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).

In graphical models, we are often interested in the marginal distribution $p(\mathbf{x})$ of the observed variables $\mathbf{x}$. The maximization of such a likelihood $p(\mathbf{x})$ in a parameterized model is closely related to the inference of $p(\mathbf{x}|\mathbf{z})$ that is involved as a subroutine, given the Bayes theorem $p(\mathbf{x}|\mathbf{z}) = p(\mathbf{z}, \mathbf{x})/p(\mathbf{x})$ where $\mathbf{z}$ is the latent variable and $p(\mathbf{x}, \mathbf{z})$ is the joint distribution.

In this paper, we focus on this graphical inference subroutine. There are two general approaches: *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 leading to satisfactory results. However, the time complexity may be unacceptable or the intractable marginal likelihood presents a difficult challenge. Moreover, the exactitude achieved by the exact inference is not worth the computational cost in some cases, *e.g.* the distribution is well determined by a small cluster of nodes in the network Jordan et al. (1999). (ii) In contrast, approximate inference, *e.g.* Markov Chain Monte-Carlo (MCMC) and variational inference, yields deterministic approximation procedures that generally provide bounds on probabilities of interest. Considering the underlying slow convergence issues of stochastic MCMC sampling procedure Salimans et al. (2015), we prefer the deterministic optimization variational inference approach to tackle with the graphical inference problem in this paper. Variational inference provides a lower bound on $p(\mathbf{x})$ and is efficiently solvable by using off-the-shelf optimization techniques, and easily applicable to large datasets Liu & Wang (2016); Kingma & Welling (2013).

Mean-field approximation Xing et al. (2012) and variational message passing Winn & Bishop (2005) are two common approaches. They both require the intractable posterior $p(\mathbf{z}|\mathbf{x})$ that can be approximated by some family of distributions. However, on one hand, this kind of approximations is often limited by the choice of distributions that can't recover the true posterior, leading to a loose lower bound; on the other hand, they often lack a flexible structure to learn an inherent disentangled latent features thus can't encode enough information to reconstruct the data. The issues become more severe when we are dealing with higher dimensional data using a graphical model.

Motivated by these limitations, we propose a new framework to uncover the latent relational structures of high dimensional data by crafting a variational hierarchical graphical flow model. Our contributions are threefold:

- We construct hierarchical latent space between variables to uncover the latent structural relations of high dimensional data, leading to a tighter lower bound.
- 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.
- Experiments....

2 Preliminaries

In this section, we will first review the principles and general notations of normalizing flows and variational inference; then introduce their relations with graphical models.

2.1 Normalizing flow

Normalizing flow (Kingma & Dhariwal, 2018; Rezende & Mohamed, 2015) defines an invertible transformation $\mathbf{f} : \mathcal{Z} \to \mathcal{X}$ between two random variables. $\mathbf{z} \sim p(\mathbf{z})$ is the latent variable which has a tractable density. $\mathbf{x} \sim p_\theta(\mathbf{x})$ is an unknown true distribution which we want to model. We usually focus on a finite sequence of transformations $\mathbf{f} = \mathbf{f}_1 \circ \mathbf{f}_2 \circ \cdots \circ \mathbf{f}_L$ such that $\mathbf{x} = \mathbf{f}(\mathbf{z})$ and $\mathbf{z} = \mathbf{f}^{-1}(\mathbf{x})$:

$$\mathbf{z} \underset{\mathbf{f}_1^{-1}}{\overset{\mathbf{f}_1}{\rightleftharpoons}} \mathbf{h}^1 \underset{\mathbf{f}_2^{-1}}{\overset{\mathbf{f}_2}{\rightleftharpoons}} \mathbf{h}^2 \cdots \underset{\mathbf{f}_L^{-1}}{\overset{\mathbf{f}_L}{\rightleftharpoons}} \mathbf{x}.$$

Under the change of variables theorem equation 1, 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.

2.2 Variational inference

Given above, the mapping $\mathbf{f}$: $\mathbf{x} \to \mathbf{z}$ can be taken as encoding process (inference or recognition), and the mapping $\mathbf{f}^{-1}$: $\mathbf{z} \to \mathbf{x}$ be taken as decoding process (generation): $\mathbf{z} \sim p(\mathbf{z}), \mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z})$. To learn the parameters θ, one typically maximizes the following marginal log-likelihood:

$$\log p_\theta(\mathbf{x}) = \int p(\mathbf{z}) p_\theta(\mathbf{x}|\mathbf{z}) d\mathbf{z}.$$

Direct optimization of the log-likelihood is usually intractable. Variational inference instead parameterizes a family of variational distribution $q_\phi(\mathbf{z}|\mathbf{x})$ to approximate the true posterior $p_\theta(\mathbf{z}|\mathbf{x}) \propto p(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})$, ending up optimizing the following evidence lower bound (ELBO):

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

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 models, each node $\mathbf{v}$ corresponds to a random variable, *e.g.* latent variables $\mathbf{z}$ and observed variables $\mathbf{x}$ in the variational framwork and the edge represents statistical dependencies between the variables, *e.g.* a function $\mathbf{f}_\theta$ parameterized by θ. 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 is to find a variational distribution, *e.g.* $q(\mathbf{z}|\mathbf{x})$, to approximate the true posterior $p(\mathbf{z}|\mathbf{x})$. This exactly coincides with the general variational inference framework as in equation 2 in the last subsection. In this paper, we focus on the factorization of the independency of 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, implying 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 the latent space and the observation space are bridged by a sequence of variables, we can build a graphical model with normalizing flow, leading to exact latent-variable inference and log-likelihood evaluation of the model. We call this model as *Variational Flow Graphical Model* (VFG).

3.1 The evidence lower bound of Variational Flow Graphical Model

Figure 1 illustrates the tree structure induced by varational flow. The hierarchical generative network has L layers, and $\mathbf{h}^l$ is the latent variable in layer l, and θ is the parameter vector of the model. The hierarchical generative process of the model is given by

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

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

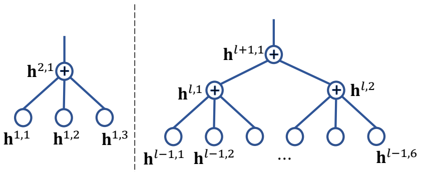

Figure 1: (Left) The structure of one node. Node $\mathbf{h}^{2,1}$ connects with its children with invertible functions. The messages from its children are aggregated at $\mathbf{h}^{2,1}$. (Right)An illustration of the latent structure from layer $l-1$ to $l+1$. $\mathbf{h}^{h,i}$ means the ith latent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

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

where $\mathbf{h} = \{\mathbf{h}^1, \cdots, \mathbf{h}^L\}$ denotes all latent variables. For node i, we use $\mathbf{h}^{(i)}$ as the forward evidence message receives from its children, and $\widehat{\mathbf{h}}^{(i)}$ as the reconstruction of $\mathbf{h}^{(i)}$ with backward message from the rood. $ch(i)$ and $pa(i)$ are node i's child set and parent set, respectively. Let $\mathbf{f}_{(i,j)}$ be the direct edge (function) from node i to j, and $\mathbf{f}^{-1}_{(i,j)}$ or $\mathbf{f}_{(j,i)}$ is its inverse function. We have

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

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

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

The derivation of the ELBO can be found in the Appendix. The first term of ELBO is the reconstruction for each layer: $\mathbf{x}$ and the latent representations $\mathbf{h}^1, ..., \mathbf{h}^{L-1}$. The second and third terms are the regularizations for the latent representation. 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 has 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 is a tree. If there are nodes have multiple parents, the graph will be a DAG (directed acyclic graph). It is easy to extend the ELBO equation 5 to DAGs with topology ordering of the nodes and thus the layer number. We have the ELBO for a DAG structure as follows

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

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 $\mathbf{x} = [\mathbf{x}^{(1)}, ..., \mathbf{x}^{(k)}]$. The terms in both equation 5 and equation 6 are computed with . We provide more details about the nodes in next subsection.

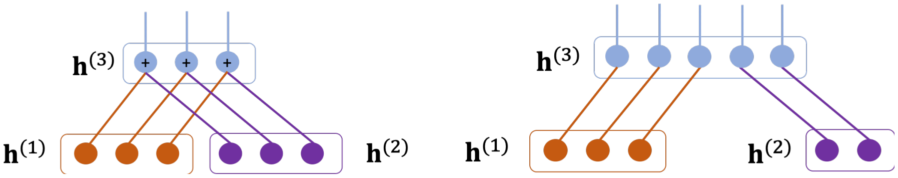

Figure 2: (Left) Aggregation with average. (Right) Aggregation with concatenation.

3.2 Node Aggregation

We use normal distribution as the prior for all the nodes. There are two approaches to aggregate signals from different nodes: average based and concatenation based aggregation. They are illustrated by the left and right plots of Figure 2, respectively. Concatenation based aggregation is simple and straight forward. We will focus on average aggregation nodes. We assume each entry of a hidden node follow normal distribution, i.e., $\mathbf{h}_j^{(i)} \sim \mathbb{N}(\mu_j^{(i)}, \sigma^2)$ for node i's jth entry. We use the same σ value across all nodes. Let's assume a model only has one average aggregation node as shown in Figure 2-left. According to with the ELBO

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

In an average 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 $\widehat{\mathbf{h}}^{(r)} = \mathbf{h}^{(r)} = \frac{1}{k}\sum_{t=1}^k \mathbf{h}^{(t)}$, and $\widehat{\mathbf{h}}^{(1)} = ... = \widehat{\mathbf{h}}^{(k)} = \widehat{\mathbf{h}}^{(r)}$. 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

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

Here $C = -dk\ln(2\pi) - \frac{dk}{2}\ln(\sigma_{\mathbf{y}}^2) - \frac{dk}{2}\ln(\sigma^2)$, and $\mathbf{f}_t$ connects $\mathbf{h}^{(t)}$ and $\mathbf{x}^{(t)}$. We use constant values for both $\sigma_{\mathbf{y}}^2$ 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 a node's state given observed nodes. Relations between variables at different nodes can also be infered with the model. The prediction of leaf node i dependents on its parents, i.e.,

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

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. For a node in a tree or DAG model, its state is updated with messages from its children with updating and then pass it to children without updating. Figure 3-right illustrates inference on a DAG. The tree and DAG structures enable the model to perform message passing among the nodes. We have the following lemma regarding the relation between two leaf nodes.

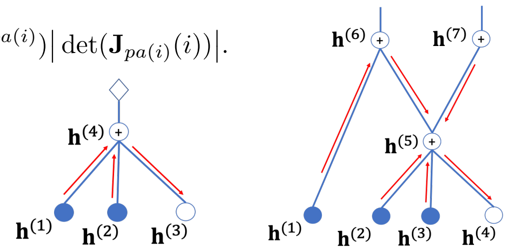

Figure 3: (Left) Inference of single aggregation node model. Node 4 aggregates messages from node 1 and 2, and then pass the updated state to node 3 for prediction. (Right) Inference on a DAG model. Observed node states are gathered in node 5 to predict the state of node 4.

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*

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

Remark 1. *Let O be the set of observed leaf nodes, j be an unobserved node, and a is the closest ancestor of O and a. Then the state of j can be imputed with $\widehat{\mathbf{x}}^j \approx \mathbf{f}_{(a,j)}(\mathbf{f}_{(O,a)}(\mathbf{x}^{(i)}))$. $\mathbf{f}_{(O,a)}$ is 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. The proof of Lemma 1 can be found in the appendix.

3.4 Algorithm and Implementation

Model parameters $\theta_{\mathbf{f}}$ are learned by maximizing the ELBO equation 5 or equation 6. The latent variables are computed in forward message passing and their reconstructions are computed in backward message passing. We use the empirical variance in a batch of training samples to approximate the entropy and $\mathbf{KL}$ terms. Ignoring explicit variance for all latent nodes enable us to use flow-based models as the encoders as well as the decoders.

4 Theory

The proposed VFG model

Algorithm 1 Inference model parameters with forward and backward message propagation

```
Input: Data distribution $\mathcal{D}$, $\mathcal{G} = \{\mathcal{V}, \mathbf{f}\}$
repeat
    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 $\mathcal{G}$ by descending the gradient $\bigtriangledown_{\theta_\mathbf{f}} \frac{1}{b} \sum_{i=1}^{b} \big[ -\mathcal{L}(\mathbf{x}_b; \theta_\mathbf{f}) \big]$ ;
until Converge
```

Table 1: Imputation Results on 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

5 Experiments

5.1 Imputation

5.1.1 Baselines

The data set is divided into training and testing sets. The model is trained with the training set, then it is used to infer the missing entries of samples in the testing set. 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.

5.1.2 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 1000 data points for model training, and each data sample has 8 dimension with 2 latent variables. The relation between the latent variables and the

Figure 4 gives the ELBO values of the proposed method.

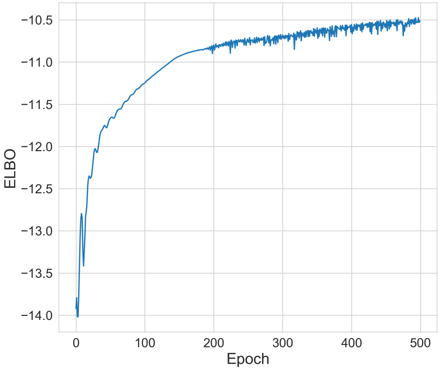

Figure 4: ELBO on the synthetic data

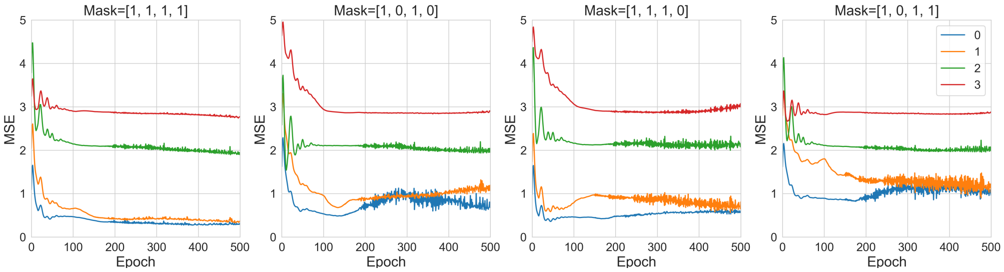

Figure 5: Imputation with Mask on the child nodes [0, 1, 2, 3] indicated by colored legends.

5.1.3 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 4 shows the anomaly detection results with different methods.

5.2 Disentanglement on MNIST

6 Conclusion

References

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`.

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

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

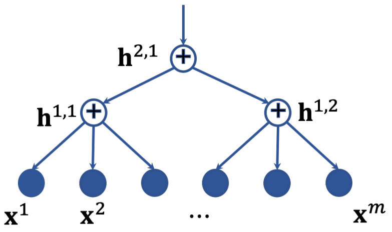

Figure 6: Tree structure.

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

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

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

$$\log p(\mathbf{x}) \geq \mathcal{L}_\theta(x) \tag{11}$$

$$=\mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\right]$$

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

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

$$=\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{h}^{1})\right] + \mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^{1:L})}{q(\mathbf{h}^{1:L}|\mathbf{x})}\right] \tag{12}$$

$$= \underbrace{\mathbb{E}_{q(\mathbf{h}^{1}|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{h}^{1})\right]}_{\substack{\text{Reconstruction of the data}\\ \text{given hidden layer 1}}} \underbrace{+\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^{1:L})}{q(\mathbf{h}^{1:L}|\mathbf{x})}\right]}_{-\mathrm{KL}^{1:L}}. \tag{13}$$

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

$$-\,\mathrm{KL}^{1:L} \tag{14}$$

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

$$=\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^1|\mathbf{h}^{2:L})p(\mathbf{h}^{2:L})}{\underbrace{q(\mathbf{h}^1|\mathbf{x})q(\mathbf{h}^{2:L}|\mathbf{h}^1)}_{\text{Due to } equation\ 10}}\right]$$

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

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 the feed message from the child layer $\overrightarrow{\mathbf{h}}^{(i)}$ and the reconstruct message $\overleftarrow{\mathbf{h}}^{(i)}$ from the parent layer, we can derive the ELBO term for the likelihood of $\overrightarrow{\mathbf{h}}^{(i)}$. With equation 10, given the hidden states $\mathbf{h}^1$ samples from layer 0, we have

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

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

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

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

$$=\mathbb{E}_{q(\mathbf{h}^{2}|\mathbf{h}^1)}\left[p(\mathbf{h}^1|\mathbf{h}^{2})\right] + \mathbb{E}_{q(\mathbf{h}^{2:L}|\mathbf{h}^1)}\left[\log \frac{p(\mathbf{h}^{2:L})}{q(\mathbf{h}^{2:L}|\mathbf{h}^1)}\right] \tag{16}$$

$$=\mathbb{E}_{q(\mathbf{h}^{2}|\mathbf{h}^1)}\left[p(\mathbf{h}^1|\mathbf{h}^{2})\right] - \mathrm{KL}^{2:L}.$$

For the term (b),

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

With equation 14 equation 15 equation 16 equation 17,

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

Similarly, for layer l, we have

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

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

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

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

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

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

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

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

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

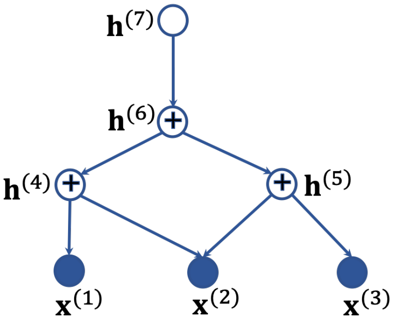

Figure 7: DAG structure. The inverse topology order is $\{$ {1,2,3}, {4,5}, {6}, {7} $\}$, and it corresponds to layers 0 to 3.

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

$$-\mathrm{KL}^{l:L} = \mathbb{E}_{q(\mathbf{h}^{pa(l)}|\mathbf{h}^{l})}\left[p(\mathbf{h}^{l}|\mathbf{h}^{pa(l)})\right] + \mathrm{H}_q(\mathbf{h}^{l}|\mathbf{h}^{ch(l)}) - \mathrm{KL}^{l+1:L}. \tag{21}$$

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

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

By recurrently applying equation 22, we have

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

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

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

Appendix C. Proof of Lemma1

Let**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,*

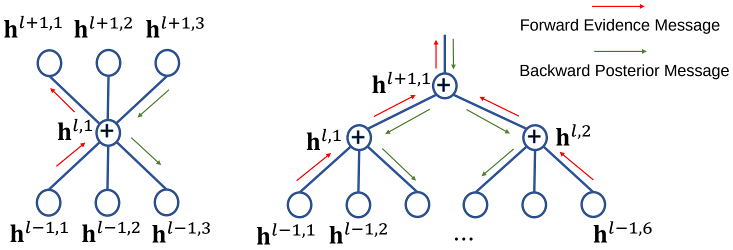

Figure 8: (Left) Message passing in a node. (Right) Message passing in a tree.

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

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