VFG: Variational Flow Graphical Model with Hierarchical Latent Structure

Anonymous Author
Anonymous Institution

Abstract

This paper introduces a novel approach to embed flow-based models with hierarchical latent data structures. The proposed model uncovers the latent relational structures of high dimensional data via a message-passing scheme through the careful integration of normalizing flows in variational graphs. Meanwhile, the model can generate data representations with reduced latent dimensions, thus overcoming the drawbacks of many flow-based models, usually requiring a high dimensional latent space involving many trivial variables. With aggregation nodes, the models provide a convenient approach for data integration. Theoretical analysis and numerical experiments on synthetic and real datasets show the benefits and broad potentials of our proposed method.

1 Introduction

Graphical models [20, 10] are potent tools to combine the particular structure of a graph and probabilistic modeling, which provides a probabilistic (and hierarchical) characterization of variables. Due to their flexibility and ability to effectively learn and perform inference in large networks [17], they have attracted lots of interest. They have been applied in many fields, *e.g.* speech recognition [3], Quick Medical Reference (QMR) model [26] and energy-based model [11]. The quantity of interest in such models is the marginal distribution of the observed data, also known as the incomplete likelihood, noted $p(\mathbf{x})$. Most statistical learning tasks involve a parameterized model and their training procedure involves computing the maximum likelihood estimate defined as $\theta^* := \arg\max_{\theta \in \mathbb{R}^d} p_\theta(\mathbf{x})$. A direct consequence of Bayes rule, which reads $p_\theta(\mathbf{x}|\mathbf{z}) = p_\theta(\mathbf{z}, \mathbf{x})/p_\theta(\mathbf{x})$, is that the maximization of such likelihood $p_\theta(\mathbf{x})$ in a parameterized model is closely related to the inference of the density $p_\theta(\mathbf{x}|\mathbf{z})$, as a subroutine during the training procedure. Note that in the above, $\mathbf{z}$ is the latent variable and $p(\mathbf{x}, \mathbf{z})$ is the joint distribution of the complete data comprised of the observations x and of z.

The focus of this paper is mostly on this graphical inference subroutine. There are two general approaches for this task: *exact inference* and *approximate inference.* (*i*) Exact inference, *e.g.* Elimination Algorithm [25] and Junction Tree Algorithm [13], resorts to an exact numerical calculation procedure of the quantity of interest. However, in most cases, exactly inferring from $p_\theta(\mathbf{x}|\mathbf{z})$ is either *computationally involved* or simply *intractable.* It is the case for modern graphical models designed for complex tasks with deep neural networks. However, it can be empirically observed that the distribution can be well determined by a small cluster of nodes in the network, see [12]. There exist a trade-off between exact inference and light computations as the accuracy achieved by the exact inference is not worth the computational cost in some cases. (*ii*) In contrast, approximate inference, *e.g.* variational inference, yields an approximation procedure that generally provides bounds on the probability density functions (pdfs) of interest without never attaining them. Despite such approximation and considering slow convergence issues of stochastic MCMC procedure [24], we opt for the deterministic Variational Inference (VI) approach to tackle the graphical inference problem. VI is computationally efficient using off-the-shelf optimization techniques and is easily applicable to large datasets [9, 15, 19]. In Variational Inference, mean-field approximation [30] and variational message passing [29] are two common approaches for graphical models. Those methods leverage families of simple and tractable distributions to approximate the intractable posterior $p(\mathbf{z}|\mathbf{x})$. However, such approximation is limited by the choice of distributions that are inherently unable to recover the true posterior, often leading to a loose lower bound. They also often lack a flexible structure to learn the intrinsic disentangled latent representation.

Preliminary work. Under review by AISTATS 2021. Do not distribute.

Dealing with high dimensional data using graphical models exacerbates this systemic inability to model the latent structure of the data efficiently. Motivated by these significant limitations, we propose a new framework, a variational hierarchical graphical flow model, and list our contributions as follows:

- **Normalizing Flows:** A normalizing flow is introduced in the variational inference task on the original hierarchical latent data model. The result is a richer and tractable posterior distribution used as an approximation of the true posterior.
- **Hierarchical and Flow-Based:** Introducing the Variational Flow Graphical (VFG) model, we propose a novel graph architecture borrowing ideas from the *hierarchical latent data* modeling and *normalizing flow* concept to uncover the underlying complex structure of high dimensional data without any posterior sampling step required in existing variational models.
- **Numerical Applications:** We highlight the benefits of our VFG model on two main applications: – the graph missing entries imputation problem and – the disentanglement learning task where we specifically demonstrate that our model achieves to disentangle the factors of variation underlying the high dimensional data given as input.

Section 2 presents concepts such as normalizing flows, VI, and variational graphical models. Section 3 introduces the Variational Flow Graphical Model (VFG) model to tackle the latent relational structure learning of high dimensional data. Section 4 corresponds to our theoretical findings. Section 5 showcases the advantage of VFG on various tasks: missing values imputation on both synthetic and real datasets, and disentanglement learning. The Appendix is devoted to proofs and further analysis.

Notations: We denote by $[L]$ the set $\{1, \cdots, L\}$, for all $L > 1$, and by $\mathbf{KL}(p||q) := \int_{\mathcal{Z}} p(z) \log(p(z)/q(z))\mathrm{d}z$ the Kullback-Leibler divergence from q to p, two probability density functions defined on the set $\mathcal{Z} \subset \mathbb{R}^d$ for any dimension $d > 0$.

2 Preliminaries

In this section, we first introduce the general principles and notations of normalizing flows and variational inference. Then, we explain how they can naturally be embedded with graphical models.

Normalizing Flows: Normalizing flows [16, 22] is a transformation of a simple probability distribution into a more complex distribution by a sequence of invertible and differentiable mappings, noted $\mathbf{f} : \mathcal{Z} \to \mathcal{X}$ between two random variables $z \in \mathcal{Z}$ of density $p(\mathbf{z})$ and $x \in \mathcal{X}$. Firstly introduced by [28] for single maps, it has been popularized in [7, 23] with deep neural networks for variational inference [22]. Flow-based models [7, 6, 5, 8, 21] are attractive approaches for density estimation as they result in better performance enjoying the exact inference capability at a *low computational cost.* The observed variable $\mathbf{x} \sim p_\theta(\mathbf{x})$ is assumed to be distributed according to an unknown distribution $p_\theta(\mathbf{x})$ parameterized by a user-designed model θ. We focus on a finite sequence of transformations $\mathbf{f} := \mathbf{f}_1 \circ \mathbf{f}_2 \circ \cdots \circ \mathbf{f}_L$ such that, $\mathbf{x} = \mathbf{f}(\mathbf{z})\,, \quad \mathbf{z} = \mathbf{f}^{-1}(\mathbf{x})$ and $\mathbf{z} \underset{\mathbf{f}_1^{-1}}{\overset{\mathbf{f}_1}{\rightleftarrows}} \mathbf{h}^1 \underset{\mathbf{f}_2^{-1}}{\overset{\mathbf{f}_2}{\rightleftarrows}} \mathbf{h}^2 \cdots \underset{\mathbf{f}_L^{-1}}{\overset{\mathbf{f}_L}{\rightleftarrows}} \mathbf{x}$. By defining the aforementioned invertible maps $\{f_\ell\}_{\ell=1}^L$, and by the chain rule and inverse function theorem, the variable $\mathbf{x} = \mathbf{f}(\mathbf{z})$ has a tractable probability density function (pdf) given as:

$$\begin{aligned}\log p_\theta(\mathbf{x}) &= \log p(\mathbf{z}) + \log|\det(\frac{\partial \mathbf{z}}{\partial \mathbf{x}})| \qquad (1)\\ &= \log p(\mathbf{z}) + \sum_{i=1}^{L} \log|\det(\frac{\partial \mathbf{h}^i}{\partial \mathbf{h}^{i-1}})|\,,\end{aligned}$$

where we have $\mathbf{h}^0 = \mathbf{x}$ and $\mathbf{h}^L = \mathbf{z}$ for conciseness. The scalar value $\log|\det(\partial \mathbf{h}^i/\partial \mathbf{h}^{i-1})|$ is the logarithm of the absolute value of the determinant of the Jacobian matrix $\partial \mathbf{h}^i/\partial \mathbf{h}^{i-1}$, also called the log-determinant. Identity (1) yields an easy mechanism to build families of distributions that, from an initial density and a succession of invertible transformations, returns tractable density functions that one can sample from (by sampling from the initial density and applying the transformations).

Variational Inference: Following the setting discussed above, the functional mapping $\mathbf{f}$: $\mathbf{x} \to \mathbf{z}$ can be viewed as an encoding process and the mapping $\mathbf{f}^{-1}$: $\mathbf{z} \to \mathbf{x}$ as a decoding one: $\mathbf{z} \sim p(\mathbf{z}), \mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z})$. To learn the vector of parameters θ, we maximize the following marginal log-likelihood $\log p_\theta(\mathbf{x}) = \log \int p(\mathbf{z}) p_\theta(\mathbf{x}|\mathbf{z}) d\mathbf{z}$. Direct optimization of the log-likelihood is usually not an option due to the intractable latent structure. Instead VI employs a parameterized family of so-called variational distributions $q_\phi(\mathbf{z}|\mathbf{x})$ to approximate the true posterior $p_\theta(\mathbf{z}|\mathbf{x}) \propto p(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})$. The goal of VI is to minimize the distance, in terms of Kullback-Leibler (KL), between the variational candidate and the true posterior $\mathbf{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x}))$. This optimization problem can be shown to be equivalent to maximizing the following evidence lower bound (ELBO) objective, noted $\mathcal{L}(\mathbf{x};\theta)$:

$$\log p_\theta(\mathbf{x}) \geqslant \mathcal{L}(\mathbf{x};\theta) = E_{p_\theta(\mathbf{x})}\{E_{q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - \mathbf{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))\}\,. \quad (2)$$

Variational Graphical Models: In Directed Acyclic Graph (DAG) models, each node $\mathbf{v}$ corresponds to a random variable, *e.g.* $\mathbf{v}$ include the latent variables $\mathbf{z}$ and observed variables $\mathbf{x}$ in the variational framework. The edges represent the statistical dependencies between the variables, *e.g.*, a function $\mathbf{f}_\theta$ parameterized by θ, which serves as a link function between two variables. The joint distribution of the model is given by $p_\theta(\mathbf{v}) = \prod_{\mathbf{v}\in\mathcal{V}} p_\theta(\mathbf{v}|pa(\mathbf{v}))$, where $\mathbf{v} = (\mathbf{z}, \mathbf{x})$, $\mathcal{V}$ is a sample space for all graph variables and $pa(\mathbf{v})$ denotes the parent node of $\mathbf{v}$. The goal of variational Bayesian networks, as a special instance of variational graphical models, is to find a variational distribution, noted $q(\mathbf{z}|\mathbf{x})$, approximating $p(\mathbf{z}|\mathbf{x})$. In this paper, we focus on the factorization of the independent and disjoint latent variables [4] such that $q(\mathbf{z}|\mathbf{x}) = \prod_i q_i(\mathbf{z}_i)$, where $\mathbf{z}_i$ is the latent variable at node i of the graph, assuming that $\mathbf{x} = pa(\mathbf{z}_i)$.

3 Variational Flow Graphical Model

Assume that there exist a sequence of variables that maps the latent variables and the observations. Then, it is possible to define a graphical model using normalizing flows, as introduced Section 2, leading to exact latent-variable inference and log-likelihood evaluation of the model. We call this model a *Variational Flow Graphical Model* (VFG) and introduce it in the following.

3.1 The Evidence Lower Bound of Variational Flow Graphical Models

We give Figure 1 an illustration of a tree structure induced by variational flows. The hierarchical generative network comprises L layers, $\mathbf{h}^l$ denotes the latent variable in layer l, and θ is the vector of model parameters. The hierarchical generative process of the model is defined as:

$$p_{\theta_\mathbf{f}}(\mathbf{x}) = \sum_{\mathbf{h}^1,\dots,\mathbf{h}^L} p_{\theta_\mathbf{f}}(\mathbf{h}^L) p_{\theta_\mathbf{f}}(\mathbf{h}^{L-1}|\mathbf{h}^L)\cdots p_{\theta_\mathbf{f}}(\mathbf{x}|\mathbf{h}^1)\,.$$

The probability density function $p_{\theta_\mathbf{f}}(\mathbf{h}^{l-1}|\mathbf{h}^l)$ is modeled with an invertible normalizing flow function. The hierarchical recognition network is factorized as

$$q_{\theta_\mathbf{f}}(\mathbf{h}|\mathbf{x}) = q_{\theta_\mathbf{f}}(\mathbf{h}^1|\mathbf{x}) q_{\theta_\mathbf{f}}(\mathbf{h}^2|\mathbf{h}^1)\cdots q_{\theta_\mathbf{f}}(\mathbf{h}^L|\mathbf{h}^{L-1})\,,$$

where $\mathbf{h} = \{\mathbf{h}^1, \cdots, \mathbf{h}^L\}$ denotes the vector of latent variables of the model. At node i, the invertible function $\mathbf{h}^{(i)}$ is used as the forward evidence message received from its children, and $\widehat{\mathbf{h}}^{(i)}$ as the reconstruction of $\mathbf{h}^{(i)}$ with backward message received from the root node. We denote by $ch(i)$ and $pa(i)$, the node i's child set and parent, respectively. Let $\mathbf{f}_{(i,j)}$ be the direct edge (function) from node i to node j, and $\mathbf{f}^{-1}_{(i,j)}$ or $\mathbf{f}_{(j,i)}$ defined as its inverse function. Then, we observe that

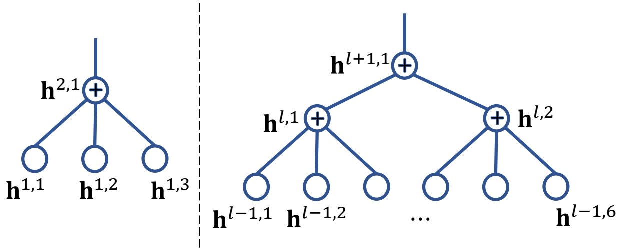

Figure 1: (Left) The structure of one node. Node $\mathbf{h}^{2,1}$ connects with its children with invertible functions. The messages from its children are aggregated at $\mathbf{h}^{2,1}$. (Right) An illustration of the latent structure from layer $l-1$ to $l+1$. $\mathbf{h}^{l,i}$ means the ith latent variable in layer l.

$$\mathbf{h}^{(j)} = \frac{1}{|ch(j)|}\sum_{i\in ch(j)} \mathbf{f}^{(i,j)}(\mathbf{h}^{(i)})\,,$$

$$\widehat{\mathbf{h}}^{(i)} = \frac{1}{|pa(i)|}\sum_{j\in pa(i)} \mathbf{f}^{-1}_{(i,j)}(\widehat{\mathbf{h}}^{(j)})\,.$$

The inference procedure includes forward and backward message passing corresponding to the encoding and decoding steps, respectively. With $\mathbf{h}^0 = \mathbf{x}$, the layer-wise ELBO, considering latent states in each layer, can be derived as

$$\mathcal{L}(\mathbf{x};\theta_\mathbf{f}) = \sum_{l=0}^{L-1} \mathbb{E}_{q(\mathbf{h}^{l+1}|\mathbf{h}^l)}\Big[\log p(\mathbf{h}^l|\widehat{\mathbf{h}}^{l+1})\Big] + \sum_{l=1}^{L-1} \mathbf{H}(\mathbf{h}^l|\mathbf{h}^{l-1}) - \mathbf{KL}\big(q(\mathbf{h}^L|\mathbf{h}^{L-1})||p(\mathbf{h}^L)\big). \quad (3)$$

More details on this expression can be found in the Appendix. The first term of the ELBO is the reconstruction term factorized over the layers, where the model pushes the variational distribution, over the latent representations $\mathbf{h}^1, ..., \mathbf{h}^{L-1}$, to fit the observed data $\mathbf{x}$. At layer l, the reconstruction $\widehat{\mathbf{h}}^l$ is generated based on $\widehat{\mathbf{h}}^{l+1}$. Optimizing the reconstruction term $\log p(\mathbf{h}^l|\widehat{\mathbf{h}}^{l+1})$ is equivalent to force the latent value $\mathbf{h}^l$ close to its reconstruction $\widehat{\mathbf{h}}^l$, i.e., $\mathbf{h}^l = \widehat{\mathbf{h}}^l$. At the root node, we have $\widehat{\mathbf{h}}^L = \mathbf{h}^L$.

The remaining terms are regularization terms for the latent representation where the negated **KL** quantity in the right-hand side keeps the model near the prior distribution of the nodes. A trade-off is thus performed here. Invertible functions are employed to connect the studied graph nodes as in flow-based models [7] to achieve tractable message passing. As shown in Figure 1-(Left), a node in a flow-graph can have multiple children and multiple parents. Each node has the forward messages from the input data and the backward messages from the root. If all the nodes only have one parent, then the structure becomes a tree. If several nodes have multiple parents, the graph will be a DAG. It is easy to extend the computation of the ELBO (3) to DAGs with topology ordering of the nodes and thus the layer number. Indeed, the ELBO for a DAG structure reads:

$$\begin{aligned}
&\log p(\mathbf{x}) \geqslant \mathcal{L}(\mathbf{x};\theta_{\mathbf{f}}) \\
=& \sum_{i\in\mathcal{V}\backslash\mathcal{R}_{\mathcal{G}}} \mathbb{E}_{q(\mathbf{h}^{pa(i)}|\mathbf{h}^{ch(pa(i))})}\Big[\log p(\mathbf{h}^{(i)}|\widehat{\mathbf{h}}^{pa(i)})\Big] \qquad (4)\\
&+ \sum_{i\in\mathcal{V}\backslash\mathcal{R}_{\mathcal{G}}} \mathbf{H}(\mathbf{h}^{(i)}|\mathbf{h}^{ch(i)}) \\
&- \sum_{i\in\mathcal{R}_{\mathcal{G}}} \mathbf{KL}\big(q(\mathbf{h}^{(i)}|\mathbf{h}^{ch(i)})|p(\mathbf{h}^{(i)})\big).
\end{aligned}$$

Here $\mathcal{V}$ stands for the node set of DAG $\mathcal{G}=\{\mathcal{V},\mathbf{f}\}$, and $\mathcal{R}_{\mathcal{G}}$ is the set of root or source nodes.

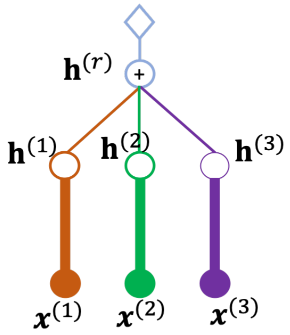

Figure 2: Aggregation with average. $\mathbf{h}^{(r)}$ has three children, $\mathbf{h}^{(1)}$, $\mathbf{h}^{(2)}$, and $\mathbf{h}^{(3)}$.

Assume there are k leaf nodes on a tree or a DAG model, corresponding to k sections of the input sample $\mathbf{x}=[\mathbf{x}^{(1)},...,\mathbf{x}^{(k)}]$, then the hidden variables in both (3) and (4) are computed with forward and backward message passing. We provide more details about the nodes in the next subsection.

3.2 Node Aggregation

In the sequel, we consider that all nodes latent variables, noted $\mathbf{h}^{l,i}$, for all $l\in[L]$ and $i\in\mathbb{N}$, are distributed according to Gaussian prior distributions or more generally exponential distributions. There are two approaches to aggregate signals from different nodes: – Average-based and – Concatenation-based aggregation. While concatenation-based aggregation is simple and straightforward, we rather focus on average-based aggregation, see Figure 2, in this paper. We assume that each entry of a hidden node follows a normal distribution, i.e., $\mathbf{h}_j^{(i)}\sim\mathcal{N}(\mu_j^{(i)},\sigma^2)$ for node i's jth entry, or an exponential distribution, i.e., $\mathbf{h}_j^{(i)}\sim\mathrm{Exp}(\lambda_j^{(i)})$. To avoid cumbersome notations, we use the same standard deviation σ across all nodes. Extending to different values for each node does not affect the rest of the paper. Assume a model only has one average aggregation node as shown in Figure 2, then (3) yields

$$\begin{aligned}
&\log p(\mathbf{x}) \\
&\geqslant \mathcal{L}(\mathbf{x};\theta_{\mathbf{f}}) = \mathbb{E}_{q(\mathbf{h}^1|\mathbf{x})}\big[\log p(\mathbf{x}|\widehat{\mathbf{h}}^1)\big] + \mathbf{H}(\mathbf{h}^1|\mathbf{x}) \qquad (5)\\
&+\mathbb{E}_{q(\mathbf{h}^2|\mathbf{h}^1)}\big[\log p(\mathbf{h}^1|\widehat{\mathbf{h}}^2)\big] - \mathbf{KL}\big(q(\mathbf{h}^2|\mathbf{h}^1)|p(\mathbf{h}^2)\big).
\end{aligned}$$

There are two aggregation rules for node i: (a) the parent value is the mean of its children, i.e., $\mathbf{h}^{(i)}=\frac{1}{|ch(i)|}\sum_{j\in ch(i)}\mathbf{h}^{(j)}$; (b) the children have the same reconstruction value with its parent, i.e., $\widehat{\mathbf{h}}^{(j)}=\widehat{\mathbf{h}}^{(i)},\forall j\in ch(i)$. Consider a single aggregation node model. Let $\mathbf{h}^{(r)}$ be the root, and it has k children, $\mathbf{h}^{(t)}, t=1,...,k$. With $\mathbf{f}_t$ as the flow function connecting $\mathbf{h}^{(t)}$ and $\mathbf{x}^{(t)}$, according to the aggregation rules we represent, in Figure 2 with $k=3$, the following identities:

$$\begin{aligned}
&\mathbf{h}^{(t)} = \mathbf{f}_t(\mathbf{x}^{(t)})\,, \\
&\widehat{\mathbf{h}}^{(r)} = \mathbf{h}^{(r)} = \frac{1}{k}\sum_{t=1}^{k}\mathbf{h}^{(t)}\,, \qquad (6)\\
&\widehat{\mathbf{h}}^{(t)} = \widehat{\mathbf{h}}^{(r)},\ t=1,...,k\,.
\end{aligned}$$

Under aforementioned prior distributions choices, the reconstruction terms in (5) are computed with

$$\begin{aligned}
&\log p(\mathbf{x}|\widehat{\mathbf{h}}^1) + \log p(\mathbf{h}^1|\widehat{\mathbf{h}}^2) \\
=& -\sum_{t=1}^{k}\Bigg\{\underbrace{\frac{1}{2\sigma_{\mathbf{x}}^2}\Big|\Big|\mathbf{x}^{(t)}-\mathbf{f}_t^{-1}(\widehat{\mathbf{h}}^{(r)})\Big|\Big|^2}_{\text{By }\widehat{\mathbf{x}}^{(t)}=\mathbf{f}_t^{-1}(\widehat{\mathbf{h}}^{(t)})=\mathbf{f}_t^{-1}(\widehat{\mathbf{h}}^{(r)})} \\
&+\underbrace{\frac{1}{2\sigma^2}\Big|\Big|\mathbf{f}_t(\mathbf{x}^{(t)})-\widehat{\mathbf{h}}^{(r)}\Big|\Big|^2}_{\text{By }\widehat{\mathbf{h}}^2=\widehat{\mathbf{h}}^{(r)},\ \mathbf{h}^{(t)}=\mathbf{f}_t(\mathbf{x}^{(t)})}\Bigg\} + C\,, \qquad (7)
\end{aligned}$$

where $C=-dk\ln(2\pi)-\frac{dk}{2}\ln(\sigma_{\mathbf{x}}^2)-\frac{dk}{2}\ln(\sigma^2)$ is a constant since both $\sigma_{\mathbf{x}}^2$ and σ^2 are constant. If all hidden nodes are exponentially distributed, the conditional distributions in reconstruction terms (7) will be Laplace distribution $\mathrm{Laplace}(0,\lambda)$, and the regularization terms will be expressed in terms of L_1 norm. We use the latent variables from a batch of training samples to approximate the entry $\mathbf{H}$ and $\mathbf{KL}$ terms in (5). We note that maximizing the ELBO will force the average aggregation node to satisfy aggregation rule (b).

3.3 Inference on Sub-graphs

Given a trained VFG model, our goal in this subsection is to infer the state of any node given observed ones.

Relations between variables at different nodes can also be inferred via our flow-based graphical model. The hidden state of the parent node j in a single aggregation model can be approximated by the observed children as follows

$$\mathbf{h}^{(j)} = \frac{1}{|ch(j) \cap O|} \sum_{i \in ch(j) \cap O} \mathbf{h}^{(i)}, \tag{8}$$

where O is the set of observed leaf nodes, see Figure 3-left for an illustration. Observe that for either a tree or a DAG, the state of any given node is updated via messages received from its children. The message passing firstly occurs from the children to the parent with updating action and then pass it back to the children without updating. Figure 3 illustrates this inference mechanism for trees in which the structure enables us to perform message passing among the nodes.

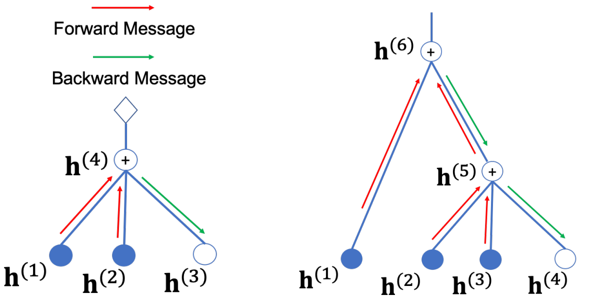

Figure 3: (Left) Inference of single aggregation node model. Node 4 aggregates from node 1 and 2, and pass the updated state to node 3 for prediction. (Right) Inference on a tree model. Observed node states are gathered in node 5 to predict the state of node 4.

We derive the following Lemma establishing the relation between two leaf nodes:

Lemma 1. *Let $\mathcal{G}$ be a trained tree structured variational flow graphical model with L layers, and i and j are two leaf nodes with a as the closest common ancestor. Given observed value at node i, the value of node j can be approximated by $\widehat{\mathbf{x}}^j \approx \mathbf{f}_{(a,j)}(\mathbf{f}_{(i,a)}(\mathbf{x}^{(i)}))$. Here $\mathbf{f}_{(i,a)}$ is the flow function path from node i to node a. The conditional density of $\mathbf{x}^{(j)}$ given $\mathbf{x}^{(i)}$ can be approximated by:*

$$\begin{aligned} &\log p(\mathbf{x}^{(j)}|\mathbf{x}^{(i)}) \\ &\approx \log p(\widehat{\mathbf{h}}^L) - \frac{1}{2}\log\left(\det\left(\mathbf{J}_{\widehat{\mathbf{x}}^{(j)}}(\widehat{\mathbf{h}}^L)^\top \mathbf{J}_{\widehat{\mathbf{x}}^{(j)}}(\widehat{\mathbf{h}}^L)\right)\right). \end{aligned} \tag{9}$$

Considering the normalizing flow (1), we have the following identity for each node of the graph structure:

$$\begin{aligned} p(\mathbf{h}^{(i)}|\mathbf{h}^{pa(i)}) &= p(\mathbf{h}^{pa(i)})\Big|\det(\frac{\partial \mathbf{h}^{pa(i)}}{\partial \mathbf{h}^{(i)}})\Big| \\ &= p(\mathbf{h}^{pa(i)})\Big|\det(\mathbf{J}_{\mathbf{h}^{pa(i)}}(\mathbf{h}^{(i)}))\Big|. \end{aligned}$$

Algorithm 1 Inference model parameters with forward and backward message propagation

```
1:  Input: Data distribution $\mathcal{D}$, $\mathcal{G} = \{\mathcal{V}, \mathbf{f}\}$
2:  for $s = 0, 1, ...$ do
3:    Sample minibatch $b$ samples $\{\mathbf{x}_1, ..., \mathbf{x}_b\}$ from $\mathcal{D}$;
4:    for $i \in \mathcal{V}$ do
5:      // forward message passing
6:      $\mathbf{h}^{(i)} = \frac{1}{|ch(i)|}\sum_{j \in ch(i)} \mathbf{f}_{(j,i)}(\mathbf{h}^{(j)})$;
7:    end for
8:    $\widehat{\mathbf{h}}^{(i)} = \mathbf{h}^{(i)}$  if $i \in \mathcal{R}_{\mathcal{G}}$ or $i \in$ layer L;
9:    for $i \in \mathcal{V}$ do
10:     // backward message passing
11:     $\widehat{\mathbf{h}}^{(i)} = \frac{1}{|pa(i)|}\sum_{j \in pa(i)} \mathbf{f}^{-1}_{(i,j)}(\widehat{\mathbf{h}}^{(j)})$;
12:   end for
13:   $\mathbf{h} = \{\mathbf{h}^{(t)} | t \in \mathcal{V}\}$, $\widehat{\mathbf{h}} = \{\widehat{\mathbf{h}}^{(t)} | t \in \mathcal{V}\}$;
14:   Approximate the entropy H and KL terms in ELBO for each layer with b samples;
15:   Updating VFG model $\mathcal{G}$ with gradient ascending: $\theta_{\mathbf{f}}^{(s+1)} = \theta_{\mathbf{f}}^{(s)} + \nabla_{\theta_{\mathbf{f}}} \frac{1}{b}\sum_{i=1}^{b} \mathcal{L}(\mathbf{x}_b; \theta_{\mathbf{f}}^{(s)})$.
16: end for
```

Remark 1. *Let O be the set of observed leaf nodes, j be an unobserved node, and a the closest ancestor of $O \cup \{a\}$. Then the state of j can be imputed with $\widehat{\mathbf{x}}^j \approx \mathbf{f}_{(a,j)}(\mathbf{f}_{(O,a)}(\mathbf{x}^{(i)}))$, where $\mathbf{f}_{(O,a)}$ is the flow function path from all nodes in O to a. Note that approximation (9) still holds for $p(\mathbf{x}^{(j)}|\mathbf{x}^O)$.*

Those results can naturally be extended to DAG models.

3.4 Algorithm and Implementation

In this section, we develop the training algorithm, see Algorithm 1, that outputs the fitted vector of parameters resulting from the maximization of the ELBO objective function (3) or (4) depending on what graph structure is used. In Algorithm 1, the inference of the latent variables is performed via forwarding message passing, cf. Line 6, and their reconstructions are computed in backward message passing, cf. Line 11.

Unlike Variational Autoencoders (VAE), the variance of latent variables in a VFG is set to be a fixed value rather than parameterized with neural networks. A VFG is a deterministic network passing latent variable values between nodes. The reconstruction (likelihood) terms in each layer are computed with forwarding and backward node states. We use the empirical variance in a batch of training samples to approximate the entropy and the regularization **KL** terms. Ignoring explicit neural network parameterized variances for all latent nodes enables us to use flow-based models as both the encoders and decoders. A direct benefit of such modeling choices (normalizing flows) is the ability to get rid of the conditional sampling steps which are costly and induce approximation noise. We thus obtain a deterministic ELBO objective (3)- (4) that can efficiently be optimized with standard stochastic optimizers.

3.4.1 Layer-wise Training

From a practical perspective, layer-wise training strategy can improve the efficiency of a model especially when it is constructed of more than two layers. In this case, we update the parameter of only one layer with backpropagation of the gradient of the loss function while keeping the other layers fixed at each optimization step. By maximizing the ELBO (3) or (4) with the above algorithm, the aggregation rules in Section 3.2 are expected to be satisfied. We can improve the inference on sub-graphs discussed in Section 3.3 by using the random masking method introduced in the sequel.

3.4.2 Random Masking

Inference on a VFG model requires the aggregation node's state to be imputed from observed children's, as shown in (8). Then, unobserved children's state can be inferred from their parent. The inference capability of VFG via imputation can be reinforced by *masking out* some sections of the training samples. The training objective can be changed to force the model to impute the value of masked sections. For example in a tree model, the alternative objective function reads

$$\begin{aligned}
&\mathcal{L}(\mathbf{x}, O_{\mathbf{x}}; \theta_{\mathbf{f}}) \\
=&\sum_{t=1}^{k} \mathbb{E}_{q(\mathbf{h}^1|\mathbf{x}^{O_{\mathbf{x}}})}\left[\log p(\mathbf{x}^{(t)}, t \notin O_{\mathbf{x}}|\widehat{\mathbf{h}}^1)\right] \\
&+ \sum_{l=1}^{L-1} \mathbb{E}_{q(\mathbf{h}^{l+1}|\mathbf{h}^l)}\left[\log p(\mathbf{h}^l|\widehat{\mathbf{h}}^{l+1})\right] \\
&+ \sum_{l=1}^{L-1} \mathbf{H}(\mathbf{h}^l|\mathbf{h}^{l-1}) - \mathbf{KL}\big(q(\mathbf{h}^L|\mathbf{h}^{L-1})|p(\mathbf{h}^L)\big).
\end{aligned} \tag{10}$$

Here $O_{\mathbf{x}}$ is the set of leaf nodes with observation. With a batch of training samples, the objectives of (3) and (10) can be maximized sequentially to optimize the ELBO and enhance inference capability as well. Training procedures with random masking is presnted in Algorithm 2.

Algorithm 2 Inference model parameters with random masking

```
Input: Data distribution $\mathcal{D}$, $\mathcal{G} = \{\mathcal{V}, \mathbf{f}\}$
for $s = 0, 1, ...$ do
    Sample minibatch $b$ samples $\{\mathbf{x}_1, ..., \mathbf{x}_b\}$ from $\mathcal{D}$;
    Optimize (3) with steps 4 to 15 in Algorithm 1;
    Sample a subset of the $k$ data sections as data observation set $O_{\mathbf{x}}$; $O \leftarrow O_{\mathbf{x}}$;
    for $i \in \mathcal{V}$ do
        // forward message passing
        $\mathbf{h}^{(i)} = \frac{1}{|ch(i) \cap O|} \sum_{j \in ch(i) \cap O} \mathbf{f}_{(j,i)}(\mathbf{h}^{(j)})$;
        $O \leftarrow O \cup \{i\}$ if $ch(i) \cap O \neq \emptyset$;
    end for
    $\widehat{\mathbf{h}}^{(i)} = \mathbf{h}^{(i)}$ if $i \in \mathcal{R}_{\mathcal{G}}$ or $i \in$ layer L;
    for $i \in \mathcal{V}$ do
        // backward message passing
        $\widehat{\mathbf{h}}^{(i)} = \frac{1}{|pa(i)|} \sum_{j \in pa(i)} \mathbf{f}^{-1}_{(i,j)}(\widehat{\mathbf{h}}^{(j)})$;
    end for
    $\mathbf{h} = \{\mathbf{h}^{(t)} \big| t \in \mathcal{V} \cap O\}$, $\widehat{\mathbf{h}} = \{\widehat{\mathbf{h}}^{(t)} \big| t \in \mathcal{V}\}$;
    Approximate the entropy $\mathbf{H}$ and $\mathbf{KL}$ terms in ELBO for each layer with b samples;
    Updating VFG with gradient of (10): $\theta_{\mathbf{f}}^{(s+1)} = \theta_{\mathbf{f}}^{(s)} + \nabla_{\theta_{\mathbf{f}}} \frac{1}{b} \sum_{i=1}^{b} \mathcal{L}(\mathbf{x}_b, O_{\mathbf{x}}; \theta_{\mathbf{f}}^{(s)})$ .
end for
```

4 Theoretical Justifications for Latent Representation Learning

The proposed Variational Flow Graphical models provide approaches to integrate multi-modal (multiple nature of data) or multi-source (collected from various sources) data. With invertible flow functions, we analyze the identifiability [14, 27] of the VFG in this section. We assume that each data point has k sections, and denote by $\mathbf{h}^{(t)}$, the latent variable for section t, namely $\mathbf{x}^{(t)}$. Suppose the distribution of the latent variable $\mathbf{h}^{(t)}$, conditioned on $\mathbf{u}$, is a factorial member of the exponential family with $m > 0$ sufficient statistics, see [?] for more details on exponential families. Here $\mathbf{u}$ is an additional observed variable which can be considered as covariates. The general form of the exponential distribution can be expressed as

$$\begin{aligned}
&p_{\mathbf{h}^{(t)}}(\mathbf{h}^{(t)}|\mathbf{u}) \\
&= \Pi_{i=1}^{d} \frac{Q_i(h^{(t,i)})}{Z_i(\mathbf{u})} \exp\left[\sum_{j=1}^{m} T_{i,j}(h^{(t,i)})\lambda_{i,j}(\mathbf{u})\right],
\end{aligned} \tag{11}$$

where Q_i is the base measure, $Z_i(\mathbf{u})$ is the normalizing constant, $T_{i,j}$ are the component of the sufficient statistic and $\lambda_{i,j}$ the corresponding parameters, depending on the variable $\mathbf{u}$. Data section variable $\mathbf{x}^{(t)}$ is generated with some complex, invertible, and deterministic function from the latent space as in:

$$\mathbf{x}^{(t)} = \mathbf{f}_t^{-1}(\mathbf{h}^{(t)}, \epsilon) .$$

Let $\mathbf{T} = [\mathbf{T}_1, ..., \mathbf{T}_l]$, and $\lambda = [\lambda_1, ..., \lambda_l]$. We define the domain of the inverse flow $\mathbf{f}_t^{-1}$ as $\mathcal{H} = \mathcal{H}_1 \times ... \times \mathcal{H}_l$. The parameter set $\widehat{\Theta} = \{\widehat{\theta} := (\widehat{\mathbf{T}}, \widehat{\lambda}, \mathbf{g})\}$ is defined in order to represent the model learned by a piratical algorithm. In the limit of infinite data and algorithm convergence, we establish the following theoretical result regarding the identifiability of the sufficient statistics $\mathbf{T}$ in our model (11).

Theorem 1. *Assume that we observe data distributed according to the model given by (11) and that* $\mathbf{x}^{(t)} = \mathbf{f}_t^{-1}(\mathbf{h}^{(t)}, \epsilon)$*. Let the following assumptions holds,*

(a) The sufficient statistics $T_{ij}(h)$ *are differentiable almost everywhere and their derivatives* $\partial T_{i,j}/\partial_h$ *are nonzero almost surely for all* $h \in \mathcal{H}_i$, $1 \leq i \leq d$ *and* $1 \leq j \leq m$.

(b) There exist $(dm+1)$ *distinct conditions* $\mathbf{u}^{(0)}$, ..., $\mathbf{u}^{(dm)}$ *such that the matrix*

$$\mathbf{L} = [\lambda(\mathbf{u}^{(1)}) - \lambda(\mathbf{u}^{(0)}), ..., \lambda(\mathbf{u}^{(dm)}) - \lambda(\mathbf{u}^{(0)})]$$

of size $dm \times dm$ *is invertible.*

Then the model parameters $\mathbf{T}(\mathbf{h}_k) = \mathbf{A}\widehat{\mathbf{T}}(\mathbf{h}_k) + \mathbf{c}$*. Here* $\mathbf{A}$ *is a* $dm \times dm$ *invertible matrix and* $\mathbf{c}$ *is a vector of size* dm.

The proof of Theorem 1 and further analysis can be found in the supplementary file.

5 Numerical Experiments

We present in this section several numerical experiments to highlight the benefits of our VFG model. The first main application we present is the imputation of missing values. We compare our method with several baseline models are compared with on both synthetic and real datasets. The second application we present is to learn the disentangled latent representations that separate the explanatory factors of variations in the data, see [2]. For that latter application, the model is trained and evaluated on the MNIST handwritten digits dataset.

5.1 Missing Entries Imputation

We now focus on the task of imputing missing entries in a graph structure. For all the following experiments, the models are trained on the training set and are used to infer the missing entries of samples in the testing set. We use MSE regarding the prediction and ground truth as the imputation metric in the comparison of different methods.

Baseline Methods: We use the following baselines for data imputation:

- *Mean Value:* Using training set mean values to replace the missing entries in the testing set.
- *Iterative Imputation:* A strategy for imputing missing values by modeling each feature with missing values as a function of other features in a Round-Robin fashion.
- *KNN:* To use K-Nearest Neighbor for data imputation, we compare the non-missing entries of each sample to the training set and use the average of top k samples as imputation values
- *Multivariate Imputation by Chained Equation (MICE):* This method impute the missing entries with multiple rounds of inference. This method can handle different type of data.

Synthetic Data: In this set of experiments, we study the proposed model with synthetic datasets. We generate a synthetic dataset of 1 300 data points, 1 000 for the training phase of the model, 300 for imputation testing. Each data sample has 8 dimensions with 2 latent variables. Let $z_1 \sim \mathcal{N}(0, 1.0^2)$ and $z_2 \sim \mathcal{N}(1.0, 2.0^2)$ be the latent variables. For a sample $\mathbf{x}$, we have $x_1 = x_2 = z_1, x_3 = x_4 = 2\sin(z_1), x_5 = x_6 = z_2$, and $x_7 = x_8 = z_2^2$. In the testing dataset, x_3, x_4, x_7, and x_8 are missing. We use a VFG model with a single average aggregation node that has four children, and each child connects the parent with a flow function consisting of 3 coupling layers [7]. Each child takes 2 variables as input data section, and the latent dimension of the VFG is 2. Figure 4 presents the imputation MSE values through different methods. We can see that the proposed VFG model performs much better than mean value, iterative, and MICE methods on synthetic dataset.

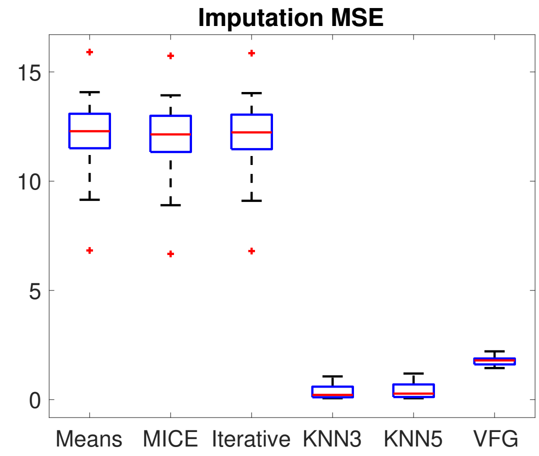

Figure 4: Imputation MSE of different methods on synthetic data.

California Housing: We further investigate the method on a real dataset. The California Housing [1] dataset has 8 feature entries and 20 640 data samples. We use the first 20 000 samples for training and 100 of the rest for testing. We get 4 data sections, and each section contains 2 variables. In the testing set, the second section is assumed missing for illustration purposes. For this set of experiments, we construct a tree structure VFG with 2 layers. The first layer has two aggregation nodes, and each of them has 2 children. The second layer consists of one aggregation node that has two children connecting with the first layer. As shown in Table 1, VFG model can outperform the other models on California Housing dataset.

Methods	*Imputation MSE*
Mean Value	1.993
MICE	1.951
Iterative Imputation	1.966
KNN (k=3)	1.974
KNN (k=5)	1.969
VFG	**1.356**

Table 1: Imputation Mean Squared Error (MSE) results on California Housing dataset.

5.2 Latent Representation Learning on MNIST

In this set of experiments, we evaluate Variational Flow Graphical Models on latent representation learning of the MNIST dataset [18]. We construct a two layer VFG model, and use exponential distribution assumption for the latent nodes, and set $\lambda = 1$. The first layer consists of one aggregation node with four children, and each child has an input dimension 14×14. The second layer is a single flow model. The latent dimension for this model is 196. Following [27], the VFG model is trained with image labels to improve the disentanglement of the latent representation of the input data. Based on the theoretical result introduced in Section 4, we set the parameters of $\mathbf{h}^L$'s prior distribution as a function of image label, i.e., $\lambda^L(u)$, where u denotes the image label. In practice, we use 10 trainable λ^Ls regarding the 10 digits. The images in the second row of Figure 5 are reconstructions of MNIST samples extracted from the testing set, displayed in the first row of the same Figure, using our proposed VFG model.

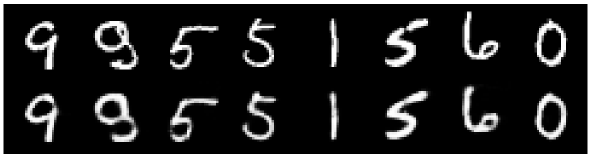

Figure 5: First row is original images, and the second row is reconstructions with VFG.

Figure 6 shows the t-distributed stochastic neighbor embedding (t-SNE) plot of 2,000 testing images's latent variables learned with our model, and 200 for each digit. We observe from Figure 6, that VFG can learn separated latent representations to distinguish different hand-written numbers.

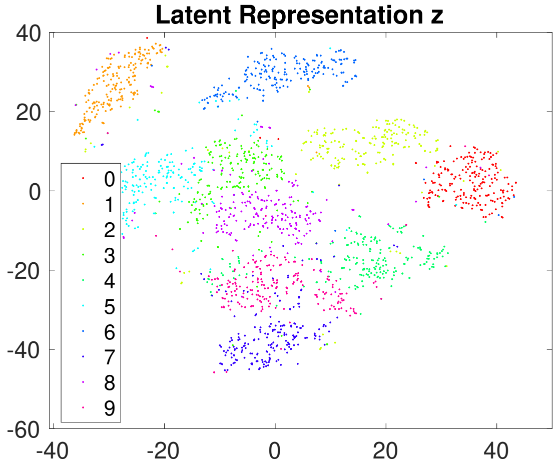

Figure 6: t-SNE plot of latent variables from VFG learned with labels.

6 Conclusion

In this paper, we propose VFG, a variational flow graphical model that aims at bridging the gap between normalizing flows and the paradigm of graphical models. Our VFG model learns the hierarchical latent structure of the input data through message passing between latent nodes, assumed to be random variable. The posterior inference, of the latent nodes given input observations, is facilitated by the careful embedding of normalizing flow in the general graph structure, thus bypassing the stochastic sampling step. Experiments on missing data imputation and disentangled representation learning illustrate the effectiveness of the model. Future work includes applying our VFG model to fine grained data relational structure learning and reasoning.

References

[1] California housing on sklearn. https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_california_housing.html.

[2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[3] Jeff A Bilmes and Chris Bartels. Graphical model architectures for speech recognition. *IEEE signal processing magazine*, 22(5):89–100, 2005.

[4] Christopher M Bishop, David Spiegelhalter, and John Winn. Vibes: A variational inference engine for bayesian networks. In *Advances in neural information processing systems*, pages 793–800, 2003.

[5] Nicola De Cao, Wilker Aziz, and Ivan Titov. Block neural autoregressive flow. In *Uncertainty in Artificial Intelligence*, pages 1263–1273. PMLR, 2020.

[6] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[7] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *ArXiv*, abs/1605.08803, 2016.

[8] Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. *arXiv preprint arXiv:1902.00275*, 2019.

[9] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[10] Estevam R Hruschka, Eduardo R Hruschka, and Nelson FF Ebecken. Bayesian networks for imputation in classification problems. *Journal of Intelligent Information Systems*, 29(3):231–252, 2007.

[11] Michael I. Jordan, editor. *Learning in Graphical Models*. MIT Press, Cambridge, MA, USA, 1999.

[12] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[13] David Kahle, Terrance Savitsky, Stephen Schnelle, and Volkan Cevher. Junction tree algorithm. *Stat*, 631, 2008.

[14] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. volume 108 of *Proceedings of Machine Learning Research*, pages 2207–2217, Online, 26–28 Aug 2020. PMLR.

[15] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[16] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018.

[17] Daphne Koller, Nir Friedman, Lise Getoor, and Ben Taskar. Graphical models in a nutshell. *Introduction to statistical relational learning*, 43, 2007.

[18] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[19] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in neural information processing systems*, pages 2378–2386, 2016.

[20] David Madigan, Jeremy York, and Denis Allard. Bayesian graphical models for discrete data. *International Statistical Review/Revue Internationale de Statistique*, pages 215–232, 1995.

[21] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*, 2019.

[22] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

[23] Oren Rippel and Ryan Prescott Adams. High-dimensional probability estimation with deep density models. *arXiv preprint arXiv:1302.5125*, 2013.

[24] Tim Salimans, Diederik Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pages 1218–1226, 2015.

[25] Scott Sanner and Ehsan Abbasnejad. Symbolic variable elimination for discrete and continuous graphical models. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[26] Michael Shwe, Blackford Middleton, David Heckerman, Max Henrion, Eric Horvitz, Harold Lehmann, and Gregory Cooper. A probabilistic reformulation of the quick medical reference system. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 790. American Medical Informatics Association, 1990.

[27] Peter Sorrenson, Carsten Rother, and Ullrich Köthe. Disentanglement by nonlinear ica with general incompressible-flow networks (gin). In *Ninth International Conference on Learning Representations*, 2020.

[28] Esteban G Tabak, Eric Vanden-Eijnden, et al. Density estimation by dual ascent of the log-likelihood. *Communications in Mathematical Sciences*, 8(1):217–233, 2010.

[29] John Winn and Christopher M Bishop. Variational message passing. *Journal of Machine Learning Research*, 6(Apr):661–694, 2005.

[30] Eric P Xing, Michael I Jordan, and Stuart Russell. A generalized mean field algorithm for variational inference in exponential families. *arXiv preprint arXiv:1212.2512*, 2012.

Appendix

A Derivation of the ELBO for both Tree and DAG structures

A.1 ELBO of Tree Models

The hierarchy generative network as given in Figure 7. For each pair of connected nodes, the edge is linked with an invertible function. We use θ to represent the parameters for all the edges. The forward message passing starts from $\mathbf{x}$ and ends at $\mathbf{h}^L$, and backward message passing is in the reverse direction. Then the likelihood for the data is given by

$$p(\mathbf{x}|\theta) = \sum_{\mathbf{h}^1,\ldots,\mathbf{h}^L} p(\mathbf{h}^L|\theta)p(\mathbf{h}^{L-1}|\mathbf{h}^L,\theta)\cdots p(\mathbf{x}|\mathbf{h}^1,\theta).$$

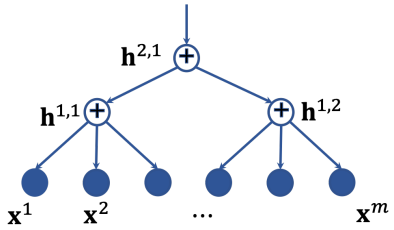

Figure 7: Tree structure.

With the flow-based ensemble model, each edge is invertible. The hierarchy of recognition network is the procedure from top to down of the structure as shown in Figure 7. Similarly, with the Markov property of the structure, the posterior density of the latent variables is given by

$$q(\mathbf{h}|\mathbf{x},\theta) = q(\mathbf{h}^1|\mathbf{x},\theta)q(\mathbf{h}^2|\mathbf{h}^1,\theta)\cdots q(\mathbf{h}^L|\mathbf{h}^{L-1},\theta)\,,$$

which can be simplified as

$$q(\mathbf{h}|\mathbf{x}) = q(\mathbf{h}^1|\mathbf{x})q(\mathbf{h}^2|\mathbf{h}^1)\cdots q(\mathbf{h}^L|\mathbf{h}^{L-1})\,.$$

Note that we also have

$$q(\mathbf{h}|\mathbf{x}) = q(\mathbf{h}^1|\mathbf{x})q(\mathbf{h}^{2:L}|\mathbf{h}^1)\,. \tag{12}$$

To derive the ELBO of a hierarchy model, we take all layers of latent variables as the latent vector in conventional VAE, and we have

$$\begin{aligned}\log p(\mathbf{x}) &= \mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\log\frac{p(\mathbf{x},\mathbf{h})}{p(\mathbf{h}|\mathbf{x})}\right]\\ &= \mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\log\frac{p(\mathbf{x},\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\frac{q(\mathbf{x},\mathbf{h})}{p(\mathbf{h}|\mathbf{x})}\right]\\ &= \underbrace{\mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\log\frac{p(\mathbf{x},\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\right]}_{\substack{\mathcal{L}_\theta(x)\\ \text{(ELBO)}}} + \underbrace{\mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\log\frac{q(\mathbf{h}|\mathbf{x})}{p(\mathbf{h}|\mathbf{x})}\right]}_{\mathbf{KL}\left(q(\mathbf{h}|\mathbf{x})|p(\mathbf{h}|\mathbf{x})\right)}\,.\end{aligned}$$

Since $\mathbf{KL}\big(q(\mathbf{h}|\mathbf{x})|p(\mathbf{h}|\mathbf{x})\big) \geq 0$ as a distance between two distributions, we obtain

$$\begin{aligned}
&\log p(\mathbf{x}) \geq \mathcal{L}_\theta(x) &(13)\\
=&\mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\right]\\
=&\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{x}|\mathbf{h}^{1:L})p(\mathbf{h}^{1:L})}{q(\mathbf{h}^{1:L}|\mathbf{x})}\right]\\
=&\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{h}^{1:L})\right] + \mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^{1:L})}{q(\mathbf{h}^{1:L}|\mathbf{x})}\right]\\
=&\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{h}^{1})\right] + \mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^{1:L})}{q(\mathbf{h}^{1:L}|\mathbf{x})}\right] &(14)\\
=&\underbrace{\mathbb{E}_{q(\mathbf{h}^{1}|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{h}^{1})\right]}_{\text{Reconstruction of the data given hidden layer 1}} + \underbrace{\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^{1:L})}{q(\mathbf{h}^{1:L}|\mathbf{x})}\right]}_{-\mathbf{KL}^{1:L}}. &(15)
\end{aligned}$$

The first term in (14) is due to $p(\mathbf{x}|\mathbf{h}^{1:L}) = p(\mathbf{x}|\mathbf{h}^1)$. The first term in (15) is due to that the expectation is regarding $\mathbf{h}^1$. The hidden variables $\mathbf{h}^{l+1:L}$ can be taken as the parameters for $\mathbf{h}^l$'s prior distribution . We expand the minus KL term in (15) as follows

$$\begin{aligned}
-\mathbf{KL}^{1:L} =&\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^{1:L})}{q(\mathbf{h}^{1:L}|\mathbf{x})}\right] &(16)\\
=&\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^1|\mathbf{h}^{2:L})p(\mathbf{h}^{2:L})}{\underbrace{q(\mathbf{h}^1|\mathbf{x})q(\mathbf{h}^{2:L}|\mathbf{h}^1)}_{\text{Due to (12)}}}\right]\\
=&\underbrace{\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^1|\mathbf{h}^{2:L})p(\mathbf{h}^{2:L})}{q(\mathbf{h}^{2:L}|\mathbf{h}^1)}\right]}_{(a)} + \underbrace{\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{1}{q(\mathbf{h}^1|\mathbf{x})}\right]}_{(b)}.
\end{aligned}$$

Given a batch of data, we take the inference in each layer as encoding and decoding procedures. In forward message passing, the hidden layer $\mathbf{h}^l$ only depends on its previous layer $l-1$. The logarithm term in (a) only relates to hidden states $\mathbf{h}^{1:L}$. With (12), given the hidden states $\mathbf{h}^1$ samples from layer 0, we have

$$(a) = \mathbb{E}_{q(\mathbf{h}^1|\mathbf{x})}\left[\mathbb{E}_{q(\mathbf{h}^{2:L}|\mathbf{h}^1)}\left[\log \frac{p(\mathbf{h}^1|\mathbf{h}^{2:L})p(\mathbf{h}^{2:L})}{q(\mathbf{h}^{2:L}|\mathbf{h}^1)}\right]\right]. \tag{17}$$

The inner expectation is actually the ELBO for layer hidden variable $\mathbf{h}^1$. Hence

$$\begin{aligned}
&\mathbb{E}_{q(\mathbf{h}^{2:L}|\mathbf{h}^1)}\left[\log \frac{p(\mathbf{h}^1|\mathbf{h}^{2:L})p(\mathbf{h}^{2:L})}{q(\mathbf{h}^{2:L}|\mathbf{h}^1)}\right]\\
=&\mathbb{E}_{q(\mathbf{h}^{2:L}|\mathbf{h}^1)}\left[\log p(\mathbf{h}^1|\mathbf{h}^{2:L})\right] + \mathbb{E}_{q(\mathbf{h}^{2:L}|\mathbf{h}^1)}\left[\log \frac{p(\mathbf{h}^{2:L})}{q(\mathbf{h}^{2:L}|\mathbf{h}^1)}\right]\\
=&\mathbb{E}_{q(\mathbf{h}^{2}|\mathbf{h}^1)}\left[\log p(\mathbf{h}^1|\mathbf{h}^{2})\right] + \mathbb{E}_{q(\mathbf{h}^{2:L}|\mathbf{h}^1)}\left[\log \frac{p(\mathbf{h}^{2:L})}{q(\mathbf{h}^{2:L}|\mathbf{h}^1)}\right] &(18)\\
=&\mathbb{E}_{q(\mathbf{h}^{2}|\mathbf{h}^1)}\left[\log p(\mathbf{h}^1|\mathbf{h}^{2})\right] - \mathbf{KL}^{2:L}.
\end{aligned}$$

For the term (b) we develop as follows:

$$(b) = \mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{1}{q(\mathbf{h}^1|\mathbf{x})}\right] = \mathbb{E}_{q(\mathbf{h}^{1}|\mathbf{x})}\left[\log \frac{1}{q(\mathbf{h}^1|\mathbf{x})}\right] = \mathbf{H}(\mathbf{h}^1|\mathbf{x}). \tag{19}$$

With (16) (17) (18) (19),

$$-\,\mathbf{KL}^{1:L} = \mathbb{E}_{q(\mathbf{h}^1|\mathbf{x})}\Big[\mathbb{E}_{q(\mathbf{h}^2|\mathbf{h}^1)}\big[\log p(\mathbf{h}^1|\mathbf{h}^2)\big] - \mathbf{KL}^{2:L}\Big] + \mathbf{H}(\mathbf{h}^1|\mathbf{x})\,.$$

Similarly, for layer l, we have

$$\begin{aligned}-\mathbf{KL}^{l:L} =&\mathbb{E}_{q(\mathbf{h}^l|\mathbf{h}^{l-1})}\Big[\mathbb{E}_{q(\mathbf{h}^{l+1}|\mathbf{h}^l)}\big[\log p(\mathbf{h}^l|\mathbf{h}^{l+1})\big] - \mathbf{KL}^{l+1:L}\Big] + \mathbf{H}(\mathbf{h}^l|\mathbf{h}^{l-1})\\ =&\mathbb{E}_{q(\mathbf{h}^l|\mathbf{h}^{l-1})}\Big[\mathbb{E}_{q(\mathbf{h}^{l+1}|\mathbf{h}^l)}\big[\log p(\mathbf{h}^l|\mathbf{h}^{l+1})\big]\Big] + \mathbf{H}(\mathbf{h}^l|\mathbf{h}^{l-1}) - \mathbf{KL}^{l+1:L}\,.\end{aligned}$$

Given a batch of samples, we compute and store the forward message and the backward message for each node in the forward and backward message passing procedures. The above KL term can be simplified as

$$-\mathbf{KL}^{l:L} =\mathbb{E}_{q(\mathbf{h}^{l+1}|\mathbf{h}^l)}\big[\log p(\mathbf{h}^l|\mathbf{h}^{l+1})\big] + \mathbf{H}(\mathbf{h}^l|\mathbf{h}^{l-1}) - \mathbf{KL}^{l+1:L}\,. \tag{20}$$

For a hierarchy model with L layers, we can recursively expand the KL term regarding the ELBO for each layer. Thus

$$\begin{aligned}&\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\Big[\log\frac{p(\mathbf{h}^{1:L})}{q(\mathbf{h}^{1:L}|\mathbf{x})}\Big] \qquad (21)\\ =&\sum_{l=1}^{L-1}\Big\{\mathbb{E}_{q(\mathbf{h}^{l+1}|\mathbf{h}^l)}\Big[\log p(\mathbf{h}^l|\mathbf{h}^{l+1})\Big] + \mathbf{H}(\mathbf{h}^l|\mathbf{h}^{l-1})\Big\}\\ &+\mathbb{E}_{q(\mathbf{h}^L|\mathbf{h}^{L-1})}\Big[\log p(\mathbf{h}^{L-1}|\mathbf{h}^L))\Big] - \mathbf{KL}\big(q(\mathbf{h}^L|\mathbf{h}^{L-1})|p(\mathbf{h}^L)\big)\,.\end{aligned}$$

With $\mathbf{h}^0 = \mathbf{x}$, with the ELBO can be written as

$$\log p(\mathbf{x}) \geq \sum_{l=0}^{L-1}\mathbb{E}_{q(\mathbf{h}^{l+1}|\mathbf{h}^l)}\Big[\log p(\mathbf{h}^l|\mathbf{h}^{l+1})\Big] + \sum_{l=1}^{L-1}\mathbf{H}(\mathbf{h}^l|\mathbf{h}^{l-1}) - \mathbf{KL}\big(q(\mathbf{h}^L|\mathbf{h}^{L-1})|p(\mathbf{h}^L)\big).$$

The hidden variables are computed with forward message passing with encoders $q(\mathbf{h}^l|\mathbf{h}^{l-1}), l = 1, ..., L$. The reconstructed hidden variables are computed with decoders $p(\mathbf{h}^l|\mathbf{h}^{l+1}), l = L-1, ..., 0$. We use $\widehat{\mathbf{h}}^l$ to represent the reconstruction of $\mathbf{h}^l$. Only at the root level L, we have $\widehat{\mathbf{h}}^L = \mathbf{h}^L$. Each latent variable is reconstructed with messages from higher layer. Hence the ELBO can be rewritten as

$$\log p(\mathbf{x}) \geq \sum_{l=0}^{L-1}\mathbb{E}_{q(\mathbf{h}^{l+1}|\mathbf{h}^l)}\Big[\log p(\mathbf{h}^l|\widehat{\mathbf{h}}^{l+1})\Big] + \sum_{l=1}^{L-1}\mathbf{H}(\mathbf{h}^l|\mathbf{h}^{l-1}) - \mathbf{KL}\big(q(\mathbf{h}^L|\mathbf{h}^{L-1})|p(\mathbf{h}^L)\big)\,.$$

A.2 ELBO for DAG Models

If we reverse the edge directions in a DAG, the result graph is still a DAG graph. The nodes can be listed in a topological order regarding the DAG structure as shown in Figure 8. By taking the topology order as the layers in tree structures, we can derive the ELBO for DAG structures. Assume the DAG structure has L layers, and the root nodes are in layer L. With $\mathbf{h}$ to represent the whole latent variables, following (13) we have the ELBO for the log-likelihood of data

$$\begin{aligned}\log p(\mathbf{x}) \geq \mathcal{L}_\theta(x) =&\mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\Big[\log\frac{p(\mathbf{x},\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\Big] \qquad (22)\\ =&\underbrace{\mathbb{E}_{q(\mathbf{h}^{pa(\mathbf{x})}|\mathbf{x})}\Big[\log p(\mathbf{x}|\mathbf{h}^{pa(\mathbf{x})})\Big]}_{\text{Reconstruction of the data given the parent nodes of the data}} + \underbrace{\mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\Big[\log\frac{p(\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\Big]}_{-\mathbf{KL}}\,.\end{aligned}$$

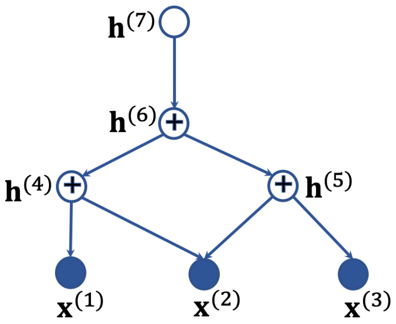

Figure 8: DAG structure. The inverse topology order is $\big\{$ {1,2,3}, {4,5}, {6}, {7} $\big\}$, and it corresponds to layers 0 to 3.

Similarly the KL term can be expanded as in the tree structures. For nodes in layer l

$$-\mathbf{KL}^{l:L} = \mathbb{E}_{q(\mathbf{h}^{pa(l)}|\mathbf{h}^l)}\big[\log p(\mathbf{h}^l|\mathbf{h}^{pa(l)})\big] + \mathbf{H}(\mathbf{h}^l|\mathbf{h}^{ch(l)}) - \mathbf{KL}^{l+1:L}\,. \tag{23}$$

The forward and backward messages or latent state of a node are stored in the message passing procedures. They can be used by the node's parents and children to compute the ELBO. It enables the calculation even the parents or children are not in layer $l+1$ or $l-1$. For the node i in layer l, $pa(i)$ may have children in layers below l. Some nodes in l may not have parent, and combining with the prior, the entropy term will become an KL term in this case. Thus, we have

$$\begin{aligned}-\mathbf{KL}^{l:L} = &\sum_{i:i\in l, i\notin\mathcal{R}_\mathcal{G}}\Big\{\mathbb{E}_{q(\mathbf{h}^{pa(i)}|\mathbf{h}^{ch(pa(i))})}\big[\log p(\mathbf{h}^i|\mathbf{h}^{pa(i)})\big] + \mathbf{H}_q(\mathbf{h}^i|\mathbf{h}^{ch(i)})\Big\} \\ &- \sum_{i\in l\bigcap\mathcal{R}_\mathcal{G}}\mathbf{KL}\big(q(\mathbf{h}^{(i)}|\mathbf{h}^{ch(i)})|p(\mathbf{h}^{(i)})\big) - \mathbf{KL}^{l+1:L}\,.\end{aligned} \tag{24}$$

Recurrently applying (24) yields

$$\begin{aligned}\mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\log\frac{p(\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\right] = &\sum_{l=1}^{L-1}\sum_{i:i\in l, i\notin\mathcal{R}_\mathcal{G}}\Big\{\mathbb{E}_{q(\mathbf{h}^{pa(i)}|\mathbf{h}^{(i)})}\Big[\log p(\mathbf{h}^{(i)}|\mathbf{h}^{pa(i)})\Big] + \mathbf{H}(\mathbf{h}^i|\mathbf{h}^{ch(i)})\Big\} \\ &-\sum_{l=1}^{L-1}\sum_{i\in l\bigcap\mathcal{R}_\mathcal{G}}\mathbf{KL}\big(q(\mathbf{h}^{(i)}|\mathbf{h}^{ch(i)})|p(\mathbf{h}^{(i)})\big) - \mathbf{KL}\big(q(\mathbf{h}^L|\mathbf{h}^{L-1})|p(\mathbf{h}^L)\big)\,.\end{aligned} \tag{25}$$

Since $L\subseteq\mathcal{R}_\mathcal{G}$, with $\mathbf{h}^{(0)}=\mathbf{x}$, (22), and (25) we have

$$\begin{aligned}\log p(\mathbf{x}) \geqslant \mathcal{L}(\mathbf{x};\theta) = &\sum_{i\in\mathcal{G}\setminus\mathcal{R}_\mathcal{G}}\mathbb{E}_{q(\mathbf{h}^{pa(i)}|\mathbf{h}^{ch(pa(i))})}\Big[\log p(\mathbf{h}^{(i)}|\mathbf{h}^{pa(i)})\Big] \\ &+\sum_{i\in\mathcal{G}\setminus\mathcal{R}_\mathcal{G}}\mathbf{H}(\mathbf{h}^{(i)}|\mathbf{h}^{ch(i)}) - \sum_{i\in\mathcal{R}_\mathcal{G}}\mathbf{KL}\big(q(\mathbf{h}^{(i)}|\mathbf{h}^{ch(i)})|p(\mathbf{h}^{(i)})\big)\,.\end{aligned}$$

B Theoretical Proofs

B.1 Proof of Lemma1

Lemma 1. *Let $\mathcal{G}$ be a well trained tree structured variational flow graphical model with L layers, and i and j are two leaf nodes with a as the closest common ancestor. Given observed value at node i, the value of node j can be approximated with $\widehat{\mathbf{x}}^j \approx \mathbf{f}_{(a,j)}(\mathbf{f}_{(i,a)}(\mathbf{x}^{(i)}))$. Here $\mathbf{f}_{(i,a)}$ is the flow function path from node i to node a. The conditional density of $\mathbf{x}^{(j)}$ given $\mathbf{x}^{(i)}$ can be approximated with*

$$\log p(\mathbf{x}^{(j)}|\mathbf{x}^{(i)}) \approx \log p(\widehat{\mathbf{h}}^L) - \frac{1}{2}\log\big(\det\big(\mathbf{J}_{\widehat{\mathbf{x}}^{(j)}}(\widehat{\mathbf{h}}^L)^\top\mathbf{J}_{\widehat{\mathbf{x}}^{(j)}}(\widehat{\mathbf{h}}^L)\big)\big)\,.$$

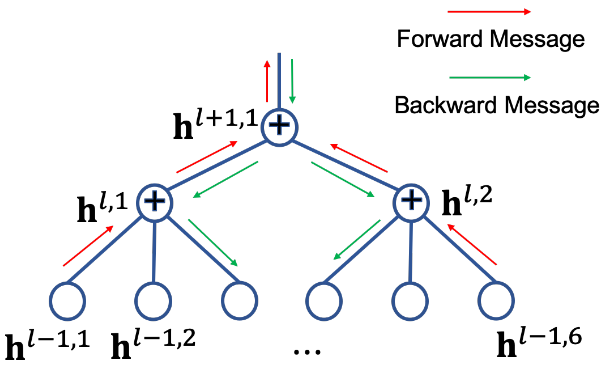

Figure 9: Message passing on a tree.

B.2 Proof of Theorem 1

Theorem 1. *Assume we observe data distributed according to the generative model given by (11) and* $\mathbf{x}^{(t)} = \mathbf{f}_t^{-1}(\mathbf{h}^{(t)}, \epsilon)$*, we further have the following assumptions,*

(a) The sufficient statistics $T_{ij}(h)$ *are differentiable almost everywhere and their derivatives* $\frac{dT_{i,j}}{dh}$ *are nonzero almost surely for all* $h \in \mathcal{H}_i$ *and all* $1 \leq i \leq d$ *and* $1 \leq j \leq m$*.*

(b) There exist $dm + 1$ *distinct conditions* $\mathbf{u}^{(0)}$*, ...,* $\mathbf{u}^{(dm)}$ *such that the matrix*

$$\mathbf{L} = [\lambda(\mathbf{u}^{(1)}) - \lambda(\mathbf{u}^{(0)}), ..., \lambda(\mathbf{u}^{(dm)}) - \lambda(\mathbf{u}^{(0)})]$$

of size $dm \times dm$ *is invertible. Then the model parameters* $\mathbf{T}(\mathbf{h}_k) = \mathbf{A}\widehat{\mathbf{T}}(\mathbf{h}_k) + \mathbf{c}$*. Here* $\mathbf{A}$ *is an* $dm \times dm$ *invertible matrix and* $\mathbf{c}$ *is a vector of size* dm*.*

Proof. The conditional probabilities of $p_{\mathbf{T},\lambda,\mathbf{f}_t^{-1}}(\mathbf{x}^{(t)}|\mathbf{u})$ and $p_{\widehat{\mathbf{T}},\widehat{\lambda},\mathbf{g}}(\mathbf{x}^{(t)}|\mathbf{u})$ are assumed to be the same in the limit of infinity data. By expanding two pdfs with change of variable rule, we have

$$\log p_{\mathbf{T},\lambda}(\mathbf{h}^{(t)}|\mathbf{u}) + \log \big| \det \mathbf{J}_{\mathbf{f}_t}(\mathbf{x}^{(t)}) \big| = \log p_{\widehat{\mathbf{T}},\widehat{\lambda}}((\mathbf{h}^{(t)})^\top|\mathbf{u}) + \log \big| \det \mathbf{J}_{g^{-1}}(\mathbf{x}^{(t)}) \big|.$$

Let $\mathbf{u}^{(0)}, ..., \mathbf{u}^{(dm)}$ be from condition (b). We can subtract this expression for $\mathbf{u}^{(0)}$ for some condition $\mathbf{u}^{(v)}$. The Jacobian terms will be removed since they do not depend $\mathbf{u}$,

$$\log p_{\mathbf{h}^{(t)}}(\mathbf{h}^{(t)}|\mathbf{u}^{(v)}) - \log p_{\mathbf{h}^{(t)}}(\mathbf{h}^{(t)}|\mathbf{u}^{(0)}) = \log p_{(\mathbf{h}^{(t)})^\top}((\mathbf{h}^{(t)})^\top|\mathbf{u}^{(v)}) - \log p_{(\mathbf{h}^{(t)})^\top}((\mathbf{h}^{(t)})^\top|\mathbf{u}^{(0)}). \tag{26}$$

Both conditional distributions of $\mathbf{h}^{(t)}$ given $\mathbf{u}$ belong to exponential family. Eq. (26) can be rewritten as

$$\begin{aligned} &\sum_{i=1}^{l} \left[\log \frac{Z_i(\mathbf{u}^{(0)})}{Z_i(\mathbf{u}^{(v)})} + \sum_{j=1}^{m} T_{i,j}(\mathbf{h}^{(t)}) \big(\lambda_{i,j}(\mathbf{u}^{(v)}) - \lambda_{i,j}(\mathbf{u}^{(0)})\big) \right] \\ =&\sum_{i=1}^{l} \left[\log \frac{\widehat{Z}_i(\mathbf{u}^{(0)})}{\widehat{Z}_i(\mathbf{u}^{(v)})} + \sum_{j=1}^{m} \widehat{T}_{i,j}(\mathbf{h}^{(t)}) \big(\widehat{\lambda}_{i,j}(\mathbf{u}^{(v)}) - \widehat{\lambda}_{i,j}(\mathbf{u}^{(0)})\big) \right]. \end{aligned} \tag{27}$$

Here the base measures Q_i are cancelled out as they do not depend on $\mathbf{u}$. Let $\bar{\lambda}(\mathbf{u}) = \lambda(\mathbf{u}) - \lambda(\mathbf{u}^{(0)})$. The above equation can be rewritten with inner products as

$$\langle \mathbf{T}(\mathbf{h}^{(t)}), \bar{\lambda} \rangle + \sum_i \log \frac{Z_i(\mathbf{u}^{(0)})}{Z_i(\mathbf{u}^{(v)})} = \langle \widehat{\mathbf{T}}((\mathbf{h}^{(t)})^\top), \bar{\widehat{\lambda}} \rangle + \sum_i \log \frac{\widehat{Z}_i(\mathbf{u}^{(0)})}{\widehat{Z}_i(\mathbf{u}^{(v)})}, \quad \forall l, 1 \leq v \leq dm.$$

Combine dm equations together and we can rewrite in matrix equation form as following

$$\mathbf{L}^\top \mathbf{T}(\mathbf{h}^{(t)}) = \widehat{\mathbf{L}}^\top \widehat{\mathbf{T}}((\mathbf{h}^{(t)})^\top) + \mathbf{b}.$$

Here $b_v = \sum_i \log \frac{\widehat{Z}_i(\mathbf{u}^{(0)})Z_i(\mathbf{u}^{(v)})}{\widehat{Z}_i(\mathbf{u}^{(v)})Z_i(\mathbf{u}^{(0)})}$. We can multiply $\mathbf{L}^\top$'s inverse with both sized of the equation,

$$\mathbf{T}(\mathbf{h}^{(t)}) = \mathbf{A}\widehat{\mathbf{T}}((\mathbf{h}^{(t)})^\top) + \mathbf{c}. \tag{28}$$

Here $\mathbf{A} = \mathbf{L}^{-1\top}\widehat{\mathbf{L}}^\top$, and $\mathbf{c} = \mathbf{L}^{-1\top}\mathbf{b}$. By a Lemma 1 from [14], there exist m distinct values $h_1^{(t,i)}$ to $h_m^{(t,i)}$ such that $\left[\frac{dT_i}{dh^{(t,i)}}(h_1^{(t,i)}), ..., \frac{dT_i}{dh^{(t,i)}}(h_m^{(t,i)})\right]$ are linear independent in $\mathbb{R}^m$, for all $1 \leq i \leq d$. Define m vectors $\mathbf{h}_v^{(t)} = [h_v^{(t,1)}, ..., h_v^{(t,d)}]$ from points given by this lemma. We obtain the Jacobian $\mathbf{Q} = [\mathbf{J}_\mathbf{T}(\mathbf{h}_1^{(t)}), ..., \mathbf{J}_\mathbf{T}(\mathbf{h}_m^{(t)})]$ with each entry as Jacobian with size $dm \times d$ from the derivative of Eq. (28) regarding these m vectors. Hence $\mathbf{Q}$ is a $dm \times dm$ invertible by the lemma and the fact that each component of $\mathbf{T}$ is univariate. We can construct a corresponding matrix $\widehat{\mathbf{Q}}$ with the Jabocian $\widehat{\mathbf{T}}(\mathbf{g}^{-1} \circ \mathbf{f}_t^{-1}(\mathbf{h}^{(t)}))$ computed at the same points and get

$$\mathbf{Q} = \mathbf{A}\widehat{\mathbf{Q}}.$$

Here $\widehat{\mathbf{Q}}$ and $\mathbf{A}$ are both full rank as $\mathbf{Q}$ is full rank. □

C Additional Numerical Experiments

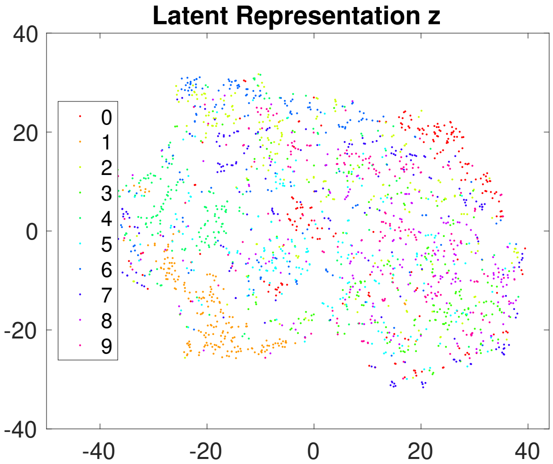

Figure 10: t-SNE plot of latent variables learned with VFG without labels.