Successive Synthesis of Latent Gaussian Trees

Ali Moharrer, Shuangqing Wei, George T. Amariucai, and Jing Deng

Abstract—A new synthesis scheme is proposed to effectively generate a random vector with prescribed joint density that induces a (latent) Gaussian tree structure. The quality of synthesis is measured by total variation distance between the synthesized and desired statistics. The proposed layered and successive encoding scheme relies on the learned structure of tree to use minimal number of common random variables to synthesize the desired density. We characterize the achievable rate region for the rate tuples of multi-layer latent Gaussian tree, through which the number of bits needed to simulate such Gaussian joint density are determined. The random sources used in our algorithm are the latent variables at the top layer of tree, through which the number of bits needed to simulate such Gaussian joint density are determined. The random sources used in our algorithm are the latent variables at the top layer of tree, through which the number of bits needed to simulate such Gaussian joint density are determined. The random sources used in our algorithm are the latent variables at the top layer of tree, through which the number of bits needed to simulate such Gaussian joint density are determined. The random sources used in our algorithm are the latent variables at the top layer of tree, through which the number of bits needed to simulate such Gaussian joint density are determined. The random sources used in our algorithm are the latent variables at the top layer of tree, through which the number of bits needed to simulate such Gaussian joint density are determined.

Index Terms—Latent Gaussian Trees, Synthesis of Random Vectors, Correlation Signs, Successive Encoding

I. INTRODUCTION

Consider the problem of simulating a random vector with prescribed joint density. Such method can be used for prediction applications, i.e., given a set of inputs we may want to compute the output response statistics. This can be achieved by generating an appropriate number of random input bits to a stochastic channel whose output vector has its empirical statistics meeting the desired one measured by a given metric.

We aim to address such synthesis problem for a case where the prescribed output statistics induces a (latent) Gaussian tree structure, i.e., the underlying structure is a tree and the joint density of the variables is captured by a Gaussian density. The Gaussian graphical models are widely studied in the literature, because of a direct correspondence between conditional independence relations occurring in the model with zeros in the inverse of covariance matrix, known as the concentration matrix. They have diverse applications in social networks, biology, and economics [1], [2], to name a few. Gaussian trees in particular have attracted much attention [2] due to their sparse structures, as well as existing computationally efficient algorithms in learning the underlying topologies [3], [4]. In this paper we assume that the parameters and structure information of the latent Gaussian tree is provided.

Our primary concern in such synthesis problem is about efficiency in terms of the amount of random bits required at the input, as well as the modeling complexity of given stochastic system through which the Gaussian vector is synthesized. We use Wyner’s common information [5] to quantify the information theoretic complexity of our scheme. Such quantity defines the necessary number of common random bits to generate two correlated outputs, through a single common source of randomness, and two independent channels.

In [6], Han and Verdu introduced the notion of resolvability of a given channel, which is defined as the minimal required randomness to generate output statistics in terms of a vanishing total variation distance between the synthesized and prescribed joint densities. In [7], [8], the authors aim to define the common information of n dependent random variables, to further address the same question in this setting. Although they generalize previous works from a bi-variate case to n random variables, but, they still resort to the same scenario as Wyner [5] did, i.e., considering a single source to define such common randomness. As we will discuss, this is a very special case to our simulation scheme, in which all the variables are interconnected through a single common variable, where we define such structure as a star tree. In [9] the authors characterize the common information between two jointly Gaussian vectors, as a function of some singular values that are related to both joint and marginal covariance matrices of two Gaussian random vectors. However, they still divide the random vector into two groups, which makes it similar to a bi-variate case in Wyner’s paper [5].

In this paper, our scope generalizes all previous studies by not only considering an input vector (and not a single variable) to produce common random bits, but also, by adopting a specific (but natural) structure to our synthesis scheme we decrease the number parameters to model the synthesis scheme. It is worthy to point that the achievability results given in this paper are under the assumed structured encoding framework. Hence, although through defining an optimization problems, we show that the proposed method is efficient in terms of both modeling and codebook rates, the converse proof, which shows the optimality of such scheme and rate regions is postponed to future studies.

II. PROBLEM FORMULATION

A. The signal model of a multi-layer latent Gaussian tree

Here, we suppose a latent graphical model, with \( Y = [Y_1, Y_2, ..., Y_k]' \) as the set of inputs (hidden variables), \( B = [B_1, ..., B_m] \), with each \( B_i \in \{-1, 1\} \) being a binary Bernoulli random variable with parameter \( \pi_i = p(B_i = 1) \) to introduce sign variables, and \( X = [X_1, X_2, ..., X_n]' \) as the set of Gaussian outputs (observed variables) with \( p_B(X) \). We also assume that the underlying network structure is a latent Gaussian tree, therefore, making the joint probability (under
each sign realization) $p_{XY|B}$ be a Gaussian joint density $N(\mu, \Sigma_{XY|b})$, where the covariance matrix $\Sigma_{XY|b}$ induces tree structure $G_T(V, E, W)$, where $V$ is the set of nodes consisting of both vectors $X$ and $Y$. $E$ is the set of edges; and $W$ is the set of edge-weights determining the pairwise covariances between any adjacent nodes. We consider normalized variances for all variables $X_i \in X$, $i \in \{1, 2, ..., n\}$ and $Y_j \in Y$, $j \in \{1, 2, ..., k\}$. Such constraints do not affect the tree structure, and hence the independence relations captured by $\Sigma_{XY|b}$. Without loss of generality, we also assume $\mu = 0$, this constraint does not change the amount of information carried by the observed vector.

In [10] we showed that the vectors $X$ and $B$ are independent. We argued the intrinsic sign singularity in Gaussian trees by $\Sigma_Y$. By $\Sigma_X$ each sign realization $p_X$ chances between any adjacent nodes. We consider normalized ability distribution the noisy channel to be characterized by the conditional probability distribution of pairwise correlations. Such constraints do not affect the tree structure, beginning from a given latent Gaussian density $p_X$, which induces the same joint Gaussian distribution $\Sigma_Y$ at each layer $l$. Hence, roughly speaking, one can carefully change the sign of several correlations, and still maintain the same value for $\rho_{x_i,x_j}$. Although, this results in variation on the correlation values $\rho_{x_i,x_m}$, $n, m \in V$. We showed that if the cardinality of the input vector $Y$ is $k$, then $2^k$ minimal Gaussian trees (that only differ in sign of pairwise correlations) may induce the same joint Gaussian density $p_X$ [10].

In order to propose the successive synthesis scheme, we need to characterize the definition of layers in a latent Gaussian tree. We define latent vector $Y^{(l)}$, to be at layer $l$, if the shortest path between each latent input $Y^{(l)}_i \in Y^{(l)}$ and the observed layer (consisting the output vector $X$) is through $l$ edges. In other words, beginning from a given latent Gaussian tree, we assume the output to be at layer $l = 0$, then find its immediate latent inputs and define $Y^{(1)}$ to include all of them. We iterate such procedure till we included all the latent nodes up to layer $L$, i.e., the top layer. In such setting, the sign input vector $B^{(l)}$ with Bernoulli sign random variables $B^{(l)}_i \in B^{(l)}$ is assigned to the latent inputs $Y^{(l)}$.

We adopt a communication channel to feature the relationship between each pair of successive layers. Assume $Y^{(l+1)}$ and $B^{(l+1)}$ as the input vectors, $Y^{(l)}$ as the output vector, and the noisy channel to be characterized by the conditional probability distribution $P_{Y^{(l+1)}|Y^{(l)}, B^{(l+1)}}(y^{(l+1)}, b^{(l+1)})$, the signal model for such a channel can be written as follows,

$$Y^{(l)} = A^{(l)}_B y^{(l+1)} + Z^{(l+1)}$$

where $A_B^{(l)}$ is the $|Y^{(l)}| \times |Y^{(l+1)}|$ transition matrix that also carries the sign information vector $B^{(l)}$, and $Z^{(l+1)} \sim N(0, \Sigma_G)$ is the additive Gaussian noise vector with independent elements, each corresponding to a different communication link from the input layer $l+1$ to the output layer $l$. Hence, the outputs $Y^{(l)}$ at each layer $l$, are generated using the inputs $Y^{(l+1)}$ at the upper layer. The case $l = 0$, is essentially for the outputs in $X$, which will be produced using their upper layer inputs at $Y^{(1)}$. As we will see next, such modeling will be the basis for our successive encoding scheme. In fact, by starting from the top layer inputs $L$, at each step we generate the outputs at the lower layer, this will be done till we reach the observed layer to synthesize the Gaussian vector $X$. Finally, note that in order to take all possible latent tree structures, we need to revise the ordering of layers in certain situations, which will be taken care of in their corresponding subsections. For now, the basic definition for layers will be satisfactory.

B. Synthesis Approach Formulation

In this section we provide mathematical formulations to address the following fundamental problem: using channel inputs $Y$ and $B$, what are the rate conditions under which we can synthesize the Gaussian channel output $X$, with a given $p_X(x)$. We propose a successive encoding scheme on multiple layers that together induce a latent Gaussian tree, as well as the corresponding bounds on achievable rate tuples. The encoding scheme is efficient because it utilizes the latent Gaussian tree structure to simulate the output. In particular, without resorting to such learned structure we need to characterize $O(kn)$ parameters (one for each link between a latent and output node), while by considering the sparsity reflected in a tree we only need to consider $O(k + n - 1)$ parameters (the edges of a tree).

Suppose we transmit input messages through $N$ channel uses, in which $t \in \{1, 2, ..., N\}$ denotes the time index. We define $Y^{(l)}_t$ to be the $t$-th symbol of the $l$-th codeword, with $i \in \{1, 2, ..., M_{Y^{(l)}}\}$ where $M_{Y^{(l)}} = 2^{NR_{Y^{(l)}}}$ is the codebook cardinality, transmitted from the existing $k_l$ sources at layer $l$. We assume there are $k_l$ sources $Y^{(l)}_j$ present at the $l$-th layer, and the channel has $L$ layers. We can similarly define $B^{(l)}_l[k]$ to be the $t$-th symbol of the $k$-th codeword, with $k \in \{1, 2, ..., M_{B^{(l)}}\}$ where $M_{B^{(l)}} = 2^{NR_{B^{(l)}}}$ is the codebook cardinality, transmitted from the existing $k_l$ sources at layer $l$. For sufficiently large rates $R_Y = [R_{Y^{(1)}}, R_{Y^{(2)}}, ..., R_{Y^{(L)}}]$ and $R_B = [R_{B^{(1)}}, R_{B^{(2)}}, ..., R_{B^{(L)}}]$ and as $N$ grows the output density of synthesized channel converges to $p_X^{\infty}$, i.e., $N$ i.i.d realization of the given output density $p_X(x)$. In other words, the average total variation between the two joint densities vanishes as $N$ grows $\infty$.

$$\lim_{N \to \infty} E(||q(x_1, ..., x_N) - \prod_{t=1}^N p_X(x_t)||_TV) \to 0 \tag{2}$$

where $q(x_1, ..., x_N)$ is the synthesized channel output, and $E||\cdot||_TV$ represents the average total variation. In this situation, we say that the rates $(R_Y, R_B)$ are achievable $[11]$.

In what follows we provide the achievable rate regions to synthesize the output statistics $p_X(x)$ for two distinct cases that together cover possible varieties that may happen in latent Gaussian tree structures. For simplicity of notation, we drop the symbol index and use $Y^{(l)}_t$ and $B^{(l)}_l$ instead of $Y^{(l)}_i$ and $B^{(l)}_l[k]$, respectively, since they can be understood from the context.

III. Main Results

We derived an interesting result in [10] that shows for any Gaussian tree the mutual information is only a function of
given \( p_X(x) \sim N(0, \Sigma_x) \) and if all the output variables are leaves, it takes the following form,

\[
I(X; Y, B) = \frac{1}{2} \log \frac{\prod_{i=1}^{n} (1 - \rho_{x_i,y_i})}{\rho_{x_j,x_k}}
\]

where for each \( X_i \), we choose two other nodes \( X_j, X_k \), where all three of them are connected to each other through \( Y \) (i.e., one of their common ancestors), which is one of the hidden variables adjacent to \( X_i \).

We argue that previous studies in [7], [8] on finding the optimal solution for \( (1) \) were made available to the synthesis scheme. Then, if we choose two other nodes \( Y, X \), we will be able to cover \( \mathcal{C} \) to the desired distribution.

A. The case with observables at the same layer

In this case, we assume that the nodes at each layer are only connected to the nodes at upper/lower layers, i.e., there is no edge between the nodes at the same layer. In other words, the variables at each layer are conditionally independent of each other given the variables at their upper layer. In order to follow Wyner’s setting \([5]\) to compute the common information, such conditional independence relationships are necessary. This, in turn forms a hyper-chain structure for latent Gaussian tree, where the hyper-nodes consist of every variable at the same layer, and hyper-edges are the collection of links connecting each the nodes at each adjacent layer. Hence, one may divide the codebook \( \mathcal{C} \) into at most \( 2^4 = 16 \) parts \( S_i \), \( i \in \{1, 2, \ldots, 16\} \), in which each part follows a specific Gaussian density with covariance values \( E[Y_{k,t}^{(1)}|Y_{l,t}^{(1)}] = \gamma_{kl} b_{k,l}^{(1)} \), \( k \neq l \in \{1, 2, 3, 4\} \). Then, we can show the lower bound on the possible rates in the second layer to be,

\[
R_{Y_2(B_1)} \geq I(Y_1; Y_2, B_1)
\]

\[
R_{Y_2(B_1)} + R_{B_2(B_1)} \geq I(Y_1; Y_2, B_2|B_1)
\]

This is due to the fact that we compute subsets of codebook for each realization of \( B_1 \). The formal results on general cases will be given in Theorem 2. Let us elaborate the synthesis scheme in this case, which will serve as a foundation for our successive encoding scheme proposed next.

For a moment, only consider the latent inputs at each layer \( Y \) without being concerned about the sign issues. We begin with the top layer, where using the input \( Y_2 \) we will synthesize the outputs \( Y_1 \). The necessary number of codewords needed is \( M_{Y_2} = 2^N R_{Y_2(B_1)} \), where the rate in the exponent is lower bounded and characterized using \([5]\). Now, by randomly picking a codeword and sending it through an appropriate channels and through \( N \) channel uses, we obtain a sample
output vector $(Y^{(1)})^N$. At the next step, through which using $Y^{(1)}$ inputs the outputs $X$ are generated, we know we need to generate enough codewords, i.e., $M_{Y^{(1)}}$ that will be characterized using Theorem 2 to synthesize the output vector. However, this is done indirectly using $Y^{(2)}$, namely, the top layer inputs, since the middle layer inputs $Y^{(1)}$ are generated using the top layer inputs. Hence, to achieve the needed rate (necessary codebook size) at this iteration, we randomly draw a codeword from the top layer codebook, send it through $N$ channel uses to obtain a codeword at middle layer. Note that this time, we iterate this procedure $M_{Y^{(1)}}$ times, since we need to achieve the necessary codebook size for the inputs $Y^{(2)}$ to synthesize the output $X$. Then, due to soft covering lemma [II], this codebook size is sufficient to generate the desired output statistics for $X$, with vanishing TV distance. Although under certain circumstances, i.e., $M_{Y^{(2)}} > M_{Y^{(1)}}$, we may then pick several redundant codewords from the top layer bin, however, due to channel randomness (due to presence of the Gaussian noise), it is unlikely to get exactly the same codewords for the middle layer inputs. Figure 3 shows the described successive encoding procedure.

In general, the output at the $l$-th layer $Y^{(l)}$ is synthesized by $Y^{(l+1)}$ and $B^{(l+1)}$, which are at layer $l + 1$. This is done through iterating the successive encoding scheme described above. In particular, looking from the bottom layer (output layer), the vector $X$ is synthesized using the upper layer inputs $Y^{(1)}$. However, such inputs themselves are generated using their upper layer variables $Y^{(2)}$. This procedure is continued till we reach the top layer nodes. In particular, looking at each layer outputs, we need to consider its top two layers to achieve necessary codebook sizes and appropriate codewords in our successive encoding scheme. Therefore, we only need Gaussian sources at the top layer $L$ and independent Gaussian noises to gradually synthesize the output that is close enough to the true observable output, measured by total variation. In Theorem 2 whose proof can be found in [I] we obtain the achievable rate region for multi-layered latent Gaussian tree, while taking care of sign information as well, i.e., at each layer dividing a codebook into appropriate sub-blocks capturing each realization of sign inputs.

**Theorem 2:** For a latent Gaussian tree having $L$ layers, and forming a hyper-chain structure, the achievable rate region is characterized by the following inequalities for each layer $l$,

$$R_B^{(l+1)} + R_{Y^{(l+1)}} \geq I[Y^{(l+1)}; B^{(l+1)}; Y^{(l)}|B^{(l)}]$$

$$R_{Y^{(l+1)}} \geq I[Y^{(l+1)}; Y^{(l)}|B^{(l)}], \quad l \in [0, L - 1]$$

where $l = 0$ shows the observable layer, in which there is no conditioning needed, since the output vector $X$ is already assumed to be Gaussian.

**B. The case with observables at different layers**

Consider a case where an edge is allowed between the variables at the same layer. In this situation we violate a conditional independence constraint used in achievability proof of the basic case, since due to presence of such intra-layer links, given the upper layer inputs, the conditional independence of lower layer outputs is not guaranteed. However, again by revising the proof procedure we may show the achievability results in this case as well. To address this issue we need to reform the latent Gaussian tree structure by choosing an appropriate root such that the variables in the newly introduced layers mimic the basic scenario, i.e., having no edges between the variables at the same layer. We begin with the top layer nodes and as we move to lower layers we seek each layer for the adjacent nodes at the same layer, and move them to a newly added layer in between the upper and lower layers. This way, we introduce new layers consisting of those special nodes, but this time we are dealing with a basic case. Note that such procedure might place the output variables at different layers, i.e., all the output variables are not generated using inputs at a single layer. We only need to show that using such procedure and previously defined achievable rates, one can still simulate output statistics with vanishing total variation distance. To clarify, consider the case shown in Figure 4.

As it can be seen, there are two adjacent nodes in the first layer, i.e., $Y_3^{(1)}$ and $Y_4^{(1)}$ are connected. Using the explained
inputs, i.e., the inputs at the L-th layer, the entire set of nodes in a latent Gaussian tree can be synthesized if the rates at each layer satisfy the constraints captured in (6).

In other words, we only rely on top layer inputs $Y^{(L)}$ as a source of common randomness (and additive Gaussian noises) to synthesize the entire joint distribution $p_{X|Y:B}$ (and not only the output statistics $p_{X}$), which is arbitrarily close to the desired statistics. The key to the proof is to reform the entire Gaussian tree structure, in a way that avoids intra-layer connections between the inputs. Then, beginning from the top layer we synthesize the nodes at each layer using the upper layer inputs and necessary codebook rates. We need to keep track of input-output codewords relationships, similar to what we did in Lemma 2. In particular, considering each particular layer outputs, we keep track of the corresponding input codeword that generated such output. However, such input vector is produced by a particular codeword at its upper layer. This procedure should always hold from the top to the bottom of latent Gaussian tree, in order to keep a valid joint dependency among the variables at every layer.

IV. CONCLUSION

In this paper, we proposed a new tree structure synthesis scheme, in which through layered forwarding channels certain Gaussian vectors can be efficiently generated. Our layered encoding approach is shown to be efficient and accurate in terms of reduced required number of parameters and random bits needed to simulate the output statistics, and its closeness to the desired statistics in terms of total variation distance.

REFERENCES

[1] A. Dobra, T. S. Eicher, and A. Lenkoski, “Modeling uncertainty in macroeconomic growth determinants using gaussian graphical models,” Statistical Methodology, vol. 7, no. 3, pp. 292–306, 2010.
[2] R. Mourad, C. Sinoquet, N. L. Zhang, T. Liu, P. Leray et al., “A survey on latent tree models and applications,” J. Artif. Intell. Res.(JAIR), vol. 47, pp. 157–203, 2013.
[3] M. J. Choi, V. Y. Tan, A. Anandkumar, and A. S. Willsky, “Learning latent tree graphical models,” The Journal of Machine Learning Research, vol. 12, pp. 1771–1812, 2011.
[4] N. Shiers, P. Zwiernik, J. A. Aston, and J. Q. Smith, “The correlation space of gaussian latent tree models,” arXiv preprint arXiv:1508.00436, 2015.
[5] A. D. Wyner, “The common information of two dependent random variables,” Information Theory, IEEE Transactions on, vol. 21, no. 2, pp. 163–179, 1975.
[6] T. S. Han and S. Verdú, “Approximation theory of output statistics,” IEEE Transactions on Information Theory, vol. 39, no. 3, pp. 752–772, 1993.
[7] P. Yang and B. Chen, “Wyner’s common information in gaussian channels,” in IEEE International Symposium on Information Theory (ISIT), 2014, pp. 3112–3116.
[8] G. Xu, W. Liu, and B. Chen, “A lossy source coding interpretation of wyner’s common information,” Information Theory, IEEE Transactions on, vol. 62, no. 2, pp. 754–768, 2016.
[9] S. Satpathy and P. Cuff, “Gaussian secure source coding and wyner’s common information," arXiv preprint arXiv:1506.00193, 2015.
[10] A. Moharrer, S. Wei, G. T. Amariucai, and J. Deng, “Synthesis of Gaussian Trees with Correlation Sign Ambiguity: An Information Theoretic Approach,” in Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on. IEEE, Oct. 2016. [Online]. Available: “http://arxiv.org/abs/1601.06403”
[11] P. Cuff, “Distributed channel synthesis,” Information Theory, IEEE Transactions on, vol. 59, no. 11, pp. 7071–7096, 2013.
This can be proven using simple matrix operations. Since the covariance matrix $\Sigma_x$ induces a latent star structure, we have the system of equations $\rho_{x,i} = \rho_{x,y} \rho_{x,y}$ for all $i \neq j$. We may expand $|\Sigma_x|$ and factorize $\rho_{x,y}$ from both row and column $i$, for all $i \in [1, n]$, $n \geq 3$. Then, we deduce the following,

$$|\Sigma_x| = \prod_{i=1}^{n} \left| \begin{array}{cc}
\rho_{x,y}^2 & 1 \\
1 & 1/\rho_{x,y}^2 & 1 \\
\vdots & \vdots & \ddots & \vdots \\
1 & 1 & \cdots & 1/\rho_{x,y}^2
\end{array} \right|
$$

By subtracting the first row from all other rows, and noting that the determinant remains invariant under such operation, we have:

$$|\Sigma_x| = \prod_{i=1}^{n} \rho_{x,y}^2 \prod_{i \neq j}^{n} (1/\rho_{x,y}^2 - 1) + \prod_{i \neq j}^{n} (1/\rho_{x,y}^2 - 1)
$$

By expanding the determinant using the first row, we have,

$$|\Sigma_x| = \prod_{i=1}^{n} \rho_{x,y}^2 \prod_{i \neq j}^{n} (1/\rho_{x,y}^2 - 1) + \prod_{i \neq j}^{n} (1/\rho_{x,y}^2 - 1)
$$

where the first term is the $1/\rho_{x,y}^2 \times C_{11}$, where $C_{11}$ is the first cofactor of $\Sigma_x$, and it is simply computed, since it becomes a lower triangular matrix. The second term is $C_{12}$ and it is again the determinant of a lower triangular matrix. The third term is $C_{13}$, and after exchanging the first two columns of it, it becomes a lower triangular matrix. For the fourth term, and $C_{14}$ we exchange the first three rows twice, to make it a lower triangular matrix. Other terms are obtained using similar procedure. We can rewrite the above equation as:

$$|\Sigma_x| = \left(1 - \frac{1}{\rho_{x,y}^2} \sum_{i=2}^{n} \frac{\rho_{x,y}^2}{1 - \rho_{x,y}^2} \right) \prod_{i=1}^{n} (1 - \rho_{x,y}^2)
$$

$$= \left(1 + \frac{\rho_{x,y}^2}{1 - \rho_{x,y}^2} \right) \prod_{i=1}^{n} (1 - \rho_{x,y}^2)
$$

By (7) we may deduce the mutual information as follows,

$$I(\mathbf{X}; \mathbf{Y}) = 1/2 \log \left( \prod_{i=1}^{n} (1 - \rho_{x,y}^2) \right)
$$

$$= 1/2 \log [1 + \sum_{i=1}^{n} \rho_{x,y}^2 - 1/\rho_{x,y}^2]
$$

where the results in (8) is a special case to this formula where $\rho_{x,y} = \rho$, for all $i \in [1, n]$.
\( I(\mathbf{X}, \mathbf{Y}) = h(\mathbf{X}) - h(\mathbf{X} | \mathbf{Y}) \), and knowing that given \( \Sigma \) the entropy \( h(\mathbf{X}) = 1/2 \log(2\pi e)^n | \Sigma | \) is fixed, we only need to show that the conditional entropy \( h(\mathbf{X} | \mathbf{Y}) \) is a concave function of \( \eta \). Using definition of entropy and by replacing for \( p_{\mathbf{X} \mathbf{Y}} \) and \( p_{\mathbf{Y}} \) using equations (7) and (8), respectively, we may characterize the conditional entropy. By taking second order derivative, we deduce the following,

\[
\frac{\partial^2 h(\mathbf{X} | \mathbf{Y})}{\partial \eta_i \partial \eta_j} = -\int \int \frac{f_i(\mathbf{x}, \mathbf{y}) f_j(\mathbf{x}, \mathbf{y})}{p_{\mathbf{X} \mathbf{Y}}} dx dy + \int \int \frac{\tilde{g}_i(\mathbf{y}) \tilde{g}_j(\mathbf{y})}{p_{\mathbf{Y}}} dy
\]

where for simplicity of notations we write \( \eta_i \) instead of \( \eta_{\mathbf{B}_i} \). Also, \( \tilde{g}_i(\mathbf{y}) = \bar{g}_i(\mathbf{y}) = g_i(\mathbf{y}) \) for \( i \in [0, 2^{k-1} - 1] \). Note the following relation,

\[
\int \int \frac{\tilde{g}_i(\mathbf{y}) f_j(\mathbf{x}, \mathbf{y})}{p_{\mathbf{X} \mathbf{Y}}} dx dy = \int \int \frac{\tilde{g}_i(\mathbf{y}) f_j(\mathbf{x}, \mathbf{y})}{p_{\mathbf{Y}}} dx dy = \int \int \frac{\bar{g}_i(\mathbf{y}) f_j(\mathbf{x}, \mathbf{y})}{p_{\mathbf{Y}}} dx dy
\]

The matrix \( H = [h_{ij}] \), \( i, j \in [0, 2^{k-1} - 1] \) characterizes the Hessian matrix the conditional entropy \( h(\mathbf{X} | \mathbf{Y}) \). To prove the concavity, we need to show \( H \) is non-positive definite. Define a non-zero real row vector \( \mathbf{c} \in \mathbb{R}^{2^k} \), then we need to form \( \mathbf{c}^T H \mathbf{c} \) as follows and show that it is non-positive.

\[
\mathbf{c}^T H \mathbf{c} = -\int \int \frac{1}{p_{\mathbf{X} \mathbf{Y}}} \sum_{i=0}^{2^k-1} \sum_{j=0}^{2^k-1} c_i c_j [f_i(\mathbf{x}, \mathbf{y}) - \tilde{g}_i(\mathbf{y}) p_{\mathbf{X} | \mathbf{Y}}] [f_j(\mathbf{x}, \mathbf{y}) - \tilde{g}_j(\mathbf{y}) p_{\mathbf{X} | \mathbf{Y}}] dx dy
\]

\[
= -\int \int \frac{1}{p_{\mathbf{Y}}} \sum_{i=0}^{2^k-1} [c_i (f_i(\mathbf{x}, \mathbf{y}) - \tilde{g}_i(\mathbf{y}) p_{\mathbf{X} | \mathbf{Y}})]^2 dx dy \leq 0
\]

Now that we showed the concavity of the conditional entropy with respect to \( \eta_i \), we only need to find the optimal solution. The formulation is defined in (14), where \( \lambda \) is the Lagrange multiplier.

\[
L = h(\mathbf{X} | \mathbf{Y}) - \lambda \sum_{i=0}^{2^k-1} \eta_i
\]
By negating all \(y_1, \ldots, y_i\) into \(-y_1, \ldots, -y_i\), it is apparent that \(t, \sum_{i \in \mathcal{L}_p} y_i\), and \(s\) do not change. Also, the terms in \(q\) either remain intact or doubly negated, hence, overall \(q\) remains intact also. However, by definition, \(p_i, i \in \mathcal{L}\) will be negated, hence overall the sum \(\sum_{i \in \mathcal{L}_p} y_i\) will be negated. The same thing can be argued for \(q\)', since exactly one variable \(y_i\) or \(y_j\) in the summation, will change its sign, so \(q'\) also will be negated. Overall, we can see that by negating \(y_1, \ldots, y_i\), we will negate \(f_i - f_j\). It remains to show that such negation does not impact \(p_{XY}\). Note that since \(p_{XY}\) includes all \(2k\) sign combinations and all of \(f_i(x, y)\) are equi-probable since \(\sum_{i=1} \sum_{j=1}^{k} f_i(x, y) \log p_{XY}(x, y)\) we can find its negation, hence making the integrand an odd function, and the corresponding integral zero. Hence, making the solution \(\eta = 1/2^k\), for all \(i \in [0, 2^k - 1]\) an optimal solution.

The only thing remaining is to show that from \(\eta = 1/2^k\) we may conclude that \(\pi_j = 1/2\) for all \(j \in [1, k]\). By definition, we may write,

\[
\eta_j = \prod_{j=1}^{k} \pi_j \sum_{i=1}^{2^k} (1 - \pi_j)^{1-b_{ij}},
\]

where \(b_{ij} \in B_i\). Assume all \(\eta_j = 1/2^k\). Consider \(\eta_j\) and find \(\eta_{j'}\) such that the two are different in only one expression, say at the \(l\)-th place. Since, all \(\eta_j\) are equal, one may deduce \(1 - \pi_l = \pi_l\), so \(\pi_l = 1/2\). Note that such \(\pi_l\) can always be found since \(\eta_j\)'s are covering all possible combinations of \(k\)-bit vector. Now, find another \(\eta_{j''}\), which is different from \(\eta_j\) at some other spot, say \(l''\), again using similar arguments, we may show \(\pi_{l''} = 1/2\). This can be done \(k\) times to show that, if all \(\eta_j = 1/2^k\), then \(\pi_1 = \ldots = \pi_k = 1/2\). This completes the proof.

**APPENDIX C**

**PROOF OF THEOREM**

The signal model can be directly written as follows,

\[
Y^{(l)} = A_{B^{(l+1)}} Y^{(l+1)} + Z^{(l)} \tag{18}
\]

Here, we show the encoding scheme to generate \(Y^{(l)}\) from \(Y^{(l+1)}\). Note that \(Y^{(l)}\) is a vector consisting of the variables \(Y_i^{(l)}\). Also, \(Y^{(l+1)}\) is a vector consisting of variables \(Y_i^{(l+1)}\). The proof relies on the procedure taken in [11]. Note that our encoding scheme should satisfy the following constraints, 1) \(Y_i^{(l+1)} Y_{-i}^{(l+1)} = Y_i^{(l+1)} Y_{-i}^{(l+1)}\), 2) \(Y_i^{(l)} Y_{-i}^{(l)} B^{(l)} Y_{-i}^{(l)}\), 3) \(Y_i^{(l)} B^{(l)} Y_{-i}^{(l)}\) are i.i.d ∼ \(P_{Y_{i}}(Y_i^{(l)})\) 4) \(Y_i^{(l+1)} = 2^N R_{Y_{i}^{(l+1)}}\), 5) \(B^{(l+1)} = 2^N R_{Y_{i}^{(l+1)}}\), 6) \(|g_{Y_{i}^{(l)}} - \prod_{i=1}^{N} P_{Y_{i}^{(l)}}(Y_i^{(l)})||TV < \epsilon\)

where the first constraint is conditioned to the conditional independence assumption characterized in the signal model [13]. The second one is to capture the intrinsic ambiguity of the latent Gaussian tree to capture the sign information. Condition 3) is due to the assumption of a given Gaussian density \(P_{Y_{i}^{(l)}} B^{(l)} Y_{-i}^{(l)} ~ N(0, \Sigma_{Y_{i}^{(l)}} B^{(l)} Y_{-i}^{(l)})\) for the output vector. Conditions 4) and 5) are due to corresponding rates for each of the inputs \(Y_i^{(l)}\) and \(B^{(l)}\). And finally, condition 6) is the synthesis requirement to be satisfied. First, we generate a codebook \(C\) of \(y_{b_{i}^{(l)}}\) sequences, with indices \(y \in C_{Y_{i}^{(l)}} = \{1, 2, \ldots, 2^N R_{Y_{i}^{(l+1)}}\}\) and \(b \in C_{B_{i}^{(l+1)}} = \{1, 2, \ldots, 2^N R_{B_{i}^{(l+1)}}\}\) according to \(\prod_{i=1}^{N} P_{Y_{i}^{(l)}}(Y_i^{(l)})\) as depicted by Figure 6

![Fig. 6: Construction of the joint density \(\gamma(Y_i^{(l)} N, Y_{i}^{(l+1)}, B^{(l+1)})\)](image)

The indices \(y_{i}^{(l)}\) and \(b\) are chosen independently and uniformly from the codebook \(C\). As can be seen from Figure 6 the channel \(Y_i^{(l)}|Y_{-i}^{(l)}\) is in fact consists of \(n\) independent channels \(Y_i^{(l)}|Y_{-i}^{(l)}\), \(i \in \{1, 2, \ldots, n\}\). The joint density is as follows,

\[
\gamma(Y_i^{(l)} N, Y_{i}^{(l+1)}, B^{(l+1)}) = \frac{1}{|C_{Y_{i}^{(l)}}||C_{B_{i}^{(l+1)}}|} \prod_{i=1}^{N} P_{Y_{i}^{(l)}}(Y_i^{(l)}|\hat{y}_i(y, b))\]

Note that \(\gamma(Y_i^{(l)} N, Y_{i}^{(l+1)}, B^{(l+1)})\) already satisfies the constraints 1), 4), and 5) by construction. Next, we need to show that it satisfies the constraint 6). The marginal density \(\gamma(Y_i^{(l)} N\) can be deduced by the following,

\[
\gamma(Y_i^{(l)} N = \frac{1}{|C_{Y_{i}^{(l)}}|C_{B_{i}^{(l+1)}}} \sum_{y_{i}^{(l)} \in C_{Y_{i}^{(l)}}} \sum_{b \in C_{B_{i}^{(l+1)}}} \prod_{i=1}^{N} P_{Y_{i}^{(l)}}(Y_i^{(l)}|\hat{y}_i(y, b))\]

We know if \(R_{B_{i}^{(l+1)}} + R_{Y_{i}^{(l+1)}} \geq 1[I(Y_{i}^{(l+1)}, B_{i}^{(l+1)}; Y_{i}^{(l)}|B_{i}^{(l)})\]

then by soft covering lemma [11] we have,

\[
\lim_{n \to \infty} E[\gamma(Y_i^{(l)} N) - \prod_{i=1}^{N} P_{Y_{i}^{(l)}}||TV = 0 \tag{19}
\]

which shows that \(\gamma(Y_i^{(l)} N\) satisfies constraint 6). For simplicity of notations we use \(\prod_{i=1}^{N} P_{Y_{i}^{(l)}}\) instead of \(\prod_{i=1}^{N} P_{Y_{i}^{(l)}}(Y_i^{(l)})\), since
it can be understood from the context. Next, let’s show that \( \gamma(y_0) \), nearly satisfies constraints 2 and 3). We need to show that as \( N \) grows the synthesized density \( \gamma(y_0)^N, B^{l+1} \) approaches \( \frac{1}{|C_B|} \prod P_{Y_0}(l) \), in which the latter satisfies both 2 and 3). In particular, we need to show that the total variation \( E[|\gamma(y_0)^N, B^{l+1} - \frac{1}{|C_B|} \prod P_{Y_0}(l)|] \) vanishes as \( N \) grows. After taking several algebraic steps similar to the ones in [1], we should equivalently show that the following term vanishes, as \( N \to \infty \),

\[
\frac{1}{|C_B|} \sum_{b \in C_B} E[|\gamma(y_0)^N, B^{l+1} = b - \prod P_{Y_0}(l)|] \quad \text{(20)}
\]

Note that given any fixed \( b \in C_B \) the number of Gaussian codewords is \( |C_Y| = 2^{NR_0^0(b)} \). Also, one can check by the signal model defined in [13] that the statistical properties of the output vector \( Y^{l+1} \), given any fixed signal value \( b \in C_B \) does not change. Hence, for sufficiently large rates, i.e., \( R_{Y_0(l+1)} \geq I[Y^{l+1}; Y_0^{B^{l+1}}] \), and by soft covering lemma, the term in the summation in (20) vanishes as \( N \) grows. So overall the term shown in (20) vanishes. This shows that in fact \( \gamma(y_0)^N \) nearly satisfies the constraints 2 and 3). Hence, let’s construct another distribution using \( \gamma(y_0)^N, Y^{l+1}, B^{l+1} \). Define,

\[
\gamma(y_0)^N, Y^{l+1}, B^{l+1} = \frac{1}{|C_B|} \prod P_{Y_0}(l) \gamma(y_0)^N, Y^{l+1}, B^{l+1} \quad \text{(21)}
\]

It is not hard to see that such density satisfies 1) – 5). We only need to show that it satisfies 6) as well. We have,

\[
|\gamma(y_0)^N = - \prod P_{Y_0}(l)| \quad \text{TV}
\]

\[
\leq |\gamma(y_0)^N - \gamma(y_0)^N| \quad \text{TV} + |\gamma(y_0)^N - \prod P_{Y_0}(l)| \quad \text{TV}
\]

\[
\leq |\gamma(y_0)^N, Y^{l+1}, B^{l+1} - \gamma(y_0)^N, Y^{l+1}, B^{l+1} + \gamma(y_0)^N, Y^{l+1}, B^{l+1} + | \quad \text{TV} + \epsilon_N
\]

\[
= |\gamma(y_0)^N, B^{l+1} - \gamma(y_0)^N, B^{l+1} + | \quad \text{TV} + \epsilon_N
\]

\[
= \frac{1}{|C_B|} \prod P_{Y_0}(l) - \gamma(y_0)^N, B^{l+1} + | \quad \text{TV} + \epsilon_N
\]

where \( \epsilon_N = |\gamma(y_0)^N - \prod P_{Y_0}(l)| \). Both terms in (23) vanish as \( N \) grows, due to (20) and (19), respectively. Note that, (21) is due to [1] Lemma VII. Also, (22) is due to [1] Lemma V.II, by considering the terms \( \gamma(y_0)^N, Y^{l+1}, B^{l+1} \) and \( \gamma(y_0)^N, Y^{l+1}, B^{l+1} \) as the outputs of a unique channel specified by \( \gamma(y_0)^N, Y^{l+1}, B^{l+1} \), with inputs \( P_{Y_0}(l) B^{l+1} \) and \( \gamma(y_0)^N, B^{l+1} \), respectively.

Finally, note that we synthesize each \( Y(l) \) for a given \( B(l) = b \). Hence, to obtain the overall statistics we have \( q_{Y(l)} = \sum_b q_{Y(l), B(l)} P(B(l) = b) \), where the summation is over all possible sign combinations for layer \( Y(l) \), which equals to \( 2^{2^{Y(l)}} = 2^{N} \). Certainly, this number becomes exponentially large if \( N \) is large. However, note that as \( N \to \infty \) each synthesized output (for each given \( B(l) = b \)) become arbitrarily close to zero. Hence, overall \( q_{Y(l)} \) becomes arbitrarily close to the desired statistics. This is also the case for the overall latent Gaussian tree, i.e., for \( L \) capturing the total number of layers, at each layer we can generate an output with vanishing total variation distance from the desired statistics, hence overall the final output statistics becomes arbitrarily close to the desired output statistics.

This completes the achievability proof.

**Appendix D**

**Proof of Lemma 2**

First, we need to change the latent tree structure in a way similar to Figure 2. We start from the standard latent structure, and at each layer we seek for those latent nodes that are at the same layer and they are neighbors. For each pair of adjacent nodes, we move the one that is further away from the top layer to a new added layer below the current one. Hence, make a new layer of latent nodes. We iterate this step until we reach the bottom layer. This way, we face different groups of observables being synthesized at different layers.

Define \( X_l(l), Y_l(l) \) and \( B_l(l) \) as the set of observables, latent nodes and sign variables at layer \( l \), respectively. In this new setting layer \( l = 0 \) defines the observable layer, which only consists of remaining output variables, with no latent nodes. If the rates at each layer satisfy the inequalities in (6), then by Theorem 2 we know that as \( N \) increases, the simulated density \( q_l(X_l(l), Y_l(l)) \) approaches to the desired density \( P_l(X_l(l), Y_l(l)) \). Suppose the first set of outputs are generated at layer \( L' \) then we know \( X = \bigcup_{l=0}^{L'} X_l(l) \). Each observable node \( X_l(l) \), for \( l < L' \) has a latent ancestor at each layer \( l < l' \leq L' \). We define \( Y \) as the union of latent nodes containing all those latent ancestors. Basically, the vector \( Y \) includes all the latent nodes \( Y_l(l) \) for \( 1 \leq l \leq L' \). We define \( B \), similarly, i.e., those sign inputs related to the nodes in the set \( Y \). With slightly abuse of notation, define \( Y = \{Y', B'\} \), and \( Y(l) = \{Y(l), B(l)\} \), for all possible layers \( l \). The encoding scheme looks exactly as discussed previously, except that this time we need to keep track of corresponding generated outputs at each layer and match them together. In particular, consider the generated outputs \( (X(l)) \), which lie at the bottom layer. Each output is generated using a particular input vector \( Y(l) \), which in turn along with other possible outputs \( (X(l)) \) are generated by a unique input codeword \( Y_0(l) \) that lie at the second layer. This procedure moves from the bottom to the top layer, in order to match each generated output at the bottom layer with the correct output vectors at other layers. Note that the sign information will be automatically taken care of, since similar to the previous cases, at each layer \( l+1 \) and given each realization of the sign vector \( B(l) = b(l) \), the input vector \( Y_0(l+1) \) will become Gaussian. We only need to show that the synthesized density regarding to such formed joint vectors approaches to the desired output density, as \( N \) grows.

By the underlying structure of latent tree, one may factorize the joint density \( q_N, \tilde{Y} \) as

\[
q_N, \tilde{Y} = q_l(X(l')\tilde{Y}((l'))_N \prod_{l=0}^{L'-1} q_l(X(l))_N|Y(l)+1)_{N}
\]

Note that the desired joint density \( p_{X, \tilde{Y}} \) also induces the same latent Gaussian tree, hence, we may write,

\[
p_{X, \tilde{Y}} = P_l(X(l')\tilde{Y}((l'))_N \prod_{l=0}^{L'-1} P_l(X(l))_N|Y(l)+1)_{N}
\]
However, by our encoding scheme shown in Figure 6, one may argue that $\prod_{l=0}^{L'-1} q_{X(l)}^N |(Y_l)_{(l+1)}^N = \prod_{l=0}^{L'-1} p_{X(l)}^N |(\tilde{Y}_{l+1})^N = \prod_{l=0}^{L'-1} p_{X(l)}^N |(Y_{l+1})^N$. By summing out $(B^{(L')})^N$ from both densities $p_{X_N,\tilde{Y}_N}$ and $q_{X_N,\tilde{Y}_N}$, we may replace $p_{X_N,\tilde{Y}_N}$ with $p_{X_N, (Y_{(l')})^N}$, $q_{X_N,\tilde{Y}_N}$ with $q_{X_N, (Y_{(l')})^N}$, since only these terms in the equations depend on the sign vector at layer $L'$, i.e., $(B^{(L')})^N$. Now, by previous arguments for the synthesized and desired density at layer $L'$, we know that the total variation distance $\|q_{X_N, (Y_{(l')})^N} - \prod_{l=0}^{L'-1} p_{X_N, (Y_{(l')})^N} \|_{TV}$ goes to zero as $N$ grows. Hence, one may simply deduce that $\|q_{X_N,\tilde{Y}_N/((B^{(L')})^N) - \prod_{l=0}^{L'-1} p_{X_N, (Y_{(l')})^N} \|_{TV} = \|q_{X_N, (Y_{(l')})^N} - \prod_{l=0}^{L'-1} p_{X_N, (Y_{(l')})^N} \|_{TV}$ goes to zero as $N$ grows. Due to [11, Lemma V.I], we know $\|q_{X_N} - \prod p_{X_N} \|_{TV} \leq \|q_{X_N,\tilde{Y}_N/((B^{(L')})^N) - \prod p_{X_N,\tilde{Y}_N/((B^{(L')})^N) \|_{TV} < \epsilon$, and as $N$ grows. This completes the proof.