Supplemental File for ‘Learning Latent Structural Relations with Message Passing Prior’

In the main submission file, we first propose a message-passing prior model to learn interactions among data components. The prior is then extended to a bi-level decomposable variational auto-encoders (VAEs) that can learn disentangled latent structural representations from input data. The auto-encoder for the second level or layer is parameterized with an aggregation model to perform relational inference. It also serves as the structural prior for the first layer’s global latent variable distribution.

In this paper, message passing denotes the encoding-decoding procedure through flows as shown in Figure 1, and it is used to aggregate hierarchical information and to compute the $\mathbf{KL}$ terms in the ELBO. Moreover, the proposed prior is significantly different from the sequential prior (LSTM) in Genesis [8]. As illustrated in Section 4.1 and Figure 10, the proposed aggregation prior overcomes the drawbacks of sequential dependency in Genesis through message passing and the local-global latent variable decomposition.

According to our theoretical analysis (Section 3.3), the disentangling representation in our model relies on the segmentation of different components to provide implicit supervision. The theory in Section 3.3 extends the results in nonlinear ICA [24, 21], and our results show that as long as some components can be constantly segmented by the model, the latent variables can be identified through the aggregation prior.

This supplementary file provides additional empirical results, theoretical proofs and details of the implementation.

A. Additional Experimental Results and Analysis

This section provides additional experimental results. All the experiments are conducted on NVIDIA Tesla V100S-PCI and TITAN X (Pascal) GPUs.

A.1. Additional Results on Tetrominoes Dataset

Methods	ARI ↑	MSC ↑
MONet	0.552	0.606
MPP$^{\mathrm{M}}$ (Ours)	**0.587**	**0.726**

Table 5. Segmentation results for different methods on Tetrominoes Dataset.

Table 5 gives the segmentation results for different methods on the testing set of Tetrominoes dataset. In addition to ARI score, we also include Mean Segmentation Covering (MSC) [12, 8] in the results, and both ARI and MSC are computed based on foreground components. According to Table 5, three methods achieve close ARI scores. Moreover, our method MPP^{M} achieves improved results for both metrics compared against MONet and Genesis. The proposed message passing prior helps the model to identify different components or objects in the dataset.

A.2. Visualization on CelebA Dataset

To further understand how our model can capture each data component, we conduct a visual analysis on CelebA Dataset. We randomly sample 8 images and show their reconstruction and segmentation results using our proposed method MPP^{G} in Figure 8-left and Genesis in Figure 8-right. We observe that both results show that the segmentation heavily relies on object colors and the components can provide some semantic information about the images. Nevertheless, we show the advantages of MPP^{G} under generative mode in Figure 9, where we plot the generated images from the decoders of both models. The images in the figures visually show that MPP^{G} can generate better images and components. For example, the component $k = 1$ better captures facial characteristics and the component $k = 5$ focuses more on hair styles.

We admit that our model only marginally improves the visualization results on natural images. However, the joint task of object segmentation and representation learning from multiple-object natural images through an unsupervised approach is still challenging. The additional priors and constraints for object segmentation could limit the model’s power on image reconstruction and generation even in comparison with vanilla generative models. We hope the proposed message passing prior could provide some inspiration in this direction.

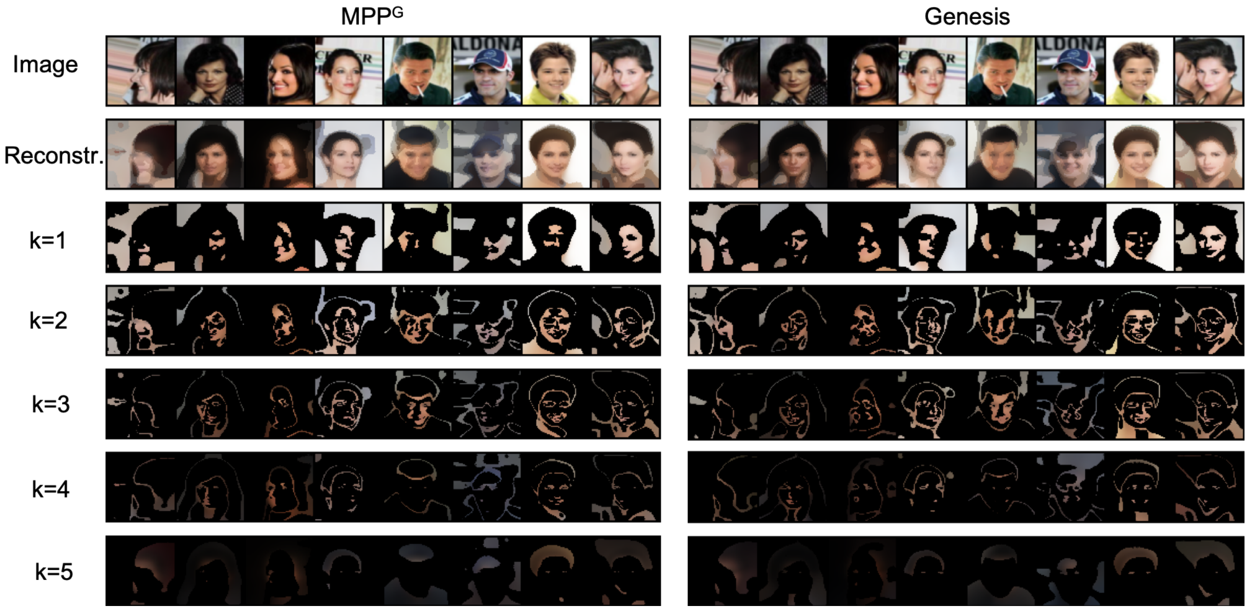

Figure 8. Input images, the reconstructions, and components from MPP^G (left) and Genesis (right). The top row is the original input images from the testing dataset, and the second row gives the reconstruction images from both methods. k is the index of component k in the lower rows. Best viewed in color.

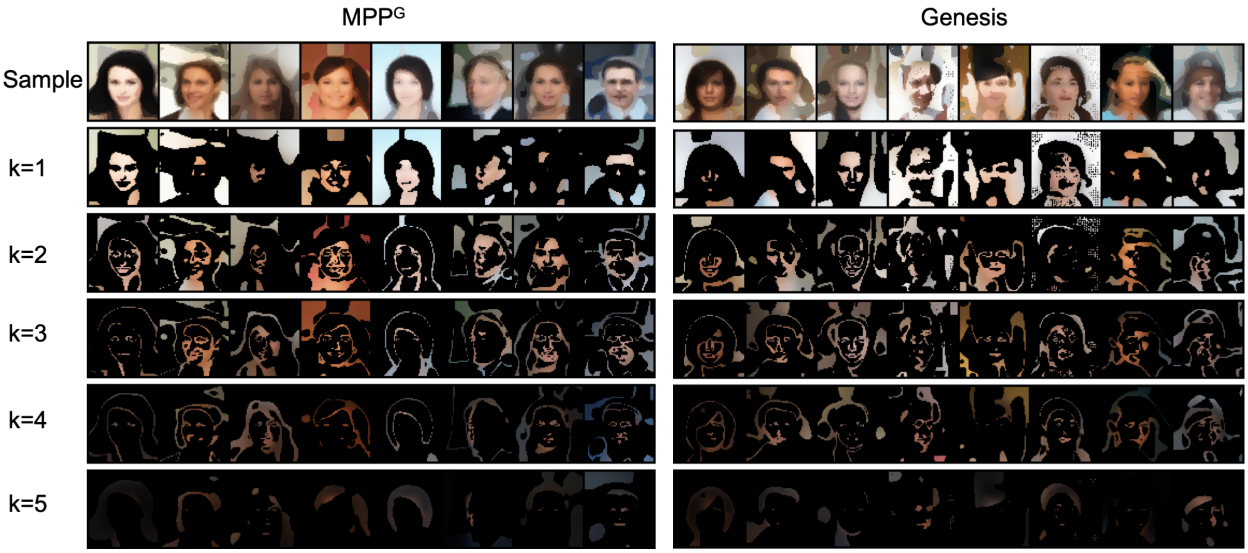

Figure 9. Randomly generated samples and components from MPP^G (left) and Genesis (right). Latent variables are randomly sampled and then the corresponding images are generated from both models respectively.

B. Evidence Lower Bound (ELBO) of Bi-Level Latent Model

To derive the ELBO in Eq. (5), we start from a bi-level variational auto-encoder (VAE) with simplified notations, and then we extend the derivation to proposed models. We use $\mathbf{h}^l$, $l \in \{1, 2\}$ to represent the latent variable in layer l. Let $\mathbf{h} = \mathbf{h}^{1,2} = \{\mathbf{h}^1, \mathbf{h}^2\}$, we have

$$
\begin{aligned}
\log p(\mathbf{x}) &= \mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\mathbf{h})}{p(\mathbf{h}|\mathbf{x})}\right] = \mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\frac{q(\mathbf{x}|\mathbf{h})}{p(\mathbf{h}|\mathbf{x})}\right] \\
&= \mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\right] + \overbrace{\mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\log \frac{q(\mathbf{h}|\mathbf{x})}{p(\mathbf{h}|\mathbf{x})}\right]}^{\geq 0} \\
&\geq \mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\right] = \mathbb{E}_{q(\mathbf{h}^{1,2}|\mathbf{x})}\left[\log \frac{p(\mathbf{x}|\mathbf{h}^{1,2})p(\mathbf{h}^{1,2})}{q(\mathbf{h}^{1,2}|\mathbf{x})}\right] \\
&= \overbrace{\mathbb{E}_{q(\mathbf{h}^{1,2}|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{h}^{1,2})\right]}^{\text{(a) Reconstruction}} + \overbrace{\mathbb{E}_{q(\mathbf{h}^{1,2}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^{1,2})}{q(\mathbf{h}^{1,2}|\mathbf{x})}\right]}^{(b)-\mathbf{KL}}.
\end{aligned}
$$

The first term is data reconstruction. We can extend the second term as follows.

$$
\begin{aligned}
&\mathbb{E}_{q(\mathbf{h}^{1,2}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^{1,2})}{q(\mathbf{h}^{1,2}|\mathbf{x})}\right] \\
=&\mathbb{E}_{q(\mathbf{h}^{1,2}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^1|\mathbf{h}^2)p(\mathbf{h}^2)}{q(\mathbf{h}^1|\mathbf{x})q(\mathbf{h}^2|\mathbf{h}^1)}\right] \\
=&\overbrace{\mathbb{E}_{q(\mathbf{h}^{1,2}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^1|\mathbf{h}^2)p(\mathbf{h}^2)}{q(\mathbf{h}^2|\mathbf{h}^1)}\right]}^{\text{(c)}} + \overbrace{\mathbb{E}_{q(\mathbf{h}^{1,2}|\mathbf{x})}\left[\log \frac{1}{q(\mathbf{h}^1|\mathbf{x})}\right]}^{\text{(d)}}
\end{aligned}
$$

Here the first term

$$
\begin{aligned}
\text{(c) } =&\mathbb{E}_{q(\mathbf{h}^{1,2}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^1|\mathbf{h}^2)p(\mathbf{h}^2)}{q(\mathbf{h}^2|\mathbf{h}^1)}\right] \\
=&\mathbb{E}_{q(\mathbf{h}^{1,2}|\mathbf{x})}\left[\log p(\mathbf{h}^1|\mathbf{h}^2)\right] + \overbrace{\mathbb{E}_{q(\mathbf{h}^{1,2}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^2)}{q(\mathbf{h}^2|\mathbf{h}^1)}\right]}^{\text{(e)}}.
\end{aligned}
$$

In the second term in the above equation, the posterior can be factorized as

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

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

$$
\text{(e)} = \mathbb{E}_{q(\mathbf{h}^{1,2}|\mathbf{x})}\left[\log \frac{p(\mathbf{h}^2)}{q(\mathbf{h}^2|\mathbf{h}^1,\mathbf{x})}\right] = -\mathbf{KL}\big(q(\mathbf{h}^2|\mathbf{h}^1,\mathbf{x})||p(\mathbf{h}^2)\big).
$$

In term (d), as the expectation is only regarding $\mathbf{h}^1$

$$
\text{(d)} = \mathbb{E}_{q(\mathbf{h}^{1,2}|\mathbf{x})}\left[\log \frac{1}{q(\mathbf{h}^1|\mathbf{x})}\right] = \mathbb{E}_{q(\mathbf{h}^1|\mathbf{x})}\left[\log \frac{1}{q(\mathbf{h}^1|\mathbf{x})}\right] = \mathbf{H}(\mathbf{h}^1|\mathbf{x}).
$$

where $H(.)$ is the entropy function. Thus, the ELBO can be written as

$$
\begin{aligned}
\log p(\mathbf{x}) \geq \mathcal{L}_{\text{ELBO}}(\mathbf{x}) =&\mathbb{E}_{q(\mathbf{h}^{1,2}|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{h}^{1,2})\right] + \mathbf{H}(\mathbf{h}^1|\mathbf{x}) \\
&+ \mathbb{E}_{q(\mathbf{h}^{1,2}|\mathbf{x})}\left[\log p(\mathbf{h}^1|\mathbf{h}^2)\right] - \mathbf{KL}\big(q(\mathbf{h}^2|\mathbf{h}^1,\mathbf{x})||p(\mathbf{h}^2)\big).
\end{aligned}
\tag{9}
$$

Due to the hierarchical bi-level structure, sampling the posterior distribution requires to follow the conditional order of variables, i.e., $q(\mathbf{h}^1|\mathbf{x})$ first then $q(\mathbf{h}^2|\mathbf{h}^1)$. Similar for the prior, $p(\mathbf{h}^2)$ first, then $p(\mathbf{h}^1|\mathbf{h}^2)$. Evaluation of the ELBO requires level-wise samples from the posterior, and the prior. We simplify the expression of ELBO (9) by omitting the sampling orders,

$$\begin{aligned}\mathcal{L}_{\text{ELBO}}(\mathbf{x}) =& \mathbb{E}_{q(\mathbf{h}^1|\mathbf{x})}\Big[\log p(\mathbf{x}|\mathbf{h}^1)\Big] + \mathbf{H}(\mathbf{h}^1|\mathbf{x}) + \mathbb{E}_{q(\mathbf{h}^1|\mathbf{x})}\Big[\log p(\mathbf{h}^1|\mathbf{h}^2)\Big] \\ & - \mathbf{KL}\big(q(\mathbf{h}^2|\mathbf{h}^1)||p(\mathbf{h}^2)\big). \end{aligned} \tag{10}$$

For the bi-level model discussed in section 4, we have level 1 latent variable $\mathbf{h}^1 = \{\mathbf{z}^c, \mathbf{z}^g, \mathbf{z}^m\}$, and level 2 latent variable $\mathbf{h}^2 = \mathbf{z}^0$. As discussed in section 4.1, the second level latent variable $\mathbf{z}^0$ controls global latent variable $\mathbf{z}^g$, with (10)

$$\begin{aligned}\mathcal{L}_{\text{ELBO}}(\mathbf{x}) =& \mathbb{E}_{q(\mathbf{z}^c,\mathbf{z}^g,\mathbf{z}^m|\mathbf{x})}\big[\log p_\theta(\mathbf{x}|\mathbf{z}^c,\mathbf{z}^g,\mathbf{z}^m)\big] - \mathbf{KL}(q(\mathbf{z}^c,\mathbf{z}^m|\mathbf{x})\|p(\mathbf{z}^c,\mathbf{z}^m)) \\ & + \mathbf{H}(\mathbf{z}^g|\mathbf{x}) + \mathbb{E}_{q(\mathbf{z}^g|\mathbf{x})}\big[\log p(\mathbf{z}^g|\mathbf{z}^0)\big] - \mathbf{KL}(q(\mathbf{z}^0|\mathbf{z}^g)\|p(\mathbf{z}^0)).\end{aligned}$$

The entropy term and fourth term are on the global latent variables, and they can be further integrated as

$$\begin{aligned}& \mathbf{H}(\mathbf{z}^g|\mathbf{x}) + \mathbb{E}_{q(\mathbf{z}^g,\mathbf{z}^0|\mathbf{x})}\big[\log p(\mathbf{z}^g|\mathbf{z}^0)\big] \\ =& -\mathbb{E}_{q(\mathbf{z}^g|\mathbf{x})}\big[\log q(\mathbf{z}^g|\mathbf{x}) - \log p(\mathbf{z}^g|\mathbf{z}^0)\big] \\ =& -\mathbf{KL}\big(q(\mathbf{z}^g|\mathbf{x})\|p(\mathbf{z}^g|\mathbf{z}^0)\big).\end{aligned}$$

It is the $\mathbf{KL}$ divergence between the posterior of $\mathbf{z}^g$, $q(\mathbf{z}^g|\mathbf{x})$, and its prior distribution $p(\mathbf{z}^g|\mathbf{z}^0)$.

C. Message Passing Prior for Genesis

$\mathbf{x}_k$	the kth component
$\widehat{\mathbf{x}}_k$	reconstruction of the kth component with decoder d
$\mathbf{z}^c_k$	local latent variable for the kth component
f_k	flow function for the kth component
$\mathbf{z}^g_k$	global latent variable for the kth component
$\widehat{\mathbf{z}}^g_k$	reconstruction of $\mathbf{z}^g_k$ with flow function f_k
$\mathbf{z}^0$	global latent variable
$\mathbf{m}_k$	mask for the kth component with attention network a
$\widehat{\mathbf{m}}_k$	reconstruction of mask k with decoder d
$\mathbf{s}_k$	scope or attention net input for the kth component
a	attention network
e	encoder network
d	decoder network
r	recurrent network for latent variables of masks
f	$f = \{f_1, f_2, ..., f_K\}$, second level encoder
f^{-1}	$f^{-1} = \{f_1^{-1}, f_2^{-1}, ..., f_K^{-1}\}$, second level decoder

Table 6. Notations for MPP^G and MPP^M.

We list the notations of MPP^G in Table 6. The neural network structure of MPP^G is given by Figure 3. Figure 10 gives the graphical illustration of the generative procedure of the variables for both MPP^G and Genesis [8].

The posterior $q_r(\mathbf{z}^m|\mathbf{x})$ is modeled with a RNN (blue blocks in Figure 3), r. The posteriors of $\mathbf{z}^c$ and $\mathbf{z}^g$, $q_e(\mathbf{z}^c|\mathbf{z}^m, \mathbf{x})$ and $q_e(\mathbf{z}^g|\mathbf{z}^m, \mathbf{x})$, are parameterized with the encoder e network. As shown in the figure, both of them also dependents on $\mathbf{z}^m$. For the bi-level auto-encoder, $(\mathbf{x}, \mathbf{m}_k)$ is the first layer's input, and $(\mathbf{z}^c_k, \mathbf{z}^g_k)$ is the first layer's latent variable. Meanwhile, $\mathbf{z}^g_k$ is also the second layer's input, and $\mathbf{z}^0$ is the second layer's latent variable. $\widehat{\mathbf{x}}_k$ and $\widehat{\mathbf{z}}^g_k$ are the reconstructions regarding the first level and second level inputs, respectively.

As shown in the graphical representation of MPP^G and Genesis (Figure 3), with $\mathbf{z}^0$ MPP^G can aggregate the information from all components simultaneously. Genesis captures the sequential dependence among the components by leveraging the

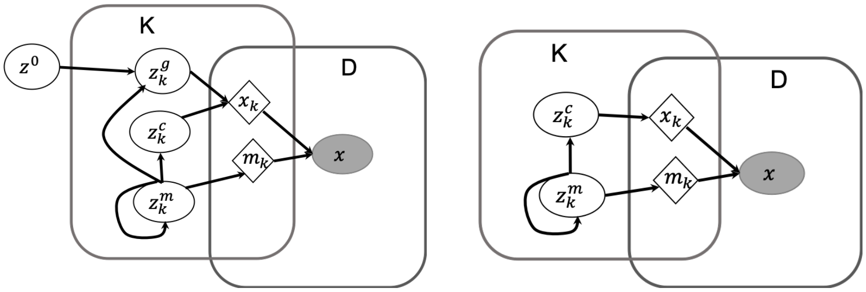

Figure 10. Graphical representation of variables for MPP^G (left) and Genesis (right).

latent variable of masks ($\mathbf{z}^m$s). The message passing prior network structure of MPP^G is given by Figure 2-Left. The second level auto-encoder is parameterized with the proposed message passing prior model $f = \{f_1, f_2, ..., f_K\}$, i.e.,

$$p_{f^{-1}}(\mathbf{z}^g|\mathbf{z}^0) = \mathbf{\Pi}_{k=1}^K p_{f_k^{-1}}(\mathbf{z}_k^g|\mathbf{z}^0),$$

and the posterior of $\mathbf{z}^0$, $q_f(\mathbf{z}^0|\mathbf{z}^g)$, is the encoding process of the model f.

D. Message Passing Prior for MONet

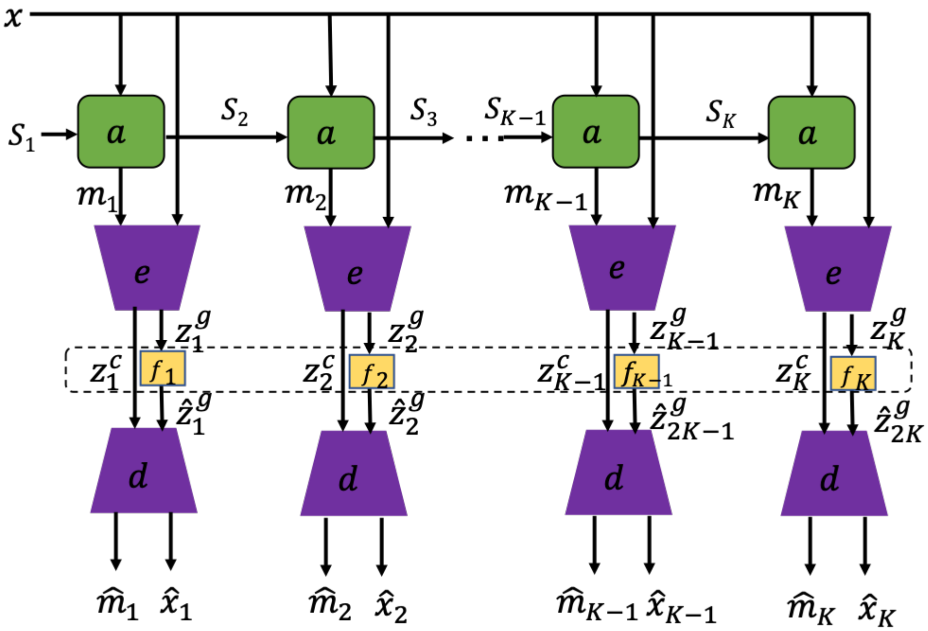

Figure 11. MONet with message passing prior. a is the attention network, e is the encoder, d is the decoder, and f_k is the flow inference network for component k. $(\mathbf{x}, \mathbf{m}_k)$ and $(\mathbf{z}_k^c, \mathbf{z}_k^g)$ are the kth component encoder's input and output; $(\mathbf{z}_k^c, \widehat{\mathbf{z}}_k^g)$ and $(\widehat{\mathbf{x}}_k, \widehat{\mathbf{m}}_k)$ are the input and output of the decoder. The input scope for kth component is defined by $\mathbf{s}_k = \mathbf{s}_{k-1} \circ (1 - \mathbf{m}_{k-1})$.

The neural network structure of MPP^M is given by Figure 11, and the corresponding graphical model is shown in Figure 12. The notations of MPP^M is listed in Table 6 as well. As shown in the Figure 11, for the bi-level auto-encoder, $(\mathbf{x}, \mathbf{m}_k)$ is the first layer's input, and $(\mathbf{z}_k^c, \mathbf{z}_k^g)$ is the first layer's latent variable. Meanwhile, $\mathbf{z}_k^g$ is also the second layer's input, and $\mathbf{z}^0$ is the second layer's latent variable. $(\widehat{\mathbf{x}}, \widehat{\mathbf{m}}_k)$ and $\widehat{\mathbf{z}}_k^g$ are the reconstructions regarding the first level and second level inputs, respectively. The second layer posterior distribution for $\mathbf{z}^0$ is $q_f(\mathbf{z}^0|\mathbf{z}_1^g\mathbf{z}_2^g...\mathbf{z}_k^g)$. As can be seen from Figure 3, the attention network generates mask $\mathbf{m}_k$ for component k. The input for the encoder e is $(\mathbf{x}_k, \mathbf{m}_k)$, and the corresponding reconstruction generated from the decoder d is $(\widehat{\mathbf{x}}_k, \widehat{\mathbf{m}}_k)$. $\mathbf{z}_k\mathbf{z}_k^g$ is the overall latent variable for component k.

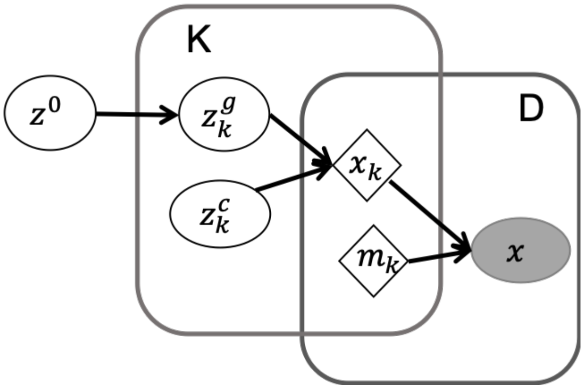

Figure 12. Graphical representation of variables in MPP^M. For component k, the corresponding latent variable is $\mathbf{z}_k^c\mathbf{z}_k^g$. $\mathbf{z}^0$ is the shared latent variable of K components. The pixel-wise data value and masks are denoted as $\mathbf{x}$ and $\bar{\mathbf{m}}$, respectively. D is the dimensionality of the input data samples.

Different from Genesis and MPP^G, MONet employs an UNet [34] as the attention network a for component segmentation [2]. Let $\mathcal{L}_k(\mathbf{x}, \mathbf{m}_k; a, e, d, f)$ be the ELBO regarding component k in the bi-level MPP^M. By omitting the latent variable for masks, and taking masks as part of the reconstruction of the decoder, equation (5) becomes

$$\begin{aligned}&\log p(\mathbf{x}, \mathbf{m}_k) \geq \mathcal{L}_k(\mathbf{x}, \mathbf{m}_k; a, e, d, f)\\ =&\mathbb{E}_{q_e(\mathbf{z}_k^c, \mathbf{z}_k^g | \mathbf{x}, \mathbf{m}_k)}\Big[\log p_d(\mathbf{x}, \mathbf{m}_k | \mathbf{z}_k^c, \mathbf{z}_k^g)\Big] - \mathbf{KL}\big(q_e(\mathbf{z}_k^c | \mathbf{x}, \mathbf{m}_k) || p(\mathbf{z}_k^c)\big) +\\ &\mathbf{H}(\mathbf{z}_k^g | \mathbf{x}, \mathbf{m}_k) + \mathbb{E}_{q_f(\mathbf{z}^0 | \mathbf{z}_k^g)}\Big[\log p_{f_k}(\mathbf{z}_k^g | \mathbf{z}^0)\Big] - \mathbf{KL}\big(q_f(\mathbf{z}^0 | \mathbf{z}_k^g) || p(\mathbf{z}^0)\big). \qquad (11)\end{aligned}$$

Here q_e is the posterior distribution for the first layer latent variable parameterized by the encoder e. p_d is the distribution for $\mathbf{x}$ and $\mathbf{m}_k$ parameterized with the decoder d. f_k is the k's flow-based model, and $f = \{f_1, ..., f_K\}$. The conditional distribution $q_f(\mathbf{z}^0|\mathbf{z}_k^g)$ captures the relationship between $\mathbf{z}_k^g$ and the other $\mathbf{z}_j^g$s, $j \neq k$. All the latent variables are assumed to follow Gaussian distribution. The variance value of posterior $q_f(\mathbf{z}^0|\mathbf{z}_1^g\mathbf{z}_2^g...\mathbf{z}_k^g)$ is set to a fixed value 1. Here a, e, d, f_k are the attention, encoder, decoder, and flow function for component k, respectively. The reconstruction term regarding $\mathbf{x}$ and $\mathbf{m}_k$ in the above ELBO (11) is $\Phi_k = \mathbb{E}_{q_e(\mathbf{z}_k^c, \mathbf{z}_k^g | \mathbf{x}, \mathbf{m}_k)}\Big[\log p_d(\mathbf{x}, \mathbf{m}_k | \mathbf{z}_k^c, \mathbf{z}_k^g)\Big]$. The reconstruction loss for the data sample $\mathbf{x}$ is weighted by the attention masks ($\mathbf{m}_k$s). The entries of the masks ($\mathbf{m}_k$s) follow Bernoulli distribution parameterized with Sigmoid functions. The reconstruction loss regarding the masks (the second term in Φ_k (12)) is tractable based on this assumption we can use the **KL** divergence between two neural network (a and d) outputs. The reconstruction term for both $\mathbf{x}$ and $\mathbf{m}_k$ is rewritten as

$$\Phi_k = \mathbb{E}_{q_e(\mathbf{z}_k^c, \mathbf{z}_k^g | \mathbf{x}, \mathbf{m}_k)}\Big[\mathbf{m}_k \log p_d(\mathbf{x} | \mathbf{z}_k^c, \mathbf{z}_k^g)\Big] - \mathbf{KL}(q_a(\mathbf{m}_k | \mathbf{x}) || p_d(\widehat{\mathbf{m}}_k | \mathbf{z}_k^c, \mathbf{z}_k^g)). \qquad (12)$$

The regularization terms for the first layer's latent variable are

$$\Psi_k = -\mathbf{KL}\big(q_e(\mathbf{z}_k^c | \mathbf{x}, \mathbf{m}_k) || p(\mathbf{z}_k^c)\big) + \mathbf{H}(\mathbf{z}_k^g | \mathbf{x}, \mathbf{m}_k). \qquad (13)$$

All the latent variables are assumed to follow Gaussian distributions. Both the **KL** and entropy terms are easy to compute with the reparameterization trick used in VAEs [26]. The objective across all components we need to maximize is given by

$$\mathcal{L}(\mathbf{x}; a, e, d, f) = \sum_{k=1}^{K} \mathcal{L}_k(\mathbf{x}, \mathbf{m}_k; a, e, d, f).$$

For component k, the terms in the ELBO (11) regarding the second layer of latent variable $\mathbf{z}^0$ is

$$\mathcal{L}_{f_k} = \mathbb{E}_{q_f(\mathbf{z}^0 | \mathbf{z}_k^g)}\Big[\log p_{f_k}(\mathbf{z}_k^g | \mathbf{z}^0)\Big] - \mathbf{KL}\big(q_f(\mathbf{z}^0 | \mathbf{z}_k^g) || p(\mathbf{z}^0)\big). \qquad (14)$$

We can see that the computation of $\mathcal{L}_{f_k}$'s values involves both encoding ($q_f(\mathbf{z}^0|\mathbf{z}^g_k)$) and decoding ($p_{f_k}(\mathbf{z}^g_k|\mathbf{z}^0)$) procedures. The first term of $\mathcal{L}_{f_k}$ is to compute the conditional log-likelihood value of $\mathbf{z}^g_k$ given $\mathbf{z}^0$. The learning of all f_k is the same as the learning of the encoder and decoder discussed in the message prior section. The prior $p(\mathbf{z}^0)$ can be standard normal distributions. Ideally, we hope the global latent variable's value can be inferred from any subset of components. That is $\mathbf{z}^0 = f(\widehat{\mathbf{z}}_0) = f_k(\mathbf{z}^g_k) = f_j(\mathbf{z}^g_j), \forall k \neq j$.

E. Recovery of Relations

In this section, we provide theoretical results and proofs regarding the proposed message passing prior.

E.1. Properties of Message Passing Prior

Given a data sample $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_K]$, we have the following lemma regarding the likelihood value computed with the message passing scheme in Figure 1.

Lemma 1. *The log-density value of* $\mathbf{y}$ *can be approximated by*

$$\log p(\mathbf{y}) \approx \log p(\mathbf{h}) - \frac{1}{2}\log\big(\det\big(\mathbf{J}_{\widehat{\mathbf{y}}}(\mathbf{h})^\top \mathbf{J}_{\widehat{\mathbf{y}}}(\mathbf{h})\big)\big). \tag{15}$$

Here $\mathbf{J}_{\widehat{\mathbf{y}}}(\mathbf{h}) = \big[\mathbf{J}^\top_{\widehat{\mathbf{y}}_1}(\mathbf{h}), \mathbf{J}^\top_{\widehat{\mathbf{y}}_2}(\mathbf{h}), ..., \mathbf{J}^\top_{\widehat{\mathbf{y}}_K}(\mathbf{h})\big]^\top$.

Proof. The structure relation between $\mathbf{h}$ and $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_K]$ is given in Figure 1, thus the Jacobian matrix regarding the functions mapping from $\mathbf{h}$ to the reconstruction $\widehat{\mathbf{y}} = [\widehat{\mathbf{y}}_1, ..., \widehat{\mathbf{y}}_K] = [f_1^{-1}(\mathbf{h}_1), ..., f_K^{-1}(\mathbf{h}_K)]$ is defined in $\mathbf{J}_{\widehat{\mathbf{y}}}(\mathbf{h})$. With the training objective (1) and (2), we can have $\widehat{\mathbf{y}} \approx \mathbf{y}$, and $\widehat{\mathbf{h}}_k \approx \mathbf{h}_k$. The change of variable theorem is known in the context of geometric measure theory as the smooth coarea formula [37, 27, 16], that is

$$p(\mathbf{y}) \approx p(\mathbf{h}) \det\big(\mathbf{J}_{\widehat{\mathbf{y}}}(\mathbf{h})^\top \mathbf{J}_{\widehat{\mathbf{y}}}(\mathbf{y})\big)^{-\frac{1}{2}}.$$

from which we obtain the log-likelihood for $\mathbf{y}$. □

We use coupling layers [6] to implement flow functions. According to the theoretical study of [35], coupling layer flows are universal approximators. Feed forward ReLU networks are implemented for the scale and shift function of coupling layers. In this section, we extend the theoretical result of feed forward neural networks [10] to relation recovery. In this paper, $\asymp$ means in the same asymptotic order.

Theorem 1. *Let the assumptions (1-2) hold, and* $|\mathcal{R}| \leq dim(\mathbf{h})$. *Let* $\widehat{g}_{u,i}$ *be the estimator that consists of deep coupling layers with width* $W \asymp n^{\frac{d}{2(\beta+d)}} \log^2 n$, *and depth* $D \asymp \log n$. *With large a enough* n, *with probability at least* $1 - \exp(-n^{\frac{d}{\beta+d}} \log^8 n)$,

a)$\|\widehat{g}_{u,i} - g^*_{u,i}\|^2_{L_2(Y)} \leq C \cdot \big\{ - n^{\frac{\beta}{\beta+d}} \log^8 n + \frac{\log\log n}{n} \big\}$ *and*

b) $\mathbb{E}_n\big[(\widehat{g}_{u,i} - g^*_{u,i})^2\big] \leq C \cdot \big\{ - n^{\frac{\beta}{\beta+d}} \log^8 n + \frac{\log\log n}{n} \big\}$. *Here* $C > 0$ *is a constant independent of* n.

Proof. The ELBO for the message passing prior model is

$$\begin{aligned}
&\log p_{f^{-1}}(\mathbf{y}) \geq \mathcal{L}(\mathbf{y}; f) \\
=&\mathbb{E}_{q_f(\mathbf{h}|\mathbf{y})}\big[\log p_{f^{-1}}(\mathbf{y}|\mathbf{h})\big] - \mathbf{KL}\big(q_f(\mathbf{h}|\mathbf{y})||p(\mathbf{h})\big) \\
=&-\frac{1}{2\sigma^2_{\mathbf{y}}}\big|\big|\mathbf{y} - f^{-1}(\mathbf{h})\big|\big|^2 - \frac{1}{2\sigma^2}\sum_{k=1}^{K}\big|\big|\mathbf{h} - \mathbf{h}_k\big|\big| - \mathbf{KL}\big(q_f(\mathbf{h}|\mathbf{y})||p(\mathbf{h})\big) + C \\
=&-\sum_{k=1}^{K}\bigg\{\frac{1}{2\sigma^2_{\mathbf{y}}}\big|\big|\mathbf{y}_k - f_k^{-1}(\mathbf{h})\big|\big|^2 + \frac{1}{2\sigma^2}\big|\big|\mathbf{h} - f_k(\mathbf{y}_k)\big|\big|^2\bigg\} - \mathbf{KL}\big(q_f(\mathbf{h}|\mathbf{y})||p(\mathbf{h})\big) + C.
\end{aligned}$$

Here $C = -lK\ln(2\pi) - \frac{lK}{2}\ln(\sigma^2_{\mathbf{y}}) - \frac{lK}{2}\ln(\sigma^2)$. The KL term regularize the distributions of all entries of $\mathbf{h}$ to be close to prior $\mathrm{N}(u, \sigma^2)$ individually and thus to be independent with each other. Without loss of generalization, we assume $\sigma_{\mathbf{y}} = \sigma = 1$.

Maximizing the ELBO is equivalent to the following optimization problem,

$$\begin{aligned}
&\min_{f} \mathcal{L}_f = \mathbb{E}_{\mathbf{y}\sim P(\mathbf{y})}\Big[\sum_{k=1}^{K}\{\big|\big|\mathbf{y}_k - f_k^{-1}(\mathbf{h})\big|\big|_2^2 + \big|\big|\mathbf{h} - f_k(\mathbf{y}_k)\big|\big|_2^2\} + \mathbf{KL}\big(q_f(\mathbf{h}|\mathbf{y})||p(\mathbf{h})\big)\Big] \\
&s.t. \quad \frac{1}{K}\sum_{k=1}^{K} f_k(\mathbf{y}_k) = \mathbf{h}
\end{aligned}$$

Inside the expectation, the objective has two parts,

$$\begin{aligned}
\mathcal{F}(\mathbf{y}) =& \sum_{k=1}^{K}\{\big|\big|\mathbf{y}_k - f_k^{-1}(\mathbf{h})\big|\big|_2^2 + \big|\big|\mathbf{h} - f_k(\mathbf{y}_k)\big|\big|_2^2\} + \mathbf{KL}\big(q_f(\mathbf{h}|\mathbf{y})||p(\mathbf{h})\big) \\
=& \mathcal{F}_{-u}(\mathbf{y}) + \mathcal{T}_u(\mathbf{y})
\end{aligned} \tag{16}$$

The first part, $\mathcal{F}_{-u}(\mathbf{y})$, can be taken as to minimize the distance between $f_u(y_u)$ and other $f_j(y_j)$s ($j \neq u$),

$$\begin{aligned}
\mathcal{F}_{-u}(\mathbf{y}) = & \sum_{j:1\leq j\leq K, j\neq u}\{\big|\big|\mathbf{y}_j - f_j^{-1}(\mathbf{h})\big|\big|_2^2 + \big|\big|\mathbf{h} - f_j(\mathbf{y}_j)\big|\big|_2^2\} + \big|\big|\mathbf{h} - f_u(\mathbf{y}_u)\big|\big|_2^2 \\
& + \mathbf{KL}\big(q_f(\mathbf{h}|\mathbf{y})||p(\mathbf{h})\big).
\end{aligned}$$

We assume the flow functions used in the models are L-Lipschitz continuous,

$$\big|\big|\mathbf{h} - f_j(\mathbf{y}_j)\big|\big|_2 = \big|\big|f_j(\mathbf{y}_j) - f_j(f_j^{-1}(\mathbf{h}))\big|\big|_2 \leq L\big|\big|\mathbf{y}_j - f_j^{-1}(\mathbf{h})\big|\big|_2.$$

Therefore, the lower bound of $\mathcal{F}_{-u}(\mathbf{y})$ reads,

$$\begin{aligned}
\mathcal{F}_{-u}(\mathbf{y}) \geq & (\frac{1}{L^2} + 1)\sum_{j:1\leq j\leq K, j\neq u}\big|\big|\mathbf{h} - f_j(\mathbf{y}_j)\big|\big|_2^2 + \big|\big|\mathbf{h} - f_u(\mathbf{y}_u)\big|\big|_2^2 \\
& + \mathbf{KL}\big(q_f(\mathbf{h}|\mathbf{y})||p(\mathbf{h})\big)
\end{aligned} \tag{17}$$

In RHS of (17),

$$\big|\big|\mathbf{h} - f_u(\mathbf{y}_u)\big|\big|_2^2 = \frac{(K-1)^2}{K^2}\Big|\Big|\frac{1}{K-1}\sum_{j:1\leq j\leq K, j\neq u} f_j(\mathbf{y}_j) - f_u(\mathbf{y}_u)\Big|\Big|_2^2. \tag{18}$$

Thus the squared terms in RHS of lower bound (17) indicate minimizing $\mathcal{F}_{-u}(\mathbf{y})$ is to minimize the mutual distances of different $f_j(\mathbf{y}_j)$s, the **KL** is to force the summation to be close to a Gaussian distribution. With the assumption that the total number of different relations is smaller than the flow dimension($|\mathcal{R}| \leq \dim(\mathbf{h})$), then there will be enough capacity for the feed forward neural network to ensure the distance between any pair of $f_j(\mathbf{y}_j)$s smaller than a very small value under high probability [10]. With a very large n, 18 will become a very small number, i.e.,

$$f_u(\mathbf{y}_u) = \frac{1}{K-1}\sum_{j:1\leq j\leq K, j\neq u} f_j(\mathbf{y}_j) + \zeta(n), \tag{19}$$

where $||\zeta(n)||_2$ is a very small value. With (19), we get

$$\mathbf{h} = \frac{1}{K}\sum_{k=1}^{K} f_u(\mathbf{y}_u) = \frac{1}{K-1}\sum_{j:1\leq j\leq K, j\neq u} f_j(\mathbf{y}_j) + \frac{1}{K}\zeta(n).$$

Hence, minimizing $\mathcal{T}_u(\mathbf{y})$ in (16) is to minimize

$$\begin{aligned}
\min_{f} \mathcal{T}_u(\mathbf{y}) =& \min_{f}\big|\big|\mathbf{y}_u - f_u^{-1}(\mathbf{h})\big|\big|_2^2 \\
=& \min_{f}\Big|\Big|\mathbf{y}_u - f_u^{-1}\Big(\frac{1}{K-1}\sum_{j:1\leq j\leq K, j\neq u} f_j(\mathbf{y}_j) + \frac{1}{K}\zeta(n)\Big)\Big|\Big|_2^2.
\end{aligned} \tag{20}$$

Let ith entry of $\mathbf{y}_u$ be a function of some entries in $\mathbf{y}_j, j \neq u$, i.e., $g^*_{u,i}$. According to Theorem 1 of [10], let all the ReLU feed forward functions in f have width $W \asymp n^{\frac{d}{2(\beta+d)}} \log^2 n$, and depth $D \asymp \log n$. Minimizing (16) with large enough data sample number n, with probability at least $1-\exp(-n^{\frac{d}{\beta+d}} \log^8 n)$, the model can achieve a) $\|\widehat{g}_{u,i} - g^*_{u,i}\|^2_{L_2(Y)} \leq C \cdot \big\{ -n^{\frac{\beta}{\beta+d}} \log^8 n + \frac{\log\log n}{n} \big\}$ and b) $\mathbb{E}_n \big[(\widehat{g}_{u,i} - g^*_{u,i})^2\big] \leq C \cdot \big\{ -n^{\frac{\beta}{\beta+d}} \log^8 n + \frac{\log\log n}{n} \big\}$. It concludes the theorem. □

E.2. Identifiability of Latent Representation

Following [24, 15], we define following equivalence relations on Θ.

Definition 1. *Let $\sim$ be the equivalence relation on Θ. We say (4) is identifiable up to $\sim$ if*

$$p(\mathbf{y}_k; \Theta) = p(\mathbf{y}_k; \widehat{\Theta}) \implies \Theta \sim \widehat{\Theta}.$$

The elements of the quotient space $\Theta / \sim$ are called the identifiability classes.

Definition 2. *Let $\sim$ be the binary relation on Θ defined by*

$$(\mathbf{T}, \lambda, f_k^{-1}) \sim (\widehat{\mathbf{T}}, \widehat{\lambda}, g_k) \leftrightarrow \exists \mathbf{A}, \mathbf{c} | \mathbf{T}(f_k(\mathbf{y}_k)) = \mathbf{A}\widehat{\mathbf{T}}(g_k^{-1}(\mathbf{y}_k)) + \mathbf{c}, \forall \mathbf{y}_k \in \mathcal{Y}_k$$

where $\mathbf{A}$ is an $lm \times lm$ matrix and $\mathbf{c}$ is a vector of size lm. If $\mathbf{A}$ is invertible, the above relation is denoted by $\sim_{\mathbf{A}}$.

Theorem 2. *Assume we observe data distributed according to the generative model given by (4) and (5), we further have the following assumptions,*

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

(b) The relations involving component k can be approximately fully recovered and can be represented with $\mathbf{u}_k(\mathbf{y}_{-k})$.

(c) There exist $lm + 1$ distinct conditions $\mathbf{u}_k^{(0)}$, ..., $\mathbf{u}_k^{(lm)}$ from $\mathbf{y}_{-k}$ such that the matrix

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

of size $lm \times lm$ is invertible. Then the model parameters $(\mathbf{T}, \lambda, f_k^{-1})$ are $\sim_{\mathbf{A}}$ identifiable.

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

$$\log p_{\mathbf{T},\lambda}(\mathbf{h}_k|\mathbf{u}_k) + \log \big| \det \mathbf{J}_f(\mathbf{y}_k) \big| = \log p_{\widehat{\mathbf{T}},\widehat{\lambda}}(\mathbf{h}'_k|\mathbf{u}_k) + \log \big| \det \mathbf{J}_{g^{-1}}(\mathbf{y}_k) \big|.$$

Different from [24, 15, 21] that use observed auxiliary variables as conditional variables, we assume the relations with component k can be recovered and use signals from other components as conditional labels. Let $\mathbf{u}_k^{(0)}, ..., \mathbf{u}_k^{(lm)}$ be from conditions (b) and (c). We can subtract this expression for $\mathbf{u}_k^{(0)}$ for some condition $\mathbf{u}_k^{(t)}$. The Jacobian terms will be removed since they do not depend $\mathbf{u}_k$,

$$\log p_{\mathbf{h}_k}(\mathbf{h}_k|\mathbf{u}_k^{(t)}) - \log p_{\mathbf{h}_k}(\mathbf{h}_k|\mathbf{u}_k^{(0)}) = \log p_{\mathbf{h}'_k}(\mathbf{h}'_k|\mathbf{u}_k^{(t)}) - \log p_{\mathbf{h}'_k}(\mathbf{h}'_k|\mathbf{u}_k^{(0)}). \tag{21}$$

Both conditional distributions of $\mathbf{h}_k$ given $\mathbf{u}_k$ belong to exponential family. Eq. (21) can be rewritten as

$$\begin{aligned} &\sum_{i=1}^{l} \left[\log \frac{Z_i(\mathbf{u}_k^{(0)})}{Z_i(\mathbf{u}_k^{(t)})} + \sum_{j=1}^{m} T_{i,j}(\mathbf{h}_k)\big(\lambda_{i,j}(\mathbf{u}_k^{(t)}) - \lambda_{i,j}(\mathbf{u}_k^{(0)})\big) \right] \\ =&\sum_{i=1}^{l} \left[\log \frac{\widehat{Z}_i(\mathbf{u}_k^{(0)})}{\widehat{Z}_i(\mathbf{u}_k^{(t)})} + \sum_{j=1}^{m} \widehat{T}_{i,j}(\mathbf{h}_k)\big(\widehat{\lambda}_{i,j}(\mathbf{u}_k^{(t)}) - \widehat{\lambda}_{i,j}(\mathbf{u}_k^{(0)})\big) \right]. \end{aligned} \tag{22}$$

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

$$\langle \mathbf{T}(\mathbf{h}_k), \bar{\lambda} \rangle + \sum_i \log \frac{Z_i(\mathbf{u}_k^{(0)})}{Z_i(\mathbf{u}_k^{(t)})} = \langle \widehat{\mathbf{T}}(\mathbf{h}'_k), \bar{\widehat{\lambda}} \rangle + \sum_i \log \frac{\widehat{Z}_i(\mathbf{u}_k^{(0)})}{\widehat{Z}_i(\mathbf{u}_k^{(t)})}, \;\; \forall l, 1 \leq l \leq lm.$$

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

$$\mathbf{L}^\top \mathbf{T}(\mathbf{h}_k) = \widehat{\mathbf{L}}^\top \widehat{\mathbf{T}}(\mathbf{h}'_k) + \mathbf{b}.$$

Here $b_t = \sum_i \log \frac{\widehat{Z}_i(\mathbf{u}_k^{(0)}) Z_i(\mathbf{u}_k^{(t)})}{\widehat{Z}_i(\mathbf{u}_k^{(t)}) Z_i(\mathbf{u}_k^{(0)})}$. We can multiply $\mathbf{L}^\top$'s inverse with both sized of the equation,

$$\mathbf{T}(\mathbf{h}_k) = \mathbf{A}\widehat{\mathbf{T}}(\mathbf{h}'_k) + \mathbf{c}. \tag{23}$$

Here $\mathbf{A} = \mathbf{L}^{-1\top}\widehat{\mathbf{L}}^\top$, and $\mathbf{c} = \mathbf{L}^{-1\top}\mathbf{b}$. By a lemma from [24], there exist m distinct values $h_{k,i}^1$ to $h_{k,i}^m$ such that $\left[\frac{dT_i}{dh_{k,i}}(h_{k,i}^1), ..., \frac{dT_i}{dh_{k,i}}(h_{k,i}^m)\right]$ are linear independent in $\mathbb{R}^m$, for all $1 \leq i \leq l$. Define m vectors $\mathbf{h}_k^t = [h_{k,1}^t, ..., h_{k,l}^t]$ from points given by this lemma. We obtain the Jacobian $\mathbf{Q} = [\mathbf{J}_\mathbf{T}(\mathbf{h}_k^1), ..., \mathbf{J}_\mathbf{T}(\mathbf{h}_k^m)]$ with each entry as Jacobian with size $lm \times l$ from the derivative of Eq. (23) regarding these m vectors. Hence $\mathbf{Q}$ is a $lm \times lm$ invertible by the lemma and the fact that each component of $\mathbf{T}$ is univariate. We can construct a corresponding matrix $\widehat{\mathbf{Q}}$ with the Jabocian $\widehat{\mathbf{T}}(g^{-1} \circ f_k^{-1}(\mathbf{h}_k))$ 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. □

F. Network Structures

The encoder, decoder, and attention network of MPP^G follow the implementation of Genesis. Flow functions used in the message passing prior for both models employ coupling layers [6], and the scale and shift function is presented in Table 7.

The network structures for the encoder and decoder of MPP^M are given in Table 8 and Table 9, respectively. The attention network of MPP^M employs one U-net [34] with 5 blocks. The decoder is a spatial broadcast decoder [40] to encourage the VAE to learn spatial features. Each flow function f_k consists of a sequence of three coupling layers. Let $\mathbf{x}$ and $\mathbf{y}$ be the input and output of a coupling layer, the scale $\mathbf{s}(\cdot)$ and shift function $\mathbf{t}(\cdot)$ is define by

$$\begin{aligned} \mathbf{y}_{1:\frac{d}{2}} &= \mathbf{x}_{1:\frac{d}{2}} \\ \mathbf{y}_{\frac{d}{2}+1:d} &= \mathbf{x}_{\frac{d}{2}+1:d} \odot \mathbf{s}(\mathbf{x}_{1:\frac{d}{2}}) + \mathbf{t}(\mathbf{x}_{1:\frac{d}{2}}). \end{aligned}$$

As shown in Table 7, there is no activation function for the $\mathbf{t}$ part in the last layer, but $\mathbf{s}$ has Sigmoid as the activation function in the last layer.

Scale and shift function			
Layer	Number of Output	Batch Normalization	Activation function
Input z	$d/2$	-	-
Fully-Connected	64	N	ReLU
Fully-Connected	64	N	ReLU
Fully-Connected	d	Y	s: Sigmoid; $\mathbf{t}$: -

Table 7. Scale and shift function of coupling layer

MPP^M Encoder (e)				
Layer	Number of Output	Kernel	Stride	Activation function
Input x	4*64*64	-	-	-
Convolution	32*32*32	3*3	2	ReLU
Convolution	32*16*16	3*3	2	ReLU
Convolution	64*8*8	3*3	2	ReLU
Convolution	64*4*4	3*3	2	ReLU
Fully-Connected	$2 \times z_{dim}$	-	-	-

Table 8. Network structure of MPP^M encoder. z_{dim} is the length of z_k plus the length of z_0.

MPP^M Decoder (d)				
Layer	Number of Output	Kernel	Stride	Activation function
Input $[z_k, z_{0(k)}]$	(z_{dim}+2)*72*72	-	-	-
Convolution	32*70*70	3*3	1	ReLU
Convolution	32*68*68	3*3	1	ReLU
Convolution	32*66*66	3*3	1	ReLU
Convolution	32*64*64	3*3	1	ReLU
Convolution	4*64*64	1*1	1	ReLU

Table 9. Network structure of MPP^M decoder