem formulation leads directly to a decomposition into sparse "filters", which are then used, together with random perturbations to solve linear equations that produce samples. Once again, for problems falling into this class, using this method is unquestionably preferred. However, for other Gaussian models for which such sparse decompositions are not directly available, other approaches need to be considered. In particular, the computation of the Cholesky decomposition has cubic complexity and a quadratic number of fills in general, even for sparse matrices as arise in graphical models [22]. While this complexity is acceptable for models of moderate size, it can be prohibitively costly for large models, e.g., those involving millions of variables.

In this paper we propose a general framework to convert iterative linear solvers based on graphical splittings to MCMC samplers by adding a random perturbation at each iteration. In particular, our algorithm can be thought of as a stochastic version of graph-based solvers and, in fact, is motivated by the use of embedded trees in [23], [24] for the computation of the mean of a GMRFs. That approach corresponds to decomposing the graph of the model into a tractable graph[1], i.e., one for which sampling is easy (e.g., a tree), and a "cut" matrix capturing the edges removed to form the tractable subgraph. The subgraphs used can have any structure for which efficient inference algorithms exist: for example, tree-structured, low treewidth, or having a small feedback vertex set (FVS) [25]. Much more importantly, in order to obtain a valid sampling algorithm, we must exercise some care, not needed or considered for the linear solvers in [23], [24], in constructing the graphical models corresponding to both the tractable subgraph and to the set of variables involved in the cut edges. We give general conditions under which graph-based iterative linear solvers can be converted into samplers and we relate these conditions to the so-called P-regularity condition [26]. We then provide a simple construction that produces a splitting satisfying those conditions. Once we have such a decomposition our algorithm proceeds at each iteration by generating a sample from the model on the subgraph and then randomly perturbing it based on the model corresponding to the cut edges. That perturbation obviously must admit tractable sampling itself and also must be shaped so that the resulting samples of the overall model are asymptotically exact. Our construction ensures both of these. As was demonstrated in [23], [24], using non-stationary splittings, i.e., different graphical decompositions in successive iterations, can lead to substantial gains in convergence speed. We extend our subgraph perturbation algorithm from stationary graphical splittings to non-stationary graphical splittings and give theoretical results for convergence guarantees. We propose an algorithm to select tractable subgraphs for stationary splittings and an adaptive method for selecting non-stationary splittings.

The authors of [27] have proposed a sampling framework that generalizes and accelerates the Gibbs sampler. Previous work in [28] has shown that the Gibbs sampler is a stochastic version of the Gauss-Seidel iteration for solving learning systems. The sampling algorithm in [27] adds additional noises corresponding to the first or second order Chebyshev coefficients to accelerate the Gibbs sampler. While the idea of converting a linear solver to a sampler is also discussed in [27], their work is different from ours because their algorithm does not consider graph structures in constructing the matrix splitting that is used (i.e., the sparsity pattern of the base matrix remains the same without considering any tractable subgraphs). Moreover, when multiple matrix splittings are used, the different splittings in [27] have differences only in the Chebyshev coefficients while in our work, different matrix splittings correspond to different graph structures.

The remainder of the paper is organized as follows. In Section II we introduce some necessary background and review some common sampling algorithms. In Section III we propose the subgraph perturbation algorithm with stationary splittings, providing efficient implementation as well as theoretical results on the convergence rate. Next in Section IV we present the use of non-stationary splittings and theoretical results on convergence. We then discuss how to select tractable subgraphs for both the stationary and the non-stationary settings in Section V. In Section VI we present experimental results using simulated data on various graph structures as well as using large-scale real data. We compare the convergence rate of our algorithm with several other techniques. Finally, we summarize the main contributions of this paper in Section VII.

## II. Background

In this section, we first introduce necessary background on GMRFs. Then we define the convergence rate used throughout this paper and review some common sampling algorithms.

### A. Gaussian Markov Random Fields

An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$ is used in an MRF to model the conditional independence structure among a set of random variables [14]. Each node $s \in \mathcal{V}$ corresponds to a random variable $x_s$. For any subset $A \subset \mathcal{V}$, the random vector $\mathbf{x}_A$ corresponds to the set of random variables $\{x_s | s \in A\}$ and we will also simply write $\mathbf{x}$ for $\mathbf{x}_{\mathcal{V}}$. This random vector has the Markov property with respect to the graph if for any subsets $A$, $B$, $S \subset \mathcal{V}$ where $S$ separates $A$ and $B$ in the graph, $\mathbf{x}_A$ and $\mathbf{x}_B$ are independent conditioned on $\mathbf{x}_S$. By the Hammersley-Clifford theorem, if the probabilistic distribution function (*p.d.f.*) $p(\mathbf{x})$ is positive everywhere, then $p(\mathbf{x})$ can be factored according to $p(\mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \phi_C(\mathbf{x}_C)$, where $\mathcal{C}$ is the collection of cliques (fully connected subgraphs) and $Z$ is the normalization factor or partition function [14].

[1]Here the subgraph is a spanning subgraph, i.e., one that includes all of the vertices and a subset of all edges.

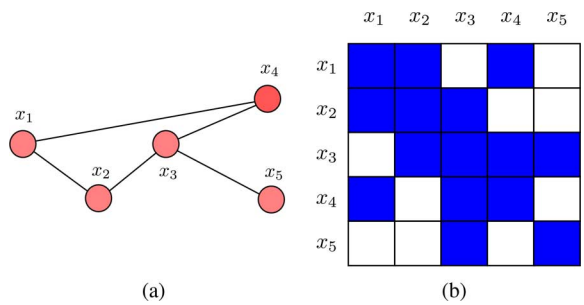


Fig. 1. (a) The sparsity pattern of the undirected graph; (b) The sparsity pattern of the information matrix.

When the random vector $\mathbf{x}_{\mathcal{V}}$ is jointly Gaussian, the model is a GMRF. The p.d.f. of a GMRF can be parametrized by

$$p(\mathbf{x}) \propto \exp\left\{-\frac{1}{2}\mathbf{x}^T J\mathbf{x} + \mathbf{h}^T\mathbf{x}\right\}, \tag{1}$$

where $J$ is the *information matrix* or *precision matrix* and $\mathbf{h}$ is the *potential vector*. The mean $\boldsymbol{\mu}$ and covariance matrix $\Sigma$ are related to $J$ and $\mathbf{h}$ by $\boldsymbol{\mu} = J^{-1}\mathbf{h}$ and $\Sigma = J^{-1}$. In this paper, we denote this distribution by either $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ or $\mathcal{N}^{-1}(\mathbf{h}, J)$. The structure of the underlying graph can be constructed using the sparsity pattern of $J$, i.e., there is an edge between $i$ and $j$ if and only if $J_{ij} \neq 0$. Hence, the conditional independence structure can be read immediately from the sparsity pattern of the information matrix as well as that of the underlying graph (See Fig. 1). Our starting point will simply be the specification of $\mathbf{h}$ and $J$ (and with it the graphical structure). One setting in which such a specification arises (and which we will illustrate with our large-scale example) is in estimation problems, that in which $\mathbf{x}$ represents a large random field, which has prior distribution $\mathcal{N}^{-1}(0, J_0)$ according to a specified graph[2] (e.g., the thin-membrane or the thin-plate model [29]) and where we have potentially sparse and noisy measurements of components of $\mathbf{x}$ given by $\mathbf{y} = C\mathbf{x} + \mathbf{v}$, $\mathbf{v} \sim \mathcal{N}(0, R)$, where $C$ is a selection matrix (a single 1 in each row, all other row elements being 0) and $R$ is a (blocked) diagonal matrix. In this case, the posterior distribution $p(\mathbf{x}|\mathbf{y})$ is $\mathcal{N}^{-1}(\mathbf{h}, J)$, where $\mathbf{h} = C^T R^{-1}\mathbf{y}$ and $J = J_0 + C^T R^{-1} C$.

### B. Sampling and Its Convergence

The sampling problem considered in this paper is to efficiently generate samples from a GMRF with underlying distribution $\mathcal{N}^{-1}(\mathbf{h}, J)$ with given model parameters $\mathbf{h}$ and $J$. We consider iterative samplers that produce a sequence of samples $\mathbf{x}^{(t)}$ for $t = 1, 2, \ldots$. An iterative sampling algorithm is correct if the samples converge in distribution to the target distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ where $\boldsymbol{\mu} = J^{-1}\mathbf{h}$ and $\Sigma = J^{-1}$. If the process to generate this sequence is Gaussian, then the marginal distribution of each iteration is fully described by its mean $\boldsymbol{\mu}^{(t)}$ and covariance matrix $\Sigma^{(t)}$. In this case, the convergence of the sampler is equivalent to $\boldsymbol{\mu}^{(t)} \to \boldsymbol{\mu}$ and $\Sigma^{(t)} \to \Sigma$ as $t \to \infty$.

[2]Without loss of generality we can assume that the prior mean of $\mathbf{x}$ is 0 simply by subtracting it from the random field and from the measurements.

As we are especially interested in fast convergence to the target distribution, we need a clear notion of convergence rate. In the study of MCMC samplers, convergence rate is often measured by the total variation of the sample distribution from the target distribution [15]. In this paper, for convenience, we instead use the Euclidean norm (denoted by $||\cdot||$) of the difference of the means and the Frobenius norm (denoted by $||\cdot||_F$) of the difference of the covariance matrices to measure the deviation of the sample distribution from the target distribution. It can be shown that for non-degenerate Gaussian models, the convergence in total variation is equivalent to the convergence in the model parameters. In particular, we define convergence rate for the mean as[3]

$$\tau_{\boldsymbol{\mu}} = -\ln \limsup_{t\to\infty} \frac{\|\boldsymbol{\mu}^{(t+1)} - \boldsymbol{\mu}\|}{\|\boldsymbol{\mu}^{(t)} - \boldsymbol{\mu}\|}, \tag{2}$$

and convergence rate for the covariance as

$$\tau_{\Sigma} = -\frac{1}{2}\ln \limsup_{t\to\infty} \frac{\|\Sigma^{(t+1)} - \Sigma\|_F}{\|\Sigma^{(t)} - \Sigma\|_F}. \tag{3}$$

### C. Commonly Used Sampling Algorithms

In this subsection, we summarize some commonly used sampling algorithms including using the Cholesky decomposition, forward sampling on trees (and beyond), and Gibbs sampling (with its variants).

*Sampling Using the Cholesky Decomposition:* The Cholesky decomposition gives a lower triangular matrix $L$ such that $J = LL^T$. Let $\mathbf{z}$ be an $n$-dimensional random vector whose entries are drawn *i.i.d.* from the standard Gaussian distribution. An exact sample $\mathbf{x}$ can be obtained by computing $\mathbf{x} = (L^T)^{-1}(\mathbf{z} + L^{-1}\mathbf{h})$. As mentioned in Section I, if such a decomposition is available and if $L$ is sparse, sampling is fast even for very large models. However, for a general sparse $J$, the computation of $L$ has cubic complexity while fill in $L$ can be quadratic in the size of the model. For very large models, the Cholesky decomposition is computationally prohibitive.[4]

*Forward Sampling for Tree-Structured Models:* For a tree-structured GMRF, an exact sample can be generated in linear time (with respect to the number of nodes) by first computing the variances and means for all nodes and covariances for the edges using BP, and then sampling the variables one by one following a root-to-leaf order where the root node can be an arbitrary node [14].

*Forward Sampling for Models With Small Feedback Vertex Sets:* There are other tractable graphical models that one can consider, including models with small FVSs, i.e., models on graphs for which there is a small set of so-called feedback nodes that, if removed, leave no cycles. In this case, using the algorithms developed in [25], one can compute the means and covariances using the feedback message passing algorithm that

[3]The notation $\limsup$ denotes the limit superior, i.e., $\limsup_{n\to\infty} x_n = \lim_{n\to\infty}(\sup_{m\geq n} x_m)$.

[4]Sparse Cholesky decomposition can be employed to reduce the computational complexity. However, even for sparse graphs, the number of fills in the worst case is still $\mathcal{O}(n^2)$ and the total computational complexity is $\mathcal{O}(n^3)$ in general [22].

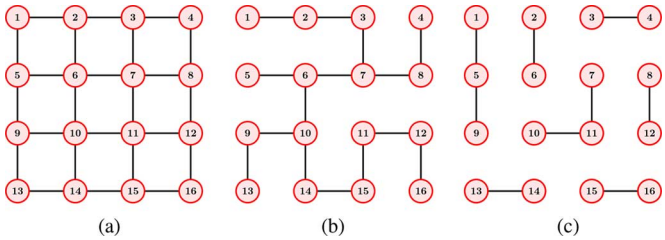


Fig. 2. The grid shown in (a) can be decomposed into a spanning tree (b) and a graph consisting of the missing edges (c). (a) Graph structure for $J$, (b) Graph structure for $J_T$, (c) Graph structure for $K$.

scales quadratically in the size of the FVS and linearly in the overall size of the graph and can then produce samples by first sampling the nodes in the FVS (perhaps using the Cholesky decomposition, with complexity cubic in the size of the FVS) and then performing forward tree sampling on the rest.

*Basic Gibbs Sampling:* The basic Gibbs sampler generates new samples, one variable at a time, by conditioning on the most recent values of its neighbors. In particular, in each iteration, a sample for all $n$ variables is drawn by performing $x_i^{(t+1)} \sim \mathcal{N}(\frac{1}{J_{ii}}h_i - \sum_{j<i,\ j\in\mathcal{N}(i)} J_{ji}x_j^{(t+1)} - \sum_{j>i,\ j\in\mathcal{N}(i)} J_{ji}x_j^{(t)}, J_{ii}^{-1})$ for $i = 1, 2, \ldots .n$, where $\mathcal{N}(i)$ denotes the set of node $i$'s neighbors in the graph. The Gibbs sampler always converges when $J \succ 0$;[5] however, the convergence can be very slow for many GMRFs, including many tree-structured models. More details on Gibbs sampling can be found in [15].

*Variants of Gibbs Sampling:* There have been many variants of the Gibbs sampler using the ideas of reordering, coloring, blocking, and collapsing, as previously mentioned in Section I. For example, in the blocked Gibbs sampler the set of nodes is partitioned into several disjoint subsets and each subset is treated as a single variable. One approach is to use graph coloring, in which variables are colored so that adjacent nodes have different colors, and then each Gibbs block is the set of nodes in one color [30]. In [20] the authors have proposed a blocking strategy where each block induces a tree-structured subgraph.

## III. Sampling by Subgraph Perturbations With Stationary Graphical Splittings

In this section, we introduce our subgraph perturbation sampling framework using stationary (fixed) splittings. First, we describe the general framework with an arbitrary graphical splitting followed by theoretical results on convergence. We then describe a local construction of the splitting that builds up the decomposition as a sum of rank-1 terms corresponding to each of the edges removed from the tractable graph. The construction of this splitting is simple to perform at run time, leads to very efficient sampling of the perturbation term required in the sampling algorithm, and ensures convergence.

[5]Throughout this paper, the notation $A \succ 0$ denotes that $A$ is positive definite and $A \succeq 0$ positive semi-definite. We also use $A \succ B$ or $A \succeq B$ to denote $A - B \succ 0$ or $A - B \succeq 0$ respectively. The symbols $\prec$ and $\preceq$ are similarly defined.

**Algorithm 1: Sampling by Subgraph Perturbations with Stationary Splittings**

Input: $J$, $\mathbf{h}$, and subgraph structure $\mathcal{T}$
Output: samples with the asymptotic distribution $\mathcal{N}^{-1}(\mathbf{h}, J)$

1) Form $J_T$ and $K$.
2) Draw an initial sample $\mathbf{x}^{(0)}$ from a Gaussian distribution.
3) At each iteration:
   a) Generate an independent sample $\mathbf{e}^{(t+1)}$ with zero mean and covariance matrix $J_T + K$.
   b) Compute $\mathbf{x}^{(t+1)}$ using the equation $\mathbf{x}^{(t+1)} = J_T^{-1}(\mathbf{h} + K\mathbf{x}^{(t)} + \mathbf{e}^{(t+1)})$.

### A. General Algorithm

Our sampling framework relies on a graph-based matrix splitting. Given the information matrix $J$ with underlying graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consider the splitting

$$J = J_T - K, \tag{4}$$

where $J_T$ has sparsity corresponding to a tractable subgraph $\mathcal{G}_T = (\mathcal{V}, \mathcal{E}_T)$ with $\mathcal{E}_T \subset \mathcal{E}$, and $K$ has sparsity corresponding to the graph with edge set $\mathcal{E}\backslash\mathcal{E}_T$ (See Fig. 2). Throughout this paper, we assume that the splittings we consider are all *graphical splittings*, i.e., both $J_T$ and $K$ are symmetric matrices corresponding to undirected graphs.

In [23] this type of splitting (although crucially without one of the further assumptions we will make) was proposed for the computation of the mean (but not covariance or samples). In particular, when $J_T$ is non-singular the mean is computed using the iterative equation

$$\boldsymbol{\mu}^{(t+1)} = J_T^{-1}\left(K\boldsymbol{\mu}^{(t)} + \mathbf{h}\right) \tag{5}$$

with the fixed-point solution $\boldsymbol{\mu} = J^{-1}\mathbf{h}$. Note that the sequence in (5) converges if and only if $\rho(J_T^{-1}K) < 1$, i.e., the spectral radius of the matrix $J_T^{-1}K$ is less than one. For this to be efficient, the computation on the right-hand side of (5) would need to be far simpler than solving for the mean directly, using the full matrix $J$. The approach in [23] took $J_T$ to have tree structure (so that the computation in (5) has linear complexity), although in principle it is possible to choose $J_T$ to have other graph structures (e.g., as in [31]) that lead to tractable computations. Moreover, while the approach is not limited to the following, the original idea in [23] and especially in [24] is simply to "cut" edges from the graph, so that $J_T$ is obtained from $J$ simply by zeroing out the elements corresponding to the cut edges, and $-K$ is the matrix whose only nonzero elements are the values corresponding to those cut edges.

The high-level idea of our sampling algorithm is to further inject noise into (5), so that the iterative linear solver becomes a stochastic process whose stationary distribution is the target distribution $\mathcal{N}^{-1}(\mathbf{h}, J)$. However, the simple idea of constructing $K$ by copying the elements of $J$ corresponding to cut edges may not be feasible for our sampling algorithm. Rather, we need to

ensure that $K$ is chosen so that $J_T + K \succ 0$. Assuming that we have a splitting that satisfies this condition, our iterative sampling algorithm is given by:

$$\mathbf{x}^{(t+1)} = J_T^{-1}\left(K\mathbf{x}^{(t)} + \mathbf{h} + \mathbf{e}^{(t+1)}\right), \tag{6}$$

where the perturbation $\mathbf{e}^{(t+1)}$ is a Gaussian random vector, independent of all other variables, with zero mean and covariance $J_T + K$. The general sampling framework is summarized in Algorithm 1.

In the next subsection we provide theoretical results showing that convergence of the iteration in (5) for a graphical splitting is equivalent to the condition $J_T + K \succ 0$, which in turn implies that the sampling method in (6) is well-defined. We also show that in this case, the sample distribution indeed converges to the correct distribution. In the last part of this section, we provide a straightforward "local" edge-by-edge method for constructing such a splitting that also directly yields an efficient generation of the perturbation $\mathbf{e}^{(t+1)}$.

### B. Correctness and Convergence

In this subsection, we present theoretical results for the general subgraph perturbation framework. Proposition 1 and Theorem 3 establish the correctness of Algorithm 1 as well as a convergence guarantee. Proposition 4 and Corollary 5 give bounds on the convergence rate.

In general, a matrix splitting $A = M - N$ is called a *P-regular splitting* if $M$ is non-singular and $M^T + N$ is positive definite [26]. The P-regularity condition has been proposed in the study of iterative linear solvers as a condition for convergence [26], [32]. In our graphical splitting $J = J_T - K$, since $J_T$ is symmetric, the P-regular condition $J_T^T + K \succ 0$ is precisely the condition that the added noise term in our perturbation framework is valid, i.e., that it corresponds to a random variable with positive definite covariance. Therefore, our sampling framework provides a new interpretation of the P-regularity condition—for graphical splittings as in (4), convergence of iterative solvers as in (5) is equivalent to the noise in $\mathbf{e}^{(t+1)}$ being valid. It has been shown in [23] that the necessary and sufficient condition for the embedded tree algorithm to converge with any initial point is $\rho(J_T^{-1}K) < 1$. In Proposition 1 we prove that this condition is equivalent to the graphical splitting being P-regular, which further guarantees the validity of the added noise in (6).

*Proposition 1:* Assuming $J \succ 0$ and that $J = J_T - K$ is a graphical splitting, the condition $\rho(J_T^{-1}K) < 1$ is satisfied if and only if the splitting is P-regular, i.e., the added noise in Algorithm 1 has a valid covariance matrix $J_T + K \succ 0$.

The proof of Proposition 1 is included in the Appendix. The following Lemma 2 is used in the proof of Theorem 3, which is our main result in this section. The proof of Lemma 2 is also deferred to the Appendix.

*Lemma 2:* Let $A$ and $B$ be square matrices. If 1) $A$ is invertible; 2) $A+B$ is symmetric and invertible, then $\Sigma = (A+B)^{-1}$ is a solution of the equation $A\Sigma A^T = B\Sigma B^T + A^T - B$.

The following Theorem 3 states that for graphical splittings, a convergent linear solver can be converted to a convergent sampler with the same convergence rate.

*Theorem 3:* For a valid GMRF with information matrix $J \succ 0$, let $J = J_T - K$ be a graphical splitting. If the corresponding linear solver converges, i.e., $\rho(J_T^{-1}K) < 1$, then the sample distribution generated by Algorithm 1 is guaranteed to converge to the target distribution and the asymptotic convergence rates $\tau_{\boldsymbol{\mu}}$ for the mean and $\tau_\Sigma$ for the covariance are both equal to $-\ln\rho(J_T^{-1}K)$.

*Proof:* From Proposition 1, we have that $J_T + K \succ 0$, i.e., the covariance matrix of the added noise is valid. It can be shown that with the initial sample distribution being Gaussian, the iterations in Algorithm 1 generate a sequence of Gaussian samples, with $\mathbf{x}^{(t)}$ having mean $\boldsymbol{\mu}^{(t)}$ and covariance matrix $\Sigma^{(t)}$. From Step 3(b) in Algorithm 1, we have

$$\begin{aligned}\boldsymbol{\mu}^{(t+1)} &= \mathbb{E}\left[\mathbf{x}^{(t+1)}\right] && (7)\\ &= \mathbb{E}\left[J_T^{-1}\left(\mathbf{h} + \mathbf{e}^{(t+1)} + K\mathbf{x}^{(t)}\right)\right] && (8)\\ &= J_T^{-1}(\mathbf{h} + K\boldsymbol{\mu}^{(t)}). && (9)\end{aligned}$$

Since $\rho(J_T^{-1}K) < 1$, the mean $\boldsymbol{\mu}^{(t+1)}$ converges to the unique fixed-point $\hat{\boldsymbol{\mu}}$ satisfying

$$\hat{\boldsymbol{\mu}} = J_T^{-1}(\mathbf{h} + K\hat{\boldsymbol{\mu}}). \tag{10}$$

So $\hat{\boldsymbol{\mu}} = (J_T - K)^{-1}\mathbf{h} = J^{-1}\mathbf{h}$, and thus $\boldsymbol{\mu}^{(t)}$ converges to the exact mean $\boldsymbol{\mu} = J^{-1}\mathbf{h}$ with convergence rate $\tau_{\boldsymbol{\mu}} = -\ln\rho(J_T^{-1}K)$.

Now we consider the convergence of the covariance matrix. From Step 3(b) in Algorithm 1, we have

$$\begin{aligned}\Sigma^{(t+1)} &= \mathrm{Cov}\left\{\mathbf{x}^{(t+1)}\right\} && (11)\\ &= \mathrm{Cov}\left\{J_T^{-1}\left(\mathbf{h} + \mathbf{e}^{(t+1)} + K\mathbf{x}^{(t)}\right)\right\} && (12)\\ &= J_T^{-1}\left(J_T + K + K\Sigma^{(t)}K\right)J_T^{-1} && (13)\\ &= \left(J_T^{-1}K\right)\Sigma^{(t)}\left(J_T^{-1}K\right)^T + J_T^{-1}(J_T + K)J_T^{-1}. && (14)\end{aligned}$$

This equation can be rewritten in vector form as

$$\begin{aligned}\mathrm{vec}(\Sigma^{(t+1)}) = &\left[\left(J_T^{-1}K\right)\otimes\left(J_T^{-1}K\right)\right]\mathrm{vec}(\Sigma^{(t)})\\ &+\mathrm{vec}\left(J_T^{-1}\left(J_T + K\right)J_T^{-1}\right),\end{aligned} \tag{15}$$

where $\mathrm{vec}(\cdot)$ denotes the column vector obtained by stacking all the columns in its argument and $A\otimes B$ denotes the Kronecker product of matrices $A$ and $B$, i.e.,

$$A\otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B\\ \vdots & \ddots & \vdots\\ a_{m1}B & \cdots & a_{mn}B\end{bmatrix}, \tag{16}$$

where $A$ is an $m$-by-$n$ matrix $[a_{ij}]_{m\times n}$. According to [33], $\rho((J_T^{-1}K)\otimes(J_T^{-1}K)) = \rho^2(J_T^{-1}K) < 1$. Hence the iterative equation (15) is guaranteed to converge to a unique fixed-point, denoted by $\mathrm{vec}(\hat{\Sigma})$, with asymptotic convergence rate $-\ln\rho^2(J_T^{-1}K)$ in the Euclidean norm. Hence, (14) converges to a unique fixed-point matrix $\hat{\Sigma}$. By Lemma 2, the fixed-point solution $\hat{\Sigma} = (J_T - K)^{-1} = J^{-1}$ is exactly the target covariance matrix. Hence, the convergence rate $\tau_\Sigma = -\frac{1}{2}\ln\rho^2(J_T^{-1}K) = -\ln\rho(J_T^{-1}K)$ since $\forall A,\ ||\mathrm{vec}(A)|| = ||A||_F$. This completes the proof of Theorem 3. ■

We have shown in Theorem 3 that the convergence rates for both the mean and the covariance are $-\ln \rho(J_T^{-1}K)$. Naturally, we want to choose a splitting with a small $\rho(J_T^{-1}K)$. This spectral radius is a highly nonlinear function of both $J_T$ and $K$, and it is useful to have bounds that are simple (and monotonic) functions of $K$ or $J_T$ alone. The following Proposition 4 is adapted from Theorem 3 in [23]. For a valid distribution with $J \succ 0$, the condition of $K \succeq 0$ in Proposition 4 is sufficient to ensure $J_T + K \succ 0$, which guarantees the convergence of Algorithm 1. In the next subsection, we provide a local implementation of Algorithm 1 where the condition $K \succeq 0$ is satisfied.

*Proposition 4:* Consider a graphical splitting $J = J_T - K$ with $J \succ 0$. If $K \succeq 0$, then $\frac{\lambda_{\max}(K)}{\lambda_{\max}(K)+\lambda_{\max}(J)} \leq \rho(J_T^{-1}K) \leq \frac{\lambda_{\max}(K)}{\lambda_{\max}(K)+\lambda_{\min}(J)} < 1$, where $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ denote the maximum and the minimum eigenvalues respectively.

*Proof:* Use Theorem 2.2 in [34] and let $\mu = 1$. ■

A simpler (and looser) bound that is much easier to compute (and hence can be used in choosing $K$) is given in the following Corollary 5. We define the weight of the $i$-th row of $K$ as $w_i(K) = \sum_j |K_{ij}|$ and let $w(K) = \max_i w_i(K)$.

*Corollary 5:* In the same setting as in Proposition 4, we have $\rho(J_T^{-1}K) \leq \frac{w(K)}{w(K)+\lambda_{\min}(J)}$.

*Proof:* Since $K \succeq 0$, we have $\lambda_{\max}(K) = \rho(K)$. By Corollary 8.1.18 in [35], we have that $\rho(K) \leq \rho(\overline{K})$, where $\overline{K}$ takes the entry-wise absolute values of $K$. By Corollary 8.1.22 in [35], $\rho(\bar{K}) \leq \max_i \Sigma_j \bar{K}_{ij}$, so $\lambda_{\max}(K) = \rho(K) \leq \rho(\overline{K}) \leq w(K)$. This corollary thus follows from Proposition 4. ■

### C. Efficient Localized Implementation

Given a graphical splitting $J = J_T - K$, Algorithm 1 requires generating noise vectors $\mathbf{e}^{(t)}$ with zero mean and covariance $J_T + K \succ 0$. Depending on the splitting, these random noise vectors may be difficult to generate. In this subsection, given a tractable subgraph $\mathcal{T}$ we provide a method to construct the splitting matrices $J_T$ and $K$ specifically so that the noise vectors $\mathbf{e}^{(t)}$ can be constructed efficiently and to guarantee convergence. Moreover, our construction is entirely local with respect to the graph. In this subsection, we focus on the construction of the splitting with a given subgraph and postpone the selection of subgraphs to Section V.

Let $\mathcal{E}_\mathcal{T}$ denote the set of edges in the subgraph $\mathcal{T}$. We construct $K$ to be sparse with respect to the subgraph with edge set $\mathcal{E}\backslash\mathcal{E}_\mathcal{T}$ as follows. For each $(i,j) \in \mathcal{E}\backslash\mathcal{E}_\mathcal{T}$, let $K^{(i,j)} = \begin{bmatrix} |J_{ij}| & -J_{ij} \\ -J_{ij} & |J_{ij}| \end{bmatrix}$, and let $[K^{(i,j)}]_{n\times n}$ be the $n$-by-$n$ matrix zero-padded from $K^{(i,j)}$, i.e., the principal submatrix corresponding to rows (and columns) $i$ and $j$ of $[K^{(i,j)}]_{n\times n}$ equals $K^{(i,j)}$ while other entries are zero. It can be easily verified that $[K^{(i,j)}]_{n\times n} \succeq 0$. We define $K$ to be the sum of these rank-one matrices as

$$K = \sum_{(i,j)\in\mathcal{E}\backslash\mathcal{E}_\mathcal{T}} \left[K^{(i,j)}\right]_{n\times n}. \tag{17}$$

The matrix $J_T$ is then obtained by

$$J_T = J + K. \tag{18}$$

Note that $J_T$ is sparse with respect to $\mathcal{T}$. Moreover, $K$ is positive semi-definite and $J_T$ is positive definite (since $J$ is positive definite for a valid model).

**Algorithm 2: Sampling by Subgraph Perturbations with Local Implementation**

Input: $J$, $\mathbf{h}$, and $\mathcal{T}$
Output: samples with the asymptotic distribution $\mathcal{N}^{-1}(\mathbf{h}, J)$

1) Construct $J_T$ and $K$ using (17) and (18).
2) Draw an initial sample $\mathbf{x}^{(0)}$ from a Gaussian distribution.
3) At each iteration:
   a) Generate an independent sample $\tilde{\mathbf{e}}^{(t+1)}$ using (20).
   b) Generate a sample $\mathbf{x}^{(t+1)}$ from $\mathcal{N}^{-1}(\mathbf{h} + K\mathbf{x}^{(t)} + \tilde{\mathbf{e}}^{(t+1)}, J_T)$.

At iteration $t+1$ of the algorithm, rather than generating the noise vector $\mathbf{e}^{(t+1)}$ directly, instead we generate a noise vector $\tilde{\mathbf{e}}^{(t+1)}$ to be Gaussian with zero mean and covariance $K$, then let $\mathbf{x}^{(t+1)}$ be a sample from the Gaussian distribution with information matrix $J_T$ and potential vector $K\mathbf{x}^{(t)} + \mathbf{h} + \tilde{\mathbf{e}}^{(t+1)}$. Hence we have

$$\mathbf{x}^{(t+1)} = J_T^{-1}\left(K\mathbf{x}^{(t)} + \mathbf{h} + \tilde{\mathbf{e}}^{(t+1)}\right) + \mathbf{n}^{(t+1)}, \tag{19}$$

where $\mathbf{n}^{(t+1)}$ is Gaussian with zero mean and covariance $J_T^{-1}$. The above procedure is equivalent to Algorithm 1 since $\tilde{\mathbf{e}}^{(t+1)} + J_T\mathbf{n}^{(t+1)}$ is equal in distribution to $\mathbf{e}^{(t+1)}$, whose covariance matrix is $J_T + K$. Note that $\boldsymbol{n}^{(t+1)}$ can be generated efficiently thanks to the assumption that $J_T$ is tractable (e.g., if it is tree-structured, the sample can be generated by forward sampling). Furthermore, the structure of $K$ allows $\tilde{\mathbf{e}}^{(t+1)}$ to be computed efficiently and locally: For each $(i,j) \in \mathcal{E}\backslash\mathcal{E}_\mathcal{T}$, let $\tilde{\mathbf{e}}^{(i,j)}$ be a two-dimensional vector sampled from a zero-mean Gaussian distribution with covariance matrix $K^{(i,j)}$. Moreover, note that since each of the matrices $K^{(i,j)}$ is rank-1, we can generate each of the $\tilde{\mathbf{e}}^{(i,j)}$ using an independent scalar sample drawn from the standard Gaussian distribution $\mathcal{N}(0,1)$ and then multiplying this by the vector $[1, -\mathrm{sgn}(J_{ij})]^T\sqrt{|J_{ij}|}$. We then obtain $\tilde{\mathbf{e}}^{(t+1)}$ by computing

$$\tilde{\mathbf{e}}^{(t+1)} = \sum_{(i,j)\in\mathcal{E}\backslash\mathcal{E}_\mathcal{T}} \left[\tilde{\mathbf{e}}^{(i,j)}\right]_n, \tag{20}$$

where $[\tilde{\mathbf{e}}^{(i,j)}]_n$ is the $n$-dimensional vector zero-padded from $\tilde{\mathbf{e}}^{(i,j)}$, i.e., the $i$-th and $j$-th entries of $[\tilde{\mathbf{e}}^{(i,j)}]_n$ take the two entries of $\tilde{\mathbf{e}}^{(i,j)}$ and all other entries of $[\tilde{\mathbf{e}}^{(i,j)}]_n$ are zero.

We have that $J_T + K \succ 0$ from our construction, so this constitutes a P-regular graphical splitting. Hence according to Proposition 1 and Theorem 3, the sample distribution converges to the target distribution. This local implementation is summarized in Algorithm 2. The computational complexity of one iteration is $C_\mathcal{T} + \mathcal{O}(|\mathcal{E}_K|)$, where $C_\mathcal{T}$ is the complexity of drawing a sample from the tractable subgraph $\mathcal{T}$ and $|\mathcal{E}_K| = |\mathcal{E}| - |\mathcal{E}_\mathcal{T}|$ is the number of edges missing from $J_T$.

## IV. Sampling by Subgraph Perturbations With Non-Stationary Graphical Splittings

In the previous section, we have introduced the subgraph perturbation algorithm with stationary splittings. It is natural to

**Algorithm 3: Sampling by Subgraph Perturbations with non-Stationary Splittings**

Input: $J$, $\mathbf{h}$
Output: samples with the asymptotic distribution $\mathcal{N}^{-1}(\mathbf{h}, J)$

1) Draw an initial sample $\mathbf{x}^{(0)}$ from a Gaussian distribution.
2) At each iteration $t$ for $t = 1, 2, 3, \ldots$
   a) Form a graphical splitting $J = J_{T_t} - K_t$, where $J_{T_t} + K_t \succ 0$.
   b) Generate an independent sample $\mathbf{e}^{(t+1)}$ with zero mean and covariance matrix $J_{T_t} + K_t$.
   c) Compute $\mathbf{x}^{(t+1)}$ using the equation $\mathbf{x}^{(t+1)} = J_{T_t}^{-1}(\mathbf{h} + K_t\mathbf{x}^{(t)} + \mathbf{e}^{(t+1)})$.

extend Algorithm 1 to using multiple subgraphs for different iterations (i.e., $J = J_{T_t} - K_t$ at iteration $t$), which we refer to as *non-stationary* graphical splittings. Using non-stationary graphical splittings for sampling is related to using non-stationary graphical splittings for inference (i.e., for computation of the mean) [23], [24], but the additional constraint $J_{T_t} + K_t \succ 0$ is needed to ensure that the added noise at each iteration is valid. In this section, we first summarize our sampling algorithm using non-stationary graphical splittings in Algorithm 3 and then present theoretical results on convergence. The results in this section provide theoretical foundations for the adaptive selection of the splittings, which will be studied in Section V.

The authors in [23] have studied the use of periodic splittings (i.e., using the set of splittings $\{J = J_{T_t} - K_t\}_{t=1}^{P}$ in a periodic manner) for inference as a special case of using an arbitrary sequence of splittings. In this case, the average convergence rate is $-\frac{1}{P}\sum_{t=1}^{P} \ln \rho(J_{T_t}^{-1}K_t)$. While a non-trivial sufficient condition guaranteeing convergence for a general $P$ is difficult to find, the authors have given a sufficient condition for the case $P = 2$. In [24] the inference problem for a GMRF is solved by adaptively selecting the next graphical splitting given the current error residual. The authors have proven that if a GMRF is walk-summable (cf. [24]), then their algorithm converges to the correct solution for an arbitrary sequence of splittings where the diagonal of each of the $K$'s is fixed to be zero.

In order for our non-stationary perturbation sampler to proceed, the noise covariance matrix at each iteration needs to be positive semidefinite (which is equivalent to the P-regularity condition according to Proposition 1). Because of this extra constraint, the conclusions for inference using non-stationary splittings do not directly apply to sampling. In the following Theorem 6, we prove that as long as we have the condition in the theorem, namely that each element in the set of splittings would produce by itself a convergent stationary perturbation sampler, the use of any arbitrary sequence from this set (including, of course, periodic selection) also leads to a convergent algorithm.

*Theorem 6:* Consider a finite collection of graphical splittings $\mathcal{S} = \{J = J_{T_i} - K_i\}_{i=1}^{N}$. The non-stationary subgraph perturbation sampling algorithm (Algorithm 3) converges to the target distribution with an arbitrary sequence of splittings chosen from $\mathcal{S}$ if and only if the stationary sampling algorithm (Algorithm 1) converges to the target distribution with each of the splittings in the sequence.

We now state several lemmas prior to proving Theorem 6. The proofs for these lemmas are provided in the Appendix.

*Lemma 7:* If $J \succ 0$ and the graphical splitting $J = J_T - K$ is P-regular, then there exists $\epsilon > 0$ such that $J - (J_T^{-1}K)^T J (J_T^{-1}K) \succeq \epsilon J$.

For a positive definite matrix $J$, we define the induced matrix norm $||A||_{J\to J}$ as $||A||_{J\to J} = \max_{\mathbf{u}\neq 0} \frac{||A\mathbf{u}||_J}{||\mathbf{u}||_J}$, where the vector norm $||\mathbf{u}||_J$ is defined by $||\mathbf{u}||_J = (\mathbf{u}^T J \mathbf{u})^{\frac{1}{2}}$.

*Lemma 8:* If $J \succ 0$ and $J = J_T - K$ is a P-regular graphical splitting, then $||J_T^{-1}K||_{J\to J} < 1$.

For a P-regular graphical splitting $J = J_T - K$, Lemma 8 states that $||J_T^{-1}K||_{J\to J} < 1$. In general it is not true that $||KJ_T^{-1}||_{J\to J} < 1$; however, the following Lemma 9 establishes that under mild conditions there exists an integer $p$ such that the $J$-induced norm of the product $\prod_{i=1}^{p} K_i J_{T_i}^{-1}$ is less than $\frac{1}{2}$.

*Lemma 9:* Consider $J \succ 0$ and a sequence of P-regular graphical splittings $\{J = J_{T_{u_t}} - K_{u_t}\}_{t=1}^{\infty}$. If the splittings are chosen from a finite number of distinct graphical splittings $\{J = J_{T_i} - K_i\}_{i=1}^{N}$, then there exists a positive integer $p$ depending only on $J$ such that $||\prod_{i=m}^{p+m-1}(K_{u_i}J_{T_{u_i}}^{-1})||_{J\to J} < \frac{1}{2}$ for any positive integer $m$.

*Proof of Theorem 6:*

*Proof:* The necessity is easy to prove since for Algorithm 3 to proceed, the noise at each iteration needs to be valid, which implies the convergence with each of the splittings according to Proposition 1.

Now we prove the sufficiency. Similarly as in the proof of Theorem 3, we use $\boldsymbol{\mu}^{(t)}$ and $\Sigma^{(t)}$ to represent the mean and covariance matrix of the sample distribution at iteration $t$. From Step 2 of Algorithm 3, we can prove that

$$\boldsymbol{\mu}^{(t+1)} - \boldsymbol{\mu} = J_{T_t}^{-1}K_t\left(\boldsymbol{\mu}^{(t)} - \boldsymbol{\mu}\right) \tag{21}$$

and

$$\Sigma^{(t+1)} - \Sigma = \left(J_{T_t}^{-1}K_t\right)\left(\Sigma^{(t)} - \Sigma\right)\left(J_{T_t}^{-1}K_t\right)^T. \tag{22}$$

From Lemma 8, we have that $||J_{T_t}^{-1}K_t||_{J\to J} < 1$. Since $\mathcal{S}$ is a finite collection of splittings, let $\sigma_{\max} = \max_{i\in\mathcal{S}} ||J_{T_i}^{-1}K_i||_{J\to J} < 1$. Hence

$$\left|\left|\boldsymbol{\mu}^{(t)} - \boldsymbol{\mu}\right|\right|_J = \left|\left|\prod_{i=1}^{t}\left(J_{T_i}^{-1}K_i\right)\left(\boldsymbol{\mu}^{(0)} - \boldsymbol{\mu}\right)\right|\right|_J \tag{23}$$

$$\leq \prod_{i=1}^{t}\left|\left|J_{T_i}^{-1}K_i\right|\right|_{J\to J}\left|\left|\boldsymbol{\mu}^{(0)} - \boldsymbol{\mu}\right|\right|_J \tag{24}$$

$$\leq \sigma_{\max}^{t}\left|\left|\boldsymbol{\mu}^{(0)} - \boldsymbol{\mu}\right|\right|_J. \tag{25}$$

Similarly,

$$\left|\left|\Sigma^{(t)} - \Sigma\right|\right|_{J\to J} \tag{26}$$

$$= \left|\left|\left(\prod_{i=1}^{t}\left(J_{T_i}^{-1}K_i\right)\right)\left(\Sigma^{(0)} - \Sigma\right)\left(\prod_{i=1}^{t}\left(J_{T_i}^{-1}K_i\right)\right)^T\right|\right|_{J\to J} \tag{27}$$

$$\leq \left|\left|\prod_{i=1}^{t}\left(J_{T_i}^{-1}K_i\right)\right|\right|_{J\to J}\left|\left|\Sigma^{(0)} - \Sigma\right|\right|_{J\to J}\left|\left|\prod_{i=1}^{t}\left(K_iJ_{T_i}^{-1}\right)\right|\right|_{J\to J} \tag{28}$$

Let $p$ be the integer in Lemma 9. Then when $t > p$, we have that

$$\left|\left|\Sigma^{(t)} - \Sigma\right|\right|_{J\to J} \leq \sigma_{\max}^t \left|\left|\Sigma^{(0)} - \Sigma\right|\right|_{J\to J} C, \qquad (29)$$

where $C = \max\{\delta_{\max}, \delta_{\max}^{p-1}, \frac{1}{2}\} \geq 0$ and $\delta_{\max} = \max_{i\in\mathcal{S}} ||K_i J_{T_i}^{-1}||_{J\to J}$.

For a positive definite symmetric matrix $J$ of finite dimension, there exist $0 < D_1 \leq D_2$ such that $D_1||\boldsymbol{v}||_2 \leq ||\boldsymbol{v}||_J \leq D_2||\boldsymbol{v}||_2$ for any vector $\boldsymbol{v}$ and $0 < C_1 \leq C_2$ such that $C_1||A||_F \leq ||A||_{J\to J} \leq C_2||A||_F$ for any matrix $A$. Hence, we have that $||\boldsymbol{\mu}^{(t)} - \boldsymbol{\mu}||_2 \leq \frac{D_2}{D_1}\sigma_{\max}^t||\boldsymbol{\mu}^{(0)} - \boldsymbol{\mu}||_2$ and $||\Sigma^{(t)} - \Sigma||_F \leq \frac{C_2 C}{C_1}\sigma_{\max}^t||\Sigma^{(0)} - \Sigma||_F$. Therefore, Algorithm 3 converges using this sequence of splittings. This concludes the proof of Theorem 6. ■

*Alternative Proof of Corollary 1 in [24]:* One of the main results in [24] states that if the graphical model is walk-summable then the embedded tree algorithm converges to the correct mean using any sequence of graphical splittings $\{J = J_{T_t} - K_t\}$ where each $J_{T_t}$ corresponds to a tree-structured graph and each $K_t$ corresponds to the cut edges and has zero diagonal. The original proof in [24] uses walk-sum diagrams. Here we give an alternative proof using results presented in this section.[6]

*Proof:* Consider the splittings used in [24], where $K_t$ has zero diagonal and the nonzero off-diagonal entries of $K_t$ take the opposite values of the corresponding entries in $J$. We define $J_t^* = J_{T_t} + K_t$ and thus the entries in $J_t^*$ have the same absolute values as the corresponding entries in $J$. Since $J$ is walk-summable, we have that $J_t^*$ is also walk-summable by the definition of walk-summability (cf. [36]). Since walk-summability implies the validity of a model, we have that $J_t^* \succ 0$. By Lemma 8, we have that $||J_{T_t}^{-1}K_t||_{J\to J} < 1$ for all $t$. Since there are a finite number of different splittings in this setting, we can show the convergence using the same arguments as in the proof of Theorem 6.

■

## V. The Selection of Tractable Subgraphs

In this section, we discuss the selection of tractable subgraphs. First, we discuss how to choose graph structures for stationary splittings; then, we propose an algorithm to adaptively select tractable subgraphs for non-stationary splittings.

### A. Select Subgraph Structures for Stationary Splittings

*1) Using Tree-Structured Subgraphs:* From the inequalities in Corollary 5, a heuristic is to choose $K$ with small absolute edge weights and at the same time ensure the rest of the graph is tree-structured. Hence, the tree-structured subgraph is encouraged to contain strong edges. An effective method is to find the maximum spanning tree (MST) with edge weights $w_{ij}$ being the absolute values of the normalized edge weights in $J$, i.e., $w_{ij} = |J_{ij}|/\sqrt{J_{ii}J_{jj}}$. The idea of using an MST has been discussed in the support graph preconditioner literature [37] as well as in the studies of the embedded tree algorithm for inference [24]. An MST can be constructed using Kruskal's algorithm in $\mathcal{O}(m \log n)$ time, where $m$ is the number of edges. This selection procedure is summarized in Algorithm 4.

**Algorithm 4: Selecting a Tree-Structured Subgraph**

**Input**: $J \succ 0$
**Output**: a tree-structured subgraph $\mathcal{T}$
1) Compute the normalized edge weights $w_{ij} = |J_{ij}|/\sqrt{J_{ii}J_{jj}}$ for all $(i,j) \in \mathcal{E}$.
2) Compute the maximum spanning tree $\mathcal{T}$ using edge weights $w_{ij}$.

In our perturbation sampling framework, the tractable subgraphs can be structures beyond trees. Here we also suggest several other tractable graph structures with existing efficient inference and sampling algorithms.

*2) Using Subgraphs With Low Tree-Width:* Graphical models with low tree-width have efficient inference and sampling algorithms and have been widely studied. We can compute a low tree-width approximation $J_T$ to $J$ using algorithms such as those in [38]–[40].

*3) Using Subgraphs With Small FVSs:* As mentioned in Section II, an FVS is a set of nodes whose removal results in a cycle-free graph. In [25] an exact inference algorithm was given for graphical models with small FVSs. This work allows one to use a graph with a small FVS as the tractable subgraph in our framework. Moreover, [25] introduced the concept of a *pseudo-FVS*, which is a set of nodes that breaks most, but not all, of the cycles in the graph. We can first use the algorithm in [25] to select a set of nodes $\mathcal{F}$ constituting a pseudo-FVS for the full graph. Then we compute a MST among the other nodes. We choose our subgraph to be the combination of nodes $\mathcal{F}$ (with all incident edges) as well as the MST of the remaining graph. Note that even though $\mathcal{F}$ is a pseudo-FVS of the original graph, it is a true FVS of the subgraph, and therefore the algorithm from [25] provides exact inference. Using this technique, there is a trade-off in choosing the size of $\mathcal{F}$: a larger set $\mathcal{F}$ means more computation per iteration but faster convergence.

*4) Using Spectrally Sparsified Subgraphs:* Many widely used GMRFs such as thin-membrane or thin-plate models have diagonally dominant information matrices. Some recent studies have shown that the graph Laplacian of a dense graph can be well-approximated by the graph Laplacian of graphs with nearly-linear number of edges [41]. These spectrally sparsified graphs have efficient inference and sampling algorithms and can also be used as tractable subgraphs.

### B. Adaptive Selection of Graph Structures for Non-Stationary Splittings

In this subsection, we propose an algorithm to adaptively select the structure of the subgraphs for non-stationary splittings. We explain our algorithm assuming that each subgraph is tree-structured, but this algorithm can be extended to other tractable subgraphs such as those mentioned in the previous subsection.

[6]Note that our sampling algorithm requires additional constraints to ensure the validity of the added noise. It is coincidental that the results in this paper lead to an alternative proof of one of the main results in [24].

From Algorithm 3, it can be shown that

$$\boldsymbol{\mu} - \boldsymbol{\mu}^{(t+1)} = \left(J^{-1} - J_{T_t}^{-1}\right)\left(\mathbf{h} - J\boldsymbol{\mu}^{(t)}\right), \tag{30}$$

which characterizes the residual error for the mean. Similarly, for the sample covariance, we have

$$\Sigma - \Sigma^{(t+1)} = \left(J^{-1} - J_{T_t}^{-1}\right)\left(J - J\Sigma^{(t)}J\right)\left(J^{-1} - J_{T_t}^{-1}\right)^T. \tag{31}$$

In [24] the authors have proposed an adaptive method using the walk-sum analysis framework: at each iteration $t+1$, choose the MST $\mathcal{T}_i$ in (30) with weights $\delta_{u,v}^{(t)}$ for edge $(u,v)$, where

$$\delta_{u,v}^{(t)} = \left(\left|h_u^{(t)}\right| + \left|h_v^{(t)}\right|\right)\frac{|J_{u,v}|}{1-|J_{u,v}|} \tag{32}$$

and $\mathbf{h}^{(t)} = \mathbf{h} - J\boldsymbol{\mu}^{(t)}$.[7] This adaptive method significantly improves the speed of convergence for inference compared with using stationary splittings. In our case of sampling, both the error for the mean and the error for the covariance matrix need to be considered. However, a similar relaxation for the covariance matrix based on (31) is too computationally costly. Hence, we resort to an auxiliary inference problem with the same information matrix $J$ and the potential vector $\mathbf{h}^*$ being the all-one vector. At each iteration of our sampling algorithm, we use the subgraph adaptively selected based on the auxiliary inference algorithm (i.e., choosing the MST with weight as in (32) but using the potential vector $\mathbf{h}^*$).

## VI. Experimental Results

In this section, we present experimental results using our perturbation sampling algorithms with both stationary graphical splittings and non-stationary graphical splittings. In the first two sets of experiments, we use simulated models on grids of various sizes; in the third example, we use standard test data of a power network of moderate size; finally, we present results using a large-scale real example for sea surface temperature estimation.

### A. Motivating Example: 3 × 10 Grids

In this motivating example, we consider a simple $3 \times 10$ grid (Fig. 3(a)). In the simulated models, the model parameters $J$ and $\mathbf{h}$ are randomly generated as follows: the entries of the potential vector $\mathbf{h}$ are generated $i.i.d.$ from a uniform distribution $U[-1,1]$; the sparsity pattern of $J$ is determined by the graph structure and the non-zero entries of $J$ are also generated *i.i.d.* from $U[-1,1]$ with a multiple of the identity matrix added to ensure $J \succ 0$. We compare several sampling algorithms, namely basic Gibbs sampling, chessboard (red-black) blocked Gibbs sampling (Fig. 3(b)), forest Gibbs sampling (Fig. 3(c), cf. [20]), and our algorithm using a stationary splitting (Fig. 3(d)) selected with Algorithm 4 (listed as "1-Tree Perturbation" in Table I). We randomly generate 100 sets of model parameters and compute the asymptotic convergence rates. The average numbers of iterations (to reduce the covariance error in half), i.e., average $\frac{\ln \frac{1}{2}}{\ln \tau_\Sigma}$, are shown in Table I.

[7]Note that here the matrix $J$ is normalized to have unit diagonal.

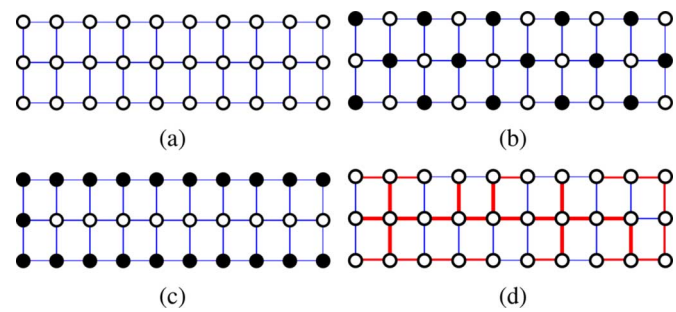


Fig. 3. Sampling from a $3 \times 10$ grid using basic Gibbs sampling, chessboard (red-black) Gibbs sampling, forest Gibbs sampling, and our subgraph perturbation sampling using a stationary splitting. (a) Graph structure of the $3 \times 10$ grid; (b) Chessboard (red-black) blocked Gibbs sampling: the set of black nodes and the set of white nodes form two blocks; (c) Forest Gibbs sampling: the set of black nodes and the set of white nodes form two separate trees. At each iteration of the forest Gibbs sampling, conditioned on one block, the other block is sampled by forward sampling; (d) Subgraph perturbation sampling using a fixed tree-structured subgraph: the thicker red edges are edges in the tree-structured subgraph while the thinner blue edges are edges in the cut matrix.

TABLE I
Convergence Rates of Various Sampling Algorithms

| | Average number of iteration (to reduce the covariance error in half) |
|---|---|
| Gibbs | 42.842 |
| Chessboard Gibbs | 42.842 |
| Forest Gibbs | 18.846 |
| 1-Tree Perturbation | 5.967 |

We also study the convergence rates using non-stationary splittings. For each generated model, we run Algorithm 3 for 20 iterations and obtain 20 tree-structured subgraphs adaptively selected using (32). Fig. 4 shows the first four tree-structured subgraphs adaptively selected on one of the generated models. We summarize the asymptotic convergence rates in Table II for the following six cases: 1) the single tree that gives the best convergence among the 20 trees[8]; 2) the worst single tree of the 20 trees; 3) alternating between the best pair of trees (by an exhaustive search among all pairs of the 20 trees); 4) alternating between the worst pair of trees; 5) using the first two adaptively selected trees (and alternating between them); and 6) using adaptively selected trees at each of the 20 iterations. From the results, we can see that using different subgraph structures give significantly different performances. On average, the best single tree can reduce the residual covariance error in half in 6 iterations while the worst single tree takes 88 iterations. The best combination of two trees gives the best

[8]The number of all spanning trees of a grid is very large (there are more than $9.41 \times 10^9$ spanning trees even for this small $3 \times 10$ grid, computed using recursive equations in [42]), which makes it intractable to do exhaustive search among all spanning trees. In addition, for a fair comparison with the adaptive method, the single tree is chosen from the 20 adaptively selected trees.

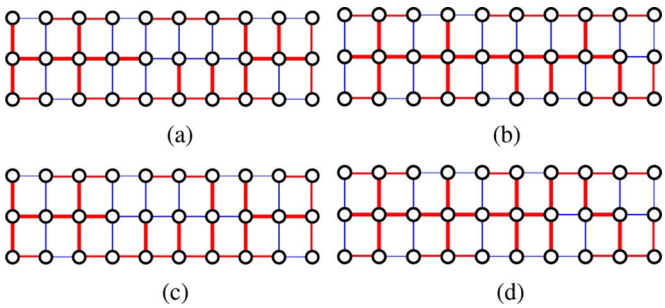


Fig. 4. Sampling from a $3 \times 10$ grid using non-stationary splittings. (a)—(d) show the first four trees adaptively selected using (32) on one run. (a) First tree, (b) Second tree, (c) Third tree, (d) Fourth tree.

TABLE II
CONVERGENCE RATES OF SUBGRAPH PERTURBATION USING NON-STATIONARY GRAPHICAL SPLITTINGS

| | Average number of iteration (to reduce the covariance error in half) |
|---|---|
| Best single tree of the first 20 trees | 5.4365 |
| Worst single tree of the first 20 trees | 87.397 |
| Best pair of trees of the first 20 trees | 3.6513 |
| Worst pair of trees of the first 20 trees | 87.397 |
| The first two trees adaptively selected trees | 5.5236 |
| All of the 20 adaptively selected trees | 4.9719 |

convergence rate, but is included only as a benchmark, as exhaustive search is not computationally feasible in practice. Using the sequence of adaptively selected trees gives the second best performance while having much less computational complexity. The sampling algorithm with non-stationary graphical splittings outperforms its stationary counterpart even using the best single tree, which demonstrates the advantages of using non-stationary graphical splittings for sampling.

Note that the comparisons between sampling methods in Table I focus on the number of iterations required for each method, even though iterations for different methods may have different run-time complexities. Indeed, each iteration of our subgraph perturbation methods requires more computation than that of a Gibbs sampler. For example, a red-black blocked Gibbs sampler may take advantage of parallelization in a computer cluster giving very fast run-time per iteration. While precise analysis of the exact run-times of different methods, including the effects of parallelization, is beyond our scope, we provide the following order-of-magnitude argument that our method is comparable to a Gibbs sampler, and in many cases faster. Suppose that we have an $N \times N$ grid with nearest-neighbor graph structure. In this case, using simple red-black coloring, a single parallelized sweep of Gibbs (consisting of parallel updating of all red nodes and then all black ones) takes $\mathcal{O}(1)$ time. However, it takes $\mathcal{O}(N)$ iterations for the effects from nodes at one extreme of the grid to reach nodes at the other extreme—hence $\mathcal{O}(N)$ time for a set of iterations to have influences propagate throughout the graph. On the other hand, our method (which may also employ parallelization for computations on the embedded tree at each iteration) takes $\mathcal{O}(N)$ time but effects from one extreme to the other occur within each iteration. Hence on a fair playing field, the two approaches have comparable time complexity for effects to propagate throughout the graph. Moreover, if graphs are weakly correlated, long-distance effects are very small, so that fully parallel Gibbs will win out. On the other hand, for strongly correlated graphs, as are needed to capture random fields with correlations at a full range of scale (e.g., as is the case for the ocean data used in our large-scale example below), those long-distance correlations are crucial, and in such cases, Gibbs, even parallelized, can take considerable time to capture those correlations. We also note that for graphs with small numbers of very high degree nodes, the complexity of Gibbs iterations, even if parallelized, increases (e.g., due to the computations and communication required to update those nodes). Such high-degree nodes can provide mechanisms for capturing long-distance correlation efficiently, and our method, exploiting small feedback vertex sets can take advantage of this structure to achieve great gains in convergence. Indeed, as we have shown in [31] models consisting of small FVS's with edges essentially to all other nodes can do an excellent job of capturing such correlations, and for such a model (small FVS plus tree) our algorithm yields exact samples directly without need for iteration.

### B. Using Subgraphs Beyond Trees

In this experiment, we study the convergence rates using different subgraph structures on grids of various sizes. For each given structure, we randomly generate model parameters using the same method as in Subsection VI.A. We compute the numbers of iterations needed to achieve an approximating error of $\epsilon = 10^{-5}$, i.e., the minimum $t$ such that$\|\Sigma^{(t)} - \Sigma\|_F \leq \epsilon$. We run the subgraph perturbation algorithm on $l$-by-$l$ grids with $l$ ranging from 3 to 30. For each grid, two different subgraphs are used: one is a tree-structured subgraph, the other is a subgraph with an FVS of size $\lceil \log l^2 \rceil$. For each size, we repeat the algorithm for 100 sets of random model parameters and the results shown are averaged over the 100 runs. Since the sizes of the simulated models are moderate, we are able to compute and compare with the exact solutions. As we can see from Fig. 5, our subgraph perturbation algorithm outperforms the Gibbs sampler and the use of subgraphs with small FVSs gives further improvement on convergence rate.[9]

### C. Power System Network: Standard Test Matrix 494 Bus

In this subsection, we use standard test data from the Harwell-Boeing Sparse Matrix Collection, which includes standard

[9]Note that more computation is involved at each iteration using FMP, but the complexity grows slowly if, as in this example, we use FVSs of sizes that are logarithmic in the size of the overall graph.

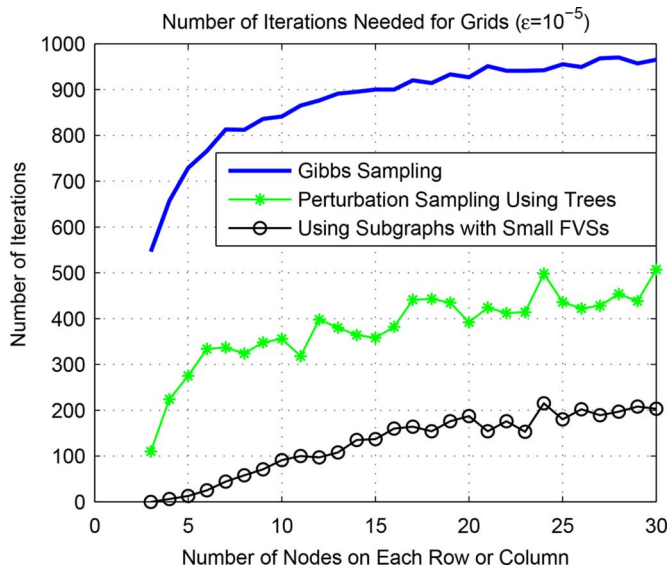


Fig. 5. The performance of subgraph perturbation sampling using various kinds of subgraphs on grids of size 3-by-3 to 30-by-30. The tractable subgraphs used include tree-structured graphs and graphs with small FVSs.

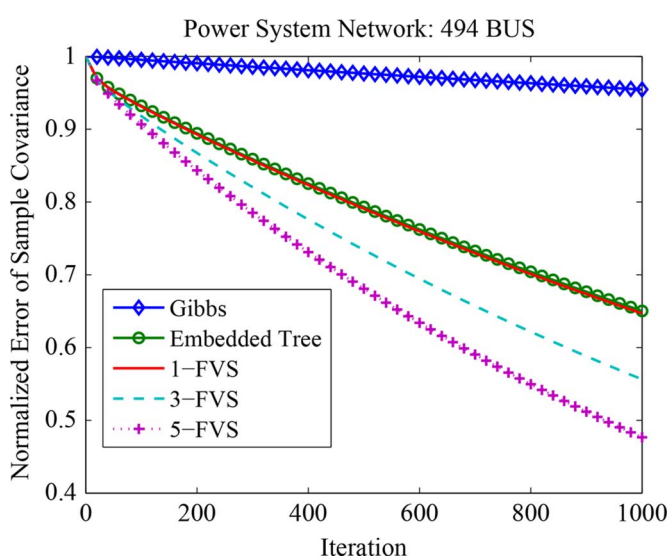


Fig. 6. Perturbation sampling using various subgraph structures on a power system network. The normalized error of the sample covariance is defined as the ratio between the sample covariance error at each iteration and the initial covariance error.

test matrices arising from a wide variety of scientific and engineering disciplines. We use the test matrix corresponding to a moderately sized (494 nodes) power system network[10]. We first add a multiple of the identity matrix to make the matrix positive definite and then normalize the matrix to have unit diagonal. Note that a diagonally dominant covariance matrix is easy to sample from (consider the extreme case of a diagonal matrix, which corresponds to independent Gaussian variables) even with the basic Gibbs sampler, but they do not represent many real applications. Hence, in order to study the models that are challenging for the Gibbs sampler or other common algorithms (which is the scenario that we focus on in this paper), we add just enough diagonal loading to make the matrix positive definite. We compare the performances of Gibbs sampling, subgraph perturbation sampling using a tree-structured subgraph and using subgraphs with FVSs of sizes one, three and five. In this experiment, we focus on stationary splittings since we are interested in comparing the performances using different types of subgraphs. The experimental results are shown in Table III and Fig. 6. As these results show, for this problem using a single tree subgraph reduces the number of iterations needed to achieve 50% error reduction by almost an order of magnitude, and using a very small size-5 FVS cuts the number down significantly further.

[10]The test matrix can be obtained from http://math.nist.gov/MatrixMarket/data/Harwell-Boeing/psadmit/494_bus.html.

TABLE III
CONVERGENCE RATES USING A SINGLE TREE AND SUBGRAPHS WITH FVS OF VARIOUS SIZES

| | Number of iterations (to reduce the covariance error in half) |
|---|---|
| Gibbs sampling (Gibbs) | 32653 |
| Subgraph perturbation with a tree (Embedded Tree) | 3491 |
| Subgraph perturbation sampling with a 1-FVS subgraph (1-FVS) | 3452 |
| Subgraph perturbation sampling with a 3-FVS subgraph (3-FVS) | 2500 |
| Subgraph perturbation sampling with a 5-FVS subgraph (5-FVS) | 1944 |

### D. Large-Scale Real Example: Sea Surface Temperature

We also run the algorithm on a large-scale GMRF built to estimate the sea surface temperature (the dataset is publicly available at http://podaac.jpl.nasa.gov/dataset/). The raw data is preprocessed to have raw measurements at $720 \times 1440$ different locations. We construct a grid of 1 036 800 nodes with additional edges connecting the eastmost and westmost nodes at the same latitudes since they are neighbors geographically. We then remove the nodes that have invalid measurements (most of which correspond to land areas). We construct a GMRF with this underlying structure using the thin-plate model [29]. Note that because of the significant number of observations, the information matrix for this model is far better conditioned than the one in the preceding section, implying that far fewer iterations are needed to reach approximate convergence. The structure of the resulting model is shown in Fig. 7(a) and the tractable subgraph used for our perturbation sampling algorithm is shown in Fig. 7(b) (for clarity, we plot a much coarser version and omit the edges connecting the eastmost and westmost nodes). A sample from the posterior distribution after 200 iterations is shown in Fig. 7(c).

## VII. CONCLUSION

The primary contributions of this paper include: (1) We provide a general framework for converting subgraph-based iterative solvers to samplers with convergence guarantees. In

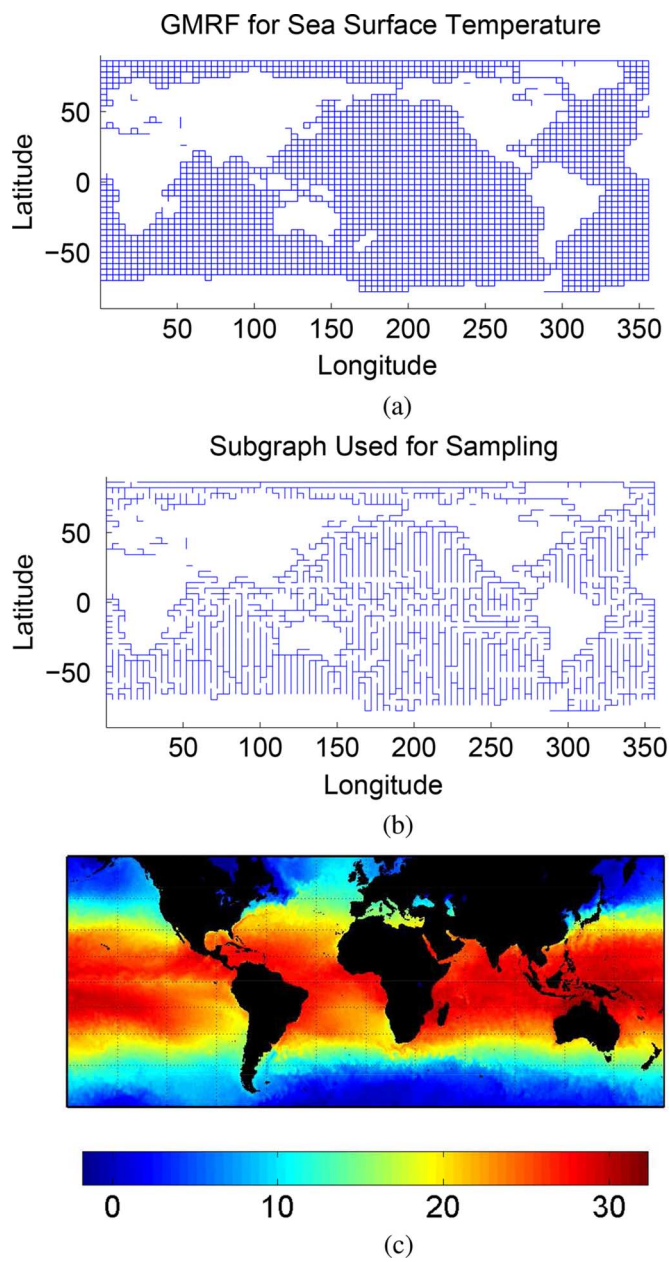


Fig. 7. Perturbation sampling from a GMRF for sea surface temperature estimation. (a) The entire GMRF for sea surface temperature, (b) The spanning tree used as a tractable subgraph, (c) Sea surface temperature in degrees (Celsius).

addition, we provide a construction where the injected noise at each iteration can be generated simply using a set of i.i.d. scalar Gaussian random variables. (2) We extend our perturbation sampling algorithm from stationary graphical splittings to non-stationary graphical splittings. In the previous studies on linear solver, it has been observed that using multiple subgraphs may give much better convergence than using any of the individual subgraphs. We prove that if we choose from a finite collection of P-regular graphical splittings, then the convergence is always guaranteed. (3) We study the use of different kinds of tractable subgraphs and we also propose an algorithm to adaptively select the subgraphs based on an auxiliary inference problem.

## Appendix A

*Proof of Proposition 1:*

*Proof:* We first prove the sufficiency. If $J_T + K \succ 0$, then $2J_T - J \succ 0$ and thus $J_T \succ 0$. Hence, $J_T^{-1} \succ 0$ has a unique positive definite square root $J_T^{-\frac{1}{2}} \succ 0$. Then we have $0 \prec J_T^{-\frac{1}{2}} J J_T^{-\frac{1}{2}} = J_T^{-\frac{1}{2}}(J_T - K)J_T^{-\frac{1}{2}} = I - J_T^{-\frac{1}{2}} K J_T^{-\frac{1}{2}}$. Hence, $\lambda_i(J_T^{-\frac{1}{2}} K J_T^{-\frac{1}{2}}) < 1$, for all $i$, where $\lambda_i(\cdot)$ denotes the $i$-th eigenvalue of the argument. From the condition $J_T + K \succ 0$, we have that $I + J_T^{-\frac{1}{2}} K J_T^{-\frac{1}{2}} = J_T^{-\frac{1}{2}}(J_T + K)J_T^{-\frac{1}{2}} \succ 0$, and thus $\lambda_i(J_T^{-\frac{1}{2}} K J_T^{-\frac{1}{2}}) > -1$, for all $i$. Because $J_T^{-1}K = J_T^{-\frac{1}{2}}(J_T^{-\frac{1}{2}} K J_T^{-\frac{1}{2}})J_T^{\frac{1}{2}}$, we have that $J_T^{-1}K$ has the same eigenvalues as $J_T^{-\frac{1}{2}} K J_T^{-\frac{1}{2}}$. Therefore, $|\lambda_i(J_T^{-1}K)| < 1$ for all $i$ and thus $\rho(J_T^{-1}K) < 1$.

We now prove the necessity. If $\rho(J_T^{-1}K) < 1$, then $I - J_T^{-1}K = J_T^{-1}J$ has positive eigenvalues. Since $J \succ 0$, $J$ has a unique positive definite square root $J^{-\frac{1}{2}} \succ 0$, and thus $0 \prec J^{\frac{1}{2}} J_T^{-1} J^{\frac{1}{2}} = J^{\frac{1}{2}}(J_T^{-1}J)J^{-\frac{1}{2}}$. So we have $J_T^{-1} \succ 0$. Hence $J_T \succ 0$ has a unique positive definite square root $J_T^{\frac{1}{2}} \succ 0$. So $J_T^{-\frac{1}{2}} K J_T^{-\frac{1}{2}}$ has the same eigenvalues as $J_T^{-1}K$ since $J_T^{-\frac{1}{2}} K J_T^{-\frac{1}{2}} = J_T^{\frac{1}{2}}(J_T^{-1}K)J_T^{-\frac{1}{2}}$, and thus $\rho(J_T^{-\frac{1}{2}} K J_T^{-\frac{1}{2}}) < 1$. Hence, $I + J_T^{-\frac{1}{2}} K J_T^{-\frac{1}{2}} \succ 0$, so $J_T + K = J_T^{\frac{1}{2}}(I + J_T^{-\frac{1}{2}} K J_T^{-\frac{1}{2}})J_T^{\frac{1}{2}} \succ 0$. Therefore, $J = J_T - K$ is a P-regular splitting. ■

*Proof of Lemma 2:*

*Proof:* It is equivalent to showing

$$A(A+B)^{-1}A^T = B(A+B)^{-1}B^T + A^T - B.$$

To do so, consider

$$\begin{aligned}
\text{LHS} &= ((A+B) - B)(A+B)^{-1}A^T \\
&= A^T - B(A+B)^{-1}A^T \\
&= A^T - B(A+B)^{-1}((A^T + B^T) - B^T) \\
&= A^T - B(A+B)^{-1}(A+B)^T + B(A+B)^{-1}B^T \\
&\overset{(a)}{=} A^T - B + B(A+B)^{-1}B^T = \text{RHS},
\end{aligned}$$

where $(a)$ is due to the assumption that $A + B$ is symmetric. ■

*Proof of Lemma 7:*

*Proof:* Since $J \succ 0$, there exists some $\delta_h \geq \delta_l > 0$ such that $\delta_h I \succ J \succ \delta_l I$. Hence, to prove Lemma 7, it is sufficient to show that there exists $\tilde{\epsilon} > 0$ such that $J - (J_T^{-1}K)^T J (J_T^{-1}K) \succeq \tilde{\epsilon} I$, which is equivalent to showing that $J - (J_T^{-1}K)^T J (J_T^{-1}K) \succ 0$.

$$\begin{aligned}
& J - \left(J_T^{-1}K\right)^T J \left(J_T^{-1}K\right) \succ 0 \\
&\Leftrightarrow J - \left(I - J_T^{-1}J\right)^T J \left(I - J_T^{-1}J\right) \succ 0 \\
&\Leftrightarrow J - \left(J + JJ_T^{-1}JJ_T^{-1}J - 2JJ_T^{-1}J\right) \succ 0 \\
&\Leftrightarrow 2JJ_T^{-1}J - JJ_T^{-1}JJ_T^{-1}J \succ 0 \\
&\overset{(a)}{\Leftrightarrow} (J^{-1}J_T)^T \left(2JJ_T^{-1}J - JJ_T^{-1}JJ_T^{-1}J\right)(J^{-1}J_T) \succ 0 \\
&\Leftrightarrow 2J_T - J \succ 0 \\
&\Leftrightarrow J_T + K \succ 0,
\end{aligned}$$

where $(a)$ is due to that $J$ and $J_T$ are both non-singular since $J \succ 0$ and $J_T = \frac{J+(J_T+K)}{2} \succ 0$.

■

*Proof of Lemma 8:*

*Proof:* For any $u \neq 0$, we have that

$$\begin{aligned}&\left(J_T^{-1}Ku\right)^T J\left(J_T^{-1}Ku\right)\\&= u^T\left(\left(J_T^{-1}K\right)^T J\left(J_T^{-1}K\right)\right)u\\&= u^T\left(\left(J_T^{-1}K\right)^T J\left(J_T^{-1}K\right) - J\right)u + u^T Ju\end{aligned}$$

From Lemma 7, there exist $\epsilon > 0$ such that $(J_T^{-1}K)^T J(J_T^{-1}K) \preceq (1-\epsilon)J$. Hence, we have $(J_T^{-1}Ku)^T J(J_T^{-1}Ku) \leq (1-\epsilon)u^T Ju$. Thus for any $u \neq 0$, $\frac{(J_T^{-1}Ku)^T J(J_T^{-1}Ku)}{u^T Ju} \leq (1-\epsilon) < 1$. Hence, by the definition of the $J$-induced norm, we have that $||J_T^{-1}K||_{J\to J} < 1$. ■

*Proof of Lemma 9:*

*Proof:* Since the sequence is arbitrary, without loss of generality, we only need to prove for $m = 1$. Since $||\cdot||_{J\to J}$ is an induced norm, there exists $0 < C_1 \leq C_2$ depending only on $J$ such that $C_1||A||_F \leq ||A||_{J\to J} \leq C_2||A||_F$ for any square matrix $A$. From Lemma 8, $||J_{T_i}^{-1}K_i||_{J\to J} < 1$ for all $i$. Since there are finitely many distinct splittings, there exists $0 \leq \sigma_{\max} < 1$ such that $||J_{T_i}^{-1}K_i||_{J\to J} \leq \sigma_{\max} < 1$ for all $i$. For induced norms, it can be shown that $||AB||_{J\to J} \leq ||A||_{J\to J}||B||_{J\to J}$. Hence, there exists integer $p$ depending only on $J$ such that $||\prod_{i=p}^1 J_{T_{u_i}}^{-1}K_{u_i}||_{J\to J} \leq \prod_{i=p}^1 ||J_{T_{u_i}}^{-1}K_{u_i}|| \leq \sigma_{\max}^p \leq \frac{C_1}{2C_2}$. Since the Frobenius norm is invariant to transposition, we have that $||\prod_{i=1}^p (J_{T_{u_i}}^{-1}K_{u_i})^T||_F = ||\prod_{i=p}^1 J_{T_{u_i}}^{-1}K_{u_i}||_F$, and thus

$$\begin{aligned}\left\|\prod_{i=1}^p\left(K_{u_i}J_{T_{u_i}}^{-1}\right)\right\|_{J\to J} &\leq C_2\left\|\prod_{i=1}^p\left(K_{u_i}J_{T_{u_i}}^{-1}\right)\right\|_F &(33)\\&= C_2\left\|\prod_{i=p}^1\left(J_{T_{u_i}}^{-1}K_{u_i}\right)\right\|_F &(34)\\&\leq \frac{C_2}{C_1}\left\|\prod_{i=p}^1\left(J_{T_{u_i}}^{-1}K_{u_i}\right)\right\|_{J\to J} &(35)\\&\leq \frac{C_2}{C_1}\frac{C_1}{2C_2} &(36)\\&= \frac{1}{2}. &(37)\end{aligned}$$

This completes the proof.

■

## REFERENCES


[1] Y. Liu, O. Kosut, and A. S. Willsky, "Sampling from Gaussian graphical models using subgraph perturbations," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2013, pp. 2498–2502.

[2] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, 4th ed. Independence, KY: Cengage Learning, 2014.

[3] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *J. Computat. Biol.*, vol. 7, no. 3–4, pp. 601–620, 2000.

[4] Y. Zhang, M. Brady, and S. Smith, "Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm," *IEEE Trans. Med. Imag.*, vol. 20, no. 1, pp. 45–57, 2001.

[5] F. R. Kschischang and B. J. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 219–230, 1998.

[6] C. Wunsch and P. Heimbach, "Practical global oceanic state estimation," *Physica D: Nonlin. Phenom.*, vol. 230, no. 1–2, pp. 197–208, 2007.

[7] L.-W. Yang, X. Liu, C. J. Jursa, M. Holliman, A. Rader, H. A. Karimi, and I. Bahar, "iGNM: A database of protein functional motions based on Gaussian network model," *Bioinform.*, vol. 21, no. 13, p. 2978, 2005.

[8] M. K. Titsias, N. D. Lawrence, and M. Rattray, "Efficient sampling for Gaussian process inference using control variables," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2008, vol. 21, pp. 1681–1688.

[9] R. Salakhutdinov, "Learning deep Boltzmann machines using adaptive MCMC," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2010, vol. 27.

[10] A. E. Gelfand, W. Gilks, R. S. , and S. D. J. , Eds., "Model determination using sampling-based methods," Markov Chain Monte Carlo in Practice 1996, pp. 145–161.

[11] S. Fine, Y. Singer, and N. Tishby, "The hierarchical hidden Markov model: Analysis and applications," *Mach. Learn.*, vol. 32, no. 1, pp. 41–62, 1998.

[12] T. Kailath, A. H. Sayed, and B. Hassibi, *Linear Estimation*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2000, vol. 1.

[13] K. Daoudi, A. B. Frakt, and A. S. Willsky, "Multiscale autoregressive models and wavelets," *IEEE Trans. Inf. Theory*, vol. 45, no. 3, pp. 828–845, 1999.

[14] M. I. Jordan, "Graphical models," *Statist. Sci.*, pp. 140–155, 2004.

[15] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York, NY, USA: Springer, 2004.

[16] Y. Amit and U. Grenander, "Comparing sweep strategies for stochastic relaxation," *J. Multivar. Anal.*, vol. 37, no. 2, pp. 197–222, 1991.

[17] A. Thomas, A. Gutin, V. Abkevich, and A. Bansal, "Multilocus linkage analysis by blocked Gibbs sampling," *Statist. Comput.*, vol. 10, no. 3, pp. 259–269, 2000.

[18] H. Rue, "Fast sampling of Gaussian Markov random fields," *J. Royal Statist. Soc. Ser. B (Statist. Methodol.)*, vol. 63, no. 2, pp. 325–338, 2001.

[19] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling, "Fast collapsed Gibbs sampling for latent Dirichlet allocation," in *Proc. Int. Conf. Knowl. Discov. Data Mining (KDD)*, 2008, pp. 569–577.

[20] F. Hamze and N. de Freitas, "From fields to trees," in *Proc. 20th Conf. Uncert. Artif. Intell. (UAI)*, 2004, pp. 243–250.

[21] G. Papandreou and A. L. Yuille, "Gaussian sampling by local perturbations," *Adv. Neural Inf. Process. Syst. (NIPS)*, pp. 1858–1866, 2010.

[22] T. A. Davis, *Direct Methods for Sparse Linear Systems*. Philadelphia, PA, USA: SIAM, 2006.

[23] E. B. Sudderth, M. J. Wainwright, and A. S. Willsky, "Embedded trees: Estimation of Gaussian processes on graphs with cycles," *IEEE Trans. Signal Process.*, vol. 52, no. 11, pp. 3136–3150, 2004.

[24] V. Chandrasekaran, J. K. Johnson, and A. S. Willsky, "Estimation in Gaussian graphical models using tractable subgraphs: A walk-sum analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1916–1930, 2008.

[25] Y. Liu, V. Chandrasekaran, A. Anandkumar, and A. S. Willsky, "Feedback message passing for inference in Gaussian graphical models," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4135–4150, 2012.

[26] J. M. Ortega, *Numerical Analysis: A Second Course*. Philadelphia, PA, USA: SIAM, 1990.

[27] C. Fox and A. Parker, "Convergence in variance of Chebyshev accelerated Gibbs samplers," *SIAM J. Scientif. Comput.*, vol. 36, pp. 124–147, 2013.

[28] A. Galli and H. Gao, "Rate of convergence of the gibbs sampler in the Gaussian case," *Math. Geol.*, vol. 33, no. 6, pp. 653–677, 2001.

[29] D. Malioutov, J. K. Johnson, M. J. Choi, and A. S. Willsky, "Lowrank variance approximation in GMRF models: Single and multiscale approaches," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4621–4634, 2008.

[30] J. Gonzalez, Y. Low, A. Gretton, and C. Guestrin, "Parallel gibbs sampling: From colored fields to thin junction trees," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2011, pp. 324–332.

[31] Y. Liu and A. S. Willsky, "Learning Gaussian graphical models with observed or latent FVSs," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2013.

[32] D. P. O'Leary and R. E. White, "Multi-splittings of matrices and parallel solution of linear systems," *SIAM J. Algebr. Discr. Methods*, vol. 6, no. 4, pp. 630–640, 1985.
[33] A. J. Laub, *Matrix Analysis for Scientists and Engineers*. Philadelphia, PA: Soc. Indust. Appl. Math., 2005.
[34] O. Axelsson, "Bounds of eigenvalues of preconditioned matrices," *SIAM J. Matrix Anal. Appl.*, vol. 13, no. 3, pp. 847–862, 1992.
[35] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1990.
[36] D. M. Malioutov, J. K. Johnson, and A. S. Willsky, "Walk-sums and belief propagation in Gaussian graphical models," *J. Mach. Learn. Res.*, vol. 7, pp. 2031–2064, 2006.
[37] M. Bern, J. R. Gilbert, B. Hendrickson, N. Nguyen, and S. Toledo, "Support-graph preconditioners," *SIAM J. Matrix Anal. Appl.*, vol. 27, no. 4, pp. 930–951, 2006.
[38] N. Srebro, "Maximum likelihood bounded tree-width Markov networks," in *Proc. 17th Conf. Uncert. Artif. Intell. (UAI)*, 2001, pp. 504–511.
[39] D. Shahaf and C. Guestrin, "Learning thin junction trees via graph cuts," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2009, pp. 113–120.
[40] D. Karger and N. Srebro, "Learning Markov networks: Maximum bounded tree-width graphs," in *Proc. 12th Ann. ACM-SIAM Symp. Discr. Algorithms. Soc. Indust. Appl. Math.*, 2001, pp. 392–401.
[41] D. A. Spielman and S.-H. Teng, "Spectral sparsification of graphs," *SIAM J. Comput.*, vol. 40, p. 981, 2011.
[42] M. Desjarlais and R. Molina, "Counting spanning trees in grid graphs," *Congressus Numerantium*, pp. 177–186, 2000.

**Ying Liu** (S'09–M'14) received the B.E. degree in electronic engineering from Tsinghua University, Beijing, China, in 2008, and the S.M. degree and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 2010 and 2014, respectively.

He was a Research Assistant with the Stochastic Systems Group under the guidance of Prof. Alan Willsky from 2008 to 2014. His research interests include machine learning, graphical models, stochastic signal processing, and distributed algorithm. He is currently with Google Inc. Cambridge office.

**Oliver Kosut** (S'06–M'10) received B.S. degrees in electrical engineering and mathematics from the Massachusetts Institute of Technology (MIT), Cambridge, in 2004, and the Ph.D. degree in electrical and computer engineering from Cornell University, Ithaca, NY, in 2010. He was a visiting student at the University of California at Berkeley during 2008–2009.

He was a Postdoctoral Research Associate in the Laboratory for Information and Decision Systems, MIT, from 2010 to 2012. Since 2012, he has been an Assistant Professor at Arizona State University, Tempe. His research interests include network information theory, security, and power systems.

**Alan S. Willsky** (S'70–M'73–SM'82–F'86–LF'13) is the M.I.T. E.S. Webster Professor of EECS (retired) and was a founder of Alphatech, Inc. He is a coauthor of *Signals and Systems*. His research interests are in the development and application of methods of estimation, machine learning, and statistical signal and image processing.

Dr. Willsky has received the following awards: the 1975 AACC Eckman Award, 1979 ASCE Alfred Noble Prize, 1980 IEEE Browder Thompson Award, 2004 IEEE Donald Fink Prize Paper Award, Doctorat Honoris Causa from the Université de Rennes, 2009 Technical Achievement, and 2014 Society Awards from the IEEE Signal Processing Society. He is a member of the National Academy of Engineering.