VFG: VARIATIONAL FLOW GRAPHICAL MODEL WITH HIERARCHICAL LATENT STRUCTURES

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper presents an approach to assemble flow-based models with hierarchical structures. With designed structures, the proposed model tries to uncover the latent relational structures of high dimensional data sets. Meanwhile, the model can generate data representations with reduced latent dimensions, and thus it overcomes the drawbacks of many flow-based models that usually require a high dimensional latent space involving many trivial variables. Experiments on synthetic and real world data sets show advantages and broad potentials of the proposed method.

1 INTRODUCTION

Graphical models (Madigan et al., 1995; Hruschka et al., 2007) are powerful tools to combine the graph structure and probabilistic modeling, which provides a structural probabilistic (and hierarchical) characterization of variables. Due to both the flexibility and power of the representation of graphical models and their ability to effectively learn and perform inference in large networks (Koller et al., 2007), they have attracted lots of interest and have been applied in many fields, *e.g.* artificial intelligence like speech recognition (Bilmes & Bartels, 2005), biology like Quick Medical Reference (QMR) model (Shwe et al., 1990) and physics like energy-based model (Jordan et al., 2004).

The quantity of interest in such models is the marginal distribution, also known as incomplete likelihood, of the observed data, noted $p(\mathbf{x})$. Most statistical learning, from a finite set of observations, tasks leverage a parameterized model and their respective training procedure involves computing the maximum likelihood estimate, *i.e.* the parameter defined as $\theta^* := \arg\max_{\theta \in \mathbb{R}^d} p_\theta(\mathbf{x})$. A direct consequence of Bayes rule, which we recall 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 pdf $p_\theta(\mathbf{x}|\mathbf{z})$, as a subroutine in the general 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 the latent variables z.

The focus of this paper is mostly on this graphical inference subroutine. There are two general approaches for the latter task: *exact inference* and *approximate inference*. (i) Exact inference, *e.g.* ELIMINATION ALGORITHM (Sanner & Abbasnejad, 2012) and JUNCTION TREE ALGORITHM (Kahle et al., 2008), resorts to an exact numerical calculation procedure of the quantity of interest and leading to satisfactory results. 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 aiming at modeling complex tasks employing for instance deep neural networks. Moreover, the exactitude achieved by the exact inference is not worth the computational cost in some cases. Indeed the distribution can be well determined by a small cluster of nodes in the network, see (Jordan et al., 1999). Thus, there exist a trade-off between exact inference and light computations. (ii) In contrast, approximate inference, *e.g.* Markov Chain Monte-Carlo (MCMC) and variational inference, yields deterministic approximation procedures that generally provide bounds on the pdfs of interest. Considering the underlying slow convergence issues of stochastic MCMC sampling procedure (Salimans et al., 2015), we ratehr opt for the deterministic Variational Inference (VI) approach to tackle the graphical inference problem. VI provides a lower bound on $p_\theta(\mathbf{x})$ and is computationally efficient using off-the-shelf optimization techniques, and easily applicable to large datasets (Hoffman et al., 2013; Kingma & Welling, 2013; Liu & Wang, 2016).

In Variational Inference, mean-field approximation (Xing et al., 2012) and variational message passing (Winn & Bishop, 2005) are two common approaches in graphical models. They both require to access the intractable posterior $p(\mathbf{z}|\mathbf{x})$. Those methods leverage families of simple and tractable distributions to approximate that latter quantity. However, on one hand, such approximation is limited by the choice of distributions that by definition do not recover the true posterior, often leading to a loose lower bound and on the other hand, they often lack a flexible structure to learn the inherent disentangled latent features. Thus, those methods cannot model the latent layer in order to accurately reconstruct the data. Dealing with high dimensional data using graphical models exacerbates that systemic flaw.

Motivated by these limitations, we propose a new framework to uncover the latent relational structures of high dimensional data. The main idea is to build a variational hierarchical graphical flow model. Our contributions read as follows:

- **Hierarchical Latent Structure:** We construct hierarchical latent space between variables to uncover the latent structural relations of high dimensional data, leading to a tighter lower bound.
- **Normalizing Flows:** Normalizing flow is introduced to impose a richer and tractable posterior to approximate the true posterior as the truth is more faithful posterior approximations do result in better performance. enjoying the exact inference capability at a low computational cost.
- **Hierarchical and Flow-Based:** Introducing VARIATIONAL FLOW GRAPHICAL (VFG) model
- **Numerical Applications:** Experiments....

The remaining of the paper is organized as follows. Section 2 presents preliminaries corresponding to important concepts such as normalizing flows, variational inference 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 theoretical findings of our model. Section 5 showcases the advantage of our model, namely VFG on two different tasks: missing values imputation on both synthetic and real dataset and disentanglement learning. Section 6 presents some conclusive remarks of our work.

Notations: We denote for all $n > 1$, $[L]$ the set $\{1, \cdots, L\}$ 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 an arbitrary dimension $d > 0$.

2 PRELIMINARIES

In this section, we first introduce the standard principles and general notations of normalizing flows and variational inference. Then, we explain how those concepts can be used with graphical models.

2.1 NORMALIZING FLOWS

Normalizing flows (Kingma & Dhariwal, 2018; Rezende & Mohamed, 2015) 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 $z \in \mathcal{Z}$ and $x \in \mathcal{X}$.

The latent variable is noted $\mathbf{z} \sim p(\mathbf{z})$ and is distributed according to a tractable density $p(\mathbf{z})$. 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 :

$$\mathbf{x} = \mathbf{f}(\mathbf{z}) \,, \quad \mathbf{z} = \mathbf{f}^{-1}(\mathbf{x}) \quad \text{and} \quad \mathbf{z} \underset{\mathbf{f}_1^{-1}}{\overset{\mathbf{f}_1}{\rightleftharpoons}} \mathbf{h}^1 \underset{\mathbf{f}_2^{-1}}{\overset{\mathbf{f}_2}{\rightleftharpoons}} \mathbf{h}^2 \cdots \underset{\mathbf{f}_L^{-1}}{\overset{\mathbf{f}_L}{\rightleftharpoons}} \mathbf{x}.$$

Using the change of variables formula, the probability density function (pdf) of the model given a data point can be written as:

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

where we have $\mathbf{h}^0 = \mathbf{x}$ and $\mathbf{h}^L = \mathbf{z}$ for conciseness. The scalar value $\log |\det(\frac{\partial \mathbf{h}^i}{\partial \mathbf{h}^{i-1}})|$ is the logarithm of the absolute value of the determinant of the Jacobian matrix ($\frac{\partial \mathbf{h}^i}{\partial \mathbf{h}^{i-1}}$), also called the log-determinant. The result of this approach is a mechanism to construct new families of distributions by choosing an initial density and then chaining together some number of parameterized, invertible and differentiable transformations. The advantage of such methods is that the new density can be sampled from (by sampling from the initial density and applying the transformations).

2.2 Variational inference

Following the setting discussed above, the functional mapping $\mathbf{f}\colon \mathbf{x} \to \mathbf{z}$ can be viewed as an encoding process (inference or recognition), and the mapping $\mathbf{f}^{-1}\colon \mathbf{z} \to \mathbf{x}$ be considered as a decoding process (generation): $\mathbf{z} \sim p(\mathbf{z}), \mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z})$. In order to learn the vector of parameters θ, one typically maximizes 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 minimized the distance, in terms of Kullback-Leibler (KL), between the variational candidate and the true posterior, noted $\mathbf{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x}))$.

It is easy to show that this optimization problem is equivalent to maximizing the following evidence lower bound (ELBO) objective, noted $\mathcal{L}(\mathbf{x};\theta)$:

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

[BK: I do not understand the sentence below]

Since the transformation f is invertible, we can simplify $q_\phi(\mathbf{z}|\mathbf{x})$ using the same set of parameters θ as in $p_\theta(\mathbf{x}|\mathbf{z})$, implying that $\phi = \theta$.

2.3 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, it can be for instance a function $\mathbf{f}_\theta$ parameterized by θ which serves as a link function between two variables. The joint distribution of the model is thus given by:

$$p_\theta(\mathbf{v}) = \prod_{\mathbf{v} \in \mathcal{V}} p_\theta(\mathbf{v}|pa(\mathbf{v})) \,, \tag{3}$$

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 $q(\mathbf{z}|\mathbf{x})$, to approximate the true posterior $p(\mathbf{z}|\mathbf{x})$. This exactly coincides with the general VI framework as in (2) in the last subsection. In this paper, we focus on the factorization of the independent and disjoint latent variables (Bishop et al., 2003):

$$q(\mathbf{z}|\mathbf{x}) = \prod_i q_i(\mathbf{z}_i) \,, \tag{4}$$

where $\mathbf{z}_i$ is the latent variable at node i of the graph, assuming that the observation $\mathbf{x}$ is the parent node: $\mathbf{x} = pa(\mathbf{z}_i)$.

3 Variational Flow Graphical Model with Hierarchical Latent Structures

Assume that there exist a sequence of variables that bridge the latent and the observation sets. Then, it is possible to define a graphical model using normalizing flows, as introduced Section 2.1, leading to exact latent-variable inference and log-likelihood evaluation of the model. We call this model a *Variational Flow Graphical Model* (VFG).

3.1 The evidence lower bound of Variational Flow Graphical Models

Figure 1 illustrates the tree structure induced by variational flows. The hierarchical generative network is comprised of L layers, $\mathbf{h}^l$ denotes the latent variable in layer l and θ is the vector of parameters of the model. 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 all 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. 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 j, and $\mathbf{f}^{-1}_{(i,j)}$ or $\mathbf{f}_{(j,i)}$ defined as its inverse function. Then, we observe that

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

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

$$\mathcal{L}(\mathbf{x};\theta) = \sum_{l=0}^{L-1} \mathbb{E}_{q(\mathbf{h}^{l+1}|\mathbf{h}^l)} \Big[\log p(\mathbf{h}^l|\widehat{\mathbf{h}}^{l+1}) \Big] + \sum_{l=1}^{L-1} \mathbf{H}(\mathbf{h}^l|\mathbf{h}^{l-1}) - \mathbf{KL}\big(q(\mathbf{h}^L|\mathbf{h}^{L-1})|p(\mathbf{h}^L)\big)\,. \quad (5)$$

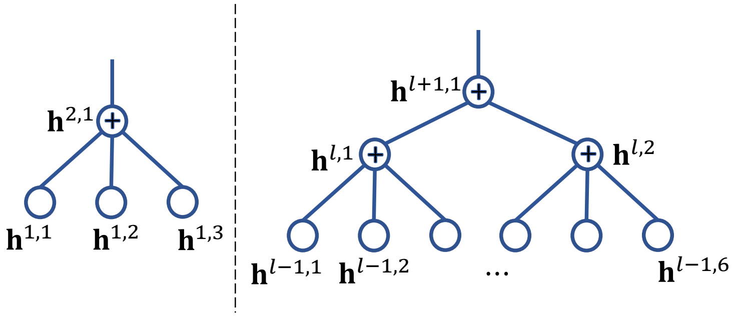

Figure 1: (Left) The structure of one node. Node $\mathbf{h}^{2,1}$ connects with its children with invertible functions. The messages from its children are aggregated at $\mathbf{h}^{2,1}$. (Right)An illustration of the latent structure from layer $l-1$ to $l+1$. $\mathbf{h}^{l,i}$ means the ith latent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

The details of the derivation of the ELBO can be found in the Appendix. The first term of ELBO is the reconstruction term for each layer: $\mathbf{x}$ and the latent representations $\mathbf{h}^1, ..., \mathbf{h}^{L-1}$ where the model pushes the variational distribution to fit the observed data. The second and third terms are some regularizations term for the latent representation where the negated **KL** term in the third position appears to keep the model near the prior. A clear trade-off is happening via such loss function. The nodes are connected with invertible functions such as flow-based models (Dinh et al., 2016) 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 samples and the backward messages from the root. If all the nodes have only one parent, then the structure becomes a tree. If there several nodes have multiple parents, the graph will be a DAG (directed acyclic graph). It is easy to extend the computation of the ELBO (5) to DAGs with topology ordering of the nodes and thus the layer number. We develop the ELBO for a DAG structure as follows:

$$\begin{aligned}\log p(\mathbf{x}) \geqslant \mathcal{L}(\mathbf{x};\theta) = &\sum_{i\in\mathcal{G}\setminus\mathcal{R}_\mathcal{G}} \mathbb{E}_{q(\mathbf{h}^{pa(i)}|\mathbf{h}^{ch(pa(i))})}\Big[\log p(\mathbf{h}^{(i)}|\widehat{\mathbf{h}}^{pa(i)})\Big]\\ &+ \sum_{i\in\mathcal{G}\setminus\mathcal{R}_\mathcal{G}} \mathbf{H}(\mathbf{h}^{(i)}|\mathbf{h}^{ch(i)}) - \sum_{i\in\mathcal{R}_\mathcal{G}} \mathbf{KL}\big(q(\mathbf{h}^{(i)}|\mathbf{h}^{ch(i)})|p(\mathbf{h}^{(i)})\big). \qquad (6)\end{aligned}$$

Here $\mathcal{G}$ stands for the node set of the GAG, and $\mathcal{R}_\mathcal{G}$ is the set of root, or source, nodes.

Assume there are k leaf nodes on a tree or a DAG model, and they correspond to k sections of the input sample $\mathbf{x} = [\mathbf{x}^{(1)}, ..., \mathbf{x}^{(k)}]$, then the terms in both (5) and (6) are computed with . [BK: incomplete sentence, what did you want to say here?] We provide more details about the nodes in next subsection.

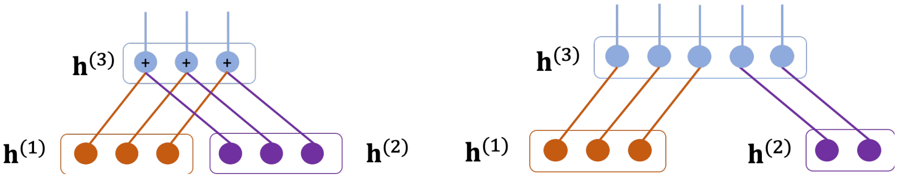

Figure 2: (Left) Aggregation with average. (Right) Aggregation with concatenation.

3.2 Node Aggregation

In the sequel, we consider that all nodes latent variables, noted $\mathbf{h}^{l,i}$, for all $l[L]$ and $i \in \mathbb{N}$, admit Gaussian distributions as prior distribution. There are two approaches to aggregate signals from different nodes: – Average-based and – Concatenation-based aggregation, see Figure 2 for an illustrrative scheme. While concatenation-based aggregation is simple and straightforward, we rather focus on Average-based aggregation for the purpose of this paper. We assume 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. To avoid cumbersome notations, we use the same standard deviation σ across all nodes. Extending to different values for each node does not affec tthe rest of the paper. Assume a model only has one average aggregation node as shown in Figure 2. According to (5), we have

$$\begin{aligned}\log p(\mathbf{x}) \geqslant \mathcal{L}(\mathbf{x};\theta_\mathbf{f}) =& \mathbb{E}_{q(\mathbf{h}^1|\mathbf{x})}\big[\log p(\mathbf{x}|\widehat{\mathbf{h}}^1)\big] + \mathbf{H}(\mathbf{h}^1|\mathbf{x})\\ &+ \mathbb{E}_{q(\mathbf{h}^2|\mathbf{h}^1)}\big[\log p(\mathbf{h}^1|\widehat{\mathbf{h}}^2)\big] - \mathbf{KL}\big(q(\mathbf{h}^2|\mathbf{h}^1)|p(\mathbf{h}^2)\big). \qquad (7)\end{aligned}$$

Note that in an average-based aggregation node i, the parent value is the mean of its children, i.e., $\mathbf{h}^{(i)} = \frac{1}{|ch(i)|}\sum_{j\in ch(i)} \mathbf{h}^{(j)}$. The children share the same reconstruction value with its parent, i.e., $\widehat{\mathbf{h}}^{(j)} = \widehat{\mathbf{h}}^{(i)}, \forall j \in ch(i)$. In an one aggregation node model with $\mathbf{h}^{(r)}$ as the root, we have

$$\widehat{\mathbf{h}}^{(r)} = \mathbf{h}^{(r)} = \frac{1}{k}\sum_{t=1}^{k} \mathbf{h}^{(t)} \quad \text{and} \quad \widehat{\mathbf{h}}^{(1)} = ... = \widehat{\mathbf{h}}^{(k)} = \widehat{\mathbf{h}}^{(r)}.$$

Here k is the children number, and $k = 3$ in Figure 2-left. Given one data sample $\mathbf{x}$, the reconstruction terms in ELBO equation 7 are computed with

$$\begin{aligned}\log p(\mathbf{x}|\widehat{\mathbf{h}}^1) + \log p(\mathbf{h}^1|\widehat{\mathbf{h}}^2) &= -\sum_{t=1}^{k}\left\{\frac{1}{2\sigma_\mathbf{x}^2}\big|\big|\mathbf{x}^{(t)} - \mathbf{f}_t^{-1}(\widehat{\mathbf{h}}^{(t)})\big|\big|^2 + \frac{1}{2\sigma^2}\big|\big|\mathbf{h}^{(t)} - \widehat{\mathbf{h}}^2\big|\big|^2\right\} + C\\ &= -\sum_{t=1}^{k}\left\{\frac{1}{2\sigma_\mathbf{x}^2}\big|\big|\mathbf{x}^{(t)} - \mathbf{f}_t^{-1}(\widehat{\mathbf{h}}^{(r)})\big|\big|^2 + \frac{1}{2\sigma^2}\big|\big|\mathbf{f}_t(\mathbf{x}^{(t)}) - \widehat{\mathbf{h}}^{(r)}\big|\big|^2\right\} + C.\end{aligned} \qquad (8)$$

Here $C = -dk\ln(2\pi) - \frac{dk}{2}\ln(\sigma_{\mathbf{x}}^2) - \frac{dk}{2}\ln(\sigma^2)$, and $\mathbf{f}_t$ connects $\mathbf{h}^{(t)}$ and $\mathbf{x}^{(t)}$. We use constant values for both $\sigma_{\mathbf{x}}^2$ and σ^2, hence the value of C is constant as well. We use the latent variables from a batch of training samples to approximate the entry $\mathbf{H}$ and $\mathbf{KL}$ terms in equation 7. We take the parent and children involved an aggregation operation as one node in the graphical figures, e.g., Figure 1.

3.3 Inference on Sub-graphs

Given a trained VFG model, we can infer the state of a node given the observed nodes. Relations between variables at different nodes can also be inferred via the flow-based graphical model that we propose. The prediction of leaf node i dependents on its parents, i.e., [BK: to complete here]

The hidden state of the parent node s in a single aggregation model can be approximated by the observed children, $\mathbf{h}^{(s)} = \frac{1}{|ch(s)|}\sum_{i\in ch(s)\cap O}\mathbf{h}^{(i)}$. Here O is the set of observed leaf nodes. Figure 3-left illustrates one example of this case.

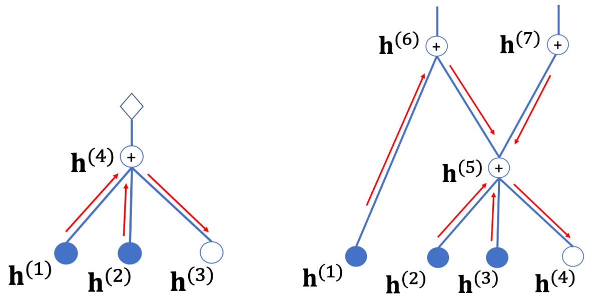

Figure 3: (Left) Inference of single aggregation node model. Node 4 aggregates from node 1 and 2, and pass the updated state to node 3 for prediction. (Right) Inference on a DAG model. Observed node states are gathered in node 5 to predict the state of node 4.

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 and then pass it back to the children without updating. Figure 3 illustrates this inference mechanism for both trees and DAGS. T he tree and DAG structures enable the model to perform message passing among the nodes. We now establish the following Lemma regarding 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 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 by:*

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

where we recall that using the normalizing flow equation (1), we have the following identity for each node of the graph structure:

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

The proof of Lemma 1 can be found in the appendix.

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

These results can be easily 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 (equation 5) or (equation 6) depending on what graph structure is used. In Algorithm 1, the inference of the latent variables is performed via forward message passing, cf. Line 5, and their reconstructions are computed in backward message passing, cf. Line 9.

[BK: To Improve. We should add a paragraph on implementation and ELBO/gradient computation] We use the empirical variance in a batch of training samples to approximate the entropy and **KL** terms. [BK: KL term between Gaussian priors is tractable, why do we approximate it?] Ignoring explicit variance for all latent nodes enable us to use flow-based models as the encoders as well as the decoders.

Algorithm 1 Inference model parameters with forward and backward message propagation

```
Input: Data distribution $\mathcal{D}$, $\mathcal{G} = \{\mathcal{V}, \mathbf{f}\}$
for $k = 0, 1, ...$ do
  Sample minibatch $b$ samples $\{\mathbf{x}_1, ..., \mathbf{x}_b\}$ from $\mathcal{D}$;
  for $i \in \mathcal{V}$ do
    $\mathbf{h}^{(i)} = \frac{1}{|ch(i)|} \sum_{j \in ch(i)} \mathbf{f}_{(j,i)}(\mathbf{h}^{(j)})$; // forward message passing
  end for
  $\mathbf{h} = \{\mathbf{h}^{(1)}, ..., \mathbf{h}^{(|\mathcal{V}|)}\}$;
  for $i \in \mathcal{V}$ do
    $\widehat{\mathbf{h}}^{(i)} = \frac{1}{|pa(i)|} \sum_{j \in pa(i)} \mathbf{f}^{-1,(i,j)}(\widehat{\mathbf{h}}^{(j)})$; // backward message passing
  end for
  $\widehat{\mathbf{h}} = \{\widehat{\mathbf{h}}^{1}, ..., \widehat{\mathbf{h}}^{(|\mathcal{V}|)}\}$;
  Updating flow-graph $\mathcal{G}$ using SGD: $\theta^{(k+1)} = \theta^{(k)} - \nabla_\theta \frac{1}{b} \sum_{i=1}^{b} \mathcal{L}(\mathbf{x}_b; \theta^{(k)})$ .
end for
```

4 Theory

The proposed VFG models provide approaches to integrate multi-modal data or data sets from different sources.

5 Experiments

We present in this section several numerical experiments to highlight the benefits of our VFG model. The first main application we present consists in missing values imputation. Several baseline models are compared with our newly introduced one on both synthetic and real datasets. The second application we present is the disentanglement learning tasks, as in finding latent representations that separate the explanatory factors of variations in the data, see (Bengio et al., 2013). For that latter application, the model is trained and evaluated on the MNIST handwritten digits dataset.

5.1 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.

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

- *Mean Value* We can directly use the mean values in the corresponding position of training set 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. We choose the KNeighborRegressor as the specific function (Pedregosa et al., 2011).
- *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 to impute the missing entries.
- *Multivariate Imputation by Chained Equation (MICE)* This method impute the missing entries with multiple rounds of inference. The method can handle different kind data types.

Evaluation with Synthetic Data: In this set of experiments, we study the proposed model with synthetic data sets. We use two latent variables, i.e. Z

We generate 1 000 data points for model training, and each data sample has 8 dimension with 2 latent variables. The relation between the latent variables and the [BK: to complete]

Figure 4 gives the ELBO values of the proposed method.

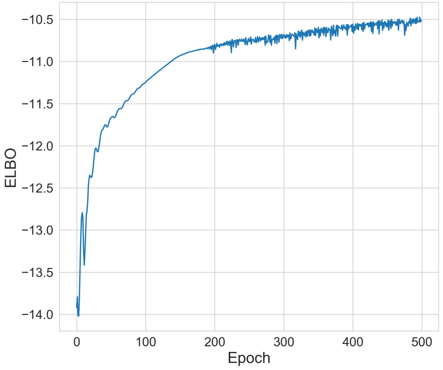

Figure 4: ELBO on the synthetic data

Methods	Imputation MSE
Mean Value	8.43
MICE	8.38
Iterative Imputation	2.64
KNN (k=3)	0.14
KNN (k=5)	0.18
Proposed	1.45

Table 1: Imputation Results on Synthetic Data.

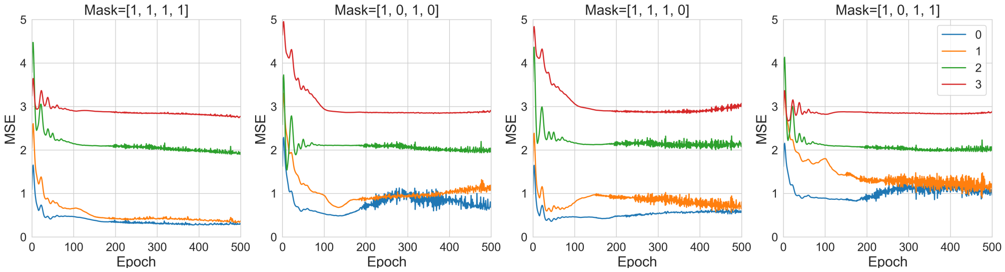

Figure 5: Imputation with Mask on the child nodes [0, 1, 2, 3] indicated by colored legends.

Arrhythmia Data Set: We further investigate the method on a tabular data set. The Arrhythmia (Dua & Graff, 2017) data set is obtained from the ODDS repository. The smallest classes, including 3, 4, 5, 7, 8, 9, 14, and 15, are combined to form the anomaly class, and the rest of the classes are combined to form the normal class. Table **??** shows the anomaly detection results with different methods.

5.2 Disentanglement on MNIST

6 Conclusion

REFERENCES

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.

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

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

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

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

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.

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.

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.

Michael I Jordan et al. Graphical models. *Statistical science*, 19(1):140–155, 2004.

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

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

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

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

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

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

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

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

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

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

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*, pp. 790. American Medical Informatics Association, 1990.

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

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. ELBO of Tree Models

The hierarchy generative network as given in Figure 6. 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

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

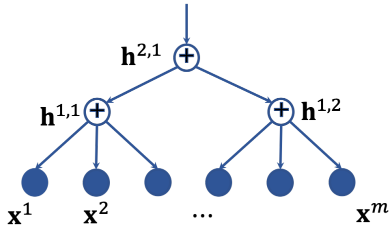

Figure 6: Tree structure.

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 6. Similarly, with the Markov property of the structure, the posterior density of the latent variables is given by

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

It can be simplified as

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

We also have

$$q(\mathbf{h}|\mathbf{x}) = q(\mathbf{h}^1|\mathbf{x})q(\mathbf{h}^{2:L}|\mathbf{h}^1). \tag{10}$$

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

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

With $\mathbf{KL}\big(q(\mathbf{h}|\mathbf{x})|p(\mathbf{h}|\mathbf{x})\big) \geq 0$, we have

$$
\begin{aligned}
&\log p(\mathbf{x}) \geq \mathcal{L}_\theta(x) && (11)\\
=&\mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\right]\\
=&\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{x}|\mathbf{h}^{1:L})p(\mathbf{h}^{1:L})}{q(\mathbf{h}^{1:L}|\mathbf{x})}\right]\\
=&\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{h}^{1:L})\right] + \mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^{1:L})}{q(\mathbf{h}^{1:L}|\mathbf{x})}\right]\\
=&\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{h}^{1})\right] + \mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^{1:L})}{q(\mathbf{h}^{1:L}|\mathbf{x})}\right] && (12)\\
=&\underbrace{\mathbb{E}_{q(\mathbf{h}^{1}|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{h}^{1})\right]}_{\substack{\text{Reconstruction of the data}\\ \text{given hidden layer 1}}} \underbrace{+\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^{1:L})}{q(\mathbf{h}^{1:L}|\mathbf{x})}\right]}_{-\mathbf{KL}^{1:L}}. && (13)
\end{aligned}
$$

The first term in equation 12 is due to $p(\mathbf{x}|\mathbf{h}^{1:L}) = p(\mathbf{x}|\mathbf{h}^1)$. The first term in equation 13 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 equation 13 as follows

$$
\begin{aligned}
&-\mathbf{KL}^{1:L} && (14)\\
=&\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^{1:L})}{q(\mathbf{h}^{1:L}|\mathbf{x})}\right]\\
=&\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^1|\mathbf{h}^{2:L})p(\mathbf{h}^{2:L})}{\underbrace{q(\mathbf{h}^1|\mathbf{x})q(\mathbf{h}^{2:L}|\mathbf{h}^1)}_{\text{Due to } equation \text{ 10}}}\right]\\
=&\underbrace{\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^1|\mathbf{h}^{2:L})p(\mathbf{h}^{2:L})}{q(\mathbf{h}^{2:L}|\mathbf{h}^1)}\right]}_{(a)} + \underbrace{\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{1}{q(\mathbf{h}^1|\mathbf{x})}\right]}_{(b)}
\end{aligned}
$$

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 equation 10, given the hidden states $\mathbf{h}^1$ samples from layer 0, we have

$$
(a) = \mathbb{E}_{q(\mathbf{h}^1|\mathbf{x})}\left[\mathbb{E}_{q(\mathbf{h}^{2:L}|\mathbf{h}^1)}\left[\log \frac{p(\mathbf{h}^1|\mathbf{h}^{2:L})p(\mathbf{h}^{2:L})}{q(\mathbf{h}^{2:L}|\mathbf{h}^1)}\right]\right]. \qquad (15)
$$

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

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

For the term (b),

$$\begin{aligned}(b) &= \mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log\frac{1}{q(\mathbf{h}^1|\mathbf{x})}\right]\\ &= \mathbb{E}_{q(\mathbf{h}^1|\mathbf{x})}\left[\log\frac{1}{q(\mathbf{h}^1|\mathbf{x})}\right]\\ &= \mathbf{H}(\mathbf{h}^1|\mathbf{x}).\end{aligned} \tag{17}$$

With equation 14 equation 15 equation 16 equation 17,

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

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

$$-\mathbf{KL}^{l:L} =\mathbb{E}_{q(\mathbf{h}^{l+1}|\mathbf{h}^l)}\big[\log p(\mathbf{h}^l|\mathbf{h}^{l+1})\big] + \mathbf{H}(\mathbf{h}^l|\mathbf{h}^{l-1}) - \mathbf{KL}^{l+1:L}. \tag{18}$$

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

$$\begin{aligned}&\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log\frac{p(\mathbf{h}^{1:L})}{q(\mathbf{h}^{1:L}|\mathbf{x})}\right] \qquad (19)\\ =&\sum_{l=1}^{L-1}\left\{\mathbb{E}_{q(\mathbf{h}^{l+1}|\mathbf{h}^l)}\left[\log p(\mathbf{h}^l|\mathbf{h}^{l+1})\right] + \mathbf{H}(\mathbf{h}^l|\mathbf{h}^{l-1})\right\}\\ &+\mathbb{E}_{q(\mathbf{h}^L|\mathbf{h}^{L-1})}\left[\log p(\mathbf{h}^{L-1}|\mathbf{h}^L))\right] - \mathbf{KL}\big(q(\mathbf{h}^L|\mathbf{h}^{L-1})|p(\mathbf{h}^L)\big)\end{aligned}$$

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

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

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

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

Appendix B. ELBO of 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 7. By taking the topology order as the layers in tree structures, we can derive the ELBO for DAG structures. Let's 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 equation 11 we have the ELBO for the log-likelihood of data

$$\begin{aligned}
&\log p(\mathbf{x}) \geq \mathcal{L}_\theta(x) \qquad (20)\\
=&\mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\right]\\
=&\underbrace{\mathbb{E}_{q(\mathbf{h}^{pa(\mathbf{x})}|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{h}^{pa(\mathbf{x})})\right]}_{\text{Reconstruction of the data given the parent nodes of the data}} + \underbrace{\mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\log \frac{p(\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\right]}_{-\mathbf{KL}}.
\end{aligned}$$

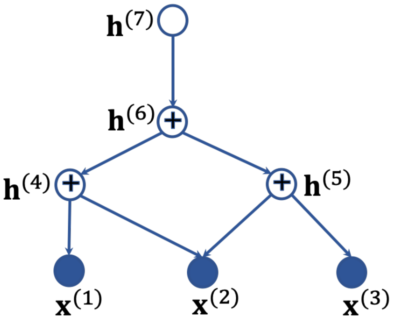

Figure 7: DAG structure. The inverse topology order is $\big\{$ {1,2,3}, {4,5}, {6}, {7} $\big\}$, and it corresponds to layers 0 to 3.

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

$$-\mathbf{KL}^{l:L} = \mathbb{E}_{q(\mathbf{h}^{pa(l)}|\mathbf{h}^l)}\big[\log p(\mathbf{h}^l|\mathbf{h}^{pa(l)})\big] + \mathbf{H}(\mathbf{h}^l|\mathbf{h}^{ch(l)}) - \mathbf{KL}^{l+1:L}. \qquad (21)$$

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

$$\begin{aligned}
-\mathbf{KL}^{l:L} = &\sum_{i:i\in l, i\notin \mathcal{R}_\mathcal{G}} \Big\{\mathbb{E}_{q(\mathbf{h}^{pa(i)}|\mathbf{h}^{ch(pa(i))})}\big[\log p(\mathbf{h}^i|\mathbf{h}^{pa(i)})\big] + \mathbf{H}_q(\mathbf{h}^i|\mathbf{h}^{ch(i)})\Big\} \qquad (22)\\
&- \sum_{i\in l\bigcap \mathcal{R}_\mathcal{G}} \mathbf{KL}\big(q(\mathbf{h}^{(i)}|\mathbf{h}^{ch(i)})|p(\mathbf{h}^{(i)})\big) - \mathbf{KL}^{l+1:L}.
\end{aligned}$$

By recurrently applying equation 22, we have

$$\begin{aligned}
&\mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\log \frac{p(\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\right] \qquad (23)\\
=&\sum_{l=1}^{L-1} \sum_{i:i\in l, i\notin \mathcal{R}_\mathcal{G}} \Big\{\mathbb{E}_{q(\mathbf{h}^{pa(i)}|\mathbf{h}^{(i)})}\Big[\log p(\mathbf{h}^{(i)}|\mathbf{h}^{pa(i)})\Big] + \mathbf{H}(\mathbf{h}^i|\mathbf{h}^{ch(i)})\Big\}\\
&- \sum_{l=1}^{L-1} \sum_{i\in l\bigcap \mathcal{R}_\mathcal{G}} \mathbf{KL}\big(q(\mathbf{h}^{(i)}|\mathbf{h}^{ch(i)})|p(\mathbf{h}^{(i)})\big) - \mathbf{KL}\big(q(\mathbf{h}^L|\mathbf{h}^{L-1})|p(\mathbf{h}^L)\big).
\end{aligned}$$

Since $L \subseteq \mathcal{R}_\mathcal{G}$, with $\mathbf{h}^{(0)} = \mathbf{x}$, equation 20, and equation 23 we have

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

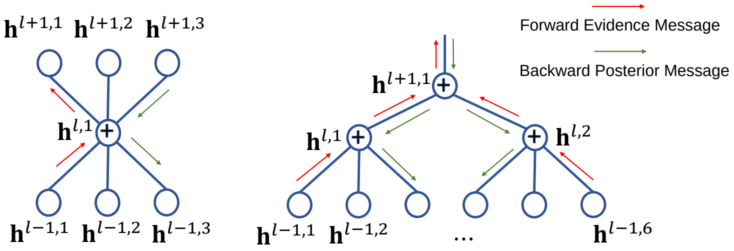

Figure 8: (Left) Message passing in a node. (Right) Message passing in a tree.

APPENDIX C. 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*

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