X. Tang et al. *Res Math Sci* (2023) 10:19
https://doi.org/10.1007/s40687-023-00381-3

Research in the Mathematical Sciences

RESEARCH

Generative modeling via tree tensor network states

Xun Tang[1*], YoonHaeng Hur[2], Yuehaw Khoo[3] and Lexing Ying[4]

*Correspondence:
xuntang@stanford.edu
[1]Institute for Computational and Mathematical Engineering, Stanford CA 94305, USA
Full list of author information is available at the end of the article

Abstract

In this paper, we present a density estimation framework based on tree tensor-network states. The proposed method consists of determining the tree topology with the Chow-Liu algorithm and obtaining a linear system of equations that defines the tensor-network components via sketching techniques. Novel choices of sketch functions are developed in order to consider graphical models that contain loops. For a wide class of d-dimensional density functions admitting the proposed ansatz, fast $O(d)$ sample complexity guarantees are provided and further corroborated by numerical experiments.

Keywords: Tensor network, Tree tensor network states, The curse of dimensionality, Generative modeling, Non-iterative method

1 Introduction

Generative modeling of a probability distribution is one of the most important tasks in machine learning, engineering, and science. In a nutshell, the goal of generative modeling is to approximate a high-dimensional distribution without the curse of dimensionality. There are generally several properties one would like to have for a generative model: (1) Can it be stored with a low memory complexity as the dimension grows? (2) Can it be determined from the given input with a low computational complexity? (3) Can it be used to generate samples with a low computational complexity? In this paper, we propose using a tree tensor network state as a generative model that enjoys these properties.

We focus on the problem of *density estimation*. More precisely, given N independent samples

$$(y_1^{(1)}, \ldots, y_d^{(1)}), \ \ldots \ (y_1^{(N)}, \ldots, y_d^{(N)}) \sim p^\star$$

drawn from some ground truth density $p^\star \colon \mathbb{R}^d \to \mathbb{R}$, our goal is to estimate $p^\star$ from the empirical distribution

$$\hat{p}(x_1, \ldots, x_d) = \frac{1}{N} \sum_{i=1}^{N} \delta_{(y_1^{(i)}, \ldots, y_d^{(i)})}(x_1, \ldots, x_d), \tag{1}$$

where $\delta_{(y_1, \ldots, y_d)}$ is the δ-measure supported on $(y_1, \ldots, y_d) \in \mathbb{R}^d$. It is hard to give a comprehensive survey of the broad field of density estimation, for this we refer readers

to [26]. Here we review several popular generative models that are related to our work. Energy-based models [14,18,27] such as graphical models represent a density by parameterizing it as the Gibbs measure of some energy function. Mixture models approximate the distribution via a composition of simple distributions. On the other hand, deep learning methods based on generative adversarial networks [12], variational auto-encoders [17] and normalizing flows [23,28] have gained tremendous popularity recently. Generally, obtaining the parameter of these parameterizations in a density estimation setting involves solving optimization problems that are often non-convex. Therefore frequently theoretical consistency guarantees of a density estimator cannot be achieved in practice. Furthermore, generating new samples from an optimized model could be difficult (for example in energy-based models) and requires running a Markov Chain Monte-Carlo.

Very recently, tensor-network methods, in particular matrix product state/tensor train, have emerged as an alternative paradigm for generative modeling [2,10,13,15]. Such methods represent the exponential size d-dimensional tensor as a network of d small tensors, achieving polynomial storage complexity in d. Moreover, for networks that can be contracted easily (e.g. tensor train), there exists an efficient strategy based on the conditional distribution method to generate independently and identically distributed samples [8]. The question is then whether one can determine the underlying tensor network efficiently for the task of density estimation. In [2,10,13], non-convex optimization approaches are applied to determine the tensor cores. Unlike these previous approaches, in [15], sketching is used to set up a parallel system of core-determining equations to determine the tensor cores of a tensor train without the use of optimization. We propose several extensions that generalize this work in terms of its practicality and reach.

We emphasize that there is another line of works in tensor literature that constructs low-rank tensor representations from sensing the entries of an order d-tensor. These include matrix completion [4] and its generalizations to tensor completion problem (e.g. [9,16,24]). Furthermore, cross-approximations [22] have been applied to the cases where one gets to choose the sensing pattern. The input data considered in these works are partial observation of the entries of the d-dimensional function, which is different from the case of density estimation, where empirical samples of the underlying distribution are given.

We now give a discussion that compares our method with other existing generative modeling methods. In Sect. 1.1, we compare TTNS with other potential tensor network architectures for generative modeling. In Sect. 1.2, we discuss the connection between TTNS and tree-based graphical models. In Sect. 1.3, we give a discussion on tensor network generative models which are based on iterative training.

1.1 Extending tensor train to tree-based tensor networks

For the case where the underlying density $p^\star$ has a tensor train (TT) format, an algorithm termed Tensor Train via Recursive Sketching (TT-RS) [15] has been introduced. In contrast to training-based modeling algorithms such as Born Machine [13], TT-RS is a generative modeling method with a rigorous sample complexity guarantee of convergence. One of the central motivations behind this paper is to obtain a convergent generative modeling method under the most general tensor network structural assump-

tion on $p^\star$. Our proposed Tree Tensor Network States (TTNS) format [21] generalizes TT. See Fig. 1 for an illustration of a tree and its corresponding TTNS tensor diagram.

We will give a short introduction to TTNS and related concepts. TTNS is a special case of Tree Tensor Network (TTN) [25]. In terms of representation power, there is a natural hierarchy

$$\mathrm{TT} \subset \mathrm{TTNS} \subset \mathrm{TTN} \subset \mathrm{TNS},$$

where TNS stands for Tensor Network States, which includes more general models such as projected entangled-pair states (PEPS) [30]. The class of the TTN models is the collection of tensor networks for which the internal bonds of the network do not form a loop, hence the word "tree". This absence of a loop means that the evaluation of a TTN scales polynomially in the number of nodes in the network. A tensor network ansatz belongs to the TTNS class if it is a TTN and each node in the network has exactly one external bond (i.e. an edge that does not connect to any other node).

We give a discussion to justify the choice of the TTNS ansatz. It is natural to choose the modeling assumption to be within the TTN class, as evaluation or sampling could not otherwise be efficient. However, to the best of our knowledge, the methods in TT-RS cannot generalize to an algorithm in the TTN ansatz because a TTN ansatz can have internal nodes, i.e. tensor components with no external bond. In contrast, a TTNS ansatz has exactly one external bond per tensor component, which allows the machinery of TT-RS to extend to TTNS ansatz. We remark that the work in [6] models the probability density by a TTN but it contains a training component and does not guarantee convergence.

1.2 Extending model inference of tree-based graphical models to tree-based tensor networks

Model inference for distributions with a TTNS ansatz is deeply related to the model inference problem of a tree-based graphical model, for which we will give an introduction in Sect. 6. The Chow-Liu algorithm [7] efficiently compresses a target density to the best tree-based graphical model in the sense of Maximum Likelihood Estimation (MLE). In terms of representational power, if the underlying distribution $p^\star$ is a tree-based graphical model, then $p^\star$ is guaranteed to have a tractable TTNS representation. On the other hand, the extra representation power of TTNS over tree-based graphical models allows the model to account for longer-range interactions between variables. To the best of our knowledge, this paper is the first instance where model inference of TTNS ansatz has been implemented.

After the model inference step, in terms of downstream tasks such as likelihood computation and sampling, both TTNS and tree-based graphical models scale linearly in the dimension d. Importantly, the samples produced from the TTNS ansatz have no autocorrelations and are i.i.d., which is a desirable property for generative modeling.

We call our main method Tree Tensor Network State via Sketching (TTNS-Sketch). By the sketching technique [32], the tensor components of the ansatz can be computed entirely with conventional linear algebraic equations, which results in a sample complexity that is quadratic in the dimension d. In terms of computational complexity scaling, the cost is linear in the sample size N, and at most quadratic in d. The method is proven to be a consistent estimator under reasonable technical assumptions.

1.3 Comparison between TTNS-Sketch and iterative algorithms

In contrast to a direct method such as TTNS-Sketch, iterative algorithm for tensor network methods (e.g. [6,10,13]) typically optimizes for the tensor components in the sense of minimal negative log-likelihood, with the maximum likelihood estimation (MLE) estimator as the optimizer. Despite the well-known Cramer-Rao bound for MLE estimators, the training in such iterative methods is non-convex, which prevents one from establishing theoretical guarantees, e.g. consistency and sample complexity. For example, [20] identifies the issue of vanishing gradient in randomly initialized quantum circuits for large qubit size. Due to the representational equivalence between tensor train and quantum circuits (see [10] for a discussion), the same issue also faces randomly initialized tensor train. This observation is also corroborated by the numerical experiments in Sect. 7, where we show the training failure of iterative methods under a setting far more modest than discussed in [20].

1.4 Main contribution

We list our main contributions as follows:

(1) We provide a simple notational system that can work well for arbitrary tree structures for TTNS ansatz. This structural flexibility is helpful for samples with an underlying tree structure but no practical path structure.
(2) We introduce perturbative sketching, motivated by randomized SVD [19]. We show that TTNS-Sketch with perturbative sketching performs well for models with short-range non-local interactions, thus exhibiting significant improvement over the graphical model given by Chow-Liu, which is only suitable for tree-based graphical models.
(3) We derive a general upper bound on the sample complexity of TTNS-Sketch. Based on the Wedin theorem and matrix Bernstein inequality, we obtain a non-asymptotic sample complexity upper bound for TTNS-Sketch under recursive sketching functions. Up to log factors and condition numbers, the sample complexity of the method scales by $N = O\left(\Delta(T)^2 d^2\right)$, where d is the number of nodes in the tree structure T, and $\Delta(T)$ stands for the maximal degree of T. This shows that TTNS-Sketch converges reasonably fast to the true model.
(4) We also identify a failure mode for iterative generative modeling methods based on the tensor train ansatz. For Born Machine (BM) [13] under a periodic spin system, we show that the training will fail unless one significantly increases the internal bond dimension. When the tensor train is of a correctly-sized internal bond dimension, the model learned by BM closely resembles that of a non-periodic spin system. In comparison, with a simple high-order Markov sketching function, TTNS-Sketch is successful at converging to $p^\star$ without over-parameterization. See Sect. 7 for detail.

1.5 Outline

The outline for the rest of this paper is as follows. Section 2 is an introduction to notations and to the basics of TTNS ansatz. Section 3 derives the essential linear equation to be used for TTNS-Sketch. Section 4 provides the main Algorithm and the condition for TTNS-Sketch to be a consistent estimator. Section 5 gives examples of sketch functions. Section 6 introduces the Chow-Liu algorithm for finding a tree structure using samples. Section 7

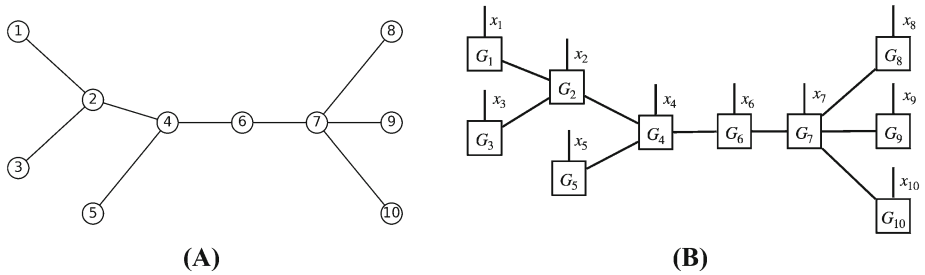


Fig. 1 **A** A tree structure $T = (V, E)$ with $V = \{1, \ldots, 10\}$. **B** Tensor Diagram representation of TTNS over T

gives the numerical result. Appendix E provides proof to the sample complexity upper bound of TTNS-Sketch under recursive sketching.

We summarize the workflow of TTNS-Sketch for the reader's convenience. When one is given a collection of samples, TTNS-Sketch breaks into three steps. In the first step, a tree structure T is determined by utilizing mutual information estimated from the samples (Sect. 6). In the second step, a tensor network structure is fixed based on the tree structure. (Sect. 2). In the third step, sketching techniques are used to solve for the tensor components of the TTNS ansatz (Sect. 4)

2 Introduction to TTNS

The aim of this section is to introduce the notation to describe a function with a TTNS ansatz. We first introduce some important notations frequently used. The letter d is reserved for the dimension of the joint distribution of interest, N is reserved for sample size, and T without a subscript is reserved for a tree graph. For any integer $q \in \mathbb{N}$, set $[q] := \{1, \ldots, q\}$.

For the TTNS ansatz, we use specific letters to label its indices. The letter x is reserved for the physical index (external bond) of the tensor core, and the letters α, β, γ are reserved for the internal bond of the tensor core.

Crucial equations are illustrated with a tensor diagram representation for convenience. To provide a concrete example, all tensor diagrams are plotted based on the specific tree structure set in Fig. 1a.

2.1 Notation for distribution

First, we introduce notations related to probabilistic distributions.

Definition 1 (Probability distribution notation) Fix a generic joint distribution p on d variables. We use $X := (X_1, \ldots, X_d)$, where $X \sim p$, to denote a random vector in d dimension. Each X_k is assumed to be a discrete random variable over $\{1, \ldots, n_k\}$. Set $n := \max_{k \in [d]} n_k$.

Definition 2 (Probability distribution for TTNS-Sketch) Suppose the distribution of interest is for the random vector $X := (X_1, \ldots, X_d)$ in d dimension, and moreover, suppose one is given samples of X. The symbol $\hat{p}$ denotes the *empirical distribution* over samples of X. The symbol $p^\star$ denotes the *underlying distribution* of X.

For this paper, we only consider discrete variables. Hence a distribution function such as $p^\star$ can be considered as a d-dimensional tensor.

2.2 Notation for tree structure

Next, we introduce notations for a tree graph. A tree graph $T = (V, E)$ is a connected undirected graph without cycles. Throughout this paper, $V = [d]$. Moreover, T is specified with a root node, and vertices will have a partial topological ordering generated by the child-parent relationship. For an undirected edge $\{w, k\}$ in T, we write it interchangeably as (w, k) or (k, w). If k is the parent of w, we also write $\{w, k\}$ as $w \to k$ with the aim of signaling the child-parent hierarchy.

The following Definition 3 contains the notation for the graph-theoretic concept one needs to define a TTNS. See Fig. 2 for an illustration.

Definition 3 (Tree topology notation) For a rooted tree structure T with nodes $V = [d]$ and any $k \in [d]$, define $\mathcal{C}(k)$, $\mathcal{P}(k)$, $\mathcal{N}(k)$ respectively as the children, parent, and neighbors of k. In particular, one has $|\mathcal{P}(k)| \leq 1$ and $\mathcal{N}(k) = \mathcal{C}(k) \cup \mathcal{P}(k)$. Moreover, define $\mathcal{E}(k)$ as the set of edges incident to k. Define $\mathcal{L}(k)$, $\mathcal{R}(k)$ respectively as the descendant, non-descendant of node k in T. In particular, $\mathcal{L}(k)$ and $\mathcal{R}(k)$ are respectively called the *left* and the *right* of node k.

Importantly, in Definition 4, we introduce several short-hands in order to write joint variables compactly.

Definition 4 (Joint variable notation) For variables indexed by nodes on T, we write $x_{\mathcal{S}}$ to denote the joint variable $(x_{i_1}, \ldots, x_{i_k})$, where $\mathcal{S} = \{i_1, \ldots, i_k\} \subset V$.

Likewise, for variables indexed by edges on T, we write $\alpha_{\mathcal{U}}$ to denote the joint index $(\alpha_{e_{i_1}}, \ldots, \alpha_{e_{i_k}})$, where $\mathcal{U} = \{e_{i_1}, \ldots, e_{i_k}\} \subset E$. In particular, $\mathcal{U} \subset E$ is typically all incident to one node w, and we write $\alpha_{(w,\mathcal{S})}$ to denote the joint variable $(\alpha_{(w,i_1)}, \ldots, \alpha_{(w,i_k)})$, where $\mathcal{S} = \{i_1, \ldots, i_k\} \subset V$. For compactness, we write $x_{\mathcal{S}\cup k} := x_{\mathcal{S}\cup\{k\}}$, where the element k is used in place of the singleton set $\{k\}$.

Frequently used symbols include $x_{\mathcal{L}(k)}$, $x_{\mathcal{R}(k)}$, $x_{\mathcal{C}(k)}$, which respectively denote the joint variable corresponding to the left, the right, and the children of k. Moreover, we use $x_{\mathcal{L}(k)\cup k}$ to denote the joint variable corresponding to nodes that are not on the right side of k. For edge-indexed variables, we use $\alpha_{(k,\mathcal{C}(k))}$ to denote the joint variables corresponding to the edges between k and its children.

2.3 Notation for TTNS ansatz

We introduce condition and notation for a generic tensor with TTNS ansatz. We will prove that having a TTNS ansatz is equivalent to satisfying the TTNS condition, i.e. having a low-rank factorization structure along a tree. See Fig. 3 for an illustration.

Condition 1 (TTNS condition) Let $T = (V, E)$ be a rooted tree graph, and let $\{r_e\}_{e\in E}$ be a collection of positive integers, where $r_{(w,k)}$ denotes internal bond rank at the edge (w, k).

A function $p\colon \prod_{k=1}^{d}[n_k] \to \mathbb{R}$ is said to satisfy the TTNS ansatz condition if for every edge $(w, k) \in E$, there exists a rank $r_{(w,k)}$ decomposition $\Phi_{w\to k}\colon \prod_{i\in\mathcal{L}(w)\cup w}[n_i]\times[r_{(w,k)}] \to \mathbb{R}$ and $\Psi_{w\to k}\colon [r_{(w,k)}] \times \prod_{i\in\mathcal{R}(w)}[n_i] \to \mathbb{R}$ such that

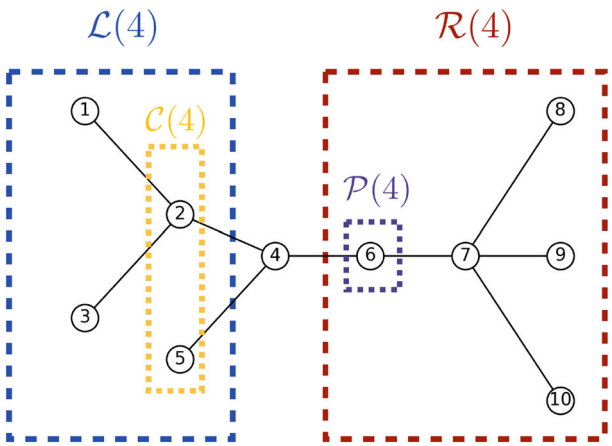


Fig. 2 Illustration of tree topology notation. For the tree in Fig. 1a, if 7 is the root, then $\mathcal{C}(4) = \{2, 5\}$, $\mathcal{P}(4) = 6$, $\mathcal{L}(4) = \{1, 2, 3, 5\}$, and $\mathcal{R}(4) = \{6, 7, 8, 9, 10\}$. In this graph, one also has $\mathcal{N}(4) = \{2, 5, 6\}$ and $\mathcal{E}(4) = \{(4, 2), (4, 5), (4, 6)\}$

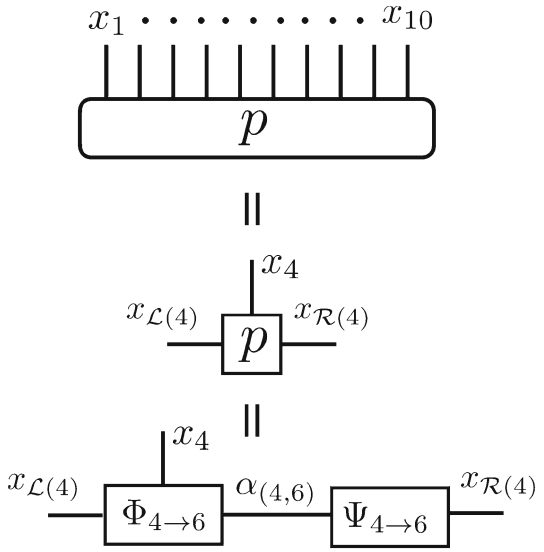


Fig. 3 Tensor diagram representation of the TTNS ansatz assumption in Condition 1. The illustration is over the rooted tree in Fig. 2 with $(w, k) = (4, 6)$. The symbol $x_{\mathcal{L}(4)} = (x_1, x_2, x_3, x_5)$ and $x_{\mathcal{R}(4)} = (x_6, x_7, \ldots, x_{10})$ is a short-hand for joint variables as defined in Definition 4

$$p(x_1, \ldots, x_d) = \sum_{\alpha_{(w,k)}=1}^{r_{(w,k)}} \Phi_{w\to k}(x_{\mathcal{L}(w)\cup w}, \alpha_{(w,k)})\Psi_{w\to k}(\alpha_{(w,k)}, x_{\mathcal{R}(w)}). \tag{2}$$

More explicitly, a TTNS ansatz can be defined in terms of tensor cores. Definition 5 shows the construction in terms of tensor cores. For illustration, see Fig. 1b.

Definition 5 (TTNS tensor core and TTNS ansatz notation) Given a tree structure $T = ([d], E)$ and corresponding ranks $\{r_e : e \in E\}$. The *TTNS tensor core* at k is denoted by G_k. Let d_k stand for the degree of k in T, and then G_k is defined as an $(d_k + 1)$-tensor of the

following shape:

$$G_k : [n_k] \times \prod_{e \in \mathcal{E}(k)} [r_e] \to \mathbb{R}.$$

We say that a function p admits a TTNS ansatz over tensor cores $\{G_i\}_{i=1}^d$ over $k = 1, \ldots, d$ if

$$p(x_1, \ldots, x_d) = \sum_{\substack{\alpha_e \in [r_e] \\ e \in E}} \left(\prod_{k=1}^{d} G_k \left(x_k, \alpha_{(k, \mathcal{N}(k))} \right) \right). \tag{3}$$

For example, when $T = (V, E)$ is a chain with $E = \{(k, k+1)\}_{k=1}^{d-1}$, a TTNS ansatz is a tensor train ansatz. In Fig. 1a, we show a tree structure over 10 vertices, and the corresponding tensor diagram for TTNS is given in Fig. 1b. For instance, $G_4 \colon [n_4] \times [r_{(4,2)}] \times [r_{(4,5)}] \times [r_{(4,6)}] \to \mathbb{R}$ in Fig. 1b, and the tensor network defines a d-dimensional function after the contraction of internal bonds.

Importantly, when working with high-dimensional functions, it is often convenient to group the variables into two subsets and think of the resulting object as a matrix. The notion is referred to as an unfolding matrix and is defined as follows:

Definition 6 (Unfolding matrix notation) For a generic D-dimensional tensor $f \colon [n_1] \times \cdots \times [n_D] \to \mathbb{R}$ and for two disjoint subsets $\mathcal{U}, \mathcal{V}$ with $\mathcal{U} \cup \mathcal{V} = [D]$, we define the corresponding *unfolding matrix* by $f(x_{\mathcal{U}}; x_{\mathcal{V}})$. Namely, group the variables indexed by $\mathcal{U}$ and the ones indexed by $\mathcal{V}$ to form rows and columns, respectively. The matrix $f(x_{\mathcal{U}}; x_{\mathcal{V}})$ is of size $\left(\prod_{i \in \mathcal{U}} n_i\right) \times \left(\prod_{j \in \mathcal{V}} n_j\right)$.

As an example, for a function p satisfying TTNS assumption in Condition 1, define the unfolding matrix of p at the edge $w \to k \in E$ as $p(x_{\mathcal{L}(w) \cup w}; x_{\mathcal{R}(w)})$, which is of size $\left(\prod_{i \in \mathcal{L}(w) \cup w} n_i\right) \times \left(\prod_{j \in \mathcal{R}(w)} n_j\right)$. Viewed in this context, Condition 1 exactly means that the unfolding matrix of p at any edge $(w, k) \in E$ is a matrix of rank $r_{(w,k)}$.

2.4 Equation for TTNS ansatz

We now show that Condition 1 implies the existence of a TTNS ansatz in the sense of Definition 5. With the information for every $\Phi_{w \to k}$ in Condition 1, there exists an equation for obtaining the TTNS tensor cores exactly. We summarize this result in Theorem 7, which shows that one can obtain cores of a TTNS by solving a recursive system of linear equations. See Fig. 6a for an illustration.

Theorem 7 *Suppose* Condition 1 *holds for a rooted tree structure* $T = ([d], E)$ *and bond information* $\{r_e\}_{e \in E}$. *For non-leaf* k, *define*

$$\Phi_{\mathcal{C}(k) \to k} = \bigotimes_{w \in \mathcal{C}(k)} \Phi_{w \to k},$$

and in terms of entries one has $\Phi_{\mathcal{C}(k) \to k} \colon \prod_{w \in \mathcal{L}(k)} [n_w] \times \prod_{w \in \mathcal{C}(k)} [r_{(w,k)}] \to \mathbb{R}$, *and*

$$\Phi_{\mathcal{C}(k) \to k}(x_{\mathcal{L}(k)}, \alpha_{(k, \mathcal{C}(k))}) = \prod_{w \in \mathcal{C}(k)} \Phi_{w \to k}(x_{\mathcal{L}(w)}, \alpha_{(k,w)}). \tag{4}$$

Then $G_k : [n_k] \times \prod_{w\in\mathcal{N}(k)}[r_{(w,k)}] \to \mathbb{R}$ *satisfies the following linear Core Determining Equations (CDE) for* $k = 1, \ldots, d$*:*

$$\begin{aligned} G_k &= \Phi_{k\to\mathcal{P}(k)} \quad \text{if } k \text{ is a leaf,} \\ \sum_{\alpha_{(k,\mathcal{C}(k))}} \Phi_{\mathcal{C}(k)\to k}(x_{\mathcal{L}(k)}, \alpha_{(k,\mathcal{C}(k))}) G_k(x_k, \alpha_{(k,\mathcal{N}(k))}) &= p(x_1, \ldots, x_d) \quad \text{if } k \text{ is the root,} \\ \sum_{\alpha_{(k,\mathcal{C}(k))}} \Phi_{\mathcal{C}(k)\to k}(x_{\mathcal{L}(k)}, \alpha_{(k,\mathcal{C}(k))}) G_k(x_k, \alpha_{(k,\mathcal{N}(k))}) &= \Phi_{k\to\mathcal{P}(k)}(x_{\mathcal{L}(k)\cup k}, \alpha_{(k,\mathcal{P}(k))}) \quad \text{otherwise.} \end{aligned} \tag{5}$$

Then, each equation of (5) *has a unique solution, and* p *has a TTNS ansatz over the cores* $\{G_i\}_{i=1}^d$ *in the sense of Definition* 5*.*

The proof is deferred to the Appendix, but we will give a rough idea of why p admits a TTNS ansatz over $\{G_i\}_{i=1}^d$. Equation (5) for when k is not root essentially shows that each $\Phi_{w\to k}$ can be represented by tensor contractions of cores in $\{G_i\}_{i\in\mathcal{L}(w)\cup w}$, and the proof is based on simple mathematical induction. From this observation, one can work with Eq. (5) for when k is the root, and replace all of the $\Phi_{w\to k}$ terms by $\{G_i\}_{i\neq\text{root}}$, and the obtained equation will be exactly (3) in Definition 5.

In summary, Theorem 7 shows how Condition 1 leads to the existence of a TTNS ansatz, and our previous remark on the construction of $\Phi_{w\to k}$ from $\{G_i\}_{i\in\mathcal{L}(w)\cup w}$ also shows a TTNS ansatz also leads to Condition 1. However, from a computational point of view, the linear system (5) in Theorem 7 is intractable and we shall address this issue using sketching in the next section.

3 Main idea of TTNS-Sketch

In the setting of this section, $p^\star$ admits a TTNS ansatz over T and $\{r_e\}_{e\in E}$ in the sense of Definition 5. We show the derivation of the linear equation which is used to solve for the TTNS tensor cores in TTNS-Sketch. However, obtaining terms in the derived linear system rely on access to $p^\star$, an assumption which we will later relax by sample approximation. To emphasize this point, all of the intermediate terms from this algorithm will be labeled with the upper-index $\star$ if it assumes access to or is derived from $p^\star$.

3.1 Gauge degree of freedom for a TTNS ansatz

By Theorem 7, a function $p^\star$ having a TTNS ansatz is equivalent to the condition that its unfolding matrix along each edge of a tree has a low-rank structure. Moreover, the ansatz is non-unique. This notion is typically called the gauge degree of freedom, which we will introduce here.

Let us view $p^\star$ by the unfolding matrix $p^\star(x_{\mathcal{L}(w)\cup w}; x_{\mathcal{R}(w)})$. For any edge $w \to k$, the TTNS condition assumes that there exists $\Phi^\star_{w\to k} : \prod_{i\in\mathcal{L}(w)\cup w}[n_i] \times [r_{(w,k)}] \to \mathbb{R}$ and $\Psi^\star_{w\to k} : [r_{(w,k)}] \times \prod_{i\in\mathcal{R}(w)}[n_i] \to \mathbb{R}$ such that

$$p^\star(x_{\mathcal{L}(w)\cup w}, x_{\mathcal{R}(w)}) = \sum_{\alpha_{(w,k)}} \Phi^\star_{w\to k}(x_{\mathcal{L}(w)\cup w}, \alpha_{(w,k)}) \Psi^\star_{w\to k}(\alpha_{(w,k)}, x_{\mathcal{R}(w)}).$$

One can view $p^\star$ as the unfolding matrix structure $p^\star(x_{\mathcal{L}(w)\cup w}; x_{\mathcal{R}(w)})$. Likewise, $\Phi^\star_{w\to k}$ as $\Phi^\star_{w\to k}(x_{\mathcal{L}(w)\cup w}; \alpha_{(w,k)})$ and $\Psi^\star_{w\to k}$ as $\Psi^\star_{w\to k}(\alpha_{(w,k)}; x_{\mathcal{R}(w)})$. Then, by using the usual matrix product notation, the TTNS assumption along the edge $w \to k$ is $p^\star = \Phi^\star_{w\to k}\Psi^\star_{w\to k}$.

Then, for any R being a nonsingular $r_{(w,k)} \times r_{(w,k)}$ matrix, one has

$$p^\star = \Phi^\star_{w\to k}\Psi^\star_{w\to k} = (\Phi^\star_{w\to k}R)(R^{-1}\Psi^\star_{w\to k}).$$

Given the information of $\{\Phi^\star_{w\to k}\}_{w\to k\in E}$, solving for the tensor core $G^\star_k$ follows from (5) in Theorem 7. Multiplying any $\Phi^\star_{w\to k}$ by a matrix R will thus result in a different TTNS ansatz for $p^\star$. In summary, a gauge degree of freedom in the low-rank decomposition of $p^\star$ leads to a gauge degree of freedom in the TTNS ansatz of $p^\star$.

The collection $\{\Phi^\star_{w\to k}, \Psi^\star_{w\to k}\}_{w\to k\in E}$ will later be chosen to have an explicit gauge, but currently, it suffices to understand gauge as fixed. The desired TTNS ansatz $\{G^\star_i\}_{i=1}^d$ as solution to (5) is also fixed.

3.2 Sketching down core determining equation

Without loss of generality, we consider the equation for $G^\star_k$ in Theorem 7 where k is neither a root nor a leaf node. We can rewrite the corresponding equation for $G^\star_k$ by substituting $\Phi^\star_{\mathcal{C}(k)\to k}$ according to definition:

$$\begin{aligned}&\sum_{\substack{\alpha_{(w,k)}\\ w\in\mathcal{C}(k)}}\left(\prod_{w\in\mathcal{C}(k)}\Phi^\star_{w\to k}(x_{\mathcal{L}(w)\cup w},\alpha_{(w,k)})\right)G^\star_k(x_k,\alpha_{(k,\mathcal{C}(k))},\alpha_{(k,\mathcal{P}(k))})\\ &=\Phi^\star_{k\to\mathcal{P}(k)}(x_{\mathcal{L}(k)\cup k},\alpha_{(k,\mathcal{P}(k))}),\end{aligned}\tag{6}$$

which is an over-determined linear system on $G^\star_k$, and the number of linear equations for $G^\star_k$ grows exponentially in d. Hence the above equation is not tractable.

The TTNS-Sketch algorithm applies the sketching operation to (6) and projects tensors of the form $\Phi_{w\to k}$ in (6) to a tensor of tractable size, which makes the equation tractable. In TTNS-Sketch, for each edge $w \to k$, we define a series of linear projection operators of the form

$$S_{w\to k}\colon [l_{(w,k)}]\times\prod_{v\in\mathcal{L}(w)\cup w}[n_v]\to\mathbb{R},$$

and they globally form an function which we call the *left-sketch function* S_k of the form

$$S_k\colon \prod_{w\in\mathcal{C}(k)}[l_{(w,k)}]\times\prod_{i\in\mathcal{L}(k)}[n_i]\to\mathbb{R}.$$

The definition of S_k is by the simple formula $S_k = \bigotimes_{w\in\mathcal{C}(k)} S_{w\to k}$, or equivalently

$$S_k(\beta_{(k,\mathcal{C}(k))},x_{\mathcal{L}(k)}) = \prod_{w\in\mathcal{C}(k)} S_{w\to k}(\beta_{(w,k)},x_{\mathcal{L}(w)\cup w}).\tag{7}$$

We remark that the factorization structure of S_k in (7) depends on a simple topological fact on trees, which is that $(\mathcal{L}(w)\cup\{w\})_{w\in\mathcal{C}(k)}$ are pairwise disjoint and their union forms $\mathcal{L}(k)$.

Now, we can apply the usual projection to (6) using S_k, i.e. multiplying both sides by S_k and summing over $x_{\mathcal{L}(k)}$. For the RHS of (6) after projection, we define a tensor $B^\star_k(\beta_{(k,\mathcal{C}(k))},x_k,\alpha_{(k,\mathcal{P}(k))})$ to represent this result, i.e.

$$B^\star_k(\beta_{(k,\mathcal{C}(k))},x_k,\alpha_{(k,\mathcal{P}(k))}) := \sum_{x_{\mathcal{L}(k)}} S_k(\beta_{(k,\mathcal{C}(k))},x_{\mathcal{L}(k)})\Phi^\star_{k\to\mathcal{P}(k)}(x_{\mathcal{L}(k)\cup k},\alpha_{(k,\mathcal{P}(k))}).\tag{8}$$

For the LHS of of (6), we define a tensor $A^\star_k(\beta_{(k,\mathcal{C}(k))}, \alpha_{(k,\mathcal{C}(k))})$ to represent the coefficient term for $G^\star_k$ under projection:

$$A^\star_k(\beta_{(k,\mathcal{C}(k))}, \alpha_{(k,\mathcal{C}(k))}) := \sum_{x_{\mathcal{L}(k)}} S_k(\beta_{(k,\mathcal{C}(k))}, x_{\mathcal{L}(k)}) \prod_{w\in\mathcal{C}(k)} \Phi^\star_{w\to k}(x_{\mathcal{L}(w)\cup w}, \alpha_{(w,k)}) \tag{9}$$

The Eq. (6) then projects to the linear equation:

$$\sum_{\alpha_{(k,\mathcal{C}(k))}} A^\star_k(\beta_{(k,\mathcal{C}(k))}, \alpha_{(k,\mathcal{C}(k))}) G^\star_k(x_k, \alpha_{(k,\mathcal{C}(k))}, \alpha_{(k,\mathcal{P}(k))}) = B^\star_k(\beta_{(k,\mathcal{C}(k))}, x_k, \alpha_{(k,\mathcal{P}(k))}), \tag{10}$$

which is an equation of the simple form of $A^\star_k G^\star_k = B^\star_k$ when viewing each tensor by appropriate unfolding matrix structures. This linear equation is illustrated in Fig. 6b.

In the sketched-down linear system, the number of linear equations for $G^\star_k$ no longer scales with d, and one can check that it is tractable. Moreover, due to the factorization structure of S_k using $S_{w\to k}$, it follows that $A^\star_k$ simplifies to

$$A^\star_k(\beta_{(k,\mathcal{C}(k))}, \alpha_{(k,\mathcal{C}(k))}) = \prod_{w\in\mathcal{C}(k)} \sum_{x_{\mathcal{L}(w)\cup w}} S_{w\to k}(\beta_{(w,k)}, x_{\mathcal{L}(w)\cup w}) \Phi^\star_{w\to k}(x_{\mathcal{L}(w)\cup w}, \alpha_{(w,k)}), \tag{11}$$

which can be readily seen from the diagram in Fig. 6b.

3.3 Derivation of $A^\star_k$ and $B^\star_k$ in TTNS-Sketch

For the time being, $A^\star_k$ and $B^\star_k$ are defined from $\Phi^\star_{w\to k}$, and we now show how one can lift this requirement. We define the *right-sketch function* T_k, which is a linear operator of the form

$$T_k : \prod_{i\in\mathcal{R}(k)} [n_i] \times [m_{(k,\mathcal{P}(k))}] \to \mathbb{R}.$$

Using T_k and S_k, one can jointly form a linear projection of $p^\star$, with the result referred to as $Z^\star_k$, as follows:

$$\begin{aligned} &Z^\star_k(\beta_{(k,\mathcal{C}(k))}, x_k, \gamma_{(k,\mathcal{P}(k))}) \\ &= \sum_{x_{\mathcal{L}(k)\cup\mathcal{R}(k)}} S_k(\beta_{(k,\mathcal{C}(k))}, x_{\mathcal{L}(k)}) p^\star(x_{\mathcal{L}(k)}, x_k, x_{\mathcal{R}(k)}) T_k(x_{\mathcal{R}(k)}, \gamma_{(k,\mathcal{P}(k))}). \end{aligned} \tag{12}$$

One then performs singular value decomposition (SVD) to $Z^\star_k$ according to the unfolding $Z^\star_k(\beta_{(k,\mathcal{C}(k))}, x_k; \gamma_{(k,\mathcal{P}(k))})$. Due to the low rank structure of $p^\star$ at $k \to \mathcal{P}(k)$, the following rank $r_{(k,\mathcal{P}(k))}$ decomposition is exact:

$$Z^\star_k(\beta_{(k,\mathcal{C}(k))}, x_k; \gamma_{(k,\mathcal{P}(k))}) = U^\star_k \Sigma^\star_k \left(V^\star_k\right)^\top. \tag{13}$$

Set $Q^\star_k = V^\star_k \left(\Sigma^\star_k\right)^{-1}$ and $\left(Q^\star_k\right)^\top = \Sigma^\star_k \left(V^\star_k\right)^\top$. Note that $\left(Q^\star_k\right)^\top$ is the pseudo-inverse of $Q^\star_k$. In particular, one has

$$Z^\star_k(\beta_{(k,\mathcal{C}(k))}, x_k; \gamma_{(k,\mathcal{P}(k))}) = U^\star_k(\beta_{(k,\mathcal{C}(k))}, x_k; \alpha_{(k,\mathcal{P}(k))}) \left(Q^\star_k\right)^\top (\alpha_{(k,\mathcal{P}(k))}; \gamma_{(k,\mathcal{P}(k))}).$$

As a summary of $Z^\star_k$, $U^\star_k$, $Q^\star_k$, see illustration in Fig. 4.

One can naturally shape $U^\star_k(\beta_{(k,\mathcal{C}(k))}, x_k; \alpha_{(k,\mathcal{P}(k))})$ as a tensor of the index $U^\star_k(\beta_{(k,\mathcal{C}(k))}, x_k, \alpha_{(k,\mathcal{P}(k))})$, i.e. $U^\star_k$: $\prod_{w\in\mathcal{C}(k)}[l_{(w,k)}] \times [n_k] \times [m_{(k,\mathcal{P}(k))}] \to \mathbb{R}$. We now write out our choice of gauge and its consequences in Condition 2:

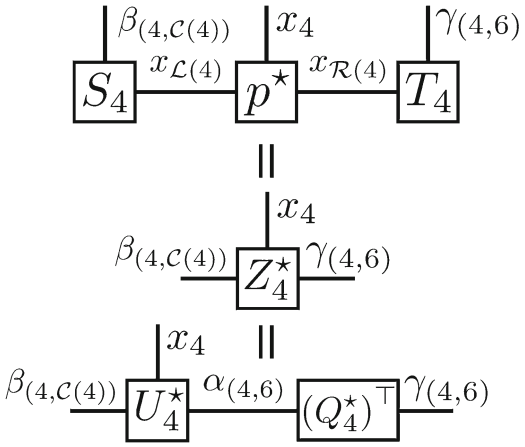


Fig. 4 Tensor diagram of the sketching step in (12) and the SVD step in (13). The illustration is over the rooted tree in Fig. 2

Condition 2 (TTNS-Sketch gauge choice) The gauge for $\Phi^\star_{k\to\mathcal{P}(k)}$ is chosen so that the following holds:

$$U_k^\star(\beta_{(k,\mathcal{C}(k))}, x_k, \alpha_{(k,\mathcal{P}(k))}) = \sum_{x_{\mathcal{L}(k)}} S_k(\beta_{(k,\mathcal{C}(k))}, x_{\mathcal{L}(k)})\Phi^\star_{k\to\mathcal{P}(k)}(x_{\mathcal{L}(k)\cup k}, \alpha_{(k,\mathcal{P}(k))}). \tag{14}$$

As a consequence of (14) and (8), one has:

$$B_k^\star = U_k^\star. \tag{15}$$

As a consequence of (13), the matrix $\left(Q_k^\star\right)^\top$ is a projection of $\Psi^\star_{k\to\mathcal{P}(k)}$ by T_k, i.e.

$$\left(Q_k^\star\right)^\top(\alpha_{(k,\mathcal{P}(k))}, \gamma_{(k,\mathcal{P}(k))}) = \sum_{x_{\mathcal{R}(k)}} \Psi^\star_{k\to\mathcal{P}(k)}(\alpha_{(k,\mathcal{P}(k))}, x_{\mathcal{R}(k)})T_k(x_{\mathcal{R}(k)}, \gamma_{(k,\mathcal{P}(k))}). \tag{16}$$

Likewise, we now show how $A_k^\star$ can be obtained. By the choice of gauge in Condition 2, one forms Corollary 8. In Fig. 5, we include an short proof using tensor diagram. As a consequence of Corollary 8, one can form a linear system for $G_k^\star$ completely in terms of the sketches $\{Z^\star_{w\to k}\}_{w\to k\in E}$, $\{Z_i^\star\}_{i=1}^d$, which can be reasonably approximated by samples. As an illustration, one can rewrite the tensor diagram in Fig. 6b as the new diagram illustrated in Fig. 6c.

Corollary 8 *Define the intermediate terms $Z^\star_{w\to k}$ and $A^\star_{w\to k}$ as follows:*

$$\begin{aligned} Z^\star_{w\to k}(\beta_{(w,k)}, \gamma_{(w,k)}) &= \sum_{x_{[d]}} S_{w\to k}(\beta_{(w,k)}, x_{\mathcal{L}(w)\cup w})p^\star(x_{\mathcal{L}(w)\cup w}, x_{\mathcal{R}(w)})T_w(x_{\mathcal{R}(w)}, \gamma_{(w,k)}) \\ A^\star_{w\to k}(\beta_{(w,k)}, \alpha_{(w,k)}) &= \sum_{\gamma_{(w,k)}} Z^\star_{w\to k}(\beta_{(w,k)}, \gamma_{(w,k)})Q_w^\star(\gamma_{(w,k)}, \alpha_{(w,k)}), \end{aligned} \tag{17}$$

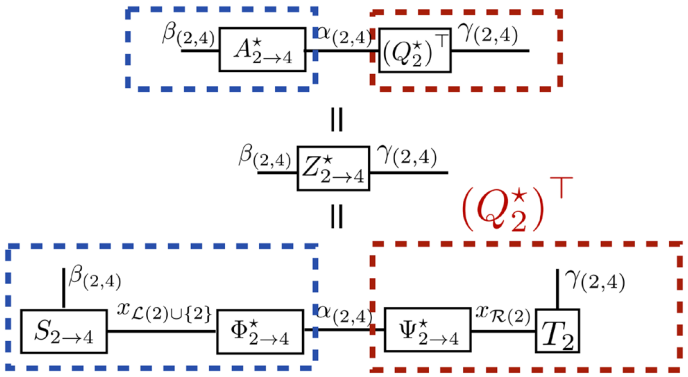


Fig. 5 Proof of Corollary 8 in terms of tensor diagram. Both equalities hold due to (17). Then, the tensors enclosed in the red box coincide due to (16), and so the tensors enclosed in the blue box coincide, which is what we need to show

Then $A^\star_k$ satisfies the following equation

$$A^\star_k(\beta_{(k,\mathcal{C}(k))}, \alpha_{(k,\mathcal{C}(k))}) = \prod_{w\in\mathcal{C}(k)} A^\star_{w\to k}(\beta_{(w,k)}, \alpha_{(w,k)}). \tag{18}$$

Proof By the factorization structure of $A^\star_k$, it suffices to show that

$$A^\star_{w\to k}(\beta_{(w,k)}, \alpha_{(w,k)}) = \sum_{x_{\mathcal{L}(w)\cup w}} S_{w\to k}(\beta_{(w,k)}, x_{\mathcal{L}(w)\cup w})\Phi^\star_{w\to k}(x_{\mathcal{L}(w)\cup w}, \alpha_{(w,k)}).$$

We use an unfolding matrix structure for $Z^\star_{w\to k}, S_{w\to k}, T_w$ in the following rewrite of (17):

$$Z^\star_{w\to k}(\beta_{(w,k)}; \gamma_{(w,k)}) = S_{w\to k}(\beta_{(w,k)}; x_{\mathcal{L}(w)\cup w})p^\star(x_{\mathcal{L}(w)\cup w}; x_{\mathcal{R}(w)})T_w(x_{\mathcal{R}(w)}; \gamma_{(w,k)})$$

Likewise, we use the unfolding matrix structure of $p^\star$, $\Phi^\star_{w\to k}$, $\Psi^\star_{w\to k}$ in

$$p^\star(x_{\mathcal{L}(w)\cup w}; x_{\mathcal{R}(w)}) = \Phi^\star_{w\to k}(x_{\mathcal{L}(w)\cup w}; \alpha_{(w,k)})\Psi^\star_{w\to k}(\alpha_{(w,k)}; x_{\mathcal{R}(w)}).$$

Using the unfolding matrix structure as just suggested, it suffices to prove

$$A^\star_{w\to k} = S_{w\to k}\Phi^\star_{w\to k}.$$

The definition for the intermediate term $Z^\star_{w\to k}$ simplifies to $Z^\star_{w\to k} = S_{w\to k}p^\star T_w$ and more importantly,

$$A^\star_{w\to k} = Z^\star_{w\to k}Q^\star_w = S_{w\to k}p^\star T_w Q^\star_w.$$

Note that one can expand according to $p^\star = \Phi^\star_{w\to k}\Psi^\star_{w\to k}$ and get

$$A^\star_{w\to k} = S_{w\to k}\Phi^\star_{w\to k}\Psi^\star_{w\to k}T_w Q^\star_w = S_{w\to k}\Phi^\star_{w\to k}\left(Q^\star_w\right)^\top Q^\star_w = S_{w\to k}\Phi^\star_{w\to k},$$

where the second equality uses (16) and last equality uses that $\left(Q^\star_w\right)^\top Q^\star_w$ is an identity matrix. □

3.4 Sample estimation of $A_k^\star$ and $B_k^\star$ in TTNS-Sketch

Practically, one only has access to the empirical distribution $\hat{p}$ via samples $\{(y_1^{(i)}, \ldots, y_d^{(i)})\}_{i=1}^N$. A finite sample approximation of $Z_k^\star$ is tractable and can be obtained by function evaluations of T_k and S_k. One has

$$\hat{Z}_k(\beta_{(k,\mathcal{C}(k))}, x_k, \gamma_{(k,\mathcal{P}(k))}) = \sum_{i=1}^N S_k(\beta_{(k,\mathcal{C}(k))}, y_{\mathcal{L}(k)}^{(i)}) \mathbf{1}(y_k^{(i)} = x_k) T_k(y_{\mathcal{R}(k)}^{(i)}, \gamma_{(k,\mathcal{P}(k))}). \tag{19}$$

Similarly, $Z_{w\to k}^\star$ can be approximated by

$$\hat{Z}_{w\to k}(\beta_{(w,k)}, \gamma_{(w,k)}) = \sum_{i=1}^N S_{w\to k}(\beta_{(w,k)}, y_{\mathcal{L}(w)\cup w}^{(i)}) T_w(y_{\mathcal{R}(w)}^{(i)}, \gamma_{(w,k)}). \tag{20}$$

One can then form $\hat{U}_k, \hat{Q}_k$ by replacing $Z_k^\star$ with $\hat{Z}_k$ in (13), noting that in this case the rank $r_{(k,\mathcal{P}(k))}$ SVD decomposition is not exact due to the presence of noise. We now explain why this algorithm is practical in the sample case. It suffices to see the linear equation in Fig. 6c. If one replaces every tensor block by a finite sample approximation in the sense discussed above (e.g. replace $Z_{w\to k}^\star$ by $\hat{Z}_{w\to k}$), then one can indeed form a linear equation, with accuracy increasing with sample size. The full algorithm for empirical distributions will be described in Sect. 4.

4 TTNS-Sketch for empirical distributions

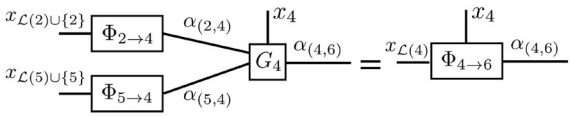
We now give the main Algorithm 1-3 for the TTNS ansatz with empirical distribution as input, which is the main use case. We include it separately from Sect. 3 due to corner cases such as when k is a leaf or root node. To emphasize the sample estimation procedure, all of the intermediate terms will be labeled with $\hat{}$ if it assumes access to $\hat{p}$. Importantly, Algorithm 3 only takes in the sketches $\{Z_{w\to k}\}_{w\to k\in E}, \{Z_i\}_{i=1}^d$ as input, which can be either noiseless or estimated, and so we do not label terms inside this subroutine.

Algorithm 1 TTNS-Sketch for empirical distribution $\hat{p}$.

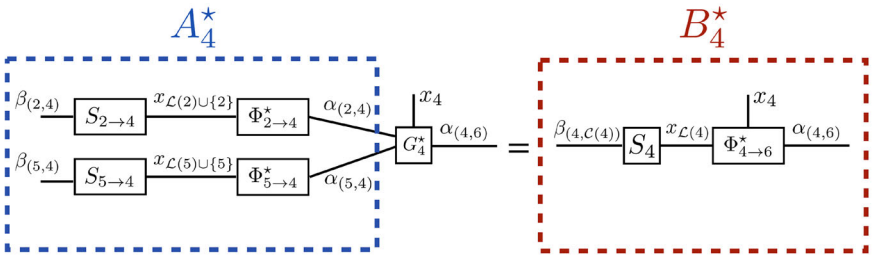
Require: Empirical distribution $\hat{p}$ formed by samples $\{(y_1^{(i)}, \ldots, y_d^{(i)})\}_{i=1}^N$.
Require: A rooted tree structure $T = ([d], E)$, and $\mathcal{C}, \mathcal{P}, \mathcal{L}, \mathcal{R}$ as in Definition 3.
Require: Target ranks $\{r_e : e \in E\} \subset \mathbb{N}$.
Require: $T_k \colon \prod_{i\in\mathcal{R}(k)}[n_i] \times [m_{(k,\mathcal{P}(k))}] \to \mathbb{R}$ for all non-root $k \in [d]$ and $T_k = 1$ if k is the root.
Require: $S_1, \ldots, S_d$ formed by $S_{w\to k}$'s as in (7).
1: $\{\hat{Z}_{w\to k}\}_{w\to k\in E}, \{\hat{Z}_i\}_{i=1}^d \leftarrow$ *Sketching*$(\hat{p}, T_1, \ldots, T_d, S_1, \ldots, S_d)$.
2: $\{\hat{A}_i, \hat{B}_i\}_{i=1}^d \leftarrow$ *SystemForming*$(\{\hat{Z}_{w\to k}\}_{w\to k\in E}, \{\hat{Z}_i\}_{i=1}^d)$.
3: Solve the following d equations via least-squares for the variables $\hat{G}_1, \ldots, \hat{G}_d$:

$$\begin{aligned} \hat{G}_k &= \hat{B}_k \quad \text{if } k \text{ is a leaf,} \\ \textstyle\sum_{\alpha_{(k,\mathcal{C}(k))}} \hat{A}_k(\beta_{(k,\mathcal{C}(k))}, \alpha_{(k,\mathcal{C}(k))}) \hat{G}_k(x_k, \alpha_{(k,\mathcal{N}(k))}) &= \hat{B}_k \quad \text{otherwise,} \end{aligned} \tag{21}$$

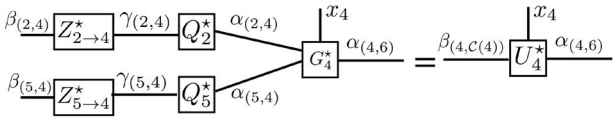
where $\hat{G}_k \colon [n_k] \times \prod_{w\in\mathcal{N}(k)}[r_{(w,k)}] \to \mathbb{R}$.
4: **return** $\hat{G}_1, \ldots, \hat{G}_d$



(A) Tensor diagram representation of the Core Determining Equation (5) for $k = 4$ in Theorem 7. The left side uses $\Phi_{2\to4}$ and $\Phi_{5\to4}$ instead of $\Phi_{\mathcal{C}(4)\to4}$, which is allowed due to (4).



(B) Tensor diagram representation of the sketched-down Core Determining Equation (10) for $k = 4$. The diagram can be thought of as a linear equation for $G_4^\star$. Note that the tensors enclosed in the blue box can be thought of as the coefficient, and the tensors enclosed in the red box can be thought of as the right-hand side. One can see that the tensor diagram equation is obtained from that of Figure 6a by a contraction according to S_4 to both sides. The left side of the diagram uses blocks corresponding to $S_{2\to4}, S_{5\to4}$, which is a consequence of (7).



(C) With $\{Z^\star_{w\to k}\}_{w\to k\in E}, \{Z_i^\star\}_{i=1}^d$ from the sketch step and $\{U_i^\star, Q_i^\star\}_{i=1}^d$ from the SVD step, one can rewrite the tensor diagram in Figure 6b into a tractable linear equation for $G_k^\star$. Note that the diagram is equivalent to that in Figure 6b due to Corollary 8.

Fig. 6 Tensor diagram representation of equations for TTNS tensor cores in TTNS-Sketch. The illustration is over the rooted tree in Fig. 2. The equation in Fig. 6a implies the rest. The diagram in Fig. 6c is derived from the diagram in Fig. 6b with a specific choice of gauge according to Condition 2. Importantly, the equation in Fig. 6c allows for finite sample approximation

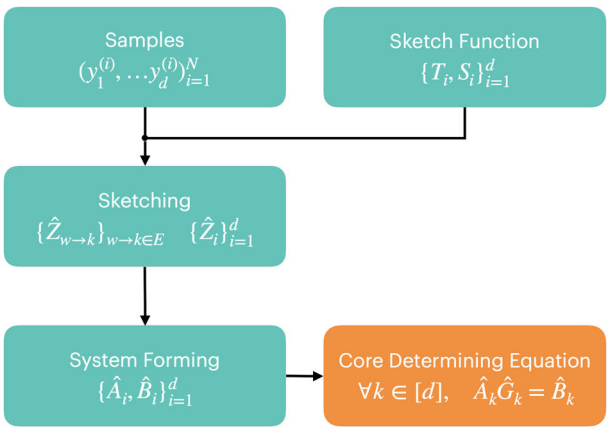


Fig. 7 Schematic of Algorithm 1

Algorithm 2 Sketching.

Require: $\hat{p}$, $T_1, \ldots, T_d$, and $S_1, \ldots, S_d$ as given in Algorithm 1.
Require: $T = ([d], E)$ as given in Algorithm 1.
for $k = 1$ to d **do**
if k is root **then**
Define $\hat{Z}_k \colon \prod_{w\in\mathcal{C}(k)}[l_{(w,k)}] \times [n_k] \to \mathbb{R}$ as

$$\hat{Z}_k(\beta_{(k,\mathcal{C}(k))}, x_k) = \sum_{i=1}^{N} S_k(\beta_{(k,\mathcal{C}(k))}, y^{(i)}_{\mathcal{L}(k)})\mathbf{1}(y^{(i)}_k = x_k).$$

else if k is leaf **then**
Define $\hat{Z}_k \colon [n_k] \times [m_{(k,\mathcal{P}(k))}] \to \mathbb{R}$ as

$$\hat{Z}_k(x_k, \gamma_{(k,\mathcal{P}(k))}) = \sum_{i=1}^{N} \mathbf{1}(y^{(i)}_k = x_k)T(y^{(i)}_{\mathcal{R}(k)}, \gamma_{(k,\mathcal{P}(k))}).$$

Define $\hat{Z}_{k\to\mathcal{P}(k)}$ according to (20)
else
Define $\hat{Z}_k \colon \prod_{w\in\mathcal{C}(k)}[l_{(w,k)}] \times [n_k] \times [m_{(k,\mathcal{P}(k))}] \to \mathbb{R}$ according to (19).
Define $\hat{Z}_{k\to\mathcal{P}(k)} \colon [l_{(k,\mathcal{P}(k))}] \times [m_{(k,\mathcal{P}(k))}] \to \mathbb{R}$ according to (20).
end if
end for
return $\{\hat{Z}_{w\to k}\}_{w\to k\in E}, \{\hat{Z}_i\}_{i=1}^d$.

4.1 Condition for consistency of TTNS-Sketch

We introduce Condition 3 on the choice of sketch functions. Essentially, Theorem 7 shows that (5) is an over-determined linear system with a unique exact solution, and one needs the "sketched-down"version of (5) to still have a unique solution:

Condition 3 Let $p^\star$ be a function that satisfies Condition 1. Moreover, let $\{\Phi^\Delta_{(w,k)}, \Psi^\Delta_{(w,k)}\}_{(w,k)\in E}$ be an arbitrary collection of tensors forming the low-rank decomposition of $p^\star$ in the sense of Condition 1, with gauge chosen arbitrarily. Let $\{T_i, S_i\}_{i=1}^d$ be the sketch functions in Algorithm 1. Define two intermediate terms $A^\Delta_k \colon \prod_{w\in\mathcal{C}(k)}[l_{w\to k}] \times \prod_{w\in\mathcal{C}(k)}[r_{w\to k}] \to \mathbb{R}$ and $B^\Delta_k \colon \prod_{w\in\mathcal{C}(k)}[l_{w\to k}] \times [n_k] \times [\gamma_{k\to\mathcal{P}(k)}] \to \mathbb{R}$ by

$$\begin{aligned} A^\Delta_k(\beta_{(k,\mathcal{C}(k))}, \alpha_{(k,\mathcal{C}(k))}) &:= \prod_{w\in\mathcal{C}(k)} \sum_{x_{\mathcal{L}(w)\cup w}} S_{w\to k}(\beta_{(w,k)}, x_{\mathcal{L}(w)\cup w})\Phi^\Delta_{w\to k}(x_{\mathcal{L}(w)\cup w}, \alpha_{(w,k)}), \\ B^\Delta_k(\beta_{(k,\mathcal{C}(k))}, x_k, \alpha_{(k,\mathcal{P}(k))}) &:= \sum_{x_{\mathcal{L}(k)}} S_k(\beta_{(k,\mathcal{C}(k))}, x_{\mathcal{L}(k)})\Phi^\Delta_{k\to\mathcal{P}(k)}(x_{\mathcal{L}(k)\cup k}, \alpha_{(k,\mathcal{P}(k))}). \end{aligned} \tag{22}$$

Moreover, define an intermediate term $\tilde{\Phi}^\star_k \colon \prod_{i\in\mathcal{L}(k)\cup k}[n_i] \times [m_{(k,\mathcal{P}(k))}] \to \mathbb{R}$ by

$$\tilde{\Phi}^\star_k(x_{\mathcal{L}(k)\cup k}, \gamma_{(k,\mathcal{P}(k))}) = \sum_{x_{\mathcal{R}(k)}} p^\star(x_1, \ldots, x_d)T_k(x_{\mathcal{R}(k)}, \gamma_{(k,\mathcal{P}(k))}).$$

Then, $\{T_i, S_i\}_{i=1}^d$ in Algorithm 1 is chosen to be such that the following conditions hold:

Algorithm 3 SystemForming.

Require: Sketches $\{Z_{w\to k}\}_{w\to k\in E}$, $\{Z_i\}_{i=1}^d$.
Require: Tree structure $T = ([d], E)$ as given in Algorithm 1.
Require: Target rank r as given in Algorithm 1.
for $k = 1$ to d **do**
$\beta_k \leftarrow \beta_{(k,\mathcal{C}(k))}$.
$\gamma_k \leftarrow \gamma_{\mathcal{P}(k)}$.
$r_k \leftarrow r_{(\mathcal{P}(k),k)}$.
$l_k \leftarrow \prod_{w\in\mathcal{C}(k)} l_{(w,k)}$.
$m_k \leftarrow m_{(k,\mathcal{P}(k))}$.
if k is a leaf **then**
Let $U_k\Sigma_k V_k^\top$, where $U_k \in \mathbb{R}^{n_k\times r_k}$, $V_k \in \mathbb{R}^{m_k\times r_k}$, $\Sigma_k \in \mathbb{R}^{r_k\times r_k}$, be the best rank-$r_k$ approximation to the matrix $Z_k(x_k;\gamma_k)$ via SVD. Define $B_k\colon [n_k]\times[r_k]\to\mathbb{R}$ where $B_k(x_k,\alpha_k) = U_k(x_k;\alpha_k)$. Set $A_k = 1$.
else if k is root **then**
Let $B_k = Z_k$. Set $Q_k = 1$.
else
Let $U_k\Sigma_k V_k^\top$, where $U_k \in \mathbb{R}^{l_k n_k\times r_k}$, $V_k \in \mathbb{R}^{m_k\times r_k}$, $\Sigma_k \in \mathbb{R}^{r_k\times r_k}$, be the best rank-$r_k$ approximation to the matrix $Z_k(\beta_k,x_k;\gamma_k)$ via SVD. Define $B_k\colon \prod_{w\in\mathcal{C}(k)}[l_{(w,k)}]\times[n_k]\times[r_k]\to\mathbb{R}$ where $B_k(\beta_k,x_k,\alpha_k) = U_k(\beta_k,x_k;\alpha_k)$.
end if
if k is non-root **then**
Let $Q_k = V_k\Sigma_k^{-1}$.
end if
end for
for $k = 1$ to d **do**
if k is non-leaf **then**
Compute $A_k\colon \prod_{w\in\mathcal{C}(k)}[l_{(w,k)}]\times\prod_{w\in\mathcal{C}(k)}[r_{(w,k)}]\to\mathbb{R}$:

$$A_k(\beta_{(k,\mathcal{C}(k))},\alpha_{(k,\mathcal{C}(k))}) = \prod_{w\in\mathcal{C}(k)}\sum_{\gamma_{(w,k)}} Z_{w\to k}(\beta_{(w,k)},\gamma_{(w,k)})Q_w(\gamma_{(w,k)},\alpha_{(w,k)}).$$

end if
end for
return $\{A_i,B_i\}_{i=1}^d$.

(i) $\tilde{\Phi}_k^\star(x_{\mathcal{L}(k)\cup k};\gamma_{(k,\mathcal{P}(k))})$ and $\Phi^\Delta_{k\to\mathcal{P}(k)}(x_{\mathcal{L}(k)\cup k};\alpha_{(k,\mathcal{P}(k))})$ have the same $r_{(k,\mathcal{P}(k))}$-dimensional column space for every non-root k.
(ii) $Z_k^\star(\beta_{(k,\mathcal{C}(k))},x_k;\gamma_{(k,\mathcal{P}(k))})$ defined in (12) is of rank $r_{(k,\mathcal{P}(k))}$ for every non-leaf and non-root k.
(iii) $A_k^\Delta(\beta_{(k,\mathcal{C}(k))};\alpha_{(k,\mathcal{C}(k))})$ has full column rank for every non-leaf k.

If one has oracle access access to $p^\star$, and moreover suppose that computing $Z^\star_{w\to k}$ and $Z_k^\star$ are tractable, then one can directly start with the SystemForming step in Algorithm 1 with the noiseless terms $Z^\star_{w\to k}$ and $Z_k^\star$ as input. Similar to Theorem 7, under a technical condition for the given sketch functions, one can solve for the tensor cores exactly. We give out the condition in Theorem 9.

Theorem 9 (Exact recovery of TTNS-Sketch) *For an underlying distribution $p^\star$ satisfying Condition 1, suppose its tree structure T and internal bond r coincides with the input for*

Algorithm 1. Let $\{T_i, S_i\}_{i=1}^d$ *satisfy Condition 3. Then, in Algorithm 1, assume one has oracle access to* $\{Z^\star_{w\to k}\}_{w\to k\in E}, \{Z^\star_i\}_{i=1}^d$. *Then, let*

$$\{A^\star_i, B^\star_i\}_{i=1}^d \leftarrow \mathit{SystemForming}(\{Z^\star_{w\to k}\}_{w\to k\in E}, \{Z^\star_i\}_{i=1}^d),$$

and let the $\{G^\star_i\}_{i=1}^d$ *be the least-squares solution the following linear system for the variables* $G_1, \ldots, G_d$:

$$\begin{aligned} G_k &= B^\star_k \quad \text{if } k \text{ is a leaf,} \\ \textstyle\sum_{\alpha_{(k,\mathcal{C}(k))}} A^\star_k(\beta_{(k,\mathcal{C}(k))}, \alpha_{(k,\mathcal{C}(k))}) G_k(x_k, \alpha_{(k,\mathcal{N}(k))}) &= B^\star_k \quad \text{otherwise,} \end{aligned} \tag{23}$$

Then linear system (23) has a unique exact solution $\{G^\star_i\}_{i=1}^d$, *which forms the cores of the TTNS ansatz of* $p^\star$ *given* $T = ([d], E)$ *and* $\{r_e : e \in E\}$. *As a consequence, the output of Algorithm 1 given an empirical distribution* $\hat{p}$ *as input satisfies* $\lim_{N\to\infty} \hat{G}_k = G^\star_k$, *i.e.* $\hat{G}_k$ *is a consistent estimator of* $G^\star_k$.

4.2 Sample complexity of TTNS-Sketch

In addition to the consistency result in Theorem 9, one can bound the rate of convergence of TTNS-Sketch in terms of the l_∞ deviation between the output ansatz and the ground truth. The main strategy underlying the proof is that one can bound the convergence of TTNS-Sketch to ground truth by bounding errors in the sample estimation of $\{Z^\star_{w\to k}\}_{w\to k\in E}, \{Z^\star_i\}_{i=1}^d$. Within the general class of recursive sketch functions (see Sect. 5.2 for a definition), we have derived a sample complexity bound for TTNS-Sketch. The proof is deferred to Appendix E. The informal version is as follows:

Theorem 10 *(Informal statement of Theorem 34)*

Let $p^\star : [n_1] \times \cdots \times [n_d] \to \mathbb{R}$ *satisfy the TTNS assumption in Condition 1. Let sketch function* $\{T_i, S_i\}_{i=1}^d$ *which satisfies Condition 3 and the recursive sketching in Condition 4 in Sect. 5.2. Let* $\{\hat{G}_i\}_{i=1}^d$ *be the output of Algorithm 1, and let* $\hat{p}_{\mathrm{TS}}$ *denote the TTNS ansatz formed by* $\{\hat{G}_i\}_{i=1}^d$. *There exists problem-dependent constant* ζ, L *such that, for* $\eta \in (0, 1)$ *and* $\epsilon \in (0, 1)$, *if*

$$N \geq \frac{18L^2d^2 + 4L\epsilon\zeta d}{\zeta^2\epsilon^2} \log\left(\frac{(ln+m)d}{\eta}\right),$$

and the constants are defined as follows:

- $l = \max_{k\in[d]} l_k$, *where* $l_k = \prod_{w\in\mathcal{C}(k)} l_{(w,k)}$
- $m = \max_{k\in[d]} m_k$, *where* $m_k = m_{(k,\mathcal{P}(k))}$.
- $n = \max_{k\in[d]} n_k$.

Then with probability at least $1 - \eta$ *one has*

$$\frac{\|\hat{p}_{\mathrm{TS}} - p^\star\|_\infty}{|||G^\star_1||| \cdots |||G^\star_d|||} \leq \epsilon,$$

where the notation $|||G^\star_i|||$ *is to be defined in Sect. 8. Moreover, the sample complexity upper bound for* N *does not explicitly depend on* $\{n_k\}_{k\in[d]}, \{m_e, n_e\}_{e\in E}$ *except in the log factor.*

In the lower bound in Theorem 10, the $N \propto \epsilon^{-2}$ rate is the Monte Carlo rate, and the sample dependency on the dimension d is only quadratic. We remark that our proof strategy readily extends to general sketch functions, where the same scaling of N in ϵ and d will hold.

4.3 Estimation of target rank *r*

If one has access to a tree T but not a target rank r, then one can define a noise threshold δ, and determine $r_{(k,\mathcal{P}(k))}$ from the SVD result in Algorithm 3 directly. Namely, one checks the singular values in Σ_k and sets $r_{(k,\mathcal{P}(k))}$ to be the number of singular values above the level set by δ.

Hence, for samples from a distribution $p^\star$ with no known structure, one can set T by the method described in Sect. 6, and set the internal bond rank by thresholding with δ in the sense described above. The result should be reasonable if $p^\star$ satisfies Condition 1 with tractable bond dimension, or if $p^\star$ can be well approximated by a TTNS ansatz. Analysis with approximated target rank is beyond the scope of this paper.

5 Choice of sketch function

We begin with Sect. 5.1, which explains that the differences in sketch functions conceptually lead to matching different statistical moments. Section 5.2 defines the concept of recursive sketch function, which is a class of sketch functions for which the sketching operation is efficient. The rest of the subsections give concrete examples of sketch functions.

5.1 Connection of sketching to moment matching

In this subsection, we give the sketched-down linear equation another interpretation as enforcing a match between the statistical moment of the output TTNS ansatz to that of the empirical distribution. Essentially, different sketch functions lead to different statistical moments to match, which conceptually leads to different optimization objectives.

First, any algorithm for solving for tensor components of a TTNS boils down to attempting to fit the following equation over $\{G_i\}_{i=1}^d$ to the sample, with the end goal of approximating $p^\star$. In terms of an equation, one can write

$$\sum_{\substack{\alpha_e \\ e\in E}} \prod_{i=1}^{d} G_i(x_i, \alpha_{(i,\mathcal{N}(i))}) \approx \hat{p}(x_1, \ldots, x_d) \approx p^\star(x_1, \ldots, x_d), \quad \forall (x_1, \ldots, x_d). \tag{24}$$

In practical settings, $\hat{p} \approx p^\star$ only weakly. Moreover, the above equation relates to the equivalence of two tensors of n^d entries, but the number of unknown parameters involved is only $O(d)$. To apply sketching, one defines a sketch function $f(\mu, x_1, \ldots, x_d)$. One then multiplies (24) by f and then sums over the joint $x_{[d]}$ variable. The sketched-down equation becomes

$$\begin{aligned} &\sum_{x_{[d]}} f(\mu, x_1, \ldots, x_d) \left(\sum_{\substack{\alpha_e \\ e\in E}} \prod_{i=1}^{d} G_i(x_i, \alpha_{(i,\mathcal{N}(i))}) \right) \\ &= \sum_{i=1}^{N} f\left(\mu, y_1^{(i)}, \ldots, y_d^{(i)}\right) \approx \mathbb{E}_{X\sim p^\star}\left[f\left(\mu, X\right)\right], \quad \forall \mu. \end{aligned} \tag{25}$$

where the first approximation sign in (24) is swapped with an equality sign, which is meant to signal that one then solves for the tensor cores using this equation. As for the approximate sign in (25), the design of f will be typically such that the variance $f(\mu, X)$ is $O(1)$ or grows slowly with d, which is why the approximation will be reasonable according to the law of large numbers.

Moreover, one can let θ stand for the TTNS tensor cores $\{G_i\}_{i=1}^d$, and let p_θ stand for the probability distribution obtained from the TTNS ansatz under such cores. The Eq. (25) is equivalent to

$$\mathbb{E}_{X\sim p_\theta}\left[f(\mu, X)\right] = \mathbb{E}_{X\sim \hat{p}}\left[f(\mu, X)\right] \approx \mathbb{E}_{X\sim p^\star}\left[f(\mu, X)\right], \quad \forall \mu. \tag{26}$$

In summary, the solution is such that p_θ is close to $\hat{p}$ in terms of statistical moments $\mathbb{E}_X\left[f(\mu, X)\right]$. For the connection to TTNS-Sketch, we consider a simple case where one has already solved for $\{G_i\}_{i\neq k}$. Consider a sketch function f_k by first defining its corresponding joint variable μ by $\mu = (\beta_{(k,\mathcal{C}(k))}, \gamma_{(k,\mathcal{P}(k))}, \iota)$. Then, let

$$f_k(\beta_{(k,\mathcal{C}(k))}, \gamma_{(k,\mathcal{P}(k))}, \iota, x_{[d]}) = S_k(\beta_{(k,\mathcal{C}(k))}, x_{\mathcal{L}(k)})\mathbf{1}\left[\iota = x_k\right] T_k(x_{\mathcal{R}(k)}, \gamma_{(k,\mathcal{P}(k))}).$$

As a result, one has

$$\mathbb{E}_{X\sim\hat{p}}\left[f_k(\mu, X)\right] = \hat{Z}_k(\beta_{(k,\mathcal{C}(k))}, \iota, \gamma_{(k,\mathcal{P}(k))}),$$

and so the sketched-down Eq. (26) tries to enforce a match between $\mathbb{E}_{X\sim p_\theta}\left[f_k(\mu, X)\right]$ and $Z_k^\star$. Different sketch functions thus lead to different sketches $Z_k^\star$ to match.

5.2 Recursive sketch functions

Storing a generic sketch function is not possible as its number of possible inputs is exponential in d. As only evaluations of the sketch functions are needed in the TTNS-Sketch algorithm, one can use sketch functions $\{S_i, T_i\}_{i=1}^d$ which have an explicit formula. Alternatively, one can use sketch functions which are themselves derived by a TTNS ansatz. We introduce a special case called *recursive sketch functions*, which allows for sketch functions more general than those with an analytic formula, but is still tractable for computation.

In the recursive sketching regime, each sketch function T_k and S_k has a TTNS structure and is recursively defined by the collection of sketch cores $\{(t_i, s_i)\}_{i=1}^d$. We first define a natural notion of subgraph TTNS function in Definition 11. We then fully specify what is a recursive sketching function in Condition 4.

Definition 11 (Subgraph TTNS function) Suppose that f is a function satisfying Condition 1 with tree T, and moreover suppose f admits tensor cores $\{s_i\}_{i=1}^d$ for its TTNS ansatz. Let $\mathcal{S} \subset V$ and let $T_\mathcal{S} = (\mathcal{S}, E_\mathcal{S})$ be the subgraph of T with vertex set $\mathcal{S}$. Then $\{s_i\}_{i=1}^d$ and $T_\mathcal{S}$ jointly defines the *subgraph TTNS function* $f_\mathcal{S}\colon \prod_{i\in\mathcal{S}}[n_i]\times\prod_{e\in\partial\mathcal{S}}[r_e]\to\mathbb{R}$ by

$$f_\mathcal{S}(x_\mathcal{S}, \alpha_{\partial\mathcal{S}}) = \sum_{\substack{\alpha_e \\ e\in E_\mathcal{S}}} \prod_{k\in\mathcal{S}} G_k\left(x_k, \alpha_{(k,\mathcal{N}(k))}\right).$$

where $\partial\mathcal{S} := \{(v, w) \in E \mid v \in \mathcal{S}, w \notin \mathcal{S}\}$.

Condition 4 (Recursive sketching condition) Let $T = (V, E)$ be a tree. Assume f (resp. g) are two functions with a TTNS ansatz over T in the sense of Condition 1, with internal

bond l (resp. m) and sketch cores $\{s_i\}_{i=1}^d$ (resp. $\{t_i\}_{i=1}^d$). Then S_k, T_k are subgraph TTNS function in the sense of Definition 11. Moreover, $S_k = f_{\mathcal{L}(k)}$ and $T_k = g_{\mathcal{R}(k)}$.

In particular, S_k satisfies a recursive relation

$$S_k(\beta_{(k,\mathcal{C}(k))}, x_{\mathcal{L}(k)}) = \prod_{w\in\mathcal{L}(k)} \sum_{\beta_{(w,\mathcal{C}(w))}} s_w(x_w, \beta_{(w,k)}, \beta_{(w,\mathcal{C}(w))}) S_w(\beta_{(w,\mathcal{C}(w))}, x_{\mathcal{L}(w)}), \tag{27}$$

and so S_k satisfies the factorization structure (7) with its $S_{w\to k}$ defined by

$$S_{w\to k}(\beta_{(w,k)}, x_{\mathcal{L}(w)}) = \sum_{\beta_{(w,\mathcal{C}(w))}} s_w(x_w, \beta_{(w,k)}, \beta_{(w,\mathcal{C}(w))}) S_w(\beta_{(w,\mathcal{C}(w))}, x_{\mathcal{L}(w)}).$$

Moreover, the recursive definition over the left sketch functions leads to a simplified equation for $\hat{A}_k$, which we show in the following proposition

Proposition 12 *Let $\hat{p}$ be an arbitrary distribution. Fix a sketch function $\{T_i, S_i\}_{i=1}^d$ which satisfies the recursive sketching assumption in* Condition 4. *Let $\{\hat{A}_i, \hat{B}_i, \hat{G}_i, \hat{Z}_i\}_{i=1}^d$ be as in* Algorithm 1 *with $\hat{p}$ as input. Then,*

$$\hat{A}_k(\beta_{(k,\mathcal{C}(k))}, \alpha_{(k,\mathcal{C}(k))}) = \prod_{w\in\mathcal{C}(k)} \sum_{(\beta_{(w,\mathcal{C}(w))}, x_w)} s_w(\beta_{(w,k)}; \beta_{(w,\mathcal{C}(w))}, x_w) \hat{B}_w(\beta_{(w,\mathcal{C}(w))}, x_w; \alpha_{(w,k)}). \tag{28}$$

Proof As a consequence of (27), the terms $\hat{Z}_{w\to k}$ and $\hat{Z}_w$ are connected under Condition 4:

$$\hat{Z}_{w\to k}(\beta_{(w,k)}, \gamma_{(w,k)}) = \sum_{x_w} \sum_{\gamma_{(w,k)}} s_w(\beta_{(w,k)}; \beta_{(w,\mathcal{C}(w))}, x_w) \hat{Z}_w(\beta_{(w,\mathcal{C}(w))}, x_w; \gamma_{(w,k)}). \tag{29}$$

Define a term $\hat{A}_{w\to k}$ similar to (17). As a consequence of (29),

$$\begin{aligned}\hat{A}_{w\to k}(\beta_{(w,k)}, \alpha_{(w,k)}) &= \sum_{\beta_{(w,\mathcal{C}(w))}} \sum_{x_w} \sum_{\gamma_{(w,k)}} s_w(\beta_{(w,k)}; \beta_{(w,\mathcal{C}(w))}, x_w) \hat{Z}_w(\beta_{(w,\mathcal{C}(w))}, x_w; \gamma_{(w,k)}) \hat{Q}_w(\gamma_{(w,k)}, \alpha_{(w,k)}) \\ &= \sum_{x_w} \sum_{\beta_{(w,\mathcal{C}(w))}} s_w(\beta_{(w,k)}; \beta_{(w,\mathcal{C}(w))}, x_w) \hat{B}_w(\beta_{(w,\mathcal{C}(w))}, x_w; \alpha_{(w,k)}),\end{aligned}$$

where the second equality is a consequence of Algorithm 3.

As a consequence, we conclude that for recursive sketching one can compute the left-hand side by

$$\hat{A}_k(\beta_{(k,\mathcal{C}(k))}, \alpha_{(k,\mathcal{C}(k))}) = \prod_{w\in\mathcal{C}(k)} \sum_{(\beta_{(w,\mathcal{C}(w))}, x_w)} s_w(\beta_{(w,k)}; \beta_{(w,\mathcal{C}(w))}, x_w) \hat{B}_w(\beta_{(w,\mathcal{C}(w))}, x_w; \alpha_{(w,k)}).$$

□

Note that the above result is solely due to the property of the left sketch function. Hence, (28) holds true if one replaces $\hat{A}_k, \hat{B}_k$ with $A_k^\star, B_k^\star$. Proposition 12 will simplify the subsequent error analysis for the recursive sketch function, as one only needs to account for the error in $\hat{Z}_k$.

5.3 Markov sketch function

A Markov sketch function allows the TTNS procedure to essentially solve for marginal distribution information for each node k around its neighbors, which we will show with the definition of its sketch function.

For the Markov sketch function, one has $l_k := n_{\mathcal{P}(k)}$, and the right-sketch function T_k is defined by

$$T_k(x_{\mathcal{R}(k)}, \gamma_{(k,\mathcal{P}(k))}) = \mathbf{1}\left[x_{\mathcal{P}(k)} = \gamma_{(k,\mathcal{P}(k))}\right]. \tag{30}$$

As for the left-sketch function, the form is similar, but it is complicated by the fact that a tree node can have multiple child nodes. Likewise, $m_{(w,k)} := n_w$, and

$$S_k(\beta_{(k,\mathcal{C}(k))}, x_{\mathcal{L}(k)}) = \prod_{w\in\mathcal{C}(k)} \mathbf{1}\left[x_w = \beta_{(w,k)}\right]. \tag{31}$$

By applying left-sketch and right sketch, one has

$$Z_k^\star(\beta_{(k,\mathcal{C}(k))}, x_k, \gamma_{(k,\mathcal{P}(k))}) = \mathbb{P}_{X\sim p^\star}\left[X_w = \beta_{(w,k)} \forall w \in \mathcal{C}(k), X_k = x_k, X_{\mathcal{P}(k)} = \gamma_k\right], \tag{32}$$

which is then the marginal distribution on the subset of nodes $\mathcal{N}(k)\cup\{k\} = \{v \in V \mid \mathrm{dist}(v,k) \le 1\}$.

We also use

$$(\mathcal{M}_{\mathcal{S}}p)(x_{\mathcal{S}}) := \mathbb{P}_{X\sim p}\left[X_v = x_v, \forall v \in \mathcal{S}\right] \tag{33}$$

to denote the marginalization of p to the variables given by the index set $\mathcal{S}$, which is a $|\mathcal{S}|$-dimensional function.

Due to the construction, the joint variables $(\beta_{(k,\mathcal{C}(k))}, \gamma_{(k,\mathcal{P}(k))})$ each has a natural correspondence to a node neighboring k. By identifying $\beta_{(w,k)} = x_w$ and $\gamma_{(k,\mathcal{P}(k))} = x_{\mathcal{P}(k)}$, one has the following entry-wise equality

$$Z_k^\star(\beta_{(k,\mathcal{C}(k))}, x_k, \gamma_{(k,\mathcal{P}(k))}) = \mathcal{M}_{\mathcal{S}_k}p^\star(x_{\mathcal{C}(k)}, x_k, x_{\mathcal{P}(k)}),$$

where $\mathcal{S}_k = \mathcal{N}(k)\cup\{k\}$.

By the description in Sect. 5.1, with Markov sketch function, TTNS-Sketch essentially tries to fit $\mathcal{M}_{\mathcal{N}(k)\cup k}\hat{p}$ over all $k \in V$. One can show that the Markov sketch function allows exact recovery for tree-based graphical models. We summarize the result in Lemma 13.

Lemma 13 *Assume that $p^\star$ is a graphical model over the tree structure $T = ([d], E)$. Then $p^\star$ satisfies Condition 1 with the tree structure T and $r_e = n$ for any $e \in E$. Moreover, sketches in (30) and (31) satisfies the condition in Theorem 9.*

5.4 Higher order Markov sketch function

One can use high-order Markov sketch functions to solve for marginal probability information over more nodes than in the previous cases. Let $\mathcal{S}_k \subset V$ be a choice of nodes of interest to k. By a suitable change of the sketch functions T_k and S_k in Sect. 5.3, one can make it so that one has the following entry-wise equality

$$Z_k^\star = \mathcal{M}_{\mathcal{S}_k} p^\star; \quad Z_{w\to k}^\star = \mathcal{M}_{(\mathcal{S}_k \cup \mathcal{L}(k)) \cup (\mathcal{S}_w \cap \mathcal{R}(w))} p^\star$$

If one wishes to ensure the high-order Markov sketching function satisfies recursive sketching, one needs the following constraint:

$$\{k\} \subset \mathcal{S}_k \subset \cup_{w \in \mathcal{C}(k)} \mathcal{S}_w. \tag{34}$$

For an example, for any integer $L \geq 1$, one sets L as a distance cutoff and let

$$\mathcal{S}_k = \{v \in V \mid \mathrm{dist}(v, k) \leq L\}. \tag{35}$$

By the triangle inequality, the construction in (35) satisfies (34), and one obtains the Markov sketching function when $L = 1$. For the choice of the neighborhood in (35), we refer to the corresponding sketch function as L-Markov sketch function. In particular, 2-Markov sketch function will play an important role in numerical experiments.

5.5 Perturbative sketching

The idea of perturbative sketching is to use recursive random projection to form the sketch functions. In Condition 5- 6, we define the structural assumption of perturbative sketching. In Theorem 14, we give a structural theorem for perturbative sketching, which shows that the sketch $Z_k^\star$ is a power series of tensors. Moreover, each term in the power series corresponds to a random projection of a marginal distribution of $p^\star$, i.e. tensor of the form $\mathcal{M}_{\mathcal{S}} p^\star$ for a subset $\mathcal{S} \subset [d]$.

In Condition 5, we make a significant simplification to recursive sketching by unifying left and right sketching to an equal footing, which allows for a cleaner structural analysis. In Condition 6, we make the assumption that each sketch core is made up of an all-one tensor plus a perturbation term.

Condition 5 (Directional symmetry for perturbative sketching) Assume the sketch functions $\{T_i, S_i\}$ satisfies Condition 4. Furthermore, one lets $l_e = m_e$ for any $e \in E$ and $t_i = s_i$ for any $i \in [d]$.

Condition 6 (perturbative structure for sketch cores)

Fix a constant $\epsilon > 0$ as the *perturbative scale*. The tensor core s_k is defined by tensors O_k and Δ_k of the form:

$$s_k(x_k, \beta_{(k,\mathcal{N}(k))}) = O_k(x_k, \beta_{(k,\mathcal{N}(k))}) + \epsilon \Delta_k(x_k, \beta_{(k,\mathcal{N}(k))}), \tag{36}$$

and moreover O_k is the all one tensor satisfying

$$O_k(x_k, \beta_{(k,\mathcal{N}(k))}) = 1.$$

For a concrete example, if the perturbation is formed by entry-wise i.i.d. random variable, one can use the following line in MATLAB to define a perturbative sketch core:

```
s_k = ones(size(s_k)) + epsilon*rand(size(s_k)).
```

Due to Condition 5, it follows that $Z_k^\star$ only depends on the sketch cores $\{s_i\}_{i\neq k}$. Therefore, one can identify $\gamma_{(k,\mathcal{P}(k))}$ with $\beta_{(k,\mathcal{P}(k))}$. By simple algebra, one can derive the following result:

Theorem 14 *(Structure theorem of perturbative sketching) Assume that Condition 5- 6 are satisfied for the chosen sketch function. For $k \in [d]$ and $\mathcal{S} \subset [d]-\{k\}$, let $T_\mathcal{S} = (\mathcal{S}, E_\mathcal{S})$ be the subgraph of T with vertex set $\mathcal{S}$. As in Definition 11, define $\Delta_\mathcal{S}$ by*

$$\Delta_\mathcal{S}(x_\mathcal{S}, \beta_{\partial\mathcal{S}}) := \sum_{\substack{\beta_e \\ e\in E_\mathcal{S}}} \prod_{i\in\mathcal{S}} \Delta_i(x_i, \beta_{(i,\mathcal{N}(i))}). \tag{37}$$

Then the following equation holds for $Z_k^\star$:

$$Z_k^\star(x_k, \beta_{(k,\mathcal{N}(k))}) = \sum_{l=0}^{d-1} \epsilon^l \sum_{\mathcal{S}\subset[d]-\{k\},|\mathcal{S}|=l} Z_{k;\mathcal{S}}^\star(x_k, \beta_{(k,\mathcal{N}(k))}), \tag{38}$$

where

$$Z_{k;\mathcal{S}}^\star(x_k, \beta_{(k,\mathcal{N}(k))}) = \sum_{\beta_e, k\notin e} \left(\sum_{x_\mathcal{S}} \mathcal{M}_{\mathcal{S}\cup\{k\}} p^\star(x_k, x_\mathcal{S}) \Delta_\mathcal{S}(x_\mathcal{S}, \beta_{\partial\mathcal{S}}) \right). \tag{39}$$

The proof is simple and left in the Appendix. There are a few consequences of Theorem 14. First, each $Z_{k;\mathcal{S}}^\star$ is a projection of the marginal distribution tensor $\mathcal{M}_{\mathcal{S}\cup\{k\}}p^\star$. Second, in (38), terms corresponding to $\mathcal{S}$ is scaled by $\epsilon^{|\mathcal{S}|}$, which itself means that contribution of $\mathcal{S}$ with large cardinality is insignificant. By the description in Sect. 5.1, with perturbative sketching function, TTNS-Sketch essentially tries to fit over all $\mathcal{M}_\mathcal{S}p^\star$, and a $\epsilon^{|\mathcal{S}|}$ factor is placed to ensure $\hat{Z}_k$ stabilizes quickly.

Moreover, if $\mathcal{S}\cup\{k\}$ is not a connected component of T, then according to (39), $Z_{k;\mathcal{S}}^\star$ does not vary with x_s or $\beta_{(k,\mathcal{N}(k))}$. Such $Z_{k;\mathcal{S}}^\star$ has no contribution to $Z_k^\star$ except for on a linear subspace spanned by an all-one tensor. Hence, the subsets $\mathcal{S}$ with nontrivial contribution have a one-to-one correspondence with connected components of T which contains k.

Importantly, the number of connected components of T both containing k and having a small cardinality only depends on the local topology of T around k. In the numerical examples, one can see that perturbative sketching performs quite well if the interaction is local, and it is more adaptable than Markov sketch functions or high-order Markov sketch functions.

In our numerical experiments, we keep a fixed design on the perturbative scale ϵ, but there is a slight numerical benefit in tuning ϵ. Generally, the parameter ϵ should decrease with sample size. The decay rate in N depends on how much marginal distribution information is needed to determine $G_k^\star$. In practice, one can choose a decay rate of $\epsilon := cN^{-f}$, with the parameter c, f determined by cross-validation.

As a remark, theoretically by applying Section E, one would be able to derive the dependence of sample complexity on ϵ, and tune ϵ accordingly. To capture the dependence on ϵ

accurately, the reader is advised to use the tighter original version of the Matrix Bernstein inequality (cf. Theorem 6.1.1 in [29]). The rigorous account of the choice of ϵ will be left for future works.

6 Topology finding

The aim of this section is to provide an algorithm for tree topology selection. In other words, with input being an empirical distribution $\hat{p}$, we define a procedure that outputs a tree structure T. Section 6.1 introduces the Chow-Liu algorithm and gives the theoretical rationale behind using it for tree selection. Section 6.2 discusses the sample complexity of the Chow-Liu algorithm.

6.1 The Chow-Liu algorithm for topology finding

For an arbitrary distribution $p^\star$, we discuss the tree topology specification problem. In practice, it could happen that one has no access to a pre-selected candidate tree topology, but one wishes to find a tree structure to reasonably capture the structure of $p^\star$. To find a tree structure of $p^\star$, the approach of this paper is to solve the problem of fitting $p^\star$ against the best tree graphical model. Then, one outputs the underlying tree structure of the optimal model as the proposed tree topology. For easy reference, we give a definition of the tree graphical model in Definition 15, and Proposition 16 is the main structural property of the construction.

Definition 15 For any tree graph $T = (V, E)$ such that $V = [d]$, a *tree graphical model* over T is a density p which admits the following representation for some $\{f_{i,j}\}_{(i,j)\in E}$:

$$p(x_1, \ldots, x_d) \propto \exp\left(\sum_{(i,j)\in E} f_{i,j}(x_i, x_j)\right). \tag{40}$$

Proposition 16 *Let $X = (X_1, \ldots, X_d) \sim p$, where p is a tree graphical model over $T = (V, E)$. Then X_i and X_j are independent conditioning on X_k if k is on the unique path connecting i and j.*

Proof Suppose that the subgraph of T obtained by removing k has L connected components $(\mathcal{S}_1, \ldots, \mathcal{S}_L)$. One can check from (40) that there exists $(f_1, \ldots, f_L)$ for which $p(x_{[d]-k} \mid x_k) = \prod_{l=1}^{L} f_l(x_{\mathcal{S}_l}, x_k)$. Thus, for any $l \neq l'$, we have shown that $X_{\mathcal{S}_l}$ is independent to $X_{\mathcal{S}_{l'}}$ given k. The fact that X_i and X_j are conditionally independent follow as a corollary, as i and j are on different connected components. □

Now, let $\mathcal{Y}$ denote the collection of all tree graphical models over d variables in the sense of Definition 15. Given an empirical distribution $\hat{p}$ from samples of $p^\star$, the proposed topology finding algorithm solves the optimization problem

$$p_{\mathrm{CL}} = \mathrm{argmin}_{p\in\mathcal{Y}} D_{\mathrm{KL}}(\hat{p} \parallel p), \tag{41}$$

and one outputs a tree structure T_{CL} by reading out the tree structure underlying p_{CL}. The optimization problem (41) can be efficiently computed thanks to the special structure of the tree graphical model. The algorithm for solving (41) is named Chow-Liu algorithm [7], for which we now describe the procedure.

In the first step, one computes the pairwise mutual information $I(X_i, X_j)$ over any distinct pair of $(i, j) \in [d] \times [d]$. In the second step, one forms a graph G, which is a complete graph on $V := [d]$ with edge weight given by $I(X_i, X_j)$. In the third step, Kruskal's algorithm is used to obtain the maximal spanning tree on G, i.e. a spanning tree over d nodes that maximizes the sum of mutual information over all of its edges. This maximal mutual information spanning tree is the Chow-Liu tree T_{CL}. After specifying marginal distribution over single nodes and edges to match that of $\hat{p}$, one uniquely determines the Chow-Liu model p_{CL}, which solves (40). Since p_{CL} is not needed, our proposed tree finding subroutine only needs to calculate the pairwise mutual information and then take the maximal spanning tree.

In addition to the motivation that T_{CL} is connected to the optimization program (41), we further discuss the rationale for using T_{CL} for our TTNS algorithm in three cases.

In the first case, if $p^\star$ is indeed a graphical model over a tree $T^\star$, then the Chow-Liu tree T_{CL} will be $T^\star$ with high probability (see below). Moreover, it is well-known that p_{CL} is the tree graphical model with minimum KL divergence to the input distribution. With mild constraint on the bond dimension, the class of functions representable by a TTNS format strictly covers density representable by a graphical model, and the performance of TTNS-Sketch with T_{CL} has a performance which is on par with p_{CL}.

In the second case, if $p^\star$ has a TTNS ansatz over a tree $T^\star$, then it is typically true that farther-away nodes in $T^\star$ are less correlated. By the maximal spanning tree procedure in Chow-Liu, variables that are far away in T_{CL} are also typically less correlated. If T_{CL} and $T^\star$ differ locally, then the TTNS-sketching algorithm still performs well empirically. As an important example in this category, suppose $p^\star$ is given by a graphical model over a graph with loops. In this case, T_{CL} will converge to a spanning tree of the graph, and one can form a TTNS ansatz with T_{CL} by choosing an appropriately large bond dimension.

In the third case, it may happen that $p^\star$ cannot be represented by a TTNS ansatz. In this case, one can quickly reject the TTNS model assumption by looking at the mutual information $I(X_i, X_j)$ used in the Chow-Liu algorithm. For any node $k \in [d]$, removing k will separate T_{CL} into two connected components. If one sees several pairs with a strong correlation between nodes separated by k, then the density $p^\star$ most likely fails the TTNS model assumption, and more general tensor networks might be more applicable.

6.2 Sample complexity for successful tree topology recovery

We discuss the number of samples required for Chow-Liu to pick the "correct"tree topology. By a correct tree topology, we will mean that the Chow-Liu Tree T_{CL} equals the tree one would have obtained if one forms the maximal spanning tree based on the exact mutual information. If $p^\star$ is a graphical model over $T^\star$, then this notion of correctness coincides with the intuitive notion of $T_{\mathrm{CL}} = T^\star$.

There has been considerable recent work in the past few years on the sample complexity of the Chow-Liu algorithm to infer the correct tree topology in the sample case. [1] shows that the sample complexity is bounded by $O(\frac{n^3 d}{\epsilon} \log 1/\delta)$ to ensure a $1 - \delta$ success rate, where ϵ is the gap in the sum of mutual information between the two best tree models. For the tree-based Ising model with no external field, [3] proves an upper bound that is $O(\log(d/\delta))$.

7 Numerical result

In this section, we perform a comparison of different modeling methods. There are four models of interest. The symbol $\hat{p}_{\mathrm{TS}}$ stands for the model obtained from the TTNS-Sketch method. The symbol $\hat{p}_{\mathrm{GM}}$ stands for the model one obtains from direct graphical modeling over a given tree structure T. Specifically, the $\hat{p}_{\mathrm{GM}}$ model with tree structure T refers to the graphical model over T where the parameters are chosen by maximum likelihood estimation. The symbol $\hat{p}_{\mathrm{CL}}$ stands for the Chow-Liu model, which is obtained by direct graphical modeling with the Chow-Liu tree T_{CL}. The symbol $\hat{p}_{\mathrm{BM}}$ stands for the model one obtains from modeling with Born Machine (BM). The training of BM is done by optimizing Negative Log Likelihood (NLL), with details of the training following from that of [13].

In what follows, the *error* of a model p refers to the relative l_2 error:

$$\mathrm{Error}(p) := \frac{\|p - p^\star\|}{\|p\|}.$$

For BM training, we will use the negative log-likelihood level of the model as a performance metric:

$$\mathrm{NLL}(p) := \sum_{i=1}^{N} p(y_1^{(i)}, \ldots, y_d^{(i)}).$$

7.1 Numerical case study: tree graphical model with different input tree topology

In this numerical experiment, we aim to exhibit the importance of using a correct tree topology. We will focus on testing the TTNS-Sketch algorithm on a tree-based graphical model under input tree topology misspecification. Specifically, given any fixed tree structure $T = (V, E)$, we consider the following graphical model over T:

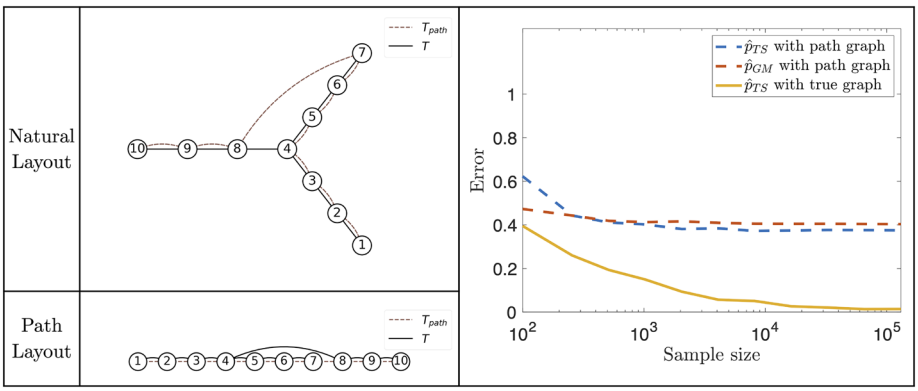
$$p^\star(x_1, \ldots, x_d) = \exp\left(-\beta \sum_{(i,j)\in E} f_{i,j}(x_i, x_j)\right), \tag{42}$$

where $\beta > 0$ is the temperature parameter. We test a simple binary model where $\beta = 1/2$ and $f_{i,j}(x_i, x_j) = -x_i x_j$ with each $x_i \in \{-1, 1\}$, which is the setting of standard ferromagnetic Ising model.

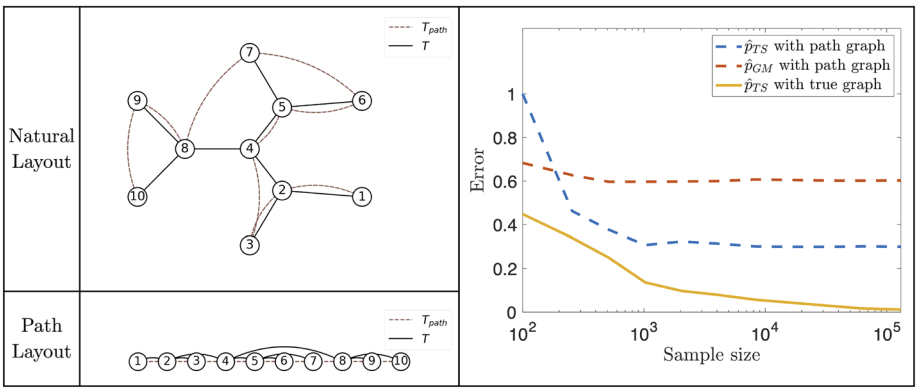
First, consider the case where T is a 10-node trident, see illustration in Fig. 8a. Even in such a simple example, one can see that there is no good way to choose a path to fit the tree model. Suggested by our deliberate choice in ordering the variables, one reasonable candidate is a path graph T_{path} that traverses from node 1 to 10 in numerical order. Indeed, with only one exception of the edge (4, 8), the tree T_{path} is almost made up of edges from T.

Likewise, we include the 10-node dendrimer graph in Fig. 8b and a bipartite graph in Fig. 8c, where we also use the numerical order to indicate the chosen T_{path} structure. While the dendrimer case also uses a reasonable path structure, one can see that the bipartite graph case uses a T_{path} structure very different from T. In Fig. 8, the natural layout draws T and T_{path} in a layout natural to T, where one can see how often T_{path} uses edge from T.

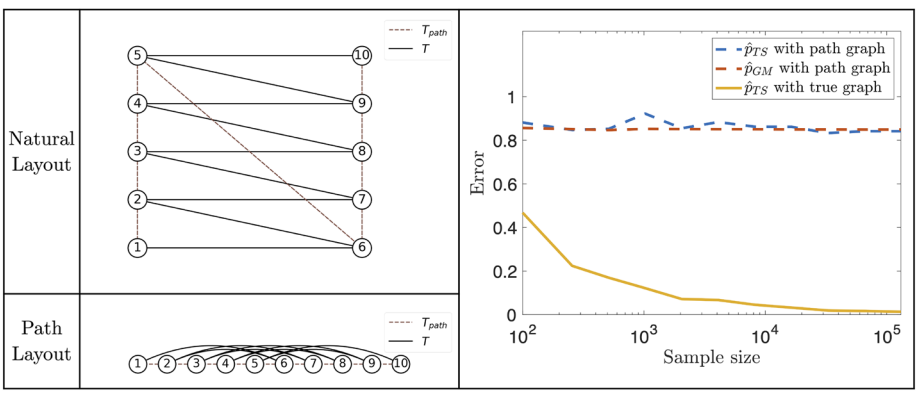
For the three tree structures, we will compare three methods (i)-(iii): (i) TTNS-Sketch over T_{path}, (ii) Graphical modeling over T_{path}, (iii) TTNS-Sketch over T. In (i) and (iii), we choose the Markov sketch function. The results are listed in Fig. 8. One can see that the error for (iii) always converges to zero with large N, which is consistent with Lemma



(A) Trident Graph, $d = 10$



(B) Dendrimer Graph, $d = 10$



(C) Bipartite Graph, $d = 10$

Fig. 8 The black solid line stands for edges on T and the dashed brown line stands for edges on T_{path}. The graph is plotted in a layout that is respectively natural for T and for T_{path}. Error plots for each case of T are also included. One can see that convergence to the true model only occurs with true tree specification

13. However, both (i) and (ii) do not converge to $p^\star$, with the performance being worst in the bipartite graph case.

In the path layout, one can more naturally "count"the internal bond dimension necessary to let $p^\star$ admit a TTNS ansatz under T_{path}. Let E_{path} stand for the edge set for T_{path}, which is essentially $E_{\text{path}} = \{(i, i+1)\}_{i=1}^{d-1}$. In this case, for any edge $e = (i, i+1)$, one can calculate the internal bond r_e by counting the number of edges one would "cut"if one

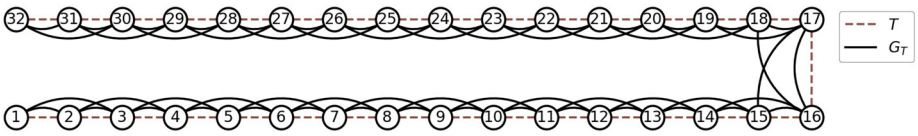


Fig. 9 Illustration of the graphical model in Sect. 7.2. The solid black line indicates edges for true graphical model G_T. The dashed brown line indicates edges for T, the tree model to be used for TTNS-Sketch

places a vertical line in between node i and $i+1$. If one counts q_i edges, then one has the upper bound $r_e \leq 2^{q_i}$ for the Ising model. More generally, suppose $X \sim p^\star$, and let each entry of X be a discrete variable over $\{1, \ldots, n\}$, then r_e is upper bounded by n^{q_i}. The description of q_i exactly coincides with the number of edges across the partition $[d] = \{1, \ldots, i\} \cup \{i+1, \ldots, d\}$, i.e. the cardinality of the cut from the partition.

Thus, one can easily extend Fig. 8 to cases with more nodes, and then the three models will have quite different behavior. The cut number q_i for the trident case is upper bounded by $q_i \leq 2$, but largest q_i for the bipartite graph is $d-1$ when d is even. Hence, in the bipartite case, the TTNS ansatz of $p^\star$ under T_{path} is not practical to compute, while the TTNS ansatz of $p^\star$ under T is simple. Importantly, the bipartite graph T is itself a path graph, and so the $p^\star$ is not complicated.

Thus, while modest tree structure misspecification can be treated with a higher bond dimension, a large structural deviation may lead to an intractable bond dimension penalty. We also note that in all of these cases, one has $T_{\text{CL}} = T$ with overwhelmingly high probability. Due to the $O(d)$ sample complexity to recover T, we will henceforth assume that one has access to the true tree model T.

7.2 Numerical case study: 1D spin system with non-local interactions

In this example, we consider a more complicated Markov random field model. Given a fixed tree structure $T = (V, E)$, we denote the shortest-path distance on T as dist_T. For any fixed positive integer l, we propose the following model

$$p^\star(x_1, \ldots, x_d) = \exp\left(-\beta \sum_{\text{dist}_T(i,j) \leq l} f_{i,j}(x_i, x_j)\right). \tag{43}$$

This is also a graphical model over $G_T = (V, E')$ by letting $E' = \{(i,j) \mid \text{dist}_T(i,j) \leq l\}$. In particular, we set $l := 2$. Moreover, we consider the case where T is a path graph with $d = 32$. See the illustration in Fig. 9.

Moreover, to demonstrate our algorithm under more general situations than binary data, we consider the 4-state clock model with $f_{i,j}(x_i, x_j) = -\cos(x_i - x_j)$ and $x_i \in \{0, \frac{2}{3}\pi, \frac{4}{3}\pi\}$. In this case, we set $\beta = 1/2$ to ensure the spin-spin correlation strength is at an appropriate level.

If one applies the Chow-Liu algorithm to the model in (43), one could obtain any one of the spanning trees of G_T. Hence, for the sake of fair comparison, we fix the path graph T as the input tree graph to TTNS-Sketch. We test TTNS-Sketch under the following sketch function (i) perturbative sketching function, (ii) Markov sketch function, and (iii) 2-Markov sketch function (defined in Sect. 5.4). For the perturbative sketch function, we

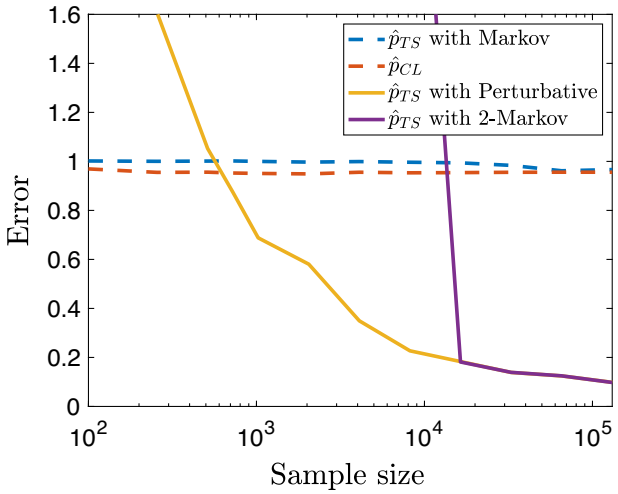


Fig. 10 Error comparison for the graphical model in Fig. 9. The model $\hat{p}_{\mathrm{CL}}$ stands for the result of direct graphical modeling with the tree structure T in Fig. 9. TTNS-Sketch with perturbative sketch function and 2-Markov sketch function converges to the true model. One can see that the two sketch functions enjoy a similar performance, while the perturbative sketch function is much more stable at a small sample size

pick $\epsilon = 1$ and a sketch core size of $l_e = 20$ for any $e \in E$. We apply singular value thresholding to ensure numerical stability.

We observe convergence for TTNS-Sketch under the perturbative sketching function and the 2-Markov sketch function. The result is shown in Fig. 10.

7.3 Numerical case study: tree graphical model with large variable dimension

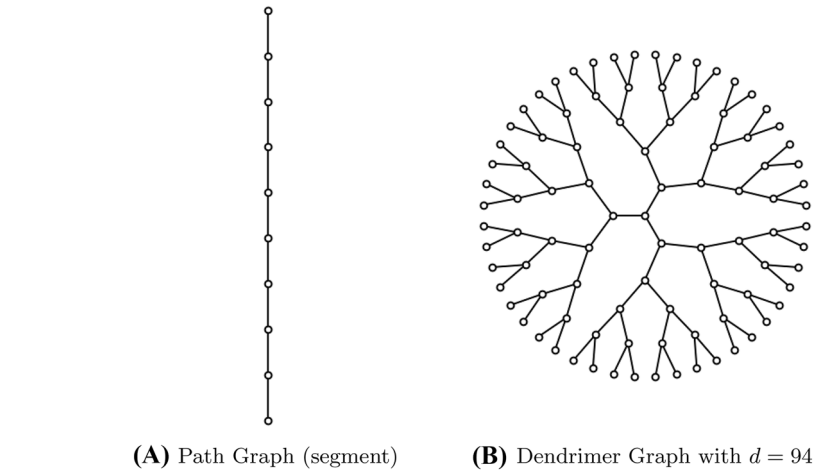
In this example, we test the performance of TTNS-Sketch in the setting of large d. We consider tree-based graphical models of the form (42) with $\beta = 1/2, f_{i,j}(x_i, x_j) = -x_i x_j$, and $x_i \in \{-1, 1\}$, which is the case of Ising model. As a remark, the cases under more general $\{f_{i,j}\}_{(i,j)\in E}$ lead to the same conclusion, and so they are excluded for the sake of brevity. For the candidate graph T, we consider a path graph with $d = 100$ nodes and the 3-fractal dendrimer graph with $d = 94$ nodes, see Fig. 11.

For samples generated from the underlying model, direct graphical modeling with the underlying tree model T provides the best tree-based graphical model in the sense of MLE, and hence we put it as the benchmark. We choose the perturbative sketching function with $\epsilon = 0.05$. The result can be seen in Fig. 12. One can see that TTNS-Sketch performs well under the large variable regime, and has the same convergence behavior as the benchmark. Due to the $N = O(d^2)$ scaling in Theorem 34 and the $O(d)$ computational scaling, one can likewise perform the same procedure up to very large d.

7.4 Numerical case study: spin system with long range interactions

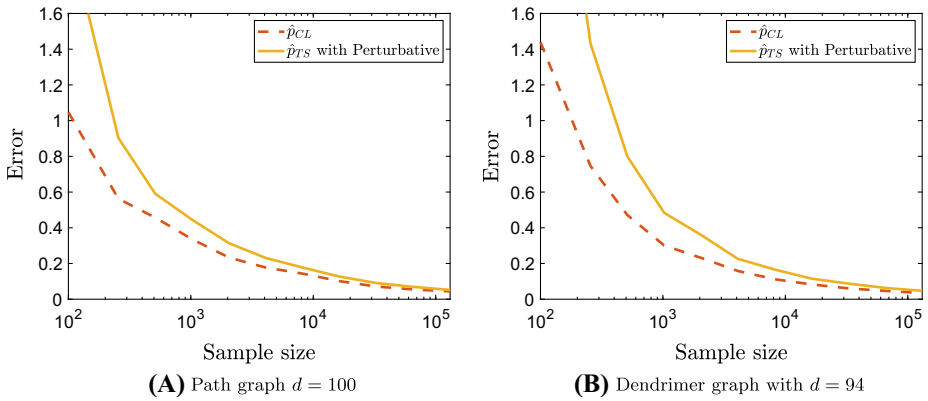
Quite importantly, in this numerical experiment, we introduce another important benchmark called Born Machine (BM). As BM is a method over the tensor train format, we restrict TTNS-Sketch to the path graph T_{path} to ensure a fair comparison.

There are two important differences between BM and TTNS-Sketch. First, BM assures the positivity of the trained model. Second, BM is based on Negative Log Likelihood (NLL) training, which is a more conventional error metric for statistical modeling. While TTNS-



(A) Path Graph (segment) **(B)** Dendrimer Graph with $d = 94$

Fig. 11 Plot of graphs considered in Sect. 7.3



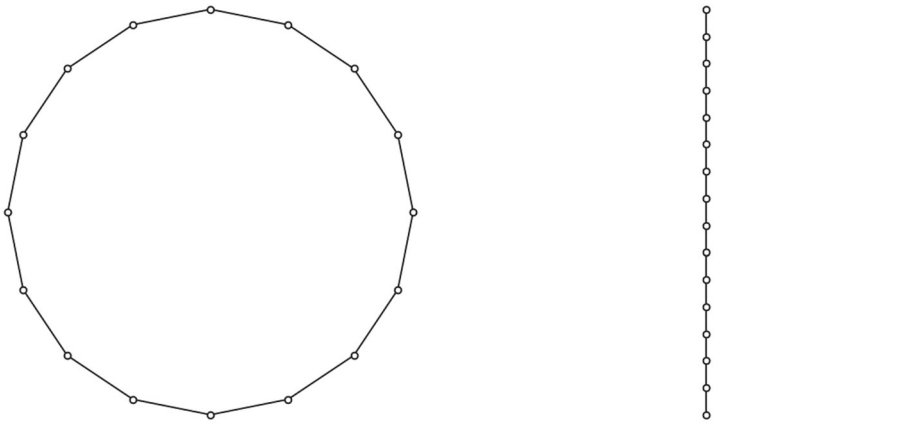
(A) Path graph $d = 100$ **(B)** Dendrimer graph with $d = 94$

Fig. 12 Error comparison for path graph with $d = 100$ and dendrimer graph with $d = 94$. The model $\hat{p}_{\mathrm{CL}}$ stands for the result of direct graphical modeling with the true Tree structure T. One can see that both methods converge to the true model

Sketch does not ensure the trained model is positive, the $\|\cdot\|_\infty$ norm error bound obtained in Section E ensures that positivity is not a major issue if the correct tree structure and sketch function are provided.

During the numerical experiment with BM, we discovered a significant failure mode of BM, which is that it fails to converge to the true model for periodic spin systems. A similar observation has been independently made in [11]. To show the failure mode in the simplest possible setting, we discuss the case of a Markov random field with a ring graph G with $d = 16$ nodes, and its graphical model representation of (44) is illustrated in Fig. 13a:

$$p^\star(x_1, \ldots, x_d) = \exp\left(- \sum_{i-j=0 \mod d} f_{i,j}(x_i, x_j)\right) \tag{44}$$



(A) Ring model with $d = 16$ **(B)** Linear path model with $d = 16$

Fig. 13 Graph representations of the true model in Sect. 7.4 and the path model learned by path-based graphical modeling

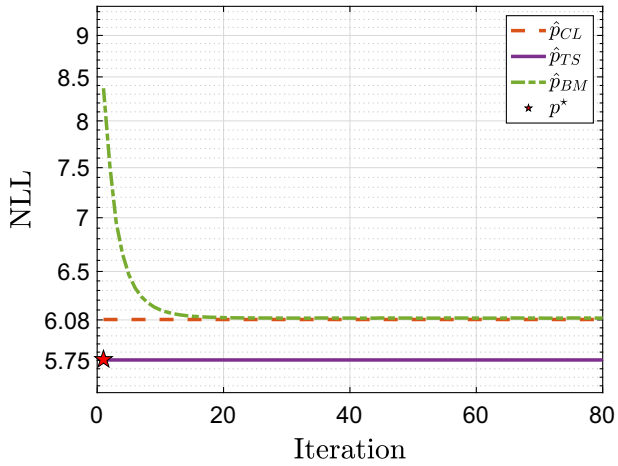


Fig. 14 Negative Log Likelihood (NLL) level comparison for the ring graph with $d = 16$ in Fig. 13a. Here, $N = 2^{15} = 32,768$. The model $\hat{p}_{\text{CL}}$ stands for the result of direct graphical modeling with the path structure in Fig. 13b. The model $\hat{p}_{\text{BM}}$ stands for the result of Born Machine with $r_{\text{max}=4}$. The model $\hat{p}_{\text{TS}}$ stands for the result of TTNS-Sketch with the special high-order Markov sketch function in (45). One can see that the output of TTNS-Sketch is close to the NLL level of the true model, while the output of BM is close to the NLL level of the path-based graphical model $\hat{p}_{\text{CL}}$

For TTNS-Sketch, we pick a special high-order Markov sketch function, with the following neighborhood of interest:

$$\mathcal{S}_k = \{k-1, k, k+1\} \cup \{1, d\}, \tag{45}$$

which is chosen to ensure convergence in the limit of sample size. Importantly, we include direct graphical modeling over the path graph T_{path} as a benchmark.

We now discuss the numerical specification for the BM modeling. We pick the sample size $N = 2^{15} = 32,768$, which is sufficiently large so that the generalization error should be small. For the BM method, there is a parameter r_{max}, which is the largest allowed

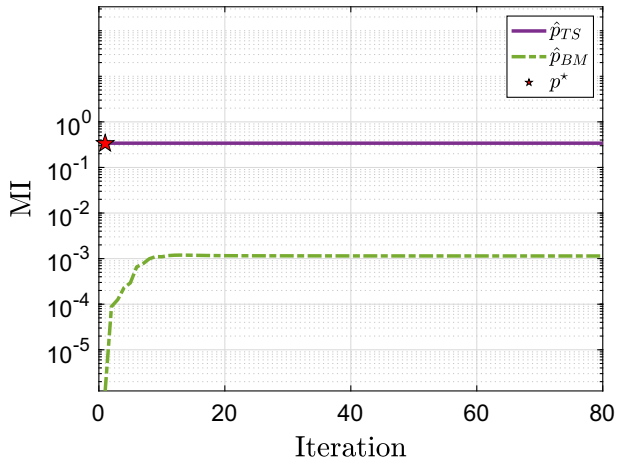


Fig. 15 Mutual Information (MI) level comparison for the ring graph with $d = 16$ in Fig. 13a, with experiment setting described in the caption of Fig. 14. The MI level of a distribution p refers to $I(X_1, X_d)$ where $X \sim p$. In other words, the MI level is the mutual information between the last and first entry of the multivariate distribution according to p. One can see that the mutual information level of BM is close to zero throughout training, which is further evidence that BM effectively learns a path graphical model

internal bond dimension, i.e. $\max_{e\in E} r_e \leq r_{\max}$. Importantly, $p^\star$ admits a tensor train ansatz with $\max_{e\in E} r_e \leq 4$, and so we pick $r_{\max} = 4$ to ensure that the BM method in this instance has no approximation error. The only training parameter is the learning rate which is picked according to cross-validation.

In Fig. 14, we plot the result of the comparison between BM, TTNS-Sketch, and direct graphical modeling. Quite surprisingly, TTNS-Sketch can converge, while the NLL level of BM stays at a sub-optimal level. Interestingly, the NLL level of the BM model $\hat{p}_{\mathrm{BM}}$ converges to that of the graphical model $\hat{p}_{\mathrm{CL}}$. Moreover, one can also show that the KL-divergence between $\hat{p}_{\mathrm{BM}}$ and $\hat{p}_{\mathrm{CL}}$ is quite small. In the converged model plotted in Fig. 14, one has $D_{\mathrm{KL}}\left(\hat{p}_{\mathrm{BM}} \,\|\, p^\star\right) \approx 0.4357$, while $D_{\mathrm{KL}}\left(\hat{p}_{\mathrm{BM}} \,\|\, \hat{p}_{\mathrm{CL}}\right) \approx 0.0266$.

Numerically, we can then conclude that the BM training implicitly leads to a sub-optimal model, and the gap to the true model can be explained by the sub-optimal model's closeness to the path-based graphical model $\hat{p}_{\mathrm{CL}}$, see Fig. 13 for illustration. As additional evidence, during the training dynamics, the learned model fails to capture the correlation between node 1 and node d. In Fig. 15, the mutual information level between node 1 and node d is plotted throughout training, and one can see the level is consistently close to zero, despite the strong correlation in $p^\star$.

In practice, when $d = 16$, this phenomenon of BM training failure persists up to $r_{\max} = 7$, and is resolved by setting $r_{\max} \geq 8$. However, if d is increased, we have observed that the smallest $r_{\max}$ for successful BM training also increases. This coupling of maximal bond dimension with d is problematic, as one could no longer expect a linear dependency between parameter size and the dimension d.

On the other hand, for TTNS-Sketch under (45), one is theoretically guaranteed convergence to true model with $r_e \leq 4$, the proof of which can be obtained by adapting the proof for Lemma 13. Moreover, for TTNS-Sketch with sketch function in 45, increasing d does not affect performance beyond the sample complexity scaling in d as discussed in Section E.

8 Conclusion

We describe an algorithm TTNS-Sketch, which obtains a Tree Tensor Network States representation of a probability density from a collection of its samples. This is done by formulating a sequence of equations, one for each core, which can be solved independently. This is a general framework that allows for arbitrary tree structures to be used. We have also compared this algorithm with similar training-based regimes in the tensor train format, in which we have shown much better performance from TTNS-Sketch even in a simple case of periodic spin systems. For models which have interaction beyond immediate neighbors, we have shown that TTNS-Sketch with perturbative sketching greatly outperforms Chow-Liu. Theoretically, we have provided conditions for TTNS-Sketch to be a consistent estimator, as well as a reasonable sample complexity upper bound.

While TTNS-Sketch might not necessarily be superior at approximating an arbitrary density, the numerical and theoretical evidence gathered point to the conclusion that it is good at the inverse problem of solving for the tensor component of a density with a TTNS ansatz. In other words, a deterministic linear algebraic subroutine is sufficient to approximate a $p^\star$ with a low-rank TTNS format. While this point has been made for tensor completion problems, it is quite remarkable that statistical inference of models with a TTNS ansatz can be shown to reduce to a linear algebraic problem.

Acknowledgements

The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request. The authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this paper.

Author details

[1]Institute for Computational and Mathematical Engineering, Stanford CA 94305, USA, [2]Department of Statistics, Chicago IL 60637, USA, [3]Computational and Applied Mathematics Initiative, Department of Statistics, University of Chicago, Chicago IL 60637, USA, [4]Department of Mathematics and Institute for Computational and Mathematical Engineering, Stanford CA 94305, USA.

Appendix A Proof of Theorem 7

Proof (Of Theorem 7) For simplicity, for the remainder of the proof, we fix a structure for all of the high-order tensors we use. For $\Phi_{w\to k}$, one reshapes it to the unfolding matrix $\Phi_{w\to k}(x_{\mathcal{L}(w)\cup w}; \alpha_{(w,k)})$. For $\Psi_{w\to k}$, one reshapes it to the unfolding matrix $\Psi_{w\to k}(\alpha_{(w,k)}; x_{\mathcal{R}(w)})$. For $\Phi_{\mathcal{C}(k)\to k}$, one reshapes it to the unfolding matrix $\Phi_{\mathcal{C}(k)\to k}(x_{\mathcal{L}(k)}; \alpha_{(k,\mathcal{C}(k))})$. For G_k, one reshapes it to the unfolding matrix $G_k(\alpha_{(k,\mathcal{C}(k))}; x_k, \alpha_{(k,\mathcal{P}(k))})$. For p, we will explicitly write out the unfolding matrix structure to avoid ambiguity.

According to Condition 1, for any edge $w \to k$, one has

$$p(x_{\mathcal{L}(w)\cup w}; x_{\mathcal{R}(w)}) = \Phi_{w\to k}\Psi_{w\to k}. \tag{46}$$

Then, define $\Phi^\dagger_{w\to k}\colon [r_{(w,k)}] \times \left[\prod_{i\in\mathcal{L}(w)\cup w} n_i\right] \to \mathbb{R}$ and $\Psi^\dagger_{w\to k}\colon \left[\prod_{i\in\mathcal{R}(w)} n_i\right] \times [r_{(w,k)}] \to \mathbb{R}$ so that $\Phi^\dagger_{w\to k}(\alpha_{(w,k)}; x_{\mathcal{L}(w)\cup w})$ denotes the pseudoinverse of $\Phi_{w\to k}$, and $\Psi^\dagger_{w\to k}(x_{\mathcal{R}(w)}; \alpha_{(w,k)})$ denotes the pseudoinverse of $\Psi_{w\to k}(\alpha_{(w,k)}; x_{\mathcal{R}(w)})$. Then,

$$\Phi^\dagger_{w\to k}\Phi_{w\to k} = \Psi_{w\to k}\Psi^\dagger_{w\to k} = \mathbb{I}_{r_{(w,k)}}$$

First, we prove the uniqueness of each equation of (5) in the sense of least squares. Note that an exact solution is guaranteed when k is a leaf, and so one only needs to consider when k is non-leaf. By assumption, $\Phi_{w\to k}$ has a full column rank of $r_{w\to k}$. In particular, the Kronecker product structure ensures that $\Phi_{\mathcal{C}(k)\to k}$ has full column rank of $\prod_{w\in\mathcal{C}(k)} r_{w\to k}$. Therefore, a unique solution to (5) exists in the sense of least squares.

Moreover, when k is non-leaf and non-root, the pseudoinverse $\Phi^{\dagger}_{\mathcal{C}(k)\to k}(\alpha_{(k,\mathcal{C}(k))}; x_{\mathcal{L}(k)})$ leads to the following explicit construction of G_k:

$$G_k = \Phi^{\dagger}_{\mathcal{C}(k)\to k}\Phi_{k\to\mathcal{P}(k)}, \tag{47}$$

and likewise when k is root, one has

$$G_k = \Phi^{\dagger}_{\mathcal{C}(k)\to k}p(x_{\mathcal{L}(k)}; x_k). \tag{48}$$

To verify that (5) holds exactly for the construction of G_k in (47), one can argue it suffices to check that

$$\Phi_{\mathcal{C}(k)\to k}\Phi^{\dagger}_{\mathcal{C}(k)\to k}p(x_{\mathcal{L}(k)}; x_{k\cup\mathcal{R}(k)}) = p(x_{\mathcal{L}(k)}; x_{k\cup\mathcal{R}(k)}), \tag{49}$$

for which we give a brief explanation. When k is the root, (48) implies that (49) coincides with (5) for when k is root. When k is non-root and non-leaf, one can multiply both sides of (49) by $\Psi^{\dagger}_{k\to\mathcal{P}(k)}$ and sum over $x_{\mathcal{R}(k)}$. According to (47), the obtained equation coincides with (5) for when k is non-root and non-leaf.

It remains to show that (49) holds. For an edge $w \to k \in E$, define a term $Q_{w\to k}$ as follows

$$Q_{w\to k}(x_{\mathcal{L}(w)\cup w}; y_{\mathcal{L}(w)\cup w}) := \sum_{\alpha_{(w,k)}} \Phi_{w\to k}(x_{\mathcal{L}(w)\cup w}; \alpha_{(w,k)})\Phi^{\dagger}_{w\to k}(\alpha_{(w,k)}; y_{\mathcal{L}(w)\cup w}).$$

Then, for a generic tensor $f\colon [n_1]\times\ldots\times[n_d]\to\mathbb{R}$, one can define a projection operator $P_{w\to k}$ as follows

$$(P_{w\to k}f)(x_1,\ldots,x_d) = \sum_{y_{\mathcal{L}(w)\cup w}} Q_{w\to k}(x_{\mathcal{L}(w)\cup w}, y_{\mathcal{L}(w)\cup w})f(y_{\mathcal{L}(w)\cup w}, x_{\mathcal{R}(w)}).$$

By commutativity of the sum operations involved, one has

$$\begin{aligned}\Phi_{\mathcal{C}(k)\to k}\Phi^{\dagger}_{\mathcal{C}(k)\to k}f &= \sum_{y_{\mathcal{L}(k)}}\left(\prod_{w\in\mathcal{C}(k)} Q_{w\to k}(x_{\mathcal{L}(w)\cup w}; y_{\mathcal{L}(w)\cup w})\right) f(x_1,\ldots,x_d)\\ &= \sum_{\substack{y_{\mathcal{L}(w)\cup w}\\ w\in\mathcal{C}(k)}}\left(\prod_{w\in\mathcal{C}(k)} Q_{w\to k}(x_{\mathcal{L}(w)\cup w}; y_{\mathcal{L}(w)\cup w})\right) f(x_1,\ldots,x_d)\\ &= \left(\prod_{w\in\mathcal{C}(k)} P_{w\to k}\right) f\end{aligned}$$

Thus, (49) holds if one can show that $P_{w\to k}p = p$ for any $w \in \mathcal{C}(k)$, but this fact is straightforward:

$$P_{w\to k}p = \Phi_{w\to k}\Phi^{\dagger}_{w\to k}p(x_{\mathcal{L}(w)\cup w}; x_{\mathcal{R}(w)})$$

$$= \Phi_{w\to k}\Phi_{w\to k}^{\dagger}\Phi_{w\to k}\Psi_{w\to k} = \Phi_{w\to k}\Psi_{w\to k} = p,$$

and thus (5) exactly holds for the constructed $\{G_i\}_{i=1}^d$.

Lastly, we prove that the solution $\{G_i\}_{i=1}^d$ forms a TTNS tensor core of p. To show this result, it will be much more convenient to use the notion of subgraph TTNS function in Definition 11. We remark that the construction in Definition 11 is only arithmetic and has no dependency on this theorem. For every node $k \in [d]$, define a subset $\mathcal{S}_k := \mathcal{L}(k) \cup \{k\}$. Then, for non-root k, we prove that $\Phi_{k\to\mathcal{P}(k)}$ is the subgraph TTNS function over $\{G_i\}_{i=1}^d$ and $T_{\mathcal{S}_k}$, i.e. one wishes to show

$$\Phi_{k\to\mathcal{P}(k)}(x_{\mathcal{L}(k)\cup k}, \alpha_{k\to\mathcal{P}(k)}) = \sum_{\substack{\alpha_e \\ e\neq(k,\mathcal{P}(k))}} \prod_{i\in\mathcal{L}(k)\cup k} G_i\left(x_i, \alpha_{(i,\mathcal{N}(i))}\right). \tag{50}$$

We prove (50) by induction. Notice that (5) proves (50) when k is a leaf node. Then, suppose that k is non-leaf and suppose by induction that $\Phi_{w\to k}$ satisfies (50) for all $w \in \mathcal{C}(k)$. Then, one can rewrite (5) by plugging in $\Phi_{\mathcal{C}(k)\to k}$ the form of each $\Phi_{w\to k}$ according to (50). The resulting equation is exactly (50) for $\Phi_{k\to\mathcal{P}(k)}$. By induction over nodes by topological order, (50) holds for every non-root k.

By the same logic, now consider (5) when k is the root. One plugs in $\Phi_{\mathcal{C}(k)\to k}$ the form of each $\Phi_{w\to k}$ according to (50). The resulting equation is exactly (3) in Definition 5, and thus $\{G_i\}_{i=1}^d$ does form a TTNS tensor core of p. □

Appendix B Proof of Theorem 9

Proof (of Theorem 9) For any non-root k, note that $Z_k^\star$ is assumed to be of rank $r_{(k,\mathcal{P}(k))}$ by (ii) in Condition 3. Let $Q_k^\star$ be as in (16). In other words, $Q_k^\star(\gamma_{(k,\mathcal{P}(k))}; \alpha_{(k,\mathcal{P}(k))})$ is formed by the rank-$r_{(k,\mathcal{P}(k))}$ SVD decomposition of $Z_k^\star$ in the SystemForming step of Algorithm 1. Thus $Q_k^\star(\gamma_{(k,\mathcal{P}(k))}; \alpha_{(k,\mathcal{P}(k))})$ is of rank $r_{(k,\mathcal{P}(k))}$, which means it has full column rank. We define

$$\Phi_{k\to\mathcal{P}(k)}^\star(x_{\mathcal{L}(k)\cup k}, \alpha_{(k,\mathcal{P}(k))}) := \sum_{\gamma_{(k,\mathcal{P}(k))}} \bar{\Phi}_k^\star(x_{\mathcal{L}(k)\cup k}, \gamma_{(k,\mathcal{P}(k))}) Q_k^\star(\gamma_{(k,\mathcal{P}(k))}, \alpha_{(k,\mathcal{P}(k))}). \tag{51}$$

Due to (i) in Condition 4 and Q_k having full rank, one can conclude that $\Phi_{k\to\mathcal{P}(k)}^\star(x_{\mathcal{L}(k)\cup k}; \alpha_{(k,\mathcal{P}(k))})$ and $\Phi_{k\to\mathcal{P}(k)}^\Delta(x_{\mathcal{L}(k)\cup k}; \alpha_{(k,\mathcal{P}(k))})$ have the same column space. Thus, there exists $\Psi_{(w,k)}^\star$'s such that $\{\Phi_{(w,k)}^\star, \Psi_{(w,k)}^\star\}_{(w,k)\in E}$ forms a collection of the low-rank decomposition of $p^\star$ in the sense of Condition 1. We make the following claim, which also justifies $\star$ upper-index in (51):

Claim: $\{\Phi_{(w,k)}^\star\}_{(w,k)\in E}$ as defined in (51) satisfies Condition 2

The proof of the claim is somewhat technical and we will defer the proof after stating how it proves this theorem.

Assume the claim is correct and $\{\Phi_{(w,k)}^\star\}_{(w,k)\in E}$ satisfies Condition 2. As a consequence, if one defines $\{A_i^\star, B_i^\star\}$ by

$$\{A_i^\star, B_i^\star\}_{i=1}^d \leftarrow \mathit{SystemForming}(\{Z_{w\to k}^\star\}_{w\to k\in E}, \{Z_i^\star\}_{i=1}^d),$$

then one can alternatively define $\{A_i^\star, B_i^\star\}_{i=1}^d$ by (22) with

$$\{\Phi_{(w,k)}^\Delta\}_{(w,k)\in E} \leftarrow \{\Phi_{(w,k)}^\star\}_{(w,k)\in E}.$$

Thus, with the alternative definition in (22), it follows that (23) is a (possibly over-determined) linear system formed by a linear projection of the linear system in (5), where chosen gauge is $\{\Phi^\star_{(w,k)}\}_{(w,k)\in E}$.

Due to Theorem 7, (5) is an over-determined linear system with a unique and exact solution. Theorem 7 guarantees an exact solution $\{G_i^\star\}_{i=1}^d$ to (5), which is then necessarily an exact solution to (23). If (23) satisfies the uniqueness of the solution, then the solution to (23) is a solution to (5), which by Theorem 7 forms a TTNS tensor core of $p^\star$. Therefore, it suffices to check uniqueness. The uniqueness of solution to (23) when k is a leaf is trivial. When k is non-leaf, note that one can apply (iii) in Condition 3 with $\{\Phi^\star_{(w,k)}\}_{(w,k)\in E}$ as the chosen gauge, which guarantees that $A_k^\star(\beta_{(k,\mathcal{C}(k))}, \alpha_{(k,\mathcal{C}(k))})$ has the full column rank for every non-leaf k. In other words, one is guaranteed the uniqueness of the solution to (23), as desired. For the assertion on the consistency of $\hat{G}_k$, note that $\lim_{N\to\infty} \hat{G}_k = G_k^\star$ follows from the fact that $\lim_{N\to\infty} \hat{A}_k = A_k^\star$ and $\lim_{N\to\infty} \hat{B}_k = B_k^\star$.

We now prove that $\{\Phi^\star_{(w,k)}\}_{(w,k)\in E}$ satisfies Condition 2. For a clear exposition, we adopt the unfolding 3-tensor structure developed in Sect. 8- 8. We remark that the 3-tensor construction is only arithmetic and does not depend on the validity of this theorem.

For $Z_k^\star$, we reshape it as $Z_k^\star(\beta_{(k,\mathcal{C}(k))}; x_k; \gamma_{(k,\mathcal{P}(k))})$. For $U_k^\star$, we reshape it as $U_k^\star(\beta_{(k,\mathcal{C}(k))}; x_k; \alpha_{(k,\mathcal{P}(k))})$. For $Q_k^\star$, we reshape it as $Q_k^\star(\gamma_{(w,k)}; 1; \alpha_{(w,k)})$. For S_k, T_k, we reshape as $S_k(\beta_{(k,\mathcal{C}(k))}; 1; x_{\mathcal{L}(k)})$ and $T_k(x_{\mathcal{R}(k)}; 1; \gamma_{(k,\mathcal{P}(k))})$. For $\Phi^\star_{k\to\mathcal{P}(k)}$, we reshape it as $\Phi^\star_{k\to\mathcal{P}(k)}(x_{\mathcal{L}(k)}; x_k; \alpha_{(k,\mathcal{P}(k))})$. For $\bar{\Phi}_k^\star(x_{\mathcal{L}(k)\cup k}, \gamma_{(k,\mathcal{P}(k))})$, we reshape it as $\bar{\Phi}_k^\star(x_{\mathcal{L}(k)}; x_k; \gamma_{(k,\mathcal{P}(k))})$. For $p^\star$, we reshape it as $p^\star(x_{\mathcal{L}(k)}; x_k; x_{\mathcal{R}(k)})$.

Then, by the construction of $Q_k^\star$ in the SystemForming step of Algorithm 1, one has $U_k^\star = Z_k^\star \circ Q_k^\star$. By (51), it follows $\Phi^\star_{k\to\mathcal{P}(k)} := \bar{\Phi}_k^\star \circ Q_k^\star$. Condition 2 is satisfied if $U_k^\star = S_k \circ \Phi^\star_{k\to\mathcal{P}(k)}$. With such a choice of unfolding 3-tensor, one obtains a simple proof as follows

$$U_k^\star = Z_k^\star \circ Q_k^\star = S_k \circ p^\star \circ T_k \circ Q_k^\star = S_k \circ \bar{\Phi}_k^\star \circ Q_k^\star = S_k \circ \Phi^\star_{k\to\mathcal{P}(k)},$$

where the first equality comes from $Z_k^\star = S_k \circ p^\star \circ T_k$, the second equality comes from $\bar{\Phi}_k^\star = p^\star \circ T_k$, and the third equality comes from $\Phi^\star_{k\to\mathcal{P}(k)} = \bar{\Phi}_k^\star \circ Q_k^\star$. Thus the claim holds and we are done.

□

Appendix C Proof of Lemma 13

Lemma 17 *Suppose p satisfies the Markov property given a rooted tree $([d], E)$. For any subsets $\mathcal{S}_1 \subset \mathcal{L}(k) \cup k$ and $\mathcal{S}_2 \subset \mathcal{R}(k)$,*

(i) $\mathcal{M}_{\mathcal{S}_1\cup\mathcal{S}_2} p(x_{\mathcal{S}_1}; x_{\mathcal{S}_2})$ *and* $\mathcal{M}_{\mathcal{S}_1\cup\mathcal{P}(k)} p(x_{\mathcal{S}_1}; x_{\mathcal{P}(k)})$ *have the same column space if* $\mathcal{P}(k) \in \mathcal{S}_2$,

(ii) $\mathcal{M}_{\mathcal{S}_1\cup\mathcal{S}_2} p(x_{\mathcal{S}_1}; x_{\mathcal{S}_2})$ *and* $\mathcal{M}_{k\cup\mathcal{S}_2} p(x_k; x_{\mathcal{S}_2})$ *have the same row space if* $k \in \mathcal{S}_1$.

Proof Define a conditional probability tensor as follows:

$$\mathcal{M}_{\mathcal{S}_1|\mathcal{S}_2} p(x_{\mathcal{S}_1}, x_{\mathcal{S}_2}) := \mathbb{P}_{X\sim p}\left[X_{\mathcal{S}_1} = x_{\mathcal{S}_1} | X_{\mathcal{S}_2} = x_{\mathcal{S}_2}\right].$$

Due to the conditional independence property for graphical models, one can write

$$\mathcal{M}_{\mathcal{S}_1\cup\mathcal{S}_2} p(x_{\mathcal{S}_1}, x_{\mathcal{S}_2}) = \mathcal{M}_{\mathcal{S}_1|\mathcal{P}(k)} p(x_{\mathcal{S}_1}, x_{\mathcal{P}(k)}) \mathcal{M}_{\mathcal{P}(k)} p(x_{\mathcal{P}(k)}) \mathcal{M}_{\mathcal{S}_2|\mathcal{P}(k)} p(x_{\mathcal{S}_2\setminus\mathcal{P}(k)}, x_{\mathcal{P}(k)})$$

$$= \mathcal{M}_{\mathcal{S}_1\cup\mathcal{P}(k)}p(x_{\mathcal{S}_1}, x_{\mathcal{P}(k)})\mathcal{M}_{\mathcal{S}_2|\mathcal{P}(k)}p(x_{\mathcal{S}_2\setminus\mathcal{P}(k)}, x_{\mathcal{P}(k)})$$

Thus, the column space of $\mathcal{M}_{\mathcal{S}_1\cup\mathcal{S}_2}p(x_{\mathcal{S}_1}; x_{\mathcal{S}_2})$ depends solely on $\mathcal{M}_{\mathcal{S}_1\cup\mathcal{P}(k)}(x_{\mathcal{S}_1}; x_{\mathcal{P}(k)})$. Therefore, (i) holds.

Similarly,

$$\mathcal{M}_{\mathcal{S}_1\cup\mathcal{S}_2}p(x_{\mathcal{S}_1}, x_{\mathcal{S}_2}) = \mathcal{M}_{\mathcal{S}_1|k}p(x_{\mathcal{S}_1\setminus k}, x_k)\mathcal{M}_{k\cup\mathcal{S}_2}p(x_k, x_{\mathcal{S}_2}),$$

which shows that the row space of $\mathcal{M}_{\mathcal{S}_1\cup\mathcal{S}_2}p(x_{\mathcal{S}_1}; x_{\mathcal{S}_2})$ depends solely on $\mathcal{M}_{k\cup\mathcal{S}_2}p(x_k; x_{\mathcal{S}_2})$. Therefore, (ii) holds. □

Proof (of Lemma 13) We will verify that Condition 3 holds. The Markov sketch function is quite special, and we often refer to a concept of natural identification. To make this concept rigorous, if two matrices $A(x; y)$ and $A'(z; w)$ are said to have a *natural identification*, it then means that $A(x; y) = A'(z; w)$ entry-wise as matrices. In particular, if one has a natural identification $A(x; y) = A'(z; w)$, then A and A' share column space, row space, and rank.

By the property of the right sketch function in the Markov sketch function, one has the natural identification $\bar{\Phi}^\star_k(x_{\mathcal{L}(k)\cup k}; \gamma_{(k,\mathcal{P}(k))}) = \mathcal{M}_{\mathcal{L}(k)\cup k\cup\mathcal{P}(k)}p^\star(x_{\mathcal{L}(k)\cup k}; x_{\mathcal{P}(k)})$. Lemma 17 then shows that the column space of $\bar{\Phi}^\star_k(x_{\mathcal{L}(k)\cup k}; \gamma_{(k,\mathcal{P}(k))})$ equals to that of $p^\star(x_{\mathcal{L}(k)\cup k}; x_{\mathcal{R}(k)})$. By Condition 1, the column space of $p^\star(x_{\mathcal{L}(k)\cup k}; x_{\mathcal{R}(k)})$ equals to that of any $\Phi^\Delta_{(w,k)}(x_{\mathcal{L}(k)\cup k}; \alpha_{(k,\mathcal{P}(k))})$, and so (i) holds.

Similarly, due to the Markov sketch function, one has the natural identification

$$Z^\star_k(\beta_{(k,\mathcal{C}(k))}, x_k; \gamma_{(k,\mathcal{P}(k))}) = \mathcal{M}_{\mathcal{C}(k)\cup k\cup\mathcal{P}(k)}p(x_{\mathcal{C}(k)\cup k}; x_{\mathcal{P}(k)}).$$

By Lemma 17 and the natural identification of $Z^\star_k$, for any every non-leaf and non-root k, it follows that $Z^\star_k$ has the same row space as that of

$$\bar{\Phi}^\star_k(x_{\mathcal{L}(k)\cup k}; \gamma_{(k,\mathcal{P}(k))}) = \mathcal{M}_{\mathcal{L}(k)\cup k\cup\mathcal{P}(k)}p^\star(x_{\mathcal{L}(k)\cup k}; x_{\mathcal{P}(k)}).$$

Hence, the rank of $Z^\star_k$ equals to the rank of $\bar{\Phi}^\star_k$. By Lemma 17, the column space of $\bar{\Phi}^\star_k$ equals to the column space of $p^\star(x_{\mathcal{L}(k)\cup k}; x_{\mathcal{R}(k)})$. Thus, the rank of $Z^\star_k$ equals to $r_{(k,\mathcal{P}(k))}$ and (ii) holds.

Because (i) and (ii) hold, the proof of Theorem 9 actually shows that there exists a gauge $\{\Phi^\star_{(w,k)}\}_{(w,k)\in E}$ which satisfies Condition 2. To verify (iii), it suffices to check (iii) for the gauge $\{\Phi^\star_{(w,k)}\}_{(w,k)\in E}$ because $A^\star_k$ having full column rank leads to any A^Δ_k having full column rank.

Moreover, it suffices to show that each $A^\star_{w\to k}(\beta_{(w,k)}; \alpha_{(w,k)})$ has full column rank of $r_{w\to k}$. If this holds, then it follows that $A^\star_k = \bigotimes_{w\in\mathcal{C}(k)} A^\star_{w\to k}$ has full column rank of $\prod_{w\in\mathcal{C}(k)} r_{w\to k}$. By the SVD step in SystemForming, recall that the column space of $Q^\star_w(\gamma_{(w,k)}; \alpha_{(w,k)})$ is the same as the column space of $\left(Z^\star_w\right)^\top(\gamma_{(w,k)}; \beta_{(w,\mathcal{C}(w))}, x_w)$. By the natural identification of

$$\left(Z^\star_w\right)^\top(\gamma_{(w,k)}; x_w, \beta_{(w,\mathcal{C}(w))}) = \mathcal{M}_{k\cup w\cup\mathcal{C}(w)}p(x_k; x_{w\cup\mathcal{C}(w)}),$$

we know that the column space of $Q^\star_w(\gamma_{(w,k)}; \alpha_{(w,k)})$ is the same as that of $\mathcal{M}_{k\cup w\cup\mathcal{C}(w)}p(x_k; x_{w\cup\mathcal{C}(w)})$. By Lemma 17, it then follows that the column space of $Q^\star_w(\gamma_{(w,k)}; \alpha_{(w,k)})$ coincides with that of $\mathcal{M}_{k\cup w}p(x_k; x_w)$.

Moreover, $Z^\star_{w\to k}$ has the natural identification $Z^\star_{w\to k}(\beta_{(w,k)}; \gamma_{(w,k)}) = \mathcal{M}_{w\cup k}p(x_w; x_k)$, and so the column space of $Z^\star_{w\to k}(\beta_{(w,k)}; \gamma_{(w,k)})$ coincides with that of $\mathcal{M}_{w\cup k}p(x_w; x_k)$.

By (17), one has

$$A^{\star}_{w\to k}(\beta_{(w,k)};\alpha_{(w,k)}) = Z^{\star}_{w\to k}(\beta_{(w,k)};\gamma_{(w,k)})Q^{\star}_{w}(\gamma_{(w,k)};\alpha_{(w,k)}),$$

and so the column space of $A^{\star}_{w\to k}$ coincides with that of

$$\mathcal{M}_{w\cup k}p(x_w;x_k)\mathcal{M}_{k\cup w}p(x_k;x_w) = \mathcal{M}_{w\cup k}p(x_w;x_k)\left(\mathcal{M}_{w\cup k}p(x_w;x_k)\right)^{\top}.$$

Thus, the rank of $A^{\star}_{w\to k}$ coincides with that of $\mathcal{M}_{w\cup k}p(x_w;x_k)\left(\mathcal{M}_{w\cup k}p(x_w;x_k)\right)^{\top}$, which in turn coincides with the rank of $\mathcal{M}_{w\cup k}p(x_w;x_k)$. By applying Lemma 17, the rank of $\mathcal{M}_{w\cup k}p(x_w;x_k)$ equals to $r_{(w,k)}$, and so (iii) holds.

□

Appendix D Proof of Theorem 14

After applying the left and right sketching, one has the following form on $Z^{\star}_{k}$:

$$Z^{\star}_{k}(x_k,\beta_{(k,\mathcal{N}(k))}) = \sum_{\substack{\beta_e\\ k\notin e}}\sum_{\substack{x_i\\ i\neq k}} p^{\star}(x_1,\ldots,x_d)\prod_{i\neq k} s_i(x_i,\beta_{(i,\mathcal{N}(i))}). \tag{52}$$

Let $\mathcal{S}_k := [d]-\{k\}$, and let $T_{\mathcal{S}_k}$ be the subgraph of T with vertex set being $\mathcal{S}_k$. Using the definition of subgraph TTNS function in Definition 11, define a tensor

$$H_k\colon \prod_{i\in[d],i\neq k}[n_i]\times\prod_{w\in\mathcal{N}(k)}[\beta_{(w,k)}]\to\mathbb{R}$$

as the subgraph TTNS function over $\{s_i\}_{i\neq k}$ and $T_{\mathcal{S}_k}$, i.e.

$$H_k(x_{[d]-\{k\}},\beta_{(k,\mathcal{N}(k))}) = \sum_{\substack{\alpha_e\\ k\notin e}}\prod_{i\neq k} s_i\left(x_i,\beta_{(i,\mathcal{N}(i))}\right). \tag{53}$$

Then (52) is equivalent to the following equation:

$$Z^{\star}_{k}(x_k,\beta_{(k,\mathcal{N}(k))}) = \sum_{\substack{x_w\\ w\neq k}} p^{\star}(x_1,\ldots,x_d)H_k(x_{[d]-\{k\}},\beta_{(k,\mathcal{N}(k))}). \tag{54}$$

From (53), one sees that H_k is multi-linear in $\{s_i\}_{i\neq k}$. We thus can apply the binomial theorem to derive a structural form on H_k as a sum of secondary terms. To do so, let $\mathcal{S}$ be an arbitrary subset of $\mathcal{S}_k$, and define a tensor

$$H_{k;\mathcal{S}}\colon \prod_{i\in[d],i\neq k}[n_i]\times\prod_{w\in\mathcal{N}(k)}[\beta_{(w,k)}]\to\mathbb{R}$$

as the subgraph TTNS function over $T_{\mathcal{S}_k}$ and $\{\Delta_i\}_{i\in\mathcal{S}_k}\cup\{O_j\}_{j\in\mathcal{S}_k-\mathcal{S}}$, i.e.

$$H_{k;\mathcal{S}}(x_{[d]-\{k\}},\beta_{(k,\mathcal{N}(k))}) = \sum_{\substack{\alpha_e\\ k\notin e}}\prod_{i\in\mathcal{S}}\Delta_i\left(x_i,\beta_{(i,\mathcal{N}(i))}\right)\prod_{j\in\mathcal{S}_k-\mathcal{S}} O_j\left(x_j,\beta_{(j,\mathcal{N}(j))}\right).$$

We now use the fact that $O_j\left(x_j,\beta_{(j,\mathcal{N}(j))}\right)=1$ in Condition 6, and so

$$H_{k;\mathcal{S}}(x_{[d]-\{k\}},\beta_{(k,\mathcal{N}(k))}) = \sum_{\substack{\alpha_e\\ k\notin e}}\prod_{i\in\mathcal{S}}\Delta_i\left(x_i,\beta_{(i,\mathcal{N}(i))}\right) = \sum_{\beta_e,k\notin e}\Delta_{\mathcal{S}}(x_{\mathcal{S}},\beta_{\partial\mathcal{S}}), \tag{55}$$

where the second equality follows from the Definition of $\Delta_{\mathcal{S}}$ in (37).

By applying the binomial theorem over the fact that $s_i = \epsilon\Delta_i + O_i$, one sees that H_k is a sum of 2^{d-1} terms, each of which formed by corresponding to one $H_{k;\mathcal{S}}$, i.e.

$$H_k(x_{[d]-\{k\}}, \beta_{(k,\mathcal{N}(k))}) = \sum_{l=0}^{d-1} \epsilon^l \sum_{\mathcal{S}\subset[d]-\{k\},|\mathcal{S}|=l} H_{k;\mathcal{S}}(x_{[d]-\{k\}}, \beta_{(k,\mathcal{N}(k))}). \tag{56}$$

Define $Z^{\star}_{k;\mathcal{S}}$ as the following tensor:

$$Z^{\star}_k(x_k, \beta_{(k,\mathcal{N}(k))}) := \sum_{\substack{x_w \\ w\neq k}} p^{\star}(x_1, \ldots, x_d) H_{k;\mathcal{S}}(x_{[d]-\{k\}}, \beta_{(k,\mathcal{N}(k))}). \tag{57}$$

The proof that $Z^{\star}_{k;\mathcal{S}}$ satisfies (39) is a simple result of exchanging summation order:

$$\begin{aligned}
&\sum_{\substack{x_w \\ w\neq k}} p^{\star}(x_1, \ldots, x_d) H_{k;\mathcal{S}}(x_{[d]-\{k\}}, \beta_{(k,\mathcal{N}(k))}) \\
&= \sum_{\substack{x_w \\ w\in\mathcal{S}}} \left(\sum_{\substack{x_w \\ w\in\mathcal{S}_k-\mathcal{S}}} p^{\star}(x_1, \ldots, x_d) \right) \left(\sum_{\beta_e, k\notin e} \Delta_{\mathcal{S}}(x_{\mathcal{S}}, \beta_{\partial\mathcal{S}}) \right) \\
&= \sum_{\beta_e, k\notin e} \left(\sum_{x_{\mathcal{S}}} \mathcal{M}_{\mathcal{S}\cup\{k\}} p^{\star}(x_k, x_{\mathcal{S}}) \Delta_{\mathcal{S}}(x_{\mathcal{S}}, \beta_{\partial\mathcal{S}}) \right).
\end{aligned}$$

Due to the linear relationship between H_k and $Z^{\star}_k$ in (54), it follows that the structural form of H_k in (56) leads to the structural form for $Z^{\star}_k$ in (38), as desired.

Appendix E Sample complexity bound of TTNS-Sketch

This section gives an upper bound for the sample complexity of TTNS-Sketch when the sketch functions satisfies recursive sketching in the sense of Condition 4. This setting is considered because one can use the alternative definition of A_k in Proposition 28, which simplifies the analysis. As an application, we obtain a sample complexity bound to the simple case where $p^{\star}$ is a graphical model over a tree T, and the sketching function is the Markov sketch function.

We give a summary of organization of this section. In Sect. 8, we introduce notations and conventions which are important for sample complexity analysis. In Sect. 8, we prove small perturbations of the cores lead to small perturbations of the obtained TTNS ansatz. In Sect. 8, we prove that small error in the estimator $\hat{Z}_k$ leads to small perturbation of the cores, leading to an upper bound for the sample complexity of TTNS-Sketch in Theorem 34. In Sect. 8, we give a proof of all the lemmas and corollaries. In Sect. 8, we remark how our derived results can be extended sample complexity bounds for total variation distance.

E.1 Preliminaries

In what follows, for a given vector v, let $\|v\|$ and $\|v\|_{\infty}$ denote its Euclidean norm and its supremum norm, respectively. For any matrix M, denote its spectral norm, Frobenius norm, and the r-th singular value by $\|M\|$, $\|M\|_F$, and $\sigma_r(M)$, respectively. Also, for a generic tensor p, let $\|p\|_{\infty}$ denote the largest absolute value of the entries of p. Lastly, the orthogonal group in dimension r is denoted by $\mathrm{O}(r)$.

A mathematical structure important in this section is 3-tensors. Similar to unfolding matrix in Definition 6, the 3-tensors we typically use come from viewing high-dimensional tensors in terms of 3-tensors by grouping joint variables:

Definition 18 (Unfolding 3-tensor Notation) For a generic d-dimensional tensor $p\colon [n_1]\times\cdots\times[n_d]\to\mathbb{R}$ and for three disjoint subsets $\mathcal{U},\mathcal{V},\mathcal{W}$ with $\mathcal{U}\cup\mathcal{V}\cup\mathcal{W}=[d]$, we define the corresponding *unfolding 3-tensor* by $p(x_{\mathcal{U}};x_{\mathcal{V}};x_{\mathcal{W}})$. The 3-tensor $p(x_{\mathcal{U}};x_{\mathcal{V}};x_{\mathcal{W}})$ is of size $\left[\prod_{i\in\mathcal{U}}n_i\right]\times\left[\prod_{j\in\mathcal{V}}n_j\right]\times\left[\prod_{k\in\mathcal{W}}n_k\right]\to\mathbb{R}$.

It is helpful to introduce a slice of the 3-tensor. In our convention, we only need to consider taking slice at the second component:

Definition 19 (Middle index slice of 3-tensor) For any 3-tensor $G\colon [r_1]\times[n_1]\times[r_2]\to\mathbb{R}$, we use $G(\cdot,x,\cdot)\colon [r_1]\times[r_2]\to\mathbb{R}$ to denote an $r_1\times r_2$ matrix obtained by fixing the second slot of G to be x.

In Definition 20- 22, we introduce a new norm and two operations for 3-tensors.

Definition 20 ($|||\cdot|||$ norm for 3-tensors) Define the norm $|||G|||$ by

$$|||G||| := \max_{x\in[n_1]}\|G(\cdot,x,\cdot)\|. \tag{58}$$

Definition 21 (contraction operator for 3-tensors) Let $G\colon [r_1]\times[n_1]\times[r_2]\to\mathbb{R}$, $G'\colon [r_3]\times[n_2]\times[r_4]\to\mathbb{R}$ be two 3-tensors. Under the assumption $r_2=r_3$, define the 3-tensor $G\circ G'\colon [r_1]\times[n_1\times n_2]\times[r_4]\to\mathbb{R}$ by

$$G\circ G'(\alpha;(x,y);\gamma)=\sum_{\beta\in[r_2]}G(\alpha,x,\beta)G'(\beta,y,\gamma). \tag{59}$$

Definition 22 (tensor product operator for 3-tensors) Let $G\colon [r_1]\times[n_1]\times[r_2]\to\mathbb{R}$, $G'\colon [r_3]\times[n_2]\times[r_4]\to\mathbb{R}$ be two 3-tensors. Define the 3-tensor $G\otimes G'\colon [r_1\times r_3]\times[n_1\times n_2]\times[r_2\times r_4]\to\mathbb{R}$ by

$$G\otimes G'((\alpha,\beta);(x,y);(\gamma,\theta))=G(\alpha,x,\beta)G'(\gamma,y,\theta). \tag{60}$$

We summarize the simple properties of the defined operation in Lemma 23, which will be useful for our derivations:

Lemma 23 *The following results hold:*

(i) Associativity of $\circ$ holds:

$$(G\circ G')\circ G''=G\circ(G'\circ G''). \tag{61}$$

(ii) Associativity of $\otimes$ holds:

$$(G\otimes G')\otimes G''=G\otimes(G'\otimes G''). \tag{62}$$

(iii) Inequality of $\circ$ under $|||\cdot|||$ norm:

$$|||G\circ G'|||\le|||G|||\cdot|||G'||| \tag{63}$$

(iv) Equality of $\otimes$ under $|||\cdot|||$ norm:

$$|||G \otimes G'||| = |||G||| \cdot |||G'||| \tag{64}$$

(v) For a three tensor $G(\alpha, x, \beta)\colon [r_1]\times[n]\times[r_2] \to \mathbb{R}$, denote $G(\alpha, x; \beta)\colon [r_1 n]\times[r_2] \to \mathbb{R}$ as the unfolding matrix by grouping the first and second index of G. One has

$$|||G||| \le \|G(\alpha, x; \beta)\| \le n|||G||| \tag{65}$$

As a consequence of associativity, given any collection of 3-tensors $\{G_i\}_{i=1}^d$, one can define $G_1 \otimes G_2 \otimes \ldots \otimes G_d$. Moreover, if the collection is such that the size of the third index of G_i coincides with the first index of G_{i+1}, then one can naturally define the 3-tensor $G_1 \circ G_2 \circ \ldots \circ G_d$.

E.2 3-tensor structure for TTNS

For cleaner analysis, one often gives unfolding matrices a 3-tensor structure:

Definition 24 (3-tensor structure for unfolding matrix) Consider a generic D-dimensional tensor $f\colon [n_1]\times\cdots\times[n_D] \to \mathbb{R}$. Moreover, suppose one picks a disjoint union $\mathcal{U}\cup\mathcal{V} = [D]$ and forms an unfolding matrix $f(x_\mathcal{U}; x_\mathcal{V})$ in the sense of Definition 6. Define $f(x_\mathcal{U}; 1; x_\mathcal{V})$ as the 3-tensor of size $\left[\prod_{i\in\mathcal{U}} n_i\right] \times \{1\} \times \left[\prod_{j\in\mathcal{V}} n_j\right] \to \mathbb{R}$. One likewise defines 3-tensor structure of $f(1; x_\mathcal{U}; x_\mathcal{V})$, whose first index is of size 1, and $f(x_\mathcal{U}; x_\mathcal{V}; 1)$, whose third index is of size 1.

A tensor core from a TTNS ansatz has a default 3-tensor view, whereby the indices are grouped according to tree topology:

Definition 25 (3-tensor structure for TTNS tensor cores) Suppose a tensor p is defined by a collection of tensor cores $\{G_i\}_{i=1}^d$ in the sense of Definition 5.

If k is neither a root node nor a leaf node, then G_k is viewed with the 3-tensor unfolding structure

$$G_k(\alpha_{(k,\mathcal{C}(k))}; x_k; \alpha_{(k,\mathcal{P}(k))})\colon \left[\prod_{w\in\mathcal{C}(k)} r_{(w,k)}\right] \times [n_k] \times [r_{(k,\mathcal{P}(k))}] \to \mathbb{R}.$$

If k is a leaf node, then G_k is viewed with the 3-tensor unfolding structure

$$G_k(1; x_k; \alpha_{(k,\mathcal{P}(k))})\colon \{1\} \times [n_k] \times [r_{(k,\mathcal{P}(k))}] \to \mathbb{R}.$$

If k is the root node, then G_k is viewed with the 3-tensor unfolding structure

$$G_k(\alpha_{(k,\mathcal{C}(k))}; x_k; 1)\colon \left[\prod_{w\in\mathcal{C}(k)} r_{(w,k)}\right] \times [n_k] \times \{1\} \to \mathbb{R}.$$

With this design of norms, one can prove the following result by simple algebra:

Lemma 26 *Suppose a generic tensor $p\colon [n_1]\times\cdots\times[n_d] \to \mathbb{R}$ is defined by a collection of tensor cores $\{G_i\}_{i=1}^d$ in the sense of Definition 5. Moreover, suppose the tensor cores are viewed by 3-tensor structures as in Definition 25. Then*

$$\|p\|_\infty \le \prod_{i=1}^d |||G_i|||.$$

Using Lemma 26, one can bound global errors by errors in tensor cores:

Lemma 27 *In Lemma 26, let ΔG_k be a perturbation of G_k. Define a tensor $p'\colon [n_1]\times\cdots\times[n_d]\to\mathbb{R}$ by tensor cores $\{G_k+\Delta G_k\}_{k=1}^d$ in the sense of Definition 5, with the tree topology T and the internal rank $\{r_e\}_{e\in E}$ the same as that of p. Suppose $|||\Delta G_k||| \le \delta_k |||G_k|||$ for all $k\in[d]$, and set $\Delta p := p' - p$. Then,*

$$\|\Delta p\|_\infty \le |||G_1|||\cdots|||G_d||| \left(\sum_{i=1}^d \delta_i\right) \exp\left(\sum_{i=1}^d \delta_i\right).$$

If $\max_{k\in[d]}\delta_k \le \epsilon/(3d)$ for some fixed $\epsilon\in(0,1)$,

$$\frac{\|\Delta p\|_\infty}{|||G_1|||\cdots|||G_d|||} \le \epsilon. \tag{66}$$

E.3 Derivation for sample complexity of TTNS-Sketch

We first give a lemma which bounds the perturbation of solutions of a linear equation $AX = B$, where in particular X, B are two 3-tensors viewed under an unfolding matrix. This result will be the main building block to form our subsequent error analysis:

Lemma 28 *Consider a matrix $A^\star(\beta,\alpha)\in\mathbb{R}^{l\times r}$ with $\operatorname{rank}(A^\star) = r \le l$ and a 3-tensor $B^\star(\beta,x,\gamma)\in\mathbb{R}^{l\times n\times m}$ with unfolding matrix structure $B^\star(\beta;(x,\gamma))\in\mathbb{R}^{l\times(nm)}$. Let $X^\star(\alpha,x,\gamma)\in\mathbb{R}^{r\times n\times m}$ be the 3-tensor with an unfolding matrix view $X^\star(\alpha;(x,\gamma))\in\mathbb{R}^{r\times(nm)}$ which uniquely solves the linear equation $A^\star X = B^\star$ in the sense of least squares:*

$$\sum_\alpha A^\star(\beta,\alpha)X(\alpha,(x,\gamma)) = B^\star(\beta,(x,\gamma)).$$

Moreover, let $\Delta B^\star\in\mathbb{R}^{l\times n\times m}$ be a perturbation of $B^\star$, and let $\Delta A^\star\in\mathbb{R}^{l\times n\times m}$ be a perturbation of $A^\star$ with $\|(A^\star)^\dagger\|\,\|\Delta A^\star\| < 1$ so that $\operatorname{rank}(A^\star+\Delta A^\star) = n$. Then, let $\Delta X^\star$ be a 3-tensor so that $X^\star+\Delta X^\star$ which uniquely solves the linear equation $(A^\star+\Delta A^\star)X = (B^\star+\Delta B^\star)$ in the sense of least squares:

$$\sum_\alpha (A^\star+\Delta A^\star)(\beta,\alpha)X(\alpha,(x,\gamma)) = (B^\star+\Delta B^\star)(\beta,(x,\gamma))$$

Under the unfolding matrix structure, suppose the column space of $B^\star$ is contained in that of $A^\star$, i.e. $X^\star$ solves $A^\star X = B^\star$ exactly, one has

$$|||\Delta X^\star||| \le \frac{\|(A^\star)^\dagger\|}{1-\|(A^\star)^\dagger\|\,\|\Delta A^\star\|}\left(\|\Delta A^\star\|\,|||X^\star||| + |||\Delta B^\star|||\right). \tag{67}$$

In particular, if $|||X^\star||| \ge \chi > 0$ for some constant χ, and $\Delta A^\star$ satisfies $\|(A^\star)^\dagger\|\,\|\Delta A^\star\| \le 1/2$, then

$$\frac{|||\Delta X^\star|||}{|||X^\star|||} \le 2\|\left(A^\star\right)^\dagger\|\,\left(\|\Delta A^\star\| + \chi^{-1}|||\Delta B^\star|||\right). \tag{68}$$

For our use case of Lemma 28, the coefficient matrix is viewed as a Kronecker product of some smaller matrices, e.g. $A_k^\star$ is formed by $\{A^\star_{w\to k}\}_{w\in\mathcal{C}(k)}$. One can bound the $\|\Delta A^\star\|$ in this case, as the following lemma shows:

Lemma 29 *Consider a collection of matrices* $\{E_i, C_i\}_{i\in[n]}$ *such that* E_i, C_i *are of the same shape. Moreover, let* $\|C_i\| \le 1$, $\|E_i\| \le \delta_i$. *Then*

$$\left\| \bigotimes_{i=1}^{n} (C_i + E_i) - \bigotimes_{i=1}^{n} C_i \right\| \le \left(\sum_{i=1}^{n} \delta_i \right) \exp\left(\sum_{i=1}^{n} \delta_i \right).$$

Lemma 28 and 29 leads to the proof strategy for obtaining sample complexity. With the particular perturbation $\Delta p^\star := \hat{p} - p^\star$, the terms $\hat{A}_k$ and $\hat{B}_k$ from the Algorithm 1 satisfies $B_k^\star + \Delta B_k^\star = \hat{B}_k$ and $A_k^\star + \Delta A_k^\star = \hat{A}_k$. The least-squares solution $G_k^\star + \Delta G_k^\star$ to the perturbed equation is thus the actual output $\hat{G}_k$ from Algorithm 1.

However, due to the SVD step that is involved in obtaining $\hat{A}_k$ and $\hat{B}_k$, one can only bound the sample estimation error in terms of the following alternative metric:

Definition 30 (dist$(\cdot,\cdot)$ operator for matrices) For any matrices $B, B^\star \in \mathbb{R}^{n\times m}$, define

$$\mathrm{dist}(B, B^\star) := \min_{R\in \mathrm{O}(m)} \|B - B^\star R\|.$$

In other words, the finite sample estimate of $\hat{A}_k, \hat{B}_k$ could be closer to a rotation of $A_k^\star, B_k^\star$, which we will denote as A_k°, B_k°. An error bound for of the type dist$(B, B^\star)$ exists through Wedin theorem, and thus the magnitude of $B_k^\circ - \hat{B}_k$ can be bounded, and $A_k^\circ - \hat{A}_k$ is bounded via (28). In Corollary 31- 32, we write out relaxed version of Wedin theorem and the Matrix Bernstein inequality, which we will use to analyze error terms of the type $\|\Delta Z_k^\star\|$, $\|\Delta B_k^\circ\|$, respectively. As a summary of all previous result, in Theorem 33, we form a quite technical proof bounding the error on the rotated cores by the sample estimation error in sketching. In Theorem 34, we form the sample complexity of TTNS-Sketch.

Corollary 31 *(Corollary to Wedin theorem, cf. Theorem 2.9 in [5]) Let* $Z^\star \in \mathbb{R}^{n\times m}$ *be a matrix of rank* r *and* $\Delta Z^\star \in \mathbb{R}^{n\times m}$ *be its perturbation with* $Z := Z^\star + \Delta Z^\star$. *Moreover, let* $B^\star, B \in \mathbb{R}^{n\times r}$ *respectively be the first* r *left singular vectors of* $Z^\star, Z$. *If* $\|\Delta Z^\star\| \le (1 - 1/\sqrt{2})\sigma_r(Z^\star)$, *then*

$$\mathrm{dist}(B, B^\star) \le \frac{2\|\Delta Z^\star\|}{\sigma_r(Z^\star)}$$

Corollary 32 *(Corollary to Matrix Bernstein inequality, cf. Corollary 6.2.1 in [29]) Let* $Z^\star \in \mathbb{R}^{n\times m}$ *be a matrix, and let* $\{Z^{(i)} \in \mathbb{R}^{n\times m}\}_{i=1}^N$ *be a sequence of i.i.d. matrices with* $\mathbb{E}\left[Z^{(i)}\right] = Z^\star$. *Denote* $\hat{Z} = \frac{1}{N}\sum_{i=1}^N Z^{(i)}$ *and* $\Delta Z^\star = \hat{Z} - Z^\star$. *Let the distribution of* $Z^{(i)}$ *be such that there exists a constant* L *with* $||Z^{(i)}|| \le L$.

Let $\gamma := \max\left(\left\| \mathbb{E}\left[Z^{(i)} \left(Z^{(i)}\right)^\top \right] \right\|, \left\| \mathbb{E}\left[\left(Z^{(i)}\right)^\top Z^{(i)} \right] \right\| \right)$, *and then*

$$\mathbb{P}\left[\|\Delta Z^\star\| \ge t \right] \le (m+n) \exp\left(\frac{-Nt^2/2}{\gamma + 2Lt/3} \right).$$

Using Jensen's inequality, one has $\gamma \le L^2$, *and*

$$\mathbb{P}\left[\|\Delta Z^\star\| \ge t \right] \le (m+n) \exp\left(\frac{-Nt^2/2}{L^2 + 2Lt/3} \right). \tag{69}$$

Theorem 33 *(Error bound over TTNS tensor cores) Let $p^\star\colon [n_1]\times\cdots\times[n_d]\to\mathbb{R}$ be a density function satisfy the TTNS assumption in Condition 1. Fix a sketch function $\{T_i, S_i\}_{i=1}^d$ which satisfies the recursive sketching assumption in Condition 4. Let $\{A_i^\star, B_i^\star, G_i^\star, Z_i^\star\}_{i=1}^d$ be as in Theorem 9. Moreover, let $\{\hat{A}_i, \hat{B}_i, \hat{G}_i, \hat{Z}_i\}_{i=1}^d$ be as in Algorithm 1 with $\hat{p}$ as input. Suppose further that for some fixed $\delta\in(0,1)$, one has*

$$\|Z_k^\star - \hat{Z}_k\| \le \zeta_k\delta, \tag{70}$$

where ζ_k is defined by a series of constants as follows:

$$\zeta_k := \left(6\frac{c_{\mathcal{C}}}{c_{k;Z}}\right)^{-1}\xi, \quad \xi := 1\wedge\min_{i\in[d]}\left(2c_{i;A}\left(c_{i;S}+c_{i;G}\right)\right)^{-1}, \tag{71}$$

and the constants are defined as follows:

- $c_{\mathcal{C}} = \max_{i\in[d]}|\mathcal{C}(i)|$,
- $c_{k;Z} = 1$ *when* $k =$ *root, and* $c_{k;Z} = \sigma_{r_{(k,\mathcal{P}(k))}}(Z_k^\star(\beta_{(k,\mathcal{C}(k))}, x_k; \gamma_{(k,\mathcal{P}(k))}))$ *otherwise.*
- $c_{k;G} = 1/\|\!|G_k^\star|\!\|$,
- $c_{k;A} = 1$ *when* $k =$ *leaf, and* $c_{k;A} = \|\left(A_k^\star\right)^\dagger\|$ *otherwise,*
- $c_{k;S} = 1$ *when* $k =$ *leaf, and* $c_{k;S} = \prod_{w\in\mathcal{C}(k)}||s_w(\beta_{(w,\mathcal{P}(w))};\beta_{(w,\mathcal{C}(w))}, x_w)||$ *otherwise.*

Then, there exists a TTNS tensor core $\{G_i^\circ\}_{i=1}^d$ for $p^\star$ in the sense of Definition 5, such that $\|\!|G_i^\circ|\!\| = \|\!|G_i^\star|\!\|$, and the following holds:

$$\frac{\|\!|\hat{G}_k - G_k^\circ|\!\|}{\|\!|G_k^\circ|\!\|} \le \delta. \tag{72}$$

We defer the proof of Theorem 33 to the end of this subsection. As a direct application, one obtain the sample complexity of TTNS-Sketch:

Theorem 34 *(Sample Complexity of TTNS-Sketch) Assume the setting and notation of Theorem 33. Let $\hat{p}_{\mathrm{TS}}$ denote the TTNS tensor formed by the TTNS tensor core $\{\hat{G}_i\}_{i=1}^d$. In particular, $\{\hat{G}_i\}_{i=1}^d$ is the output of Algorithm 1 with the empirical distribution $\hat{p}$ formed by N i.i.d. samples $(y_1^{(i)},\ldots,y_d^{(i)})_{i=1}^N$. Let $Z_k^{(i)}$ be the i-th sample estimate of $Z_k^\star$, i.e.*

$$Z_k^{(i)}(\beta_{(k,\mathcal{C}(k))}, x_k, \gamma_{(k,\mathcal{P}(k))}) := S_k(\beta_{(k,\mathcal{C}(k))}, y_{\mathcal{L}(k)}^{(i)})\mathbf{1}(y_k^{(i)} = x_k)T_k(y_{\mathcal{R}(k)}^{(i)}, \gamma_{(k,\mathcal{P}(k))}),$$

and set L_k as an upper bound of $\|Z_k^{(i)}(\beta_{(k,\mathcal{C}(k))}, x_k; \gamma_{(k,\mathcal{P}(k))})\|$. Define $L = \max_{k\in[d]}L_k$.

For $\eta\in(0,1)$ and $\epsilon\in(0,1)$, suppose

$$N \ge \frac{18L^2d^2 + 4L\epsilon\zeta d}{\zeta^2\epsilon^2}\log\left(\frac{(ln+m)d}{\eta}\right),$$

and the constants are defined as follows:

- $\zeta = \max_{k\in[d]}\zeta_k$, *with ζ_k as in Theorem 33.*
- $l = \max_{k\in[d]}l_k$, *where* $l_k = \prod_{w\in\mathcal{C}(k)}l_{(w,k)}$
- $m = \max_{k\in[d]}m_k$, *where* $m_k = m_{(k,\mathcal{P}(k))}$.
- $n = \max_{k\in[d]}n_k$.

Then with probability at least $1-\eta$ *one has*

$$\frac{\|\hat{p}_{\mathrm{TS}}-p^\star\|_\infty}{|||G_1^\star|||\cdots|||G_d^\star|||}\le\epsilon. \tag{73}$$

Proof (of Theorem 34)

Suppose that the inequality (70) holds with $\delta=\frac{\epsilon}{3d}$. In the setting of Theorem 33, note that $p^\star$ is formed by $\{G_i^\circ\}_{i=1}^d$, and $\hat{p}_{\mathrm{TS}}$ is formed by $\{G_i^\circ+\Delta G_i^\circ\}_{i=1}^d$ with $|||\Delta G_i^\circ|||\le\frac{\epsilon}{3d}|||G_i^\circ|||$. Moreover, one has $|||G_i^\star|||=|||G_i^\circ|||$. Applying (66) in Lemma 27, one thus has

$$\frac{\|\hat{p}_{\mathrm{TS}}-p^\star\|_\infty}{|||G_1^\star|||\cdots|||G_d^\star|||}=\frac{\|\hat{p}_{\mathrm{TS}}-p^\star\|_\infty}{|||G_1^\circ|||\cdots|||G_d^\circ|||}\le\epsilon.$$

By a simple union bound argument, it suffices to find a sample size that (70) is guaranteed for each individual $k\in[d]$ with $\delta=\frac{\epsilon}{3d}$ and with probability $1-\frac{\eta}{d}$. We apply (69) in Corollary 32, where $Z_k^\star$ is a matrix of size $\mathbb{R}^{l_k n_k\times m_k}$. With the choice of (l, n, m, L) as set in the theorem statement, one has

$$\mathbb{P}\left[\|\Delta Z_k^\star\|\ge t\right]\le(ln+m)\exp\left(\frac{-Nt^2/2}{L^2+2Lt/3}\right).$$

It then suffices for one to find a lower bound for N so that for $t=\zeta\frac{\epsilon}{3d}$ one has

$$(ln+m)\exp\left(\frac{-Nt^2/2}{L^2+2Lt/3}\right)\le\eta/d.$$

By simple algebra, it suffices to lower bound N by the following quantity:

$$N\ge\frac{2L^2+4Lt/3}{t^2}\log\left(\frac{(ln+m)d}{\eta}\right)=\frac{18L^2d^2+4L\epsilon\zeta d}{\zeta^2\epsilon^2}\log\left(\frac{(ln+m)d}{\eta}\right).$$

□

As a corollary, for a Markov sketch function, note that each $Z_k^{(i)}$ is a tensor with one entry being of value one, the rest being zero. Under this setting, note that $\|Z_k^{(i)}\|\le\|Z_k^{(i)}\|_F=1$, and hence one can set $L=1$. Let $\Delta(T)$ denote the maximal degree of a tree T. One has $l\le n^{\Delta(T)-1}$ and $m=n\le ln$. Thus one obtains a sample complexity for TTNS-Sketch under Markov sketching:

Corollary 35 *(Sample Complexity of TTNS-Sketch for Markov Sketch function) Suppose that* $p^\star$ *is a graphical model over a tree* T*, with the sketching function being the Markov sketch function specified in Lemma 13. Suppose*

$$N\ge\frac{18d^2+4\epsilon\zeta d}{\zeta^2\epsilon^2}\log\left(\frac{2n^{\Delta(T)}d}{\eta}\right).$$

Then, with probability at least $1-\eta$*, one has*

$$\frac{\|\hat{p}_{\mathrm{TS}}-p^\star\|_\infty}{|||G_1^\star|||\cdots|||G_d^\star|||}\le\epsilon. \tag{74}$$

In the remainder of this subsection, we give the proof of Theorem 33, which is a culmination of all previous statements, the proof of which are of secondary interest and are included in Sect. 8. For some intuition of Theorem 33, the factors in ζ_k is set such that ξ can bound the sample estimation error of the sketched down core determining equation

in Algorithm 1. One then uses Lemma 28 to derive (72). As a sanity check of the defined constants, note that $\xi_i := (2c_{i;A}\,(c_{i;S}+c_{i;G}))^{-1}$ can be thought of as a homogeneous constant. That is, for any non-zero scaling constant $\{q_i\}_{i=1}^d$, changing the sketch cores from $\{s_i\}_{i=1}^d$ to $\{q_i s_i\}_{i=1}^d$ won't affect ξ_i, which is because the resultant multiplicative change to $\{c_{i;A}, c_{i;S}, c_{i;G}\}$ will be cancelled out in ξ_i. One can think of $\xi = 1 \wedge \min_i \xi_i$ in Theorem 33 as serving the role of condition number. Moreover, because $(Z_k^\star - \hat{Z}_k) \propto c_{k;Z}$ by definition, it follows the condition in (70) will not be affected if a scaling constant is applied to sketch cores.

Proof (of Theorem 33) Following the short-hand in Algorithm 3, for the joint variables we write $\beta_k \leftarrow \beta_{(k,\mathcal{C}(k))}, \gamma_k \leftarrow \gamma_{\mathcal{P}(k)}, \alpha_k \leftarrow \alpha_{(k,\mathcal{P}(k))}$, and for the bond dimensions we write $r_k \leftarrow r_{(\mathcal{P}(k),k)}, l_k \leftarrow \prod_{w\in\mathcal{C}(k)} l_{(w,k)}, m_k \leftarrow m_{(k,\mathcal{P}(k))}$. Moreover, if k is leaf, then we understand β_k as a joint variable taking value in $\{1\}$, and $l_k = 1$. Likewise, if k is root, then we understand α_k, γ_k respectively as a joint variable taking value in $\{1\}$, and $r_k = m_k = 1$. In this notation, when k is leaf or root, the joint variables sketch Z_k is conveniently written as $Z_k(\beta_k, x_k; \gamma_k)$.

For this proof, we will fix a canonical unfolding matrix structure for the tensors used. For $Z_k(\beta_k, x_k, \gamma_k)$ being one of $\{Z_k^\star, \hat{Z}_k, \Delta Z_k^\star\}$, we reshape it as $Z_k(\beta_k, x_k; \gamma_k)$. For $B_k(\beta_k, x_k, \alpha_k)$ being one of $\{B_k^\star, B_k^\circ, \hat{B}_k, \Delta B_k^\star, \Delta B_k^\circ\}$, we reshape it as $B_k(\beta_k, x_k; \alpha_k)$. For $A_k(\beta_k, \alpha_{(k,\mathcal{C}(k))})$ being one of $\{A_k^\star, A_k^\circ, \hat{A}_k, \Delta A_k^\star, \Delta A_k^\circ\}$, we reshape it as $A_k(\beta_k; \alpha_{(k,\mathcal{C}(k))})$. For $s_k(\beta_{(k,\mathcal{P}(k))}, \beta_k, x_k)$, we reshape it as $s_k(\beta_{(k,\mathcal{P}(k))}; \beta_k, x_k)$. For $G_k(\alpha_{(k,\mathcal{C}(k))}, x_k, \alpha_k)$ being one of $\{G_k^\star, G_k^\circ, \hat{G}_k, \Delta G_k^\star, \Delta G_k^\circ\}$, we reshape it as $G_k(\alpha_{(k,\mathcal{C}(k))}; x_k, \alpha_k)$.

Fix a non-root k, recall that $B_k^\star$ and $\hat{B}_k$ are the first r_k left singular vectors of $Z_k^\star$ and $\hat{Z}_k$, respectively. One applies Corollary 31: if $\|\Delta Z_k^\star\| \le (1 - 1/\sqrt{2})\sigma_{r_k}(Z_k^\star)$, then one can find $R_k \in \mathrm{O}(r_k)$ such that one can define $B_k^\circ := B_k^\star R_k$ so that

$$\hat{B}_k = B_k^\circ + \Delta B_k^\circ, \quad \|\Delta B_k^\circ\| \le \frac{2\|\Delta Z_k^\star\|}{\sigma_{r_k}(Z_k^\star)}$$

and by (v) in Lemma 23, one has

$$|||\Delta B_k^\circ||| \le \frac{2\|\Delta Z_k^\star\|}{\sigma_{r_k}(Z_k^\star)}. \tag{75}$$

Meanwhile, if k is the root, there is no SVD step. In this case, the perturbation $\Delta B_k^\star$ is simply $\Delta Z_k^\star$. For consistency, when k is the root, we set $B_k^\circ = B_k^\star$, and the corresponding perturbation ΔB_k° is just $\Delta Z_k^\star$.

In summary, B_k° is a rotation of $B_k^\star$, and $\hat{B}_k$ differs from B_k° by a perturbation ΔB_k°, for which one has an error bound. For a "rotated" version of $A_k^\star$, define

$$A_k^\circ(\beta_{(k,\mathcal{C}(k))}, \alpha_{(k,\mathcal{C}(k))}) := \prod_{w\in\mathcal{C}(k)} \sum_{(\beta_w, x_w)} s_w(\beta_{(w,k)}, \beta_w, x_w) B_w^\circ(\beta_w, x_w, \alpha_{(w,k)}) \tag{76}$$

Viewed in the unfolding matrix structure fixed in the beginning of proof, one can write $A_k^\circ = \bigotimes_{w\in\mathcal{C}(k)} s_w B_w^\circ$. Likewise, one has $\hat{A}_k = \bigotimes_{w\in\mathcal{C}(k)} s_w \hat{B}_w = \bigotimes_{w\in\mathcal{C}(k)} \left(s_w B_w^\circ + s_w \Delta B_w^\circ\right)$.

Now, with the chosen unfolding matrix structure, consider the following "rotated" versions of (21):

$$\begin{aligned} G_k^\circ &= B_k^\circ(\beta_k; x_k, \alpha_k) \quad \text{if } k \text{ is a leaf,} \\ A_k^\circ G_k^\circ &= B_k^\circ(\beta_k; x_k, \alpha_k) \quad \text{otherwise.} \end{aligned} \tag{77}$$

We will first prove that $\{G_i^\circ\}_{i=1}^d$ forms a TTNS tensor core for $p^\star$ in the sense of Definition 5. Suppose one has $\{\Phi^\star_{k\to\mathcal{P}(k)}\}_{k\neq \text{root}}$ defined according to Condition 2. Then, Theorem 9 proves that $\{G_i^\star\}_{i=1}^d$ solves the CDE (5) in Theorem 7 for the choice of gauge as $\{\Phi^\star_{k\to\mathcal{P}(k)}\}_{k\neq \text{root}}$. Now consider (5) for a rotated choice of gauge $\{\Phi^\circ_{k\to\mathcal{P}(k)} := \Phi^\star_{k\to\mathcal{P}(k)}R_k\}_{k\neq \text{root}}$. One can directly check that the sketched down equation coincides with (77), and Theorem 9 ensures that the solution $\{G_i^\circ\}_{i=1}^d$ is unique and forms a TTNS tensor core for $p^\star$.

Next, we prove that $|||G_k^\circ||| = |||G_k^\star|||$, with the 3-tensor view for G_k°, $G_k^\star$ as in Definition 25. As the coefficients A_k°'s and right-hand sides B_k°'s are simply the rotations of $A_k^\star$'s and $B_k^\star$'s of (21), one can verify that G_k° is a rotation of $G_k^\star$. If k is not leaf nor root, the equation for G_k° can be rewritten as

$$\left(\bigotimes_{w\in\mathcal{C}(k)} s_w B_w^\star R_w\right) G_k^\circ = B_k^\star R_k, \tag{78}$$

whereas the equation for $G_k^\star$ is

$$\left(\bigotimes_{w\in\mathcal{C}(k)} s_w B_w^\star\right) G_k^\star = B_k^\star.$$

For the rotation matrix $R_k(\alpha;\beta)\in \mathrm{O}(r_k)$, one gives it a 3-tensor view as $R_k(\alpha;1;\beta)$ in the sense of Definition 24. One can directly verify that the following equation for G_k° solves (78):

$$G_k^\circ = \left(\bigotimes_{w\in\mathcal{C}(k)} R_w^\top\right) \circ G_k^\star \circ R_k. \tag{79}$$

Likewise, $G_k^\circ = G_k^\star \circ R_k$ if k is leaf, and $G_k^\circ = \left(\bigotimes_{w\in\mathcal{C}(k)} R_w^\top\right) \circ G_k^\star$ if k is root. Then $|||G_k^\circ||| = |||G_k^\star|||$ as a consequence. The constructive form in (79) also gives a more intuitive sense of why $\{G_i^\circ\}_{i=1}^d$ forms a TTNS tensor core in the same way as $\{G_i^\star\}_{i=1}^d$. The reason is that each R_k and $R_k^\top$ comes in pairs, which does not change the formed TTNS tensor itself.

Next, we prove that, for $\Delta B_k^\circ := \hat{B}_k - B_k^\circ$ and $\Delta A_k^\circ := \hat{A}_k - A_k^\circ$, the assumption (70) leads to the the following bound:

$$\|\Delta B_k^\circ\| \le \xi\delta, \quad \|\Delta A_k^\circ\| \le c_{k;S}\xi\delta. \tag{80}$$

First, for ΔB_k°'s, we will derive a tighter bound

$$\|\Delta B_k^\circ\| \le \frac{\xi\delta}{3c_{\mathcal{C}}}, \tag{81}$$

which implies $\|\Delta B_k^\circ\| \le \xi\delta$ as $3c_{\mathcal{C}} \ge 1$. To see this, using (75), one has for any non-root k,

$$\|\Delta B_k^\circ\| \le \frac{2\|\Delta Z_k^\star\|}{\sigma_{r_k}(Z_k^\star)} \le \frac{2\zeta_k\delta}{c_{k;Z}} = \frac{2}{c_{k;Z}}\frac{c_{k;Z}}{6c_{\mathcal{C}}}\xi\delta \le \frac{\xi\delta}{3c_{\mathcal{C}}}.$$

If k is the root, recall that $\Delta B_k^\circ = \Delta Z_k^\star$, hence using $c_{k;Z} = 1$,

$$\|\Delta B_k^\circ\| = \|\Delta Z_k^\star\| \le \zeta_k \delta = \frac{c_{k;Z}}{6c_{\mathcal{C}}}\xi\delta \le \frac{\xi\delta}{3c_{\mathcal{C}}}.$$

Therefore, (81) holds.

Next, for a non-leaf k, recall that

$$\begin{aligned}\Delta A_k^\circ &= \bigotimes_{w\in\mathcal{C}(k)} (s_w B_w^\circ + s_w \Delta B_w^\circ) - \bigotimes_{w\in\mathcal{C}(k)} s_w B_w^\circ \\ &= \left(\bigotimes_{w\in\mathcal{C}(k)} s_w\right)\left(\bigotimes_{w\in\mathcal{C}(k)} (B_w^\circ + \Delta B_w^\circ) - \bigotimes_{w\in\mathcal{C}(k)} B_w^\circ\right)\end{aligned}$$

By definition, one has $\|\bigotimes_{w\in\mathcal{C}(k)} s_w\| = c_{k;S}$. Note that $\|B_w^\circ\| = 1$ and $\|\Delta B_w^\circ\| \le \frac{\xi}{3c_{\mathcal{C}}}\delta$. Hence, one can apply Lemma 29, which shows

$$\|\Delta A_k^\circ\| \le c_{k;S}\left(\sum_{w\in\mathcal{C}(k)} \|\Delta B_w^\circ\|\right)\exp\left(\sum_{w\in\mathcal{C}(k)} \|\Delta B_w^\circ\|\right),$$

Using (81),

$$\sum_{w\in\mathcal{C}(k)} \|\Delta B_w^\circ\| \le c_{\mathcal{C}} \cdot \max_{w\in[d]} \|\Delta B_w^\circ\| \le c_{\mathcal{C}} \frac{\xi\delta}{3c_{\mathcal{C}}} \le \frac{\xi\delta}{3}.$$

Hence,

$$\begin{aligned}\|\Delta A_k^\circ\| &\le c_{k;S}\left(\sum_{w\in\mathcal{C}(k)} \|\Delta B_w^\circ\|\right)\exp\left(\sum_{w\in\mathcal{C}(k)} \|\Delta B_w^\circ\|\right) \\ &\le c_{k;S}\frac{\xi\delta}{3}\exp(1) \\ &\le c_{k;S}\xi\delta,\end{aligned}$$

where the last two steps hold because $\frac{\xi\delta}{3} < 1$ and $\exp(1) < 3$.

It remains to show how (80) lead to (72).

If k is a leaf,

$$\frac{\|\Delta G_k^\circ\|}{\|G_k^\circ\|} = \frac{\|\Delta B_k^\circ\|}{\|G_k^\circ\|} \le \frac{\|\Delta B_k^\circ\|}{\|G_k^\circ\|} \le c_{k;G}\xi\delta \le \delta,$$

where the first equation follows from $\Delta G_k^\circ = \Delta B_k^\circ$ in (79), the first inequality comes from (V) in Lemma 23, and the last inequality uses $c_{k;A} = c_{k;S} = 1$ and $\xi \le \left(2c_{k;A}\left(c_{k;S} + c_{k;G}\right)\right)^{-1} = \frac{1}{2}\left(1 + c_{k;G}\right)^{-1}$.

Importantly, note that

$$A_k^\circ = \left(\bigotimes_{w\in\mathcal{C}(k)} s_w B_w^\star\right)\left(\bigotimes_{w\in\mathcal{C}(k)} R_w\right),$$

and so the fact that each R_w is orthogonal implies $\| \left(A_k^\circ\right)^\dagger \| = \| \left(A_k^\star\right)^\dagger \| = c_{k;A}$. For any non-leaf k, note that $\xi, \delta \leq 1$ leads to $\| \left(A_k^\circ\right)^\dagger \| \|\Delta A_k^\circ\| \leq c_{k;A} c_{k;S} \xi \delta \leq 1/2$. From Lemma 28 and (V) in Lemma 23, it follows that

$$\begin{aligned}\frac{|||\Delta G_k^\circ|||}{|||G_k^\circ|||} &\leq 2\| \left(A_k^\circ\right)^\dagger \| \left(\|\Delta A_k^\circ\| + \|\Delta B_k^\circ\| c_{k;G}\right) \\ &\leq 2c_{k;A} \left(c_{k;S} + c_{k;G}\right) \xi \delta \\ &\leq \delta,\end{aligned}$$

and so we are done.

□

E.4 Remarks on sample complexity bound for total variation distance

Using the proof technique as outline before, one can derive a sample complexity upper bound on the total variation norm via the l_1 distance between $p^\star$ and $\hat{p}_{\mathrm{TS}}$. Note that one can define a new norm $|||\cdot|||_1$ by

$$|||G(\alpha; x; \beta)|||_1 := \sum_x \|G(\cdot, x, \cdot)\|,$$

which is a similar definition to $|||\cdot|||$.

The proofs in Sect. 8 are also written such that the adaptation to $|||\cdot|||_1$ is straightforward. First, all of the results in Lemma 23 will hold in this new norm, with only a change in the constant in (v). Second, from the $|||\cdot|||_1$ version of Lemma 23, one can bound the global $\|\cdot\|_1$ error by the core-wise $|||\cdot|||_1$ error by an adaptation of Lemma 27. Finally, for local $|||\cdot|||_1$ error on cores, the proof of Lemma 28 also proves that Lemma 28 holds if one replaces $|||\cdot|||$ by the new $|||\cdot|||_1$ norm. Importantly, the $N = O(d^2)$ scaling will still hold under the l_1-norm.

E.5 Proof of results

Proof (of Lemma 23)

In the notation of Definition 19, one can write the definition of $\circ$ by

$$G \circ G'(\cdot, (x, y), \cdot) = G(\cdot, x, \cdot) G'(\cdot, y, \cdot).$$

Associativity of $\circ$ thus follows from associativity of matrix product, and likewise the inequality for $\circ$ comes from submultiplicativity of matrix product under spectral norm:

$$\begin{aligned}&\max_{(x,y)} \|G \circ G'(\cdot, (x, y), \cdot)\| = \max_{(x,y)} \|G(\cdot, x, \cdot) G'(\cdot, y, \cdot)\| \\ &\leq \left(\max_x \|G(\cdot, x, \cdot)\|\right) \left(\max_y \|G'(\cdot, y, \cdot)\|\right)\end{aligned}$$

Likewise, by abuse of notation, also use $\otimes$ as the Kronecker product operation over matrices. Then one can simplify and write the definition of $\otimes$ by

$$G \otimes G'(\cdot, (x, y), \cdot) = G(\cdot, x, \cdot) \otimes G'(\cdot, y, \cdot).$$

Associativity of $\otimes$ likewise follows from associativity of Kronecker product over matrices. Likewise, the equality for $\otimes$ comes from multiplicativity of matrix product under

spectral norm:

$$\|G \otimes G'(\cdot,(x,y),\cdot)\| = \|G(\cdot,x,\cdot)\| \cdot \|G'(\cdot,y,\cdot)\|$$

We now prove (V). For a vector $v \in \mathbb{R}^{r_2}$, one can view the vector $G(\alpha,x;\beta)v$ as the concatenation of n smaller vectors of the form $G(\cdot,x;\cdot)v$. For the upper bound, one has

$$\|G(\alpha,x;\beta)v\| = \sum_{x\in[n]} \|G(\cdot,x;\cdot)v\| \leq n \max_{x\in[n]} \|G(\cdot,x;\cdot)\|\|v\| = n|||G|||\|v\|,$$

where is done after taking supremum over v with $\|v\| = 1$.

For the lower bound, one has

$$\|G(\alpha,x;\beta)v\| = \sum_{x\in[n]} \|G(\cdot,x;\cdot)v\| \geq \max_{x\in[n]} \|G(\cdot,x;\cdot)\|\|v\| = |||G|||\|v\|,$$

and likewise one is done after taking supremum over v with $\|v\| = 1$.

□

Proof (of Lemma 26) Suppose that in T, the maximum distance from the root node is L. At a level $l \in \{1,\ldots,L\}$, suppose there are d_l nodes in T which are of distance l to the root, denoted by the set $\{v_i^l\}_{i=1}^{d_l}$. Then, if one views p as a 3-tensor of size $\{1\}\times\left[\prod_{i=1}^d n_i\right]\times\{1\} \to \mathbb{R}$, then one has

$$p = G_{\mathrm{root}(T)} \circ \bigotimes_{i\in[d_1]} G_{v_i^1} \circ \bigotimes_{k\in[d_2]} G_{v_j^2} \circ \ldots \circ \bigotimes_{k\in[d_L]} G_{v_k^L}, \tag{82}$$

which is only a consequence of the structure of the TTNS ansatz and the 3-tensor structure of TTNS tensor core in Definition 25. Then, by the chosen 3-tensor structure of p, one has $\|p\|_\infty = |||p|||$. By Lemma 23, one has

$$\|p\|_\infty = |||p||| \leq |||G_{\mathrm{root}(T)}||| \prod_{l=1}^L ||| \bigotimes_{i\in[d_l]} G_{v_i^l} ||| = |||G_{\mathrm{root}(T)}||| \prod_{l=1}^L \prod_{i\in[d_l]} |||G_{v_i^l}||| = \prod_{i\in[d]} |||G_i|||, \tag{83}$$

where the first inequality and the second equality follows from (63) and (64) in Lemma 23.

□

Proof (of Lemma 27) Let $p_0,\ldots,p_d$ be a sequence of tensors such that $p_0 = p$, and p_k is the tensor formed by the TTNS tensor core $\{G_i + \Delta G_i\}_{i=1}^k \cup \{G_j\}_{j\notin[k]}$. One is interested in the error $\|p_d - p_0\|_\infty$, and one can bound by

$$\|p_d - p_0\|_\infty \leq \sum_{k=1}^d \|p_k - p_{k-1}\|_\infty.$$

One can then bound the magnitude of each term in this telescoping sum. Note that $p_k - p_{k-1}$ is a TTNS ansatz formed by cores $\{G_i + \Delta G_i\}_{i=1}^{k-1} \cup \{\Delta G_k\} \cup \{G_j\}_{j=k+1}^{d}$, and thus by Lemma 26

$$\|p_k - p_{k-1}\|_\infty \leq \prod_{i=1}^{k-1} |||G_i + \Delta G_i||| \, |||\Delta G_k||| \prod_{j=k+1}^{d} |||G_j||| \leq \delta_k \prod_{i=1}^{d}(1+\delta_i) \prod_{i=1}^{d} |||\Delta G_i|||.$$

Therefore, using $1 + x \leq \exp(x)$, one has

$$\|p_d - p_0\|_\infty \leq \left(\sum_{i=1}^{d} \delta_i\right) \prod_{i=1}^{d}(1+\delta_i) \prod_{i=1}^{d} |||\Delta G_i||| \leq \left(\sum_{i=1}^{d} \delta_i\right) \exp\left(\sum_{i=1}^{d} \delta_i\right) \prod_{i=1}^{d} |||\Delta G_i|||$$

□

Proof (of Lemma 28) Note that (68) is only a corollary of (67). To prove (67), it suffices to prove that for any $i \in [n]$, one has

$$\|\Delta X^\star(\cdot, i, \cdot)\| \leq \frac{\|(A^\star)^\dagger\|}{1 - \|(A^\star)^\dagger\| \|\Delta A^\star\|} \left(\|\Delta A^\star\| \|X^\star(\cdot, i, \cdot)\| + \|\Delta B^\star(\cdot, i, \cdot)\|\right), \tag{84}$$

whereby (67) is obtained by taking maximum over $i \in [n]$ on both sides.

Based on the above observation, one can simplify notation and reduce argument over $|||\cdot|||$ to regular spectral norm over matrices. For a fixed $i \in [n]$, define $C^\star := B^\star(\cdot, i, \cdot)$. Let $Y^\star$ be the matrix which is the unique *exact* solution the linear equation $A^\star Y = C^\star$. Naturally, one has $Y^\star = X^\star(\cdot, i, \cdot)$.

Likewise, define $\Delta C^\star = \Delta B^\star(\cdot, i, \cdot)$ as the corresponding perturbation to $C^\star$, and let $Y^\star + \Delta Y^\star$ be the matrix which is the unique solution the linear equation $(A^\star + \Delta A^\star)Y = (C^\star + \Delta C^\star)$ in the sense of least squares. As before, one has $Y^\star + \Delta Y^\star = X^\star(\cdot, i, \cdot) + \Delta X^\star(\cdot, i, \cdot)$. Then, (84) is equivalent to the following inequality:

$$\|\Delta Y^\star\| \leq \frac{\|(A^\star)^\dagger\|}{1 - \|(A^\star)^\dagger\| \|\Delta A^\star\|} \left(\|\Delta A^\star\| \|X^\star\| + \|\Delta C^\star\|\right). \tag{85}$$

To reduce further, for an arbitrary $v \in \mathbb{R}^m$, note that it suffices to prove the following result

$$\|\Delta Y^\star v\| \leq \frac{\|(A^\star)^\dagger\|}{1 - \|(A^\star)^\dagger\| \|\Delta A^\star\|} \left(\|\Delta A^\star\| \|Y^\star v\| + \|\Delta C^\star v\|\right), \tag{86}$$

and (85) follows by taking supremum over v with $\|v\| = 1$.

To simplify further, define $b^\star := C^\star v \in \mathbb{R}^l$, and let $x^\star := Y^\star v \in \mathbb{R}^r$ be the unique *exact* solution to $A^\star x = b^\star$. Moreover, let $\Delta b^\star := \Delta C^\star v$ and let $\Delta x^\star := \Delta Y^\star v$, and then $x^\star + \Delta x^\star$ solves the linear equation $(A^\star + \Delta A^\star)x = (b^\star + \Delta b^\star)$ in the sense of least squares. This is exactly the setting of Theorem 3.48 in [31], because of which (86) holds as a corollary. Thus we are done. □

Proof (of Lemma 29)

Let $C_i' = C_i + E_i$, then

$$\begin{aligned}\bigotimes_{i=1}^{n}(C_i+E_i)-\bigotimes_{i=1}^{n}C_i &= (C_1'\otimes\cdots\otimes C_n')-(C_1\otimes C_2'\otimes\cdots\otimes C_n')\\ &+(C_1\otimes C_2'\otimes\cdots\otimes C_n')-(C_1\otimes C_2\otimes C_3'\otimes\cdots\otimes C_n')\\ &+\cdots\\ &+(C_1\otimes\cdots\otimes C_{n-1}\otimes C_n')-(C_1\otimes\cdots\otimes C_n).\end{aligned} \tag{87}$$

The first line on the right-hand side of (87) reduces to $E_1\otimes C_2'\otimes\cdots\otimes C_n'$. Since $\|C_i'\|\le\|C_i\|+\|E_i\|\le 1+\delta_i$,

$$\|E_1\otimes C_2'\otimes\cdots\otimes C_n'\| = \|E_1\|\|C_2'\|\cdots\|C_n'\|\le\delta_1(1+\delta_2)\cdots(1+\delta_n)\le\delta_1\cdot\prod_{i=1}^{n}(1+\delta_i).$$

The norm of the j-th line on the right-hand side of (87) is upper bounded by $\delta_j\cdot\prod_{i=1}^{n}(1+\delta_i)$. Therefore, using $1+x\le\exp(x)$, one has

$$\left\|\bigotimes_{i=1}^{n}(C_i+E_i)-\bigotimes_{i=1}^{n}C_i\right\|\le\left(\sum_{i=1}^{n}\delta_i\right)\cdot\prod_{i=1}^{n}(1+\delta_i)\le\left(\sum_{i=1}^{n}\delta_i\right)\exp\left(\sum_{i=1}^{n}\delta_i\right).$$

□

Proof (of Corollary 31 and Corollary 32)

For Corollary 31, we apply Theorem 2.9, (2.26a) in [5]: if $\|\Delta Z^\star\|\le(1-1/\sqrt{2})\sigma_r(Z^\star)$, then

$$\operatorname{dist}(B,B^\star)\le\frac{2\|(B^\star)^\top\Delta Z^\star\|}{\sigma_r(Z^\star)-\sigma_{r+1}(Z^\star)}\le\frac{2\|(B^\star)^\top\|\|\Delta Z^\star\|}{\sigma_r(Z^\star)-\sigma_{r+1}(Z^\star)},$$

and we are done by applying $\sigma_{r+1}(Z^\star)=0$ and $\|B^\star\|=1$.

For Corollary 32, only (69) is new, and one only needs to justify $\gamma\le L^2$. By Jensen's theorem and sub-multiplicativity of spectral norm, one has

$$\left\|\mathbb{E}\left[Z^{(i)}\left(Z^{(i)}\right)^\top\right]\right\|\le\mathbb{E}\left[\left\|Z^{(i)}\left(Z^{(i)}\right)^\top\right\|\right]\le\mathbb{E}\left[\left\|Z^{(i)}\right\|^2\right]\le L^2.$$

□

.

Received: 17 October 2022 Accepted: 19 March 2023 Published online: 28 April 2023

References

1. Bhattacharyya, Arnab, Gayen, Sutanu, Price, Eric, Vinodchandran, NV: Near-Optimal Learning of Tree-Structured Distributions by Chow-Liu. In: 2021 Proceedings of the 53rd annual acm SIGACT symposium on theory of computing, pp 147- 160
2. Bradley, Tai-Danae., Stoudenmire, E Miles, Terilla, John: Modeling sequences with quantum states: a look under the hood. Mach. Learn. Sci. Technol. **1**(3), 035008 (2020)
3. Bresler, Guy, Karzand, Mina: Learning a tree-structured ising model in order to make predictions. Ann. Statist. **48**(2), 713–737 (2020)
4. Candes, Emmanuel J., Plan, Yaniv: Matrix completion with noise. Proc. IEEE **98**(6), 925–936 (2010)

5. Chen, Yuxin, Chi, Yuejie, Fan, Jianqing, Ma, Cong: Spectral methods for data science: a statistical perspective: ISSN=1935-8237 Foundations and Trends in Machine. Learning **14**(5), 566–806 (2021). https://doi.org/10.1561/2200000079
6. Cheng, Song, Wang, Lei, Xiang, Tao, Zhang, Pan: Tree tensor networks for generative modeling. Phys. Rev. B **99**(15), 155131 (2019)
7. Chow, C.K.C.N., Liu, Cong: Approximating discrete probability distributions with dependence trees. IEEE Trans. Inform. Theory **14**(3), 462–467 (1968)
8. Dolgov, Sergey, Anaya-Izquierdo, Karim, Fox, Colin, Scheichl, Robert: Approximation and sampling of multivariate probability distributions in the tensor train decomposition. Statist. Comput. **303**, 603–625 (2020)
9. Gandy, Silvia, Recht, Benjamin, Yamada, Isao: Tensor completion and low-n-rank tensor recovery via convex optimization. Inverse Probl. **27**(2), 025010 (2011)
10. Glasser, Ivan, Sweke, Ryan, Pancotti, Nicola, Eisert, Jens, Cirac, Ignacio: Expressive power of tensor-network factorizations for probabilistic modeling. Adv. Neural Inform. Process. Syst. 32 (2019)
11. Gomez, Abigail McClain, Yelin, Susanne F, Najafi, Khadijeh: Born machines for periodic and open XY quantum spin chains, (2021), arXiv preprint arXiv:2112.05326,
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Adv. Neural Inform. Process. Syst. **27**, 139 (2014)
13. Han, Zhao-Yu., Wang, Jun, Fan, Heng, Wang, Lei, Zhang, Pan: Unsupervised generative modeling using matrix product states. Phys. Rev. X **8**(3), 031012 (2018)
14. Hinton, Geoffrey E.: Training products of experts by minimizing contrastive divergence. Neural Comput. **14**(8), 1771–1800 (2002)
15. Hur, Yoonhaeng, Hoskins, Jeremy G, Lindsey, Michael, Stoudenmire, E Miles, Khoo, Yuehaw: Generative modeling via tensor train sketching, (2022). arXiv preprint arXiv:2202.11788,
16. Khoo, Yuehaw, Lu, Jianfeng, Ying, Lexing: Efficient construction of tensor ring representations from sampling, (2017), arXiv preprint arXiv:1711.00954,
17. Kingma, Diederik P, Welling, Max: Auto-encoding variational bayes, (2013), arXiv preprint arXiv:1312.6114,
18. LeCun, Yann, Chopra, Sumit, Hadsell, Raia, Ranzato, M., Huang, F.: A tutorial on energy-based learning. Predict Struct. data. **1**, 10 (2006)
19. Lin, Lin, Lu, Jianfeng, Ying, Lexing: Fast construction of hierarchical matrix representation from matrix-vector multiplication. J. Comput. Phys. **230**(10), 4071–4087 (2011)
20. McClean, Jarrod R., Boixo, Sergio, Smelyanskiy, Vadim N., Babbush, Ryan, Neven, Hartmut: Barren plateaus in quantum neural network training landscapes. Nat. Commun. **9**(1), 1–6 (2018)
21. Nakatani, Naoki, Chan, Garnet Kin-Lic.: Efficient tree tensor network states (TTNS) for quantum chemistry: generalizations of the density matrix renormalization group algorithm. J. Chem. Phys. **138**(13), 134113 (2013)
22. Oseledets, Ivan V.: Tensor-train decomposition. SIAM J. Sci. Comput. **33**(5), 2295–2317 (2011)
23. Rezende, Danilo, Mohamed, Shakir: Variational inference with normalizing flows. In: PMLR, 2015 International conference on machine learning. pp 1530- 1538 (2015)
24. Richard, Emile, Montanari, Andrea: A statistical model for tensor pca. Adv. Neural Inform. Process. Syst. 27 (2014)
25. Shi, Y.-Y., Duan, L.-M., Vidal, Guifre: Classical simulation of quantum many-body systems with a tree tensor network. Phys. Rev. A **74**(2), 022320 (2006)
26. Silverman, Bernard W: Density Estimation for Statistics and Data Analysis, Routledge, (2018)
27. Song, Yang, Ermon, Stefano: Generative modeling by estimating gradients of the data distribution. Adv. Neural Inform. Process. Syst. 32 (2019)
28. Tabak, Esteban G., Vanden-Eijnden, Eric: Density estimation by dual ascent of the log-likelihood. Commun. Math. Sci. **8**(1), 217–233 (2010)
29. Tropp, Joel A., et al.: An introduction to matrix concentration inequalities. Foundat. Trends. Mach. Learning **8**(1–2), 1–230 (2015)
30. Verstraete, Frank, Wolf, Michael M., Perez-Garcia, David, Cirac, J Ignacio: Criticality: the area law, and the computational power of projected entangled pair states. Phys. Rev. Lett. **96**(22), 220601 (2006)
31. Wendland, Holger: Numerical Linear Algebra: An Introduction, Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge (2017)
32. Woodruff, David P., et al.: Sketching as a tool for numerical linear algebra. Found. Trends. Theoret. Comput. Sci. **10**(1–2), 1–157 (2014)

Publisher's Note