Appendix:Guaranteed Scalable Learning of Latent Tree Models

A RELATED WORKS

There has been widespread interest in developing distributed learning techniques, e.g. the recent works of Smola and Narayanamurthy ((2010)) and Wei et al. ((2013)). These works consider parameter estimation via likelihood-based optimizations such as Gibbs sampling, while our method involves more challenging tasks where both the structure and the parameters are estimated. Simple methods such as local neighborhood selection through ℓ_1-regularization ((Meinshausen and Bühlmann, 2006)) or local conditional independence testing ((Anandkumar et al., 2012c)) can be parallelized, but these methods do not incorporate hidden variables. Finally, note that the latent tree models provide a statistical description, in addition to revealing the hierarchy. In contrast, hierarchical clustering techniques are not based on a statistical model ((Krishnamurthy et al., 2012)) and cannot provide valuable information such as the level of correlation between observed and hidden variables.

Both Chang and Hartigan ((1991)) and Chang ((1996)) provide detailed identifiability analyses of latent trees and the former ((Chang and Hartigan, 1991)) introduces maximum likelihood estimation method, different from our guaranteed parallel spectral method using tensor decomposition.

P Parikh et al. ((2011)), Song et al. ((2011, 2014)) are closely related to our paper. The P Parikh et al. ((2011)) paper does not address the structure learning. The sample complexity is polynomial in k, whereas ours is logarithmic in k. The Song et al. ((2014)) paper provides guarantees for structure learning, but not for parameter estimation. The Song et al. ((2011)) paper does not provide sample complexity theorem or analysis for recovering the latent tree structure or parameter with provable guarantees.

B SYNTHETIC EXPERIMENTS

B.1 Machine Setup

Setup Experiments are conducted on a server running the Red Hat Enterprise 6.6 with 64 AMD Opteron processors and 265 GBRAM. The program is written in C++, coupled with the multi-threading capabilities of the OpenMP environment ((Dagum and Menon, 1998)) (version 1.8.1). We use the Eigen toolkit[1] where BLAS operations are incorporated. For SVDs of large matrices, we use randomized projection methods ((Gittens and Mahoney, 2013b)) as described in Appendix L.

B.2 Synthetic Results:

We compare our method with the implementation of Choi et al. ((2011)), where a serial learning procedure is carried out for binary variables $(d = 2)$ and parameter learning is carried out via EM. We consider a latent tree model over nine observed variables four hidden nodes. We restart EM 20 times, and select the best result. The results are in Figure 3.

We measure the structure recovery error via $\left\{ \sum_{i=1}^{|\widehat{G}|} \min_j |\widehat{G}_i \notin G_j| / |\widehat{G}_i| \right\} / |\widehat{G}|$, where G and $\widehat{G}$ are the ground-truth and recovered categories. We measure the parameter recovery error via $\mathcal{E} = \|A - \hat{A}\Pi\|_F$, where A is the true parameter and $\hat{A}$ is the estimated parameter and Π is a suitable permutation matrix that aligns the columns of $\hat{A}$ with A so that they have minimum distance. Π is greedily calculated. It is shown that as number of samples increases, both methods recover the structure correctly, as predicted by the theory. However, EM is stuck in local optima and fails to recover the true parameters, while the tensor decomposition correctly recovers the true parameters.

We then present our experimental results focusing on the large p and even larger d regime. See Table 2. We achieve efficient running times with good accuracy for structure and parameter estimation.

We perform experiments on a synthetic tree with observable dimension $d = 1,000,000$, number of observable nodes $p = 3$, hidden dimension $k = 3$ and number of samples $N = 1000$ which is similar to community detection setting of

[1] http://eigen.tuxfamily.org/index.php?title=Main_Page

Huang et al. ((2013)). We note that the recovery of the structure and the parameters is done in 10.6 seconds.

C EXPERIMENTS ON NIPS AND NY TIMES DATASET

We use the entire NIPS dataset from the UCI bag-of-words repository. This consists of 1500 documents and 12419 words in the vocabulary. We estimate the hierarchical structure of the whole corpus in only 15332.4 seconds (4 hours). Additionally, we focus on the top 5000 most frequently appeared keywords and illustrate the local structures in Figure 5. The running time for the subset is only 1164.4 seconds (20 minutes).

We also use the NY Times dataset from the UCI bag-of-words repository. This consists of 300000 documents and 102660 words in the vocabulary. Our algorithm estimates the hierarchical structure of 3000 most frequently appeared keywords in the NY Times dataset in only 107.8 seconds. Note that $d = 2$, since we consider the occurrence of a word in a document as a binary variable. Below is a subset of the keywords graph we estimated in Figure 6. We note that the relationships the among the words in Figure 6 match intuition. For example, govern and secur are grouped together whereas movi, studio and produc are grouped together. The numbers represent hidden nodes.

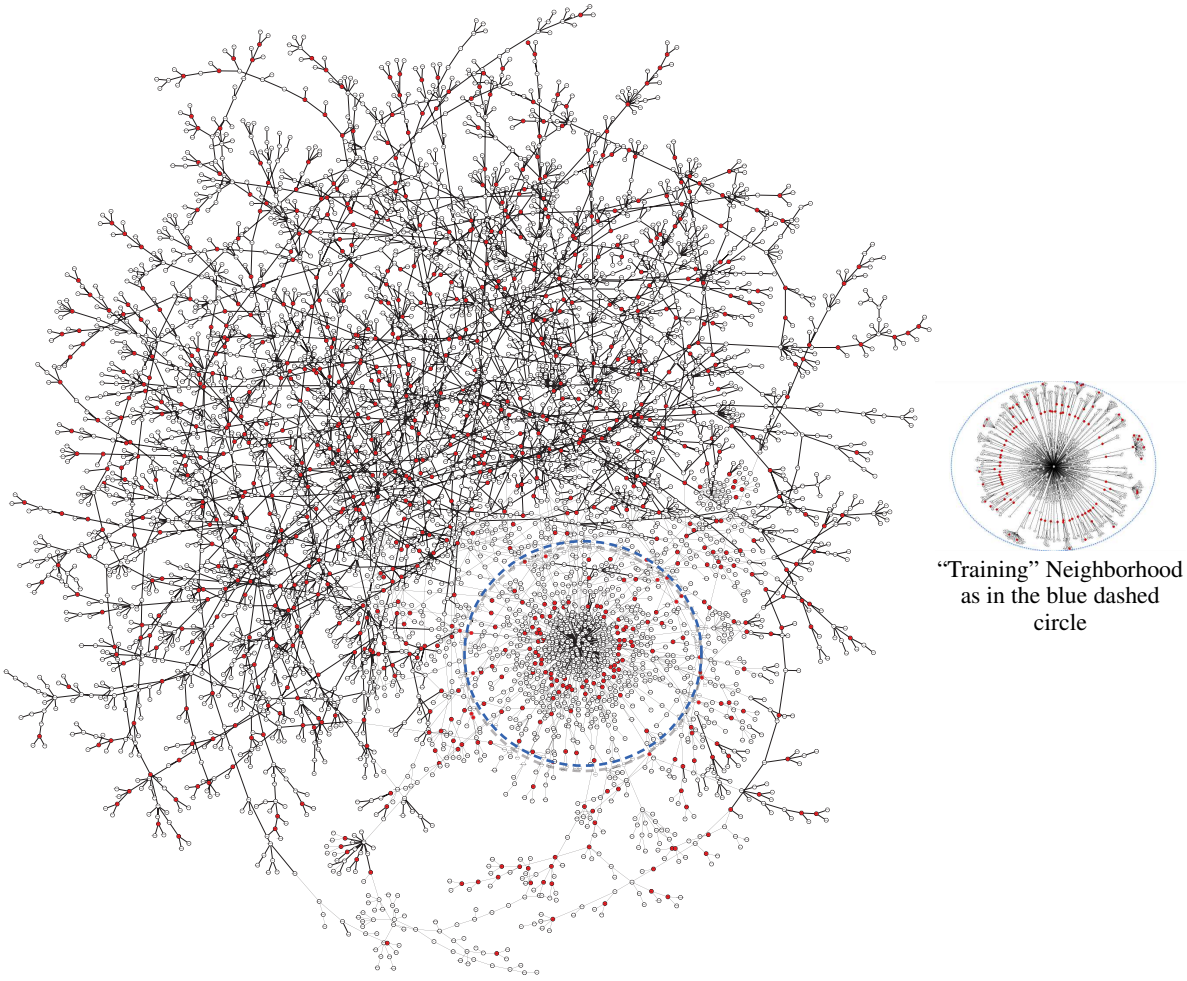

Figure 4: Estimated NIPS top 5000 keywords global hierarchical structures. Red nodes are latent nodes introduced. The blue dash circle in the global hierarchical structure is zoomed in which is the neighborhood of word "training".

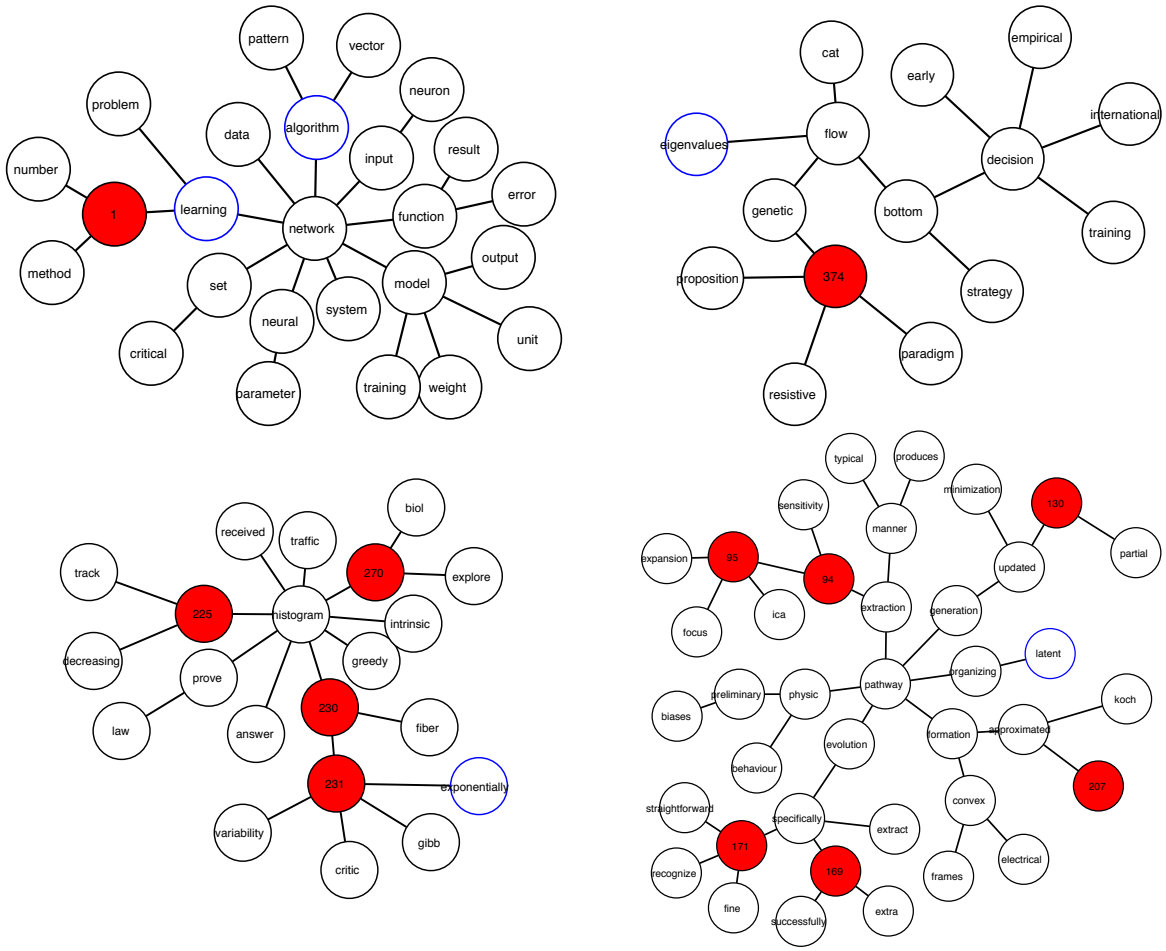

Figure 5: Extended neighborhoods of some of the words estimated from the NIPS dataset. We run our algorithm for the top 5000 keywords global and local hierarchical structures.

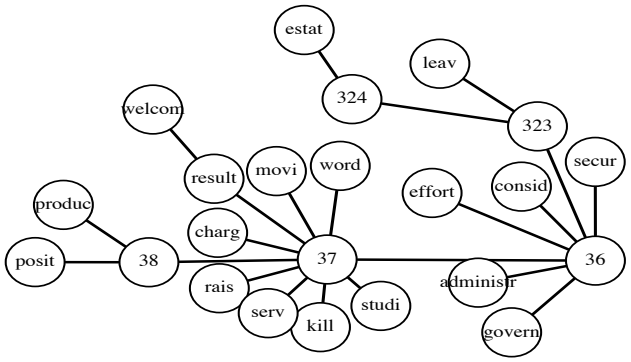

Figure 6: A local view of recovered NY Times keywords structure

D EXPERIMENTS ON HEALTHCARE DATASETS

Quantitative Analysis We evaluate our resulting hierarchy with a ground truth tree, based on medical knowledge[2]. We compare our results with a baseline: the agglomerative clustering. Evaluating the performance of tree structure recovery is indeed nontrivial as the *number* and *location* of the hidden variables vary as well as the depths. Two trees may be similar but may look different due to the difference that how refined the hierarchical clusterings are. The standard Robinson Foulds (RF) metric ((Robinson and Foulds, 1981))(between our estimated latent tree and the ground truth tree) is computed to evaluate the structure recovery in Table 3. RF is a well defined metric for evaluating the difference of two tree structures. Similar to other distance metrics, smaller RF indicates the recovered structure being closer to the ground-truth.The smaller the metric is, the better the recovered tree is. The proposed method is slightly better than the baseline and the advantage increases with more nodes. However, our proposed method provides an efficient probabilistic graphical model that can support general inference which is not feasible with the baseline.

Data	p	RF(agglo.)	RF(proposed)
MIMIC2	163	0.0061	0.0061
CMS	168	0.0060	0.0059
MIMIC2	952	0.0060	0.0011

Table 3: Robinson Foulds (RF) metric compared with the "ground-truth" tree for both MIMIC2 and CMS dataset. Our proposed results are better as we increase the number of nodes.

Qualitative analysis Here we report the results from the 2-dimensional case (i.e., observed variable is binary). In Figure 7 in appendix D.2, we show a portion of the learned tree using the MIMIC2 healthcare data. The yellow nodes are latent nodes from the learned subtrees while the blue nodes represent observed nodes(diagnosis codes) in the original dataset. Diagnoses that are similar were generally grouped together. For example, many neoplastic diseases were grouped under the same latent node (node 1135). While some dissimilar diseases were grouped together, there usually exists a known or plausible association of the diseases in the clinical setting. For example, in Figure 7 in appendix D.2, clotting-related diseases and altered mental status were grouped under the same latent node as several neoplasms. This may reflect the fact that altered mental status and clotting conditions such as thrombophlebitis can occur as complications of neoplastic diseases ((Falanga et al., 2003)). The association of malignant neoplasms of prostate and colon polyps, two common cancers in males, is captured under latent node 1136 ((Group et al., 2014)).

For both the MIMIC2 and CMS datasets, we performed a qualitative comparison of the resulting trees while varying the hidden dimension k for the algorithm. The resulting trees for different values of k did not exhibit significant differences. This implies that our algorithm is robust with different choices of hidden dimensions. The estimated model parameters are also robust for different values of k based on the results.

Scalability Our algorithm is scalable w.r.t. varying characteristics of the input data. First, it can handle a large number of patients efficiently, as shown in Figure 9(a). It has a linear scaling behavior as we vary the number observed nodes, as shown in Figure 9(b). Furthermore, even in cases where the number of observed variables is large, our method maintains an almost linear scale-up as we vary the computational power available, as shown in Figure 9(c). So, by providing the respective resources, our algorithm is practical under any variation of the input data characteristics.

D.1 Data Description

(1) MIMIC2: The MIMIC2 dataset record disease history of 29,862 patients where a overall of 314,647 diagnostic events over time representing 5675 diseases are logged. We consider patients as samples and groups of diseases as variables. We analyze and compare the results by varying the group size (therefore varying d and p).

(2) CMS: The CMS dataset includes 1.6 million patients, for whom 15.8 million medical encounter events are logged. Across all events, 11,434 distinct diseases (represented by ICD codes) are logged. We consider patients as samples and groups of diseases as variables. We consider specific diseases within each group as dimensions. We analyze and compare the results by varying the group size (therefore varying d and p). While the MIMIC2 dataset and CMS dataset both contain logged diagnostic events, the larger volume of data in CMS provides an opportunity for testing

[2]The ground truth tree is the PheWAS hierarchy provided in the clinical study ((Denny et al., 2010))

the algorithm's scalability. We qualitatively evaluate biological implications on MIMIC2 and quantitatively evaluate algorithm performance and scalability on CMS.

To learn the disease hierarchy from data, we also leverage some existing domain knowledge about diseases. In particular, we use an existing mapping between ICD codes and higher-level Phenome-wide Association Study (PheWAS) codes ((Denny et al., 2010)). We use (about 200) PheWAS codes as observed nodes and the observed node dimension is set to be binary ($d = 2$) or the maximum number of ICD codes within a pheWAS code ($d = 31$).

D.2 Results

Case d =31: We learn a tree from the MIMIC2 dataset, in which we grouped diseases into 163 pheWAS codes and up to 31 dimensions per variable. Figure 8 in appendix D.2 shows a portion of the learned tree of four subtrees which all reflect similar diseases relating to trauma. A majority of the learned subtrees reflected clinically meaningful concepts, in that related and commonly co-occurring diseases tended to group together in the same subtrees or in nearby subtrees.

We also learn the disease tree from the larger CMS dataset, in which we group diseases into 168 variables and up to 31 dimensions per variable. Similar to the case from the MIMIC2 dataset, a majority of learned subtrees reflected clinically meaningful concepts.

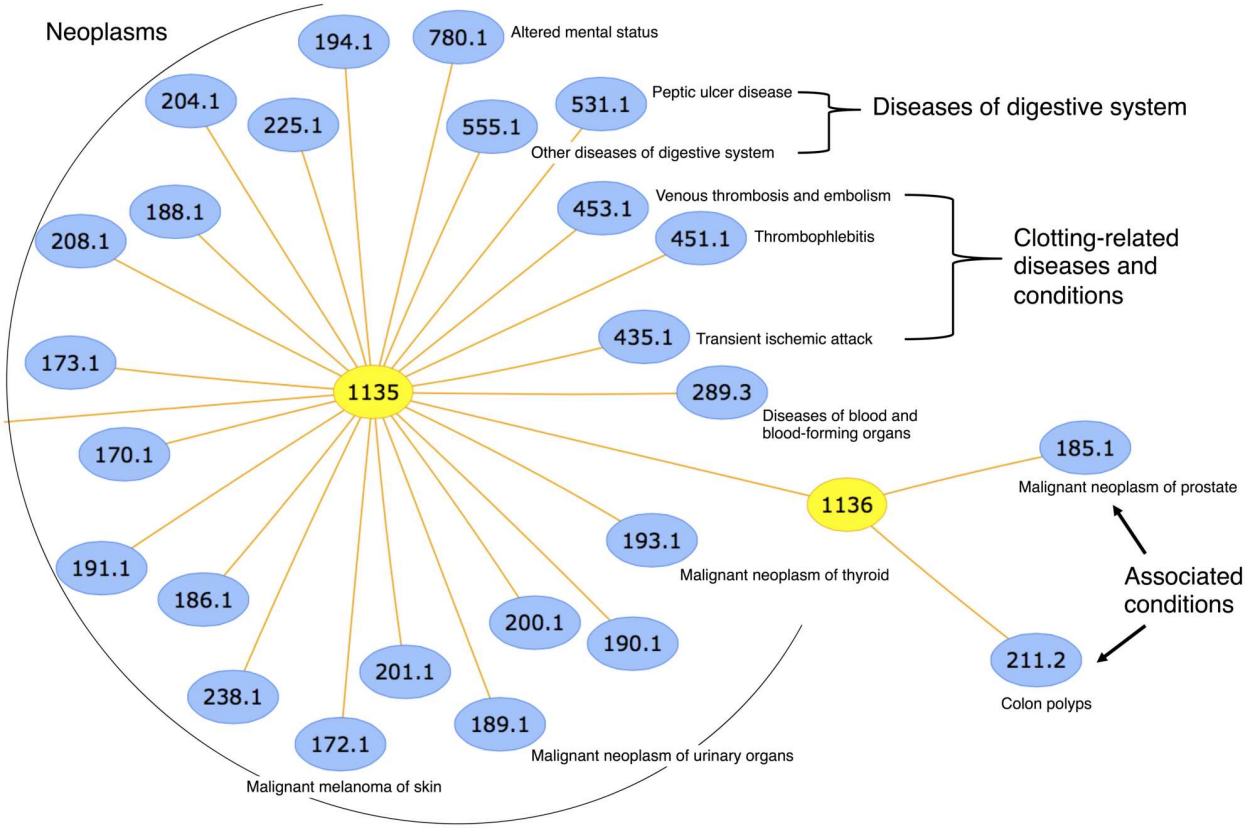

Figure 7: An example of two subtrees which represent groups of similar diseases which may commonly co-occur. Nodes colored yellow are latent nodes from learned subtrees.

E ADDITIVITY OF THE MULTIVARIATE INFORMATION DISTANCE

Recall that the additive information distance between nodes two categorical variables x_i and x_j was defined in Choi et al. ((2011)). We extend the notation of information distance to high dimensional variables via Definition 4.1 and present the proof of its additivity in Lemma 4.2 here.

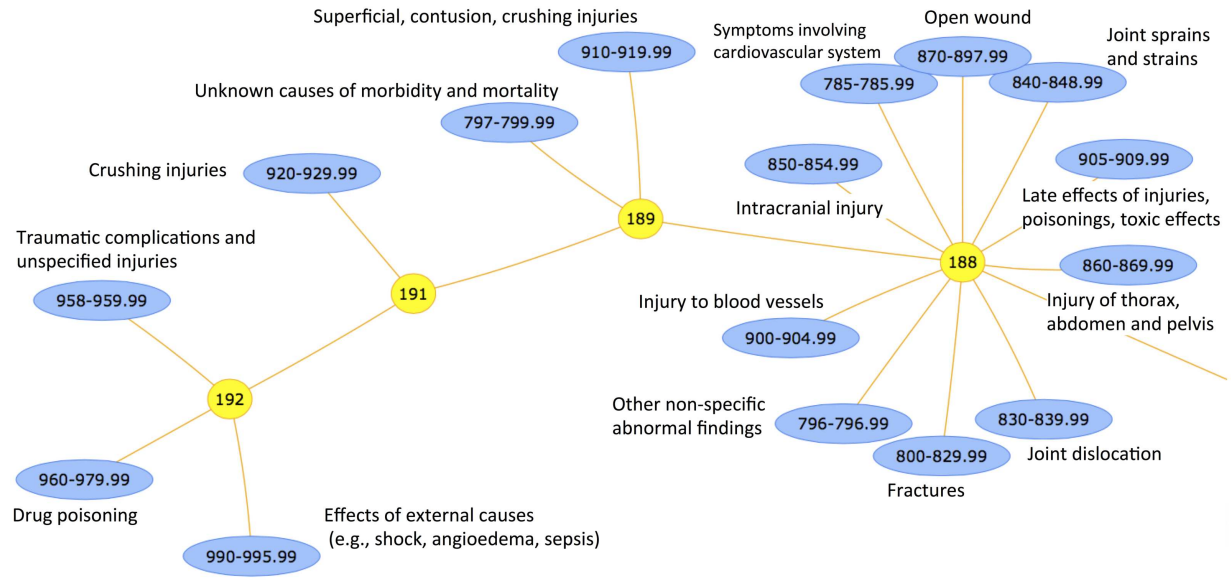

Figure 8: An example of four subtrees which represent groups of similar diseases which may commonly co-occur. Most variables in this subtree are related to trauma.

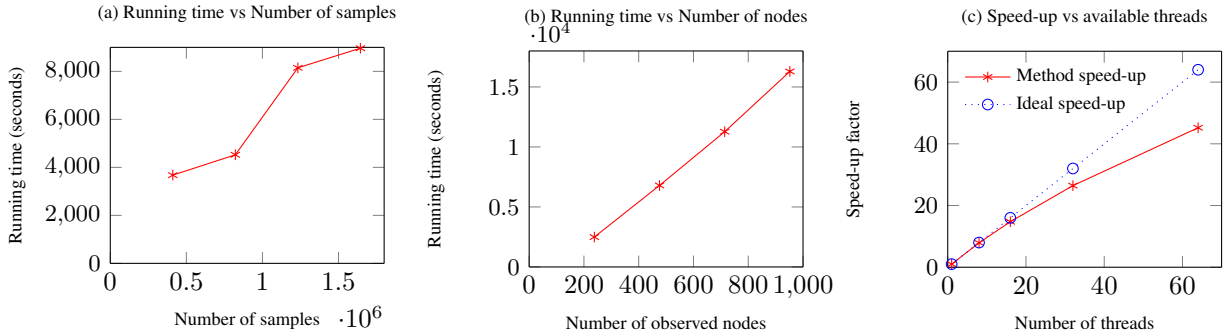

Figure 9: **(a)** CMS dataset sub-sampling w.r.t. varying number of samples. **(b)** MIMIC2 dataset sub-sampling w.r.t. varying number of observed nodes. Each one of the observed nodes is binary ($d = 2$). **(c)** MIMIC2 dataset: Scaling w.r.t. varying computational power, establishing the scalability of our method even in the large p regime. The number of observed nodes is 1083 and each one of them is binary ($p = 1083, d = 2$).

Proof.

$$\mathbb{E}[x_a x_c^\top] = \mathbb{E}[\mathbb{E}[x_a x_c^\top | x_b]] = A\mathbb{E}[x_b x_b^\top]B^\top$$

Consider three nodes a, b, c such that there are edges between a and b, and b and c. Let the $A = \mathbb{E}(x_a|x_b)$ and $B = \mathbb{E}(x_c|x_b)$. From Definition 4.1, we have, assuming that $\mathbb{E}(x_a x_a^\top)$, $\mathbb{E}(x_b x_b^\top)$ and $\mathbb{E}(x_c x_c^\top)$ are full rank.

$$\text{dist}(v_a, v_c) = -\log \frac{\prod_{i=1}^{k} \sigma_i(\mathbb{E}(x_a x_c^\top))}{\sqrt{\det(\mathbb{E}(x_a x_a^\top))\det(\mathbb{E}(x_c x_c^\top))}}$$

$$e^{-\text{dist}(v_a, v_c)} = \det\left(\mathbb{E}(x_a x_a^\top)^{-1/2} U^\top \mathbb{E}(x_a x_c^\top) V \mathbb{E}(x_c x_c^\top)^{-1/2}\right)$$

where k-SVD$((\mathbb{E}(x_a x_c^\top)) = U\Sigma V^\top)$. Similarly,

$$e^{-\text{dist}(v_a, v_b)} = \det\left(\mathbb{E}(x_a x_a^\top)^{-1/2} U^\top \mathbb{E}(x_a x_b^\top) W \mathbb{E}(x_b x_b^\top)^{-1/2}\right)$$

$$e^{-\text{dist}(v_b, v_c)} = \det\left(\mathbb{E}(x_b x_b^\top)^{-1/2} W^\top \mathbb{E}(x_b x_c^\top) V \mathbb{E}(x_c x_c^\top)^{-1/2}\right)$$

where k-SVD$((\mathbb{E}(x_a x_b^\top)) = U\Sigma W^\top)$ and k-SVD$((\mathbb{E}(x_b x_c^\top)) = W\Sigma V^\top)$.

Therefore,

$$\begin{aligned} e^{-(\text{dist}(a,b)+\text{dist}(b,c))} &= \det(\mathbb{E}(x_a x_a^\top)^{-1/2} U^\top \mathbb{E}(x_a x_b^\top)\mathbb{E}(x_b x_b^\top)^{-1/2-1/2}\mathbb{E}(x_b x_c^\top)V\mathbb{E}(x_c x_c^\top)^{-1/2}) \\ &= \det(\mathbb{E}(x_a x_a^\top)^{-1/2} U^\top A\mathbb{E}(x_b x_b^\top)B^\top V\mathbb{E}(x_c x_c^\top)^{-1/2}) = e^{-\text{dist}(v_a,v_c)} \end{aligned}$$

We conclude that the multivariate information distance is additive. Note that $\mathbb{E}\left[x_a x_b^\top\right] = \mathbb{E}\left(\mathbb{E}\left(x_a x_b^\top | x_b\right)\right) = \mathbb{E}\left(A x_b x_b^\top\right) = A\mathbb{E}(x_b x_b^\top)$. □

We note that when the second moments are not full rank, the above distance can be extended as follows:

$$\text{dist}(v_a, v_c) = -\log \frac{\prod_{i=1}^{k} \sigma_i(\mathbb{E}(x_a x_c^\top))}{\sqrt{\prod_{i=1}^{k} \sigma_i(\mathbb{E}(x_a x_a^\top)) \prod_{i=1}^{k} \sigma_i(\mathbb{E}(x_c x_c^\top))}}.$$

F LOCAL RECURSIVE GROUPING

The Local Recursive Grouping (LRG) algorithm is a local divide and conquer procedure for learning the structure and parameter of the latent tree (Algorithm 1). We perform recursive grouping simultaneously on the sub-trees of the MST. Each of the sub-tree consists of an internal node and its neighborhood nodes. We keep track of the internal nodes of the MST, and their neighbors. The resultant latent sub-trees after LRG can be merged easily to recover the final latent tree. Consider a pair of neighboring sub-trees in the MST. They have two common nodes (the internal nodes) which are neighbors on MST. Firstly we identify the path from one internal node to the other in the trees to be merged, then compute the multivariate information distances between the internal nodes and the introduced hidden nodes. We recover the path between the two internal nodes in the merged tree by inserting the hidden nodes closely to their surrogate node. Secondly, we merge all the leaves which are not in this path by attaching them to their parent. Hence, the recursive grouping can be done in parallel and we can recover the latent tree structure via this merging method.

Lemma F.1. *If an observable node v_j is the surrogate node of a hidden node h_i, then the hidden node h_i can be discovered using v_j and the neighbors of v_j in the MST.*

This is due to the additive property of the multivariate information distance on the tree and the definition of a surrogate node. This observation is crucial for a completely local and parallel structure and parameter estimation. It is also easy to see that all internal nodes in the MST are surrogate nodes.

After the parallel construction of the MST, we look at all the internal nodes $\mathcal{X}_{\text{int}}$. For $v_i \in \mathcal{X}_{\text{int}}$, we denote the neighborhood of v_i on MST as nbdsub$(v_i; \text{MST})$ which is a small sub-tree. Note that the number of such sub-trees is equal to the number of internal nodes in MST.

For any pair of sub-trees, nbd$_{\text{sub}}(v_i; \text{MST})$ and nbd$_{\text{sub}}(v_j; \text{MST})$, there are two topological relationships, namely overlapping (i.e., when the sub-trees share at least one node in common) and non-overlapping (i.e., when the sub-trees do not share any nodes).

Since we define a neighborhood centered at v_i as only its immediate neighbors and itself on MST, the overlapping neighborhood pair nbd$_{\text{sub}}(v_i; \text{MST})$ and nbd$_{\text{sub}}(v_j; \text{MST})$ can only have conflicting paths, namely path$(v_i, v_j; \mathcal{N}_i)$ and path$(v_i, v_j; \mathcal{N}_j)$, if v_i and v_j are neighbors in MST.

With this in mind, we locally estimate all the latent sub-trees, denoted as $\mathcal{N}_i$, by applying Recursive Grouping ((Choi et al., 2011)) in a parallel manner on nbdsub$(v_i; \text{MST})$, $\forall v_i \in \mathcal{X}_{\text{int}}$. Note that the latent nodes automatically introduced by RG(v_i) have v_i as their surrogate. We update the tree structure by joining each level in a bottom-up manner. The testing of the relationship among nodes ((Choi et al., 2011)) uses the additive multivariate information distance metric (Appendix E) $\Phi(v_i, v_j; k) = \text{dist}(v_i, v_k) - \text{dist}(v_i, v_k)$ to decide whether the nodes v_i and v_j are parent-child or siblings. If they are siblings, they should be joined by a hidden parent. If they are parent and child, the

child node is placed as a lower level node and we add the other node as the single parent node, which is then joined in the next level.

Finally, for each internal edge of MST connecting two internal nodes v_i and v_j, we consider merging the latent sub-trees. In the example of two local estimated latent sub-trees in Figure 1, we illustrate the complete local merging algorithm that we propose.

G PROOF SKETCH FOR THEOREM 7.1

We argue for the correctness of the method under exact moments. The sample complexity follows from the previous works. In order to clarify the proof ideas, we define the notion of *surrogate node* ((Choi et al., 2011)) as follows.

Definition G.1. *Surrogate node for hidden node h_i on the latent tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ is defined as* $Sg(h_i; \mathcal{T}) := \arg\min_{v_j \in \mathcal{X}} dist(v_i, v_j)$.

In other words, the surrogate for a hidden node is an observable node which has the minimum multivariate information distance from the hidden node. See Figure 1(a), the surrogate node of h_1, $\text{Sg}(h_1; \mathcal{T})$, is v_3, $\text{Sg}(h_2; \mathcal{T}) = \text{Sg}(h_3; \mathcal{T}) = v_5$. Note that the notion of the surrogate node is only required for analysis, and our algorithm does not need to know this information.

The notion of surrogacy allows us to relate the constructed MST (over observed nodes) with the underlying latent tree. It can be easily shown that contracting the hidden nodes to their surrogates on latent tree leads to MST. Local recursive grouping procedure can be viewed as reversing these contractions, and hence, we obtain consistent local sub-trees.

We now argue the correctness of the structure union procedure, which merges the local sub-trees. In each reconstructed sub-tree $\mathcal{N}_i$, where v_i is the group leader, the discovered hidden nodes $\{h^i\}$ form a surrogate relationship with v_i, i.e. $\text{Sg}(h^i; \mathcal{T}) = v_i$. Our merging approach maintains these surrogate relationships. For example in Figure 1(d1,d2), we have the path $v_3 - h_1 - v_5$ in $\mathcal{N}_3$ and path $v_3 - h_3 - h_2 - v_5$ in $\mathcal{N}_5$. The resulting path is $v_3 - h_1 - h_3 - h_2 - v_5$, as seen in Figure 1(e). We now argue why this is correct. As discussed before, $\text{Sg}(h_1; \mathcal{T}) = v_3$ and $\text{Sg}(h_2; \mathcal{T}) = \text{Sg}(h_3; \mathcal{T}) = v_5$. When we merge the two subtrees, we want to preserve the paths from the group leaders to the added hidden nodes, and this ensures that the surrogate relationships are preserved in the resulting merged tree. Thus, we obtain a global consistent tree structure by merging the local structures. The correctness of parameter learning comes from the consistency of the tensor decomposition techniques and careful alignments of the hidden labels across different decompositions. Refer to Appendix H, K for proof details and the sample complexity.

H PROOF OF CORRECTNESS FOR LRG

Definition H.1. *A latent tree $\mathcal{T}_{\geq 3}$ is defined to be a minimal (or identifiable) latent tree if it satisfies that each latent variable has at least 3 neighbors.*

Definition H.2. *Surrogate node for hidden node h_i in latent tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ is defined as*

$$Sg(h_i; \mathcal{T}) := \arg\min_{v_j \in \mathcal{X}} dist(v_i, v_j).$$

There are some useful observations about the MST in Choi et al. ((2011)) which we recall here.

Property H.3 (MST − surrogate neighborhood preservation)**.** *The surrogate nodes of any two neighboring nodes in $\mathcal{E}$ are also neighbors in the MST. I.e.,*

$$(h_i, h_j) \in \mathcal{E} \Rightarrow (Sg(h_i), Sg(h_j)) \in MST.$$

Property H.4 (MST − surrogate consistency along path)**.** *If $v_j \in \mathcal{X}$ and $v_h \in Sg^{-1}(v_j)$, then every node along the path connecting v_j and v_h belongs to the inverse surrogate set $Sg^{-1}(v_j)$, i.e.,*

$$v_i \in Sg^{-1}(v_j),\ \forall v_i \in Path(v_j, v_h)$$

if

$$v_h \in Sg^{-1}(v_j).$$

The MST properties observed connect the MST over observable nodes with the original latent tree $\mathcal{T}$. We obtain MST by contracting all the latent nodes to its surrogate node.

Given that the correctness of CLRG algorithm is proved in Choi et al. ((2011)), we prove the equivalence between the CLRG and PLRG.

Lemma H.5. *For any sub-tree pairs nbd$[v_i; MST]$ and nbd$[v_i; MST]$, there is at most one overlapping edge. The overlapping edge exists if and only if $v_i \in$ nbd$(v_j; MST)$.*

This is easy to see.

Lemma H.6. *Denote the latent tree recovered from nbd$[v_i; MST]$ as $\mathcal{N}_i$ and similarly for nbd$[v_j; MST]$. The inconsistency, if any, between $\mathcal{N}_i$ and $\mathcal{N}_j$ occurs in the overlapping path$(v_i, v_j; \mathcal{N}_i)$ in and path$(v_i, v_j; \mathcal{N}_j)$ after LRG implementation on each subtrees.*

We now prove the correctness of LRG. Let us denote the latent tree resulting from merging a subset of small latent trees as $T_{\text{LRG}}(S)$, where S is the set of center of subtrees that are merged pair-wisely. CLRG algorithm in Choi et al. ((2011)) implements the RG in a serial manner. Let us denote the latent tree learned at iteration i from CLRG is $T_{\text{CLRG}}(S)$, where S is the set of internal nodes visited by CLRG at current iteration . We prove the correctness of LRG by induction on the iterations.

At the initial step $S = \emptyset$: $T_{\text{CLRG}} = MST$ and $T_{\text{LRG}} = MST$, thus $T_{\text{CLRG}} = T_{\text{LRG}}$.

Now we assume that for the same set S_{i-1}, $T_{\text{CLRG}} = T_{\text{LRG}}$ is true for $r = 1, \ldots, i-1$. At iteration $r = i$ where CLRG employs RG on the immediate neighborhood of node v_i on $T_{\text{CLRG}}(S_{i-1})$, let us assume that H_i is the set of hidden nodes who are immediate neighbors of $i-1$. The CLRG algorithm thus considers all the neighbors and implements the RG. We know that the surrogate nodes of every latent node in H_i belong to previously visited nodes S_{i-1}. According to Property H.3 and H.4, if we contract all the hidden node neighbors to their surrogate nodes, CLRG thus is a RG on neighborhood of i on MST.

As for our LRG algorithm at this step, $T_{\text{LRG}}(S_i)$ is the merging between $T_{\text{LRG}}(S_{i-1})$and $\mathcal{N}_i$. The latent nodes whose surrogate node is j are introduced between the edge $(i-1, i)$. Now that we know $\mathcal{N}_i$ is the RG output from immediate neighborhood of i on MST. Therefore, we proved that $T_{\text{CLRG}}(S_i) = T_{\text{LRG}}(S_i)$.

I CROSS GROUP ALIGNMENT CORRECTION

In order to achieve cross group alignments, tensor decompositions on two cross group triplets have to be computed. The first triplet is formed by three nodes: reference node in group 1, x_1, non-reference node in group 1, x_2, and reference node in group 2, x_3. The second triplet is formed by three nodes as well: reference node in group 2, x_3, non-reference node in group 2, x_4 and reference node in group 1, x_1. Let us use h_1 to denote the parent node in group 1, and h_2 the parent node in group 2.

From Trip(x_1, x_2, x_3), we obtain $P(h_1|x_1) = \tilde{A}$, $P(x_2|h_1) = B$ and $P(x_3|h_1) = P(x_3|h_2)P(h_2|h_1) = DE$. From Trip$(x_3, x_4, x_1)$, we know $P(x_3|h_2) = D\Pi$, $P(x_4|h_2) = C\Pi$ and $P(h_2|x_1) = P(h_2|h_1)P(h_1|x_1) = \Pi E\tilde{A}$, where Π is a permutation matrix. We compute Π as $\Pi = \sqrt{(\Pi E\tilde{A})(\tilde{A})^\dagger (DE)^\dagger (D\Pi)}$ so that $D = (D\Pi)\Pi^\dagger$ is aligned with group 1. Thus, when all the parameters in the two groups are aligned by permute group 2 parameters using Π, thus the alignment is completed.

Similarly, the alignment correction can be done by calculating the permutation matrices while merging different threads.

Overall, we merge the local structures and align the parameters from LRG locla sub-trees using Procedure 2 and 3.

J COMPUTATIONAL COMPLEXITY

We recall some notations here: d is the observable node dimension, k is the hidden node dimension ($k \ll d$), N is the number of samples, p is the number of observable nodes, and z is the number of non-zero elements in each sample.

Multivariate information distance estimation involves sparse matrix multiplications to compute the pairwise second

moments. Each observable node has a $d \times N$ sample matrix with z non-zeros per column. Computing the product $x_1 x_2^T$ from a single sample for nodes 1 and 2 requires $O(z)$ time and there are N such sample pair products leading to $O(Nz)$ time. There are $O(p^2)$ node pairs and hence the degree of parallelism is $O(p^2)$. Next, we perform the k-rank SVD of each of these matrices. Each SVD takes $O(d^2 k)$ time using classical methods. Using randomized methods ((Gittens and Mahoney, 2013b)), this can be improved to $O(d + k^3)$.

Next on, we construct the MST in $O(\log p)$ time per worker with p^2 workers. The structure learning can be done in $O(\Gamma^3)$ per sub-tree and the local neighborhood of each node can be processed completely in parallel. We assume that the group sizes Γ are constant (the sizes are determined by the degree of nodes in the latent tree and homogeneity of parameters across different edges of the tree. The parameter estimation of each triplet of nodes consists of implicit stochastic updates involving products of $k \times k$ and $d \times k$ matrices. Note that we do not need to consider all possible triplets in groups but each node must be take care by a triplet and hence there are $O(p)$ triplets. This leads to a factor of $O(\Gamma k^3 + \Gamma d k^2)$ time per worker with p/Γ degree of parallelism.

At last, the merging step consists of products of $k \times k$ and $d \times k$ matrices for each edge in the latent tree leading to $O(dk^2)$ time per worker with p/Γ degree of parallelism.

K SAMPLE COMPLEXITY

From Theorem 11 of Choi et al. ((2011)), we recall the number of samples required for the recovery of the tree structure that is consistent with the ground truth (for a precise definition of consistency, refer to Definition 2 of Choi et al. ((2011))).

From Anandkumar et al. ((2012a)), we recall the sample complexity for the faithful recovery of parameters via tensor decomposition methods.

We define ϵ_P to be the noise raised between empirical estimation of the second order moments and exact second order moments, and ϵ_T to be the noise raised between empirical estimation of the third order moments and the exact third order moments.

Lemma K.1. *Consider positive constants C, C', c and c', the following holds. If*

$$\epsilon_P \leq c\frac{\frac{\lambda_k}{\lambda_1}}{k}, \qquad \epsilon_T \leq c'\frac{\lambda_k \sigma_k^{3/2}}{k}$$
$$N \geq C\left(\log(k) + \log\left(\log\left(\frac{\lambda_1 \sigma_k^{3/2}}{\epsilon_T} + \frac{1}{\epsilon_P}\right)\right)\right)$$
$$L \geq \text{poly}(k)\log(1/\delta),$$

then with probability at least $1-\delta$, tensor decomposition returns $(\widehat{v_i}, \lambda_i) : i \in [k]$ satisfying, after appropriate reordering,

$$\|\widehat{v_i} - v_i\|_2 \leq C'\left(\frac{1}{\lambda_i}\frac{1}{\sigma_k^2}\epsilon^T + \left(\frac{\lambda_1}{\lambda_i}\frac{1}{\sqrt{\sigma_k}} + 1\right)\epsilon_P\right)$$
$$|\widehat{\lambda_i} - \lambda_i| \leq C'\left(\frac{1}{\sigma_k^{3/2}}\epsilon_T + \lambda_1 \epsilon_P\right)$$

for all $i \in [k]$.

We note that $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_k > 0$ are the non-zero singular values of the second order moments, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k > 0$ are the ground-truth eigenvalues of the third order moments, and v_i are the corresponding eigenvectors for all $i \in [k]$.

L EFFICIENT SVD USING SPARSITY AND DIMENSIONALITY REDUCTION

Without loss of generality, we assume that a matrix whose SVD we aim to compute has no row or column which is fully zeros, since, if it does have zero entries, such row and columns can be dropped.

Let $A \in \mathbb{R}^{n \times n}$ be the matrix to do SVD. Let $\Phi \in R^{d \times \tilde{k}}$, where $\tilde{k} = \alpha k$ with α is a scalar, usually, in the range $[2, 3]$. For the i^{th} row of Φ, if $\sum_i |\Phi|(i, :) \neq 0$ and $\sum_i |\Phi|(:, i) \neq 0$, then there is only one non-zero entry and that entry is uniformly chosen from $[\tilde{k}]$. If either $\sum_i |\Phi|(i, :) = 0$ or $\sum_i |\Phi|(:, i) = 0$, we leave that row blank. Let $D \in \mathbb{R}^{d \times d}$ be a diagonal matrix with iid Rademacher entries, i.e., each non-zero entry is 1 or -1 with probability $\frac{1}{2}$. Now, our embedding matrix ((Clarkson and Woodruff, 2013)) is $S = D\Phi$, i.e., we find AS and then proceed with the Nystrom ((Huang et al., 2013)) method. Unlike the usual Nystrom method ((Gittens and Mahoney, 2013a)) which uses a random matrix for computing the embedding, we improve upon this by using a sparse matrix for the embedding since the sparsity improves the running time and the memory requirements of the algorithm.