Supplemental File for ‘Variational Flow Graphical Model’

The proposed variational flow graphical models assemble flow functions with tree or DAG structures via variational inference on aggregation nodes. In this supplemental file, we first present more results, then we give more details on techniques/methodology of VFG models.

A Additional Numerical Experiments

All the experiments are conducted on NVIDIA-TITAN X (Pascal) GPUs. In our experiments, we use the same coupling block [7] to construct different flow functions. The coupling block consists in three fully connected layers (of dimension 64) separated by two RELU layers along with the coupling trick. Each flow function has block number $b \geqslant 2$. All latent variables, $\mathbf{h}^i, i \in \mathcal{V}$ are forced to be non-negative via Sigmoid or RELU functions. Non-negativeness can help the model to identify sparse structures of the latent space.

A.1 California Housing Dataset

We further investigate the method on a real dataset. The California Housing 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, as the goal is to impute this missing section. Here, we construct a tree structure VFG with 2 layers. The first layer has two aggregation nodes, and each of them has two children. The second layer consists of one aggregation node that has two children connecting with the first layer. Each flow function has 4 coupling blocks. We can see Table 2 that our model yields significantly better results than any other method in terms of prediction error.

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 2: California Housing dataset: Imputation Mean Squared Error (MSE) results.

A.2 Representation Learning with MNIST

For MNIST, we construct a tree structure VFG model depicted in Figure 7. In the first layer, there are 4 flow functions, and each of them takes 14×14 image blocks as the input. Thus a 28×28 input image is divided into four 14×14 blocks as the input of VFG model. The four nodes are aggregated as the input of the upper layer flow.

A.2.1 Latent Representation Learning on MNIST

Figure 9 presents the t-SNE plot of the root latent variables from VFG trained without labels. The figure clearly shows that even without label information, different digits’ representation are roughly scattered in different areas. Compared to Figure 8 in section 6.3, label information indeed can improve the latent representation learning.

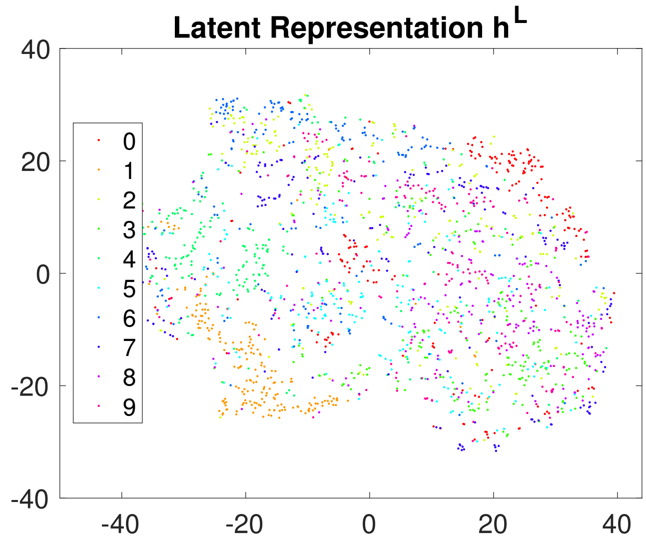
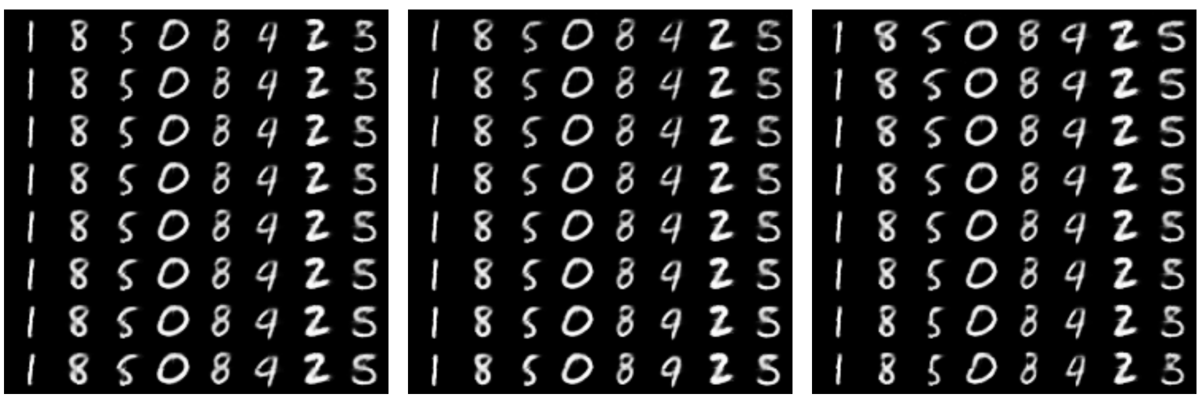
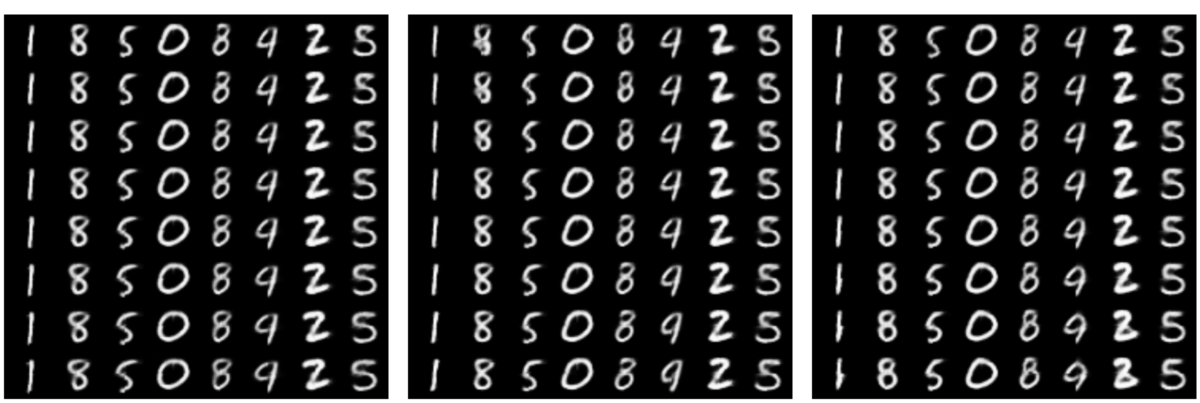


Figure 9: MNIST: t-SNE plot of latent variables from VFG learned without labels.

A.2.2 Disentanglement on MNIST



(a) i=6, Width (b) i=19, Brightness (c) i=60, Upper width



(d) i=92 (e) i=119, Angle (f) i=157

Figure 10: MNIST: Increasing each latent variable from a small value to a larger one.

We study disentanglement on MNIST with our proposed VFG model introduced in section 6.3. But different from the model in section 6.3, here, the distribution parameter λ for all latent variables are set to be trainable across all layers. Each digit has its trainable vector, $\lambda \in \mathbb{R}^d$ that is used across all layers. To show the disentanglement of learned latent representation, we first obtain the root latent variables of a set of images through forward message passing. Each latent variable's values are changed increasingly within a range centered at the value of the latent variable obtained from last step. This perturbation is performed for each image in the set. Figure 10 shows the change of images by increasing one latent variable from a small value to a larger one. The figure presents some of the latent variables that have obvious effects on images, and most of the $d = 196$ variables do not impact the generation significantly. Latent variables $i = 6$ and $i = 60$ control the digit width. Variable $i = 19$ affects the brightness. $i = 92, i = 157$ and some of the variables not displayed here control the style of the generated digits.

B ELBO Calculation

The recognition model in a VFG is the neural network (encoder) used to approximate the posterior of latent variables. With invertible neural networks (flows), the recognition model and the generative model in a VFG share the same structure and parameters. As shown in Figure 11, the recognition and generative models are realized with forward and backward message passing, respectively.

Maximize the ELBOs (5,7) requires evaluation of both the reconstruction and the $\mathbf{KL}$ terms. It involves samples from both the recognition and generative models. In this section, we first gives the conditional distributions in both generative model and the posterior, then give more details on ELBO computation. We start from tree models, and it is easy to extend to DAG models.

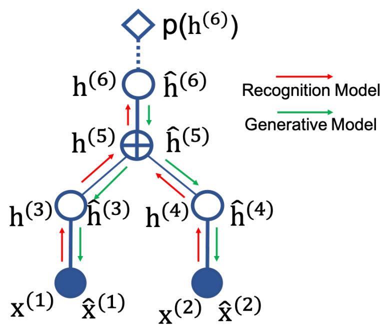


Figure 11: The recognition model consists of froward message from data to approximate the posterior distributions; the generative model is realized by backward message from the root node.

B.1 Distributions of Latent Variables

B.1.1 Generative Model

In a tree VFG, the sample reconstruction in the generative model consists of layer-wise backward message passing, i.e., latent variable generation in each layer. For any $l, 0 \leq l \leq L-1$, latent variable backward state (reconstruction) $\widehat{\mathbf{h}}^l$ is propagated from layer $l+1$ via the flow function $\mathbf{f}_l$ between the two layers with $\widehat{\mathbf{h}}^l = \mathbf{f}_l^{-1}(\widehat{\mathbf{h}}^{l+1})$.

The prior $p(\mathbf{h}^L)$ for the root latent variable $\mathbf{h}^L$ is Laplace(0,1). With a sample $\widehat{\mathbf{h}}^L$ from the posterior, i.e., $\widehat{\mathbf{h}}^L = \mathbf{h}^L \sim q(\cdot|\mathbf{h}^{L-1})$, the conditional distribution for latent variable in layer l is $p(\cdot|\widehat{\mathbf{h}}^{l+1}) :=$ Laplace$(\widehat{\mathbf{h}}^l, 1)$. Here the location parameter is generated from layer $l+1$, i.e., $\widehat{\mathbf{h}}^l = \mathbf{f}_l^{-1}(\widehat{\mathbf{h}}^{l+1})$.

For a latent variable $\mathbf{h}^l$ sampling from the posterior, its log-likelihood regarding $p(\cdot|\widehat{\mathbf{h}}^{l+1})$ in (10) is given by

$$\log p(\mathbf{h}^l|\widehat{\mathbf{h}}^{l+1}) = -\|\mathbf{h}^l - \widehat{\mathbf{h}}^l\|_1 - d \cdot \log 2.$$

Here $d = dim(\mathbf{h}^l)$. Hence, minimizing $\mathbf{KL}$s is to minimize the ℓ_1 distance between latent variables and their reconstructions.

B.1.2 Recognition Model

The forward message passing in the recognition model consists of layer-wise sample generation. In layer $l, 1 \leq l \leq L$, latent variable forward state $\mathbf{h}^l$ is propagated from layer $l-1$ via the flow function $\mathbf{f}_{l-1}$ between the two layers with $\mathbf{h}^l = \mathbf{f}_{l-1}(\mathbf{h}^{l-1})$.

We assume each entry of hidden variable $\mathbf{h}^l$ follows a Laplace distribution, i.e., $\mathbf{h}_j^l \sim \text{Laplace}(\mu_j^l, s_j^l)$ for layer l's jth entry. Here μ_j^l is the location and s_j^l is the scale. Compared with other distributions, Laplace can introduce sparsity to the model and it works well in practice. At level $l \in [L]$, we set $q(\cdot|\mathbf{h}^{l-1}) := \text{Laplace}(\mu^l, \mathbf{s}^l)$ with

$$\mu^l = \text{median}\big(H\big), \ \ \mathbf{s}^l = \frac{1}{B}\sum_{b=1}^{B} |\mathbf{h}^l(\mathbf{x}_b) - \mu^l| \ . \tag{15}$$

Here $H = \{\mathbf{h}^l(\mathbf{x}_b)|1 \leqslant b \leqslant B\}$ is a batch of latent values generated from a batch of data samples with size B, i.e., $X_B = \{\mathbf{x}_b|1 \leqslant b \leqslant B\}$. The median operation is performed element-wisely. For each $\mathbf{x}_b$, $\mathbf{h}^l(\mathbf{x}_b) = \mathbf{f}^{l-1}\big(\mathbf{h}^{l-1}(\mathbf{x}_b)\big)$.

B.2 KL Term

For any $l, 1 \leq l \leq L-1$, the calculation of the $\mathbf{KL}^l$ term (6) requires message passing and samples from both recognition and generative models, i.e.,

$$\mathbf{KL}^l = \mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\big[\log q(\mathbf{h}^l|\mathbf{h}^{l-1}) - \log p(\mathbf{h}^l|\widehat{\mathbf{h}}^{l+1})\big] \simeq \log q(\mathbf{h}^l|\mathbf{h}^{l-1}) - \log p(\mathbf{h}^l|\widehat{\mathbf{h}}^{l+1}). \tag{16}$$

Here $q(\cdot|\mathbf{h}^{l-1})$ is a Laplace with location and scale equal to the median and scale defined in equation 15; $p(\cdot|\widehat{\mathbf{h}}^{l+1})$ is a Laplace parameterized with $(\widehat{\mathbf{h}}^l, 1.0)$ as discussed in B.1.1. Hence with $\mathbf{h}^l$ we can compute the log-likelihoods on RHS of (16) and thus the $\mathbf{KL}^l$ value. When $l = L$, $\mathbf{KL}^L = \mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\big[\log q(\mathbf{h}^L|\mathbf{h}^{L-1}) - \log p(\mathbf{h}^L)\big] \simeq \log q(\mathbf{h}^L|\mathbf{h}^{L-1}) - \log p(\mathbf{h}^L)$.

Assuming 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 (5) and (7) are computed with forward and backward message passing. Next, we provide more details about the nodes.

In practice, we set $M = 1$ for efficiency. With a batch of training samples, $\mathbf{x}_b, 1 \leqslant b \leqslant B$, the structure of flow functions make the forward and backward message passing very efficient, and thus the estimation of the ELBO.

B.3 Reconstruction Term

The reconstruction term in ELBO (5) can be computed with the backward message from the generative model $p(\mathbf{x}|\widehat{\mathbf{h}}^1)$, i.e.,

$$\begin{aligned}&\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\big[\log p(\mathbf{x}|\mathbf{h}^{1:L})\big] = \mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\big[\log p(\mathbf{x}|\widehat{\mathbf{h}}^{1:L})\big]\\ &\simeq \frac{1}{M}\sum_{m=1}^{M}\log p(\mathbf{x}|\widehat{\mathbf{h}}_m^{1:L}) = \frac{1}{M}\sum_{m=1}^{M}\log p(\mathbf{x}|\widehat{\mathbf{h}}_m^{1}) \simeq \log p(\mathbf{x}|\widehat{\mathbf{h}}^1).\end{aligned}$$

For a VFG model, we set $M = 1$. In the last term, $p(\mathbf{x}|\widehat{\mathbf{h}}^1)$ is either Gaussian or Binary distribution parameterized with $\widehat{\mathbf{x}}$ generated via the flow function with $\widehat{\mathbf{h}}^1$ as the input.

C Aggregation Node

Let $\mathbf{f}_{(i,j)}$ be the direct edge (function) from node i to node j, and $\mathbf{f}_{(i,j)}^{-1}$ or $\mathbf{f}_{(j,i)}$ defined as its inverse function. Then, at an aggregation node i that has multiple ($|ch(i)|$) children, its latent variable in forward message passing is the mean of all children's output, i.e.,

$$\mathbf{h}^{(i)} = \frac{1}{|ch(i)|}\sum_{j \in ch(i)} \mathbf{f}_{(j,i)}(\mathbf{h}^{(j)}). \tag{17}$$

On the other hand, if node i in a DAG has multiple parents, the reconstruction of its latent variable is the mean of all parents' output , i.e.,

$$\widehat{\mathbf{h}}^{(i)} = \frac{1}{|pa(i)|} \sum_{j \in pa(i)} \mathbf{f}^{-1}_{(i,j)}(\widehat{\mathbf{h}}^{(j)})\,. \tag{18}$$

Notice that the above two equations hold even when node i has only one child or parent.

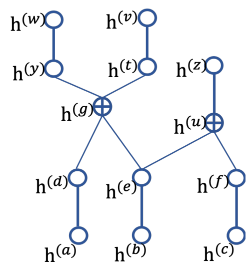


Figure 12: Aggregation node on a DAG.

Besides averaging, the aggregation nodes also ensure the latent variable on the two ends of an identity function are *consistent*. We use node i in the DAG presented in Figure 12 as an example. Node i has two parents, u and v; and two children, d and e. Node i connects its parents and children with identity functions. According to (17) and (18), we have $\mathbf{h}^{(i)} = (\mathbf{h}^{(d)} + \mathbf{h}^{(e)})/2$ and $\widehat{\mathbf{h}}^{(i)} = (\widehat{\mathbf{h}}^{(u)} + \widehat{\mathbf{h}}^{(v)})/2$. Here aggregation consistent means, for i's children, their forward state should be consistent with i's backward state, i.e.,

$$\mathbf{h}^{(d)} = \mathbf{h}^{(e)} = \widehat{\mathbf{h}}^{(i)}. \tag{19}$$

For i's parents, their backward state should be consistent with i's forward state, i.e.,

$$\widehat{\mathbf{h}}^{(u)} = \widehat{\mathbf{h}}^{(v)} = \mathbf{h}^{(i)}. \tag{20}$$

We utilize the $\mathbf{KL}$ term in the ELBOs to ensure (19) and (20) can be satisfied during parameter updating. The $\mathbf{KL}$ term regarding node i is

$$\mathbf{KL}^{(i)} = \mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\big[\log q(\mathbf{h}^{(i)}|\mathbf{h}^{ch(i)}) - \log p(\mathbf{h}^{(i)}|\widehat{\mathbf{h}}^{pa(i)})\big] \simeq \log q(\mathbf{h}^{(i)}|\mathbf{h}^{ch(i)}) - \log p(\mathbf{h}^{(i)}|\widehat{\mathbf{h}}^{pa(i)}).$$

Here

$$\begin{aligned}\log p(\mathbf{h}^{(i)}|\widehat{\mathbf{h}}^{pa(i)}) =& \frac{1}{2}\big(\log p(\mathbf{h}^{(i)}|\widehat{\mathbf{h}}^{(u)}) + p(\mathbf{h}^{(i)}|\widehat{\mathbf{h}}^{(v)})\big) \\ =& \frac{1}{2}\big(-\|\mathbf{h}^{(i)} - \widehat{\mathbf{h}}^{(u)}\|_1 - \|\mathbf{h}^{(i)} - \widehat{\mathbf{h}}^{(v)}\|_1 - 2d \cdot \log 2\big).\end{aligned}$$

Hence minimizing $\mathbf{KL}^{(i)}$ is equal to minimize $\{\|\mathbf{h}^{(i)} - \widehat{\mathbf{h}}^{(u)}\|_1 + \|\mathbf{h}^{(i)} - \widehat{\mathbf{h}}^{(v)}\|_1\}$ which achieves the objective in equation 20.

Similarly, $\mathbf{KL}$s of i's children intend to realize consistency given in equation 19. We use node d as an example. The $\mathbf{KL}$ term regarding d is

$$\mathbf{KL}^{(d)} = \mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\big[\log q(\mathbf{h}^{(d)}|\mathbf{h}^{ch(d)}) - \log p(\mathbf{h}^{(d)}|\widehat{\mathbf{h}}^{pa(d)})\big] \simeq \log q(\mathbf{h}^{(d)}|\mathbf{h}^{ch(d)}) - \log p(\mathbf{h}^{(d)}|\widehat{\mathbf{h}}^{pa(d)}).$$

With

$$\log p(\mathbf{h}^{(d)}|\widehat{\mathbf{h}}^{pa(d)}) = \log p(\mathbf{h}^{(d)}|\widehat{\mathbf{h}}^{(i)}) = -\|\mathbf{h}^{(d)} - \widehat{\mathbf{h}}^{(i)}\|_1 - d \cdot \log 2,$$

minimizing $\mathbf{KL}^{(d)}$ is to minimize $\|\mathbf{h}^{(d)} - \widehat{\mathbf{h}}^{(i)}\|_1$ that targets at equation 19. In summary, by maximizing the ELBO of a VFG, the aggregation consistency can be attained along with fitting the model to the data.

D More Details on Inference

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.*

Proof. Without loss generality, we assume that there are relationships among different data sections, and the value of one section can be partially or approximately imputed by other sections. According to the aggregation rule (b) discussed in section 3.3, at an aggregation node a, the latent value of a child node j has the same reconstruction value as the parent node. The reconstruction of the child node j can be approximated with the reconstruction of the parent node, i.e., $\widehat{\mathbf{h}}^{(j)} \approx \mathbf{f}_{(a,j)}(\widehat{\mathbf{h}}^{a)})$. Recalling the reconstruction term in the ELBO (5), at each node we have $\mathbf{h}^{(a)} \approx \widehat{\mathbf{h}}^{(a)}$. Hence for node a's descendent j, we have $\widehat{\mathbf{h}}^{(j)} \approx \mathbf{f}_{(a,j)}(\mathbf{h}^{(a)})$, and $\mathbf{f}_{(a,j)}$ is the flow function path from a to j. The value of node a can be approximated by the value of its descendent i that has observation, i.e., $\mathbf{h}^{(a)} \approx \mathbf{f}_{(i,a)}(\mathbf{h}^{(i)})$. Hence, we have $\widehat{\mathbf{x}}^{(j)} \approx \mathbf{f}_{(a,j)}(\mathbf{f}_{(i,a)}(\mathbf{x}^{(i)}))$. $\square$

Lemma 1 and Remark 1 provide an approach to conduct inference on a tree and impute missing values in the dataset. It is easy to extend the inference method to DAG VFG models.

E Derivation of the ELBOs for Trees and DAGs

E.1 ELBO of Tree Models

Let each data sample has k sections, i.e., $\mathbf{x} = [\mathbf{x}^{(1)}, ..., \mathbf{x}^{(k)}]$. VFGs are graphical models that can integrate different sections or components of the dataset. We assume that for each pair of connected nodes, the edge is an invertible flow function. The vector of parameters for all the edges is denoted by θ. The forward message passing starts from $\mathbf{x}$ and ends at $\mathbf{h}^L$, and backward message passing in the reverse direction. We start with the hierarchical generative tree network structure illustrated by an example in Figure 13. Then the marginal likelihood term of the data reads

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

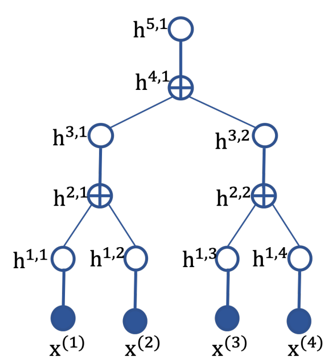


Figure 13: A tree VFG with $L = 5$ and three aggregation nodes.

The hierarchical prior distribution is given by factorization

$$p(\mathbf{h}) = p(\mathbf{h}^L)\mathbf{\Pi}_{l=1}^{L-1}p(\mathbf{h}^l|\mathbf{h}^{l+1}). \tag{21}$$

The probability density function $p(\mathbf{h}^{l-1}|\mathbf{h}^l)$ in the prior is modeled with one or multiple invertible normalizing flow functions. The hierarchical posterior (recognition network) is factorized as

$$q_\theta(\mathbf{h}|\mathbf{x}) = q(\mathbf{h}^1|\mathbf{x})q(\mathbf{h}^2|\mathbf{h}^1)\cdots q(\mathbf{h}^L|\mathbf{h}^{L-1}). \tag{22}$$

Draw samples from the prior (21) involves sequential conditional sampling from the top of the tree to the bottom, and computation of the posterior (22) takes the reverse direction. Notice that

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

With the hierarchical structure of a tree, we further have

$$q(\mathbf{h}^{l:L}|\mathbf{h}^{l-1}) = q(\mathbf{h}^l|\mathbf{h}^{l-1})q(\mathbf{h}^{l+1:L}|\mathbf{h}^l\mathbf{h}^{l-1}) = q(\mathbf{h}^l|\mathbf{h}^{l-1})q(\mathbf{h}^{l+1:L}|\mathbf{h}^l) \tag{23}$$

$$p(\mathbf{h}^{l:L}) = p(\mathbf{h}^l|\mathbf{h}^{l+1:L})p(\mathbf{h}^{l+1:L}) = p(\mathbf{h}^l|\mathbf{h}^{l+1})p(\mathbf{h}^{l+1:L}) \tag{24}$$

By leveraging the conditional independence in the chain structures of both posterior and prior, the derivation of trees' ELBO becomes easier.

$$\begin{aligned}\log p(\mathbf{x}) &= \log \int p(\mathbf{x}|\mathbf{h})p(\mathbf{h})d\mathbf{h}\\ &= \log \int \frac{q(\mathbf{h}|\mathbf{x})}{q(\mathbf{h}|\mathbf{x})}p(\mathbf{x}|\mathbf{h})p(\mathbf{h})d\mathbf{h}\\ &\geqslant \mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\big[\log p(\mathbf{x}|\mathbf{h}) - \log q(\mathbf{h}|\mathbf{x}) + \log p(\mathbf{h})\big] = \mathcal{L}(x;\theta).\end{aligned}$$

The last step is due to the Jensen inequality. With $\mathbf{h} = \mathbf{h}^{1:L}$,

$$\begin{aligned}&\log p(\mathbf{x}) \geqslant \mathcal{L}(x;\theta)\\ =&\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\big[\log p(\mathbf{x}|\mathbf{h}^{1:L}) - \log q(\mathbf{h}^{1:L}|\mathbf{x}) + \log p(\mathbf{h}^{1:L})\big]\\ =&\underbrace{\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\big[\log p(\mathbf{x}|\mathbf{h}^{1:L})\big]}_{\text{(a) Reconstruction of the data}} - \underbrace{\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\big[\log q(\mathbf{h}^{1:L}|\mathbf{x}) - \log p(\mathbf{h}^{1:L})\big]}_{\mathbf{KL}^{1:L}}\end{aligned} \tag{25}$$

With conditional independence in the hierarchical structure, we have

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

The second term of (25) can be further expanded as

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

Similarly, with conditional independence of the hierarchical latent variables, $p(\mathbf{h}^1|\mathbf{h}^{2:L}) = p(\mathbf{h}^1|\mathbf{h}^2)$. Thus

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

We can further expand the $\mathbf{KL}^{2:L}$ term following similar conditional independent rules regarding the tree structure. At level l, we get

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

With (23) and (24), it is easy to show that

$$\mathbf{KL}^{l:L} = \underbrace{\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\big[\log q(\mathbf{h}^l|\mathbf{h}^{l-1}) - \log p(\mathbf{h}^l|\mathbf{h}^{l+1})\big]}_{\mathbf{KL}^l} + \underbrace{\mathbb{E}_{q(\mathbf{h}^{l:L}|\mathbf{x})}\big[\log q(\mathbf{h}^{l+1:L}|\mathbf{h}^l) - \log p(\mathbf{h}^{l+1:L})\big]}_{\mathbf{KL}^{l+1:L}}. \tag{26}$$

The ELBO (25) can be written as

$$\mathcal{L}(\mathbf{x};\theta) = \mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\big[\log p(\mathbf{x}|\mathbf{h}^{1:L})\big] - \sum_{l=1}^{L-1}\mathbf{KL}^l - \mathbf{KL}^L. \tag{27}$$

When $1 \leqslant l \leqslant L-1$

$$\mathbf{KL}^l = \mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\big[\log q(\mathbf{h}^l|\mathbf{h}^{l-1}) - \log p(\mathbf{h}^l|\mathbf{h}^{l+1})\big]. \tag{28}$$

As discussed in section B, evaluation of the terms in (27) requires samples of both the posterior and the prior in each layer of the tree structure. According to conditional independence, the expectation regarding variational distribution layer l just depends on layer $l-1$. We can simplify the expectation each term of (27) with the default assumption that all latent variables are generated regarding data sample $\mathbf{x}$. Therefore the ELBO (27) can be simplified as

$$\mathcal{L}(\mathbf{x};\theta) = \mathbb{E}_{q(\mathbf{h}^1|\mathbf{x})}\big[\log p(\mathbf{x}|\widehat{\mathbf{h}}^1)\big] - \sum_{l=1}^{L} \mathbf{KL}^l. \tag{29}$$

The $\mathbf{KL}$ term (28) becomes

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

When $l = L$,

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

E.2 Improve ELBO Estimation with Flows

In this paper we follow the approach in [25, 17, 2] using normalizing flows to further improve posterior estimation on a tree VFG model. At each layer, minimizing $\mathbf{KL}$ term is to is to optimize the parameters of the network so that the posterior is closer to the prior. As shown in Figure 11, for layer l, we can take the encoding-decoding procedures (discussed in section B) as transformation of the posterior distribution from layer l to $l+1$, and then transform it back. By counting in the transformation difference [25, 17, 2], the $\mathbf{KL}$ at layer l becomes

$$\begin{aligned}\mathbf{KL}^l =& \mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\left[\log q(\mathbf{h}^l|\mathbf{h}^{l-1}) + \log\left|\det\frac{\partial \mathbf{h}^l}{\partial \mathbf{h}^{l+1}}\right| + \log\left|\det\frac{\partial \widehat{\mathbf{h}}^{l+1}}{\partial \widehat{\mathbf{h}}^{l}}\right| - \log p(\mathbf{h}^l|\widehat{\mathbf{h}}^{l+1})\right] \\ \simeq& \frac{1}{M}\sum_{m=1}^{M}\left[\log q(\mathbf{h}^l_m|\mathbf{h}^{l-1}) + \log\left|\det\frac{\partial \mathbf{h}^l_m}{\partial \mathbf{h}^{l+1}_m}\right| + \log\left|\det\frac{\partial \widehat{\mathbf{h}}^{l+1}_m}{\partial \widehat{\mathbf{h}}^{l}_m}\right| - \log p(\mathbf{h}^l_m|\widehat{\mathbf{h}}^{l+1}_m)\right].\end{aligned}$$

E.3 ELBO of DAG Models

Note that if we reverse the edge directions in a DAG, the resulting graph is still a DAG graph. The nodes can be listed in a topological order regarding the DAG structure as shown in Figure 14.

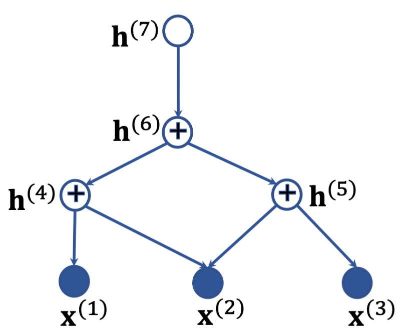


Figure 14: A DAG with inverse topology order $\{$ {1,2,3}, {4,5}, {6}, {7} $\}$, and they correspond to layers 0 to 3.

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. We denote by $\mathbf{h}$ the vector of latent variables, then following (25) we develop the ELBO as

$$\begin{aligned}\log p(\mathbf{x}) \geqslant \mathcal{L}(x;\theta) =& \mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\log\frac{p(\mathbf{x},\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\right] \\ =& \underbrace{\mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\Big[\log p(\mathbf{x}|\mathbf{h})\Big]}_{\text{Reconstruction of the data}} - \underbrace{\mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\Big[\log q(\mathbf{h}|\mathbf{x}) - \log p(\mathbf{h})\Big]}_{\mathbf{KL}}.\end{aligned} \tag{30}$$

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

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

Note that $ch(l)$ may include nodes from layers lower than $l-1$, and $pa(l)$ may include nodes from layers higher than l. Some nodes in l may not have parent. Based on conditional independence with the topology order of a DAG, we have

$$q(\mathbf{h}^{l:L}|\mathbf{h}^{1:l-1}) = q(\mathbf{h}^{l}|\mathbf{h}^{1:l-1})q(\mathbf{h}^{l+1:L}|\mathbf{h}^{l}) = q(\mathbf{h}^{l}|\mathbf{h}^{1:l-1})q(\mathbf{h}^{l+1:L}|\mathbf{h}^{1:l}) \tag{31}$$

$$p(\mathbf{h}^{l:L}) = p(\mathbf{h}^{l}|\mathbf{h}^{l+1:L})p(\mathbf{h}^{l+1:L}) \tag{32}$$

Following (26) and with (31-32), we have

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

Furthermore,

$$q(\mathbf{h}^{l}|\mathbf{h}^{1:l-1}) = q(\mathbf{h}^{l}|\mathbf{h}^{ch(l)}), \qquad p(\mathbf{h}^{l}|\mathbf{h}^{l+1:L}) = p(\mathbf{h}^{l}|\mathbf{h}^{pa(l)}).$$

Hence,

$$\mathbf{KL}^{l:L} = \underbrace{\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\big[\log q(\mathbf{h}^{l}|\mathbf{h}^{ch(l)}) - \log p(\mathbf{h}^{l}|\mathbf{h}^{pa(l)})\big]}_{\mathbf{KL}^{l}} + \mathbf{KL}^{l+1:L} \tag{33}$$

For nodes in layer l,

$$\mathbf{KL}^{l} = \sum_{i\in l} \underbrace{\mathbb{E}_{q(\mathbf{h}^{1:L}|\mathbf{x})}\big[\log q(\mathbf{h}^{(i)}|\mathbf{h}^{ch(i)}) - \log p(\mathbf{h}^{(i)}|\mathbf{h}^{pa(i)})\big]}_{\mathbf{KL}^{(i)}}.$$

Recurrently applying (33) to (30) yields

$$\mathcal{L}(\mathbf{x};\theta) = \mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\big[\log p(\mathbf{x}|\mathbf{h})\big] - \sum_{i\in\mathcal{V}\setminus\mathcal{R}_\mathcal{G}} \mathbf{KL}^{(i)} - \sum_{i\in\mathcal{R}_\mathcal{G}} \mathbf{KL}\big(q(\mathbf{h}^{(i)}|\mathbf{h}^{ch(i)})||p(\mathbf{h}^{(i)})\big).$$

For node i,

$$\mathbf{KL}^{(i)} = \mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\big[\log q(\mathbf{h}^{(i)}|\mathbf{h}^{ch(i)}) - \log p(\mathbf{h}^{(i)}|\mathbf{h}^{pa(i)})\big].$$

F Theoretical Justifications for Latent Representation Learning

The proposed Variational Flow Graphical models provide approaches to integrate multi-modal (multiple natures of data) or multi-source (collected from various sources) data. With invertible flow functions, we analyze the identifiability [15, 30] of the VFG in this section. We assume that each input 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 [8] 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

$$p_{\mathbf{h}^{(t)}}(\mathbf{h}^{(t)}|\mathbf{u}) = \Pi_{i=1}^{d}\frac{Q_i(h^{(t,i)})}{Z_i(\mathbf{u})}\exp\Bigg[\sum_{j=1}^{m} T_{i,j}(h^{(t,i)})\lambda_{i,j}(\mathbf{u})\Bigg], \tag{34}$$

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:

$$\mathbf{x}^{(t)} = \mathbf{f}_t^{-1}(\mathbf{h}^{(t)}, \epsilon), \tag{35}$$

where ϵ is some additional random noise in the generation of $\mathbf{x}^{(t)}$. Let $\mathbf{T} = [\mathbf{T}_1, ..., \mathbf{T}_d]$, and $\lambda = [\lambda_1, ..., \lambda_d]$. We define the domain of the inverse flow $\mathbf{f}_t^{-1}$ as $\mathcal{H} = \mathcal{H}_1 \times ... \times \mathcal{H}_d$. 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. Let $\mathbf{z}^{(t)}$ be one sample's latent variable recovered by the algorithm regarding $\mathbf{h}^{(t)}$. 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 (34).

Theorem 1. *Assume that the observed data is distributed according to the model given by (34) and (35). 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 $h \in \mathcal{H}_i$, $1 \leqslant i \leqslant d$ and $1 \leqslant j \leqslant m$.

(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 $\mathbf{T}(\mathbf{h}^{(t)}) = \mathbf{A}\widehat{\mathbf{T}}(\mathbf{z}^{(t)}) + \mathbf{c}$. Here $\mathbf{A}$ is a $dm \times dm$ 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 infinite data. By expanding the probability density functions with the correct change of variable, we have

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

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

$$\log p_{\mathbf{h}^{(t)}}(\mathbf{h}^{(t)}|\mathbf{u}^{(v)}) - \log p_{\mathbf{h}^{(t)}}(\mathbf{h}^{(t)}|\mathbf{u}^{(0)}) = \log p_{\mathbf{z}^{(t)}}(\mathbf{z}^{(t)}|\mathbf{u}^{(v)}) - \log p_{\mathbf{z}^{(t)}}(\mathbf{z}^{(t)}|\mathbf{u}^{(0)}). \tag{36}$$

Both conditional distributions in equation 36 belong to the exponential family. Eq. (36) thus reads

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

Here the base measures Q_is are canceled out. Let $\bar{\lambda}(\mathbf{u}) = \lambda(\mathbf{u}) - \lambda(\mathbf{u}^{(0)})$. The above equation can be expressed, with inner products, as follows

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

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

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

Here $b_v = \sum_{i=1}^{d} \log \frac{\widehat{Z}_i(\mathbf{u}^{(0)})Z_i(\mathbf{u}^{(v)})}{\widehat{Z}_i(\mathbf{u}^{(v)})Z_i(\mathbf{u}^{(0)})}$. We can multiply $\mathbf{L}^\top$'s inverse with both sized of the equation,

$$\mathbf{T}(\mathbf{h}^{(t)}) = \mathbf{A}\widehat{\mathbf{T}}(\mathbf{z}^{(t)}) + \mathbf{c}. \tag{37}$$

Here $\mathbf{A} = \mathbf{L}^{-1\top}\widehat{\mathbf{L}}^\top$, and $\mathbf{c} = \mathbf{L}^{-1\top}\mathbf{b}$. By Lemma 1 from [15], 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 linearly independent in $\mathbb{R}^m$, for all $1 \leqslant i \leqslant d$. Define m vectors $\mathbf{h}_v^{(t)} = [h_v^{(t),1}, ..., h_v^{(t),d}]$ from points given by this lemma. We obtain the following Jacobian matrix

$$\mathbf{Q} = \left[\mathbf{J}_{\mathbf{T}}(\mathbf{h}_1^{(t)}), ..., \mathbf{J}_{\mathbf{T}}(\mathbf{h}_m^{(t)})\right],$$

where each entry is the Jacobian of size $dm \times d$ from the derivative of Eq. (37) regarding the m vectors $\{\mathbf{h}_j^{(t)}\}_{j=1}^m$. 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 Jacobian of $\widehat{\mathbf{T}}(\mathbf{g}^{-1} \circ \mathbf{f}_t^{-1}(\mathbf{h}^{(t)}))$ 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. $\square$

According to Theorem 1, the proposed model not only can identify global latent factors, but also identify the latent factors for each section with enough auxiliary information. VFG provides a potential approach to learn the latent hierarchical structures from datasets.