A. Proof of Theorem 3.1

In this section, we prove Theorem 3.1 which gives closed-form formulation for the prior conditional layer PC in the Gaussian prior case.

We first introduce a powerful tool named partition of unity in Lemma in order to prove Theorem 3.1. We adopt the notations in Section 3 here.

Lemma A.1 *Assume $A \in \mathbb{R}^{d_c \times d}$ ($d_c < d$) has full row-rank, i.e. $rank(A) = d_c$, there exists a matrix $\tilde{A} \in \mathbb{R}^{(d-d_c)\times d}$ such that $A\tilde{A}^T = 0 \in \mathbb{R}^{d_c \times (d-d_c)}$. And for any symmetric positive definite matrix Σ, we have the following decomposition of the identity (unit) matrix $I_d \in \mathbb{R}^{d\times d}$:*

$$\begin{aligned} I_d =& \Sigma^{\frac{1}{2}} A^T (A\Sigma A^T)^{-1} A \Sigma^{\frac{1}{2}} \\ &+ \Sigma^{-\frac{1}{2}} \tilde{A}^T (\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-1} \tilde{A}\Sigma^{-\frac{1}{2}} \end{aligned}$$

Proof: The matrix $\tilde{A}$ is in fact the orthogonal complement of A. Let $V \in \mathbb{R}^d$ be the row space of A, then $\dim(V) = d_c < d$, so the orthogonal complement $V^\perp$ of the subspace $V \subset \mathbb{R}^d$ is non-trivial: $\dim(V^\perp) = d - d_c > 0$. Collect a basis of $V^\perp$ and pack them in rows, we have a matrix $\tilde{A} \in \mathbb{R}^{(d-d_c)\times d}$. By construction we know $A\tilde{A}^T = 0$, because V and $V^\perp$ are orthogonal to each other.

Now consider the following matrix $\Omega \in \mathbb{R}^{d\times d}$:

$$\Omega := \begin{bmatrix} (A\Sigma A^T)^{-\frac{1}{2}} A\Sigma^{\frac{1}{2}} \\ (\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}} \tilde{A}\Sigma^{-\frac{1}{2}} \end{bmatrix}^T .$$

We have

$$\begin{aligned} &\Omega^T\Omega \\ =& \begin{bmatrix} (A\Sigma A^T)^{-\frac{1}{2}} A\Sigma^{\frac{1}{2}} \\ (\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}} \tilde{A}\Sigma^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} (A\Sigma A^T)^{-\frac{1}{2}} A\Sigma^{\frac{1}{2}} \\ (\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}} \tilde{A}\Sigma^{-\frac{1}{2}} \end{bmatrix}^T , \\ =& \begin{bmatrix} \left(\Omega^T\Omega\right)_{11} & \left(\Omega^T\Omega\right)_{12} \\ \left(\Omega^T\Omega\right)_{21} & \left(\Omega^T\Omega\right)_{22} \end{bmatrix} , \end{aligned}$$

where, since $A\tilde{A}^T = 0$ and Σ is symmetric: $\Sigma = \Sigma^T$,

$$\begin{aligned} \left(\Omega^T\Omega\right)_{11} &= (A\Sigma A^T)^{-\frac{1}{2}} A\Sigma A^T (A\Sigma A^T)^{-\frac{1}{2}} = I_{d_c} , \\ \left(\Omega^T\Omega\right)_{12} &= (A\Sigma A^T)^{-\frac{1}{2}} A\Sigma^{\frac{1}{2}}\Sigma^{-\frac{1}{2}}\tilde{A}^T(\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}} \\ &= (A\Sigma A^T)^{-\frac{1}{2}} A\tilde{A}^T(\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}} = 0 , \\ \left(\Omega^T\Omega\right)_{21} &= (\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}} \tilde{A}\Sigma^{-\frac{1}{2}}\Sigma^{\frac{1}{2}} A^T (A\Sigma A^T)^{-\frac{1}{2}} \\ &= (\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}} \tilde{A} A^T (A\Sigma A^T)^{-\frac{1}{2}} = 0 , \\ \left(\Omega^T\Omega\right)_{22} &= (\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}} \tilde{A}\Sigma^{-1}\tilde{A}^T (\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}} \\ &= I_{d-d_c} . \end{aligned}$$

So Ω is in fact a $d\times d$ orthonormal matrix, because

$$\Omega^T\Omega = \begin{bmatrix} I_{d_c} & \\ & I_{d-d_c} \end{bmatrix} = I_d .$$

The orthonormality of Ω also implies $\Omega\Omega^T = I_d$, which can expand as

$$\begin{aligned} I_d &= \Omega\Omega^T \\ &= \begin{bmatrix} (A\Sigma A^T)^{-\frac{1}{2}} A\Sigma^{\frac{1}{2}} \\ (\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}} \tilde{A}\Sigma^{-\frac{1}{2}} \end{bmatrix}^T \begin{bmatrix} (A\Sigma A^T)^{-\frac{1}{2}} A\Sigma^{\frac{1}{2}} \\ (\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}} \tilde{A}\Sigma^{-\frac{1}{2}} \end{bmatrix} \\ &= \Sigma^{\frac{1}{2}} A^T (A\Sigma A^T)^{-1} A\Sigma^{\frac{1}{2}} \\ &\quad + \Sigma^{-\frac{1}{2}} \tilde{A}^T (\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-1} \tilde{A}\Sigma^{-\frac{1}{2}} . \end{aligned}$$

And this proves Lemma A.1. □

Now we give the proof to Theorem 3.1.

Theorem 3.1 *Suppose that ρ is a Gaussian with density $\mathcal{N}(x; 0, \Sigma)$ where the covariance Σ is positive definite, then with $U^c := \Sigma A^T (A\Sigma A^T)^{-1} \in \mathbb{R}^{d\times d_c}$ and $\Sigma^c := \Sigma - \Sigma A^T (A\Sigma A^T)^{-1} A\Sigma \in \mathbb{R}^{d\times d}$, we have*

$$\rho(x|Ax = x_c) = \mathcal{N}(x; U^c x_c, \Sigma^c) .$$

Furthermore, there exists a matrix $W \in \mathbb{R}^{d\times(d-d_c)}$ such that $\Sigma^c = WW^T$, and the prior conditioning layer PC can be given as, with $z \in \mathbb{R}^{d-d_c}$ being standard Gaussian

$$x = PC(x_c, z) = U^c x_c + Wz ,$$

and PC is invertible between x and (x_c, z).

Proof: The conditional probability rule suggests that

$$\rho(x|Ax = x_c) = \rho(x) \bigg/ \left(\int_{\{x' : Ax' = x_c\}} \rho(x') \mathrm{d}x' \right)$$

When x_c is given and fixed, the denominator in the above is a constant with respect to x. Therefore, since we recall that the prior ρ is a Gaussian $\mathcal{N}(0, \Sigma)$, we have

$$\log \rho(x|Ax = x_c) = \log \rho(x) - C' = -\frac{1}{2} x^T \Sigma^{-1} x + C ,$$

where C is a constant that only depends on x_c and Σ. Since $\log \rho(x|Ax = x_c)$ is a quadratic function of x, $\rho(x|Ax = x_c)$ should also be a Gaussian distribution. To determine this distribution we only need to calculate its mean $\mathbb{E}[x|Ax = x_c]$ and covariance $\mathrm{Cov}[x|Ax = x_c]$.

With $U^c := \Sigma A^T (A\Sigma A^T)^{-1}$, we decompose $x = (x - U^c Ax) + U^c Ax$. We will prove later that $x - U^c Ax$ is independent from Ax, so that,

$$\begin{aligned} &\mathbb{E}[x|Ax = x_c] = \mathbb{E}[(x - U^c Ax) + U^c Ax | Ax = x_c] \\ =& \mathbb{E}[x - U^c Ax | Ax = x_c] + \mathbb{E}[U^c Ax | Ax = x_c] \\ =& 0 + U^c x_c = U^c x_c . \end{aligned}$$

To show $x - U^c Ax = (I_d - U^c A)x$ is independent from Ax, where $I_d \in \mathbb{R}^{d\times d}$ is the identity (unit) matrix, we notice that they are both linear transformation of the Gaussian variable x, so their joint distributions should also be a Gaussian, and their covariance can be computed as

$$\mathrm{Cov}[(I_d - U^c A)x, Ax] = (I_d - U^c A)\Sigma A^T .$$

Notice that $U^c A\Sigma A^T = \Sigma A^T(A\Sigma A^T)^{-1}A\Sigma A^T = \Sigma A^T$, so $(I_d - U^c A)\Sigma A^T = \Sigma A^T - \Sigma A^T = 0$. Thus, $x - U^c Ax = (I_d - U^c A)x$ is independent from Ax.

Finally, since $\mathbb{E}\left[x|Ax = x_c\right] = U^c x_c$, we calculate

$$\mathrm{Cov}\left[x|Ax = x_c\right] = \mathrm{Cov}\left[x - U^c Ax|Ax = x_c\right] .$$

Because $x - U^c Ax = (I_d - U^c A)x$ is independent from Ax, we can drop the condition and write:

$$\begin{aligned}
&\mathrm{Cov}\left[x|Ax = x_c\right] = \mathrm{Cov}\left[x - U^c Ax\right] \\
=&(I_d - U^c A)\Sigma(I_d - U^c A)^T \\
=&\Sigma - U^c A\Sigma - \Sigma A^T\left(U^c\right)^T + U^c A\Sigma A^T\left(U^c\right)^T .
\end{aligned}$$

Plug in the definition of U^c, we find

$$\mathrm{Cov}\left[x|Ax = x_c\right] = \Sigma - \Sigma A^T(A\Sigma A^T)^{-1}A\Sigma = \Sigma^c .$$

Therefore we can conclude that

$$\rho(x|Ax = x_c) = \mathcal{N}(x; U^c x_c, \Sigma^c) .$$

For the close form of PC, we first notice that

$$\begin{aligned}
\Sigma^c &= \Sigma - \Sigma A^T(A\Sigma A^T)^{-1}A\Sigma \\
&= \Sigma^{\frac{1}{2}}\left(I_d - \Sigma^{\frac{1}{2}}A^T(A\Sigma A^T)^{-1}A\Sigma^{\frac{1}{2}}\right)\Sigma^{\frac{1}{2}} .
\end{aligned}$$

Using the identity decomposition in Lemma A.1, we have

$$\begin{aligned}
\Sigma^c &= \Sigma^{\frac{1}{2}}\Sigma^{-\frac{1}{2}}\tilde{A}^T(\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-1}\tilde{A}\Sigma^{-\frac{1}{2}}\Sigma^{\frac{1}{2}} \\
&= \tilde{A}^T(\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-1}\tilde{A} .
\end{aligned}$$

Now set $W = \tilde{A}^T(\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}}$, then $W \in \mathbb{R}^{d\times(d-d_c)}$ and $\Sigma^c = WW^T$. With the existence of W, it remains to show that $U^c x_c + Wz$ follows the same distribution as $\rho(x|Ax = x_c)$ for a given x_c, and Gaussian noise z.

We first check if the condition $Ax = x_c$ is satisfied,

$$\begin{aligned}
&A\left(U^c x_c + Wz\right) = AU^c x_c + AWz \\
=&AU^c x_c + A\tilde{A}^T(\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}}z = AU^c x_c \\
=&A\Sigma A^T(A\Sigma A^T)^{-1}x_c = x_c .
\end{aligned} \tag{12}$$

Thus the condition is satisfied. On the other hand, with x_c given and fixed, and z being Gaussian noise, $U^c x_c + Wz$ follows a Gaussian distribution with mean $U^c x_c$ and covariance $WW^T = \Sigma^c$. Therefore, the prior conditioning layer PC can be given as

$$x = PC(x_c, z) = U^c x_c + Wz .$$

Finally, to show the invertibility of PC between x and (x_c, z), it remains to show how to map x back to x_c and z. We claim that the inversion is given by

$$(x_c, z) = PC^{-1}(x) = (Ax, (\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}}\tilde{A}\Sigma^{-1}x) .$$

The first part holds true because

$$Ax = A\left(U^c x_c + Wz\right) = x_c$$

as shown in (12). The second part holds because, when plug in $x = U^c x_c + Wz$, we notice that

$$\begin{aligned}
&(\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}}\tilde{A}\Sigma^{-1}U^c \\
=&(\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}}\tilde{A}\Sigma^{-1}\Sigma A^T(A\Sigma A^T)^{-1} = 0 ,
\end{aligned}$$

and similarly

$$\begin{aligned}
&(\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}}\tilde{A}\Sigma^{-1}W \\
=&(\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}}\tilde{A}\Sigma^{-1}\tilde{A}^T(\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}} = I .
\end{aligned}$$

Therefore, $(\tilde{A}\Sigma^{-1}\tilde{A}^T)^{-\frac{1}{2}}\tilde{A}\Sigma^{-1}x = 0x_c + Iz = z$. So the invertibility of PC is guaranteed. □

We remark that PC is not unique, as for any orthonormal matrix $P \in \mathbb{R}^{(d-d_c)\times(d-d_c)}$, the map $x = U^c x_c + WPz$ is also a valid candidate for PC.

B. Comparison of the Jeffreys divergence and Kullback-Leibler divergence

The KL divergence sometimes can be inefficient to detect multi-modes: it could be easily trapped by a local minimum that misses some modes or is far from the ground-truth. We support our claim by a concrete example below.

Given $\sigma > 0$, let q be a 1-D Gaussian mixture model, with parameters μ_1 and μ_2 unknown but fixed:

$$q(x) = \frac{1}{2}\left(\mathcal{N}(x; \mu_1, \sigma^2) + \mathcal{N}(x; \mu_2, \sigma^2)\right) .$$

Our parametric model p is also a 1-D Gaussian mixture model with parameter $\theta = (\theta_1, \theta_2)$:

$$p_\theta(x) = \frac{1}{2}\left(\mathcal{N}(x; \theta_1, \sigma^2) + \mathcal{N}(x; \theta_2, \sigma^2)\right) .$$

Setting $\mu_1 = -\mu_2 = 1.5$, and $\sigma = 0.25$, we plot the landscape of single-sided KL divergences $D_{\mathrm{KL}}(p_\theta\|q)$ and $D_{\mathrm{KL}}(q\|p_\theta)$, and the Jeffreys divergence $D_{\mathrm{J}}(p_\theta\|q)$ as functions of $\theta = (\theta_1, \theta_2)$ in Figure 8.

It is now clear that $D_{\mathrm{KL}}(p\|q)$ alone might guide the training towards the local minima around $(1.5, 1.5)$ or $(-1.5, -1.5)$, where only one mode of q is captured, see Figure 8. We explain this phenomenon as $D_{\mathrm{KL}}(p\|q) = \mathbb{E}_p\left[\log(p/q)\right] = \int p(x)\left(\log p(x) - \log q(x)\right)\mathrm{d}x$ becomes small as long as p is close to zero wherever q close to zero. (Nielsen & Nock, 2009) describes this property as "zero-forcing", and observes that $D_{\mathrm{KL}}(p\|q)$ will be small when high-density region of p is covered by that of q. However, it doesn't strongly enforce p to capture all high-density region of q. In our example, when $(\theta_1, \theta_2) = (1.5, 1.5)$ or $(-1.5, -1.5)$, the only high-density region of p (around 1.5 **or** -1.5) is a strict subset of high-density region of q (around both 1.5 and -1.5), and thus it attains a local minimum of $D_{\mathrm{KL}}(p\|q)$.

We also argue that the other KL divergence $D_{\mathrm{KL}}(q\|p)$ alone faces the risk as well. Similarly, $D_{\mathrm{KL}}(q\|p) = \mathbb{E}_q\left[\log(q/p)\right] = \int q(x)\left(\log q(x) - \log p(x)\right)\mathrm{d}x$ becomes small as long as q is close to zero wherever p is close to zero.

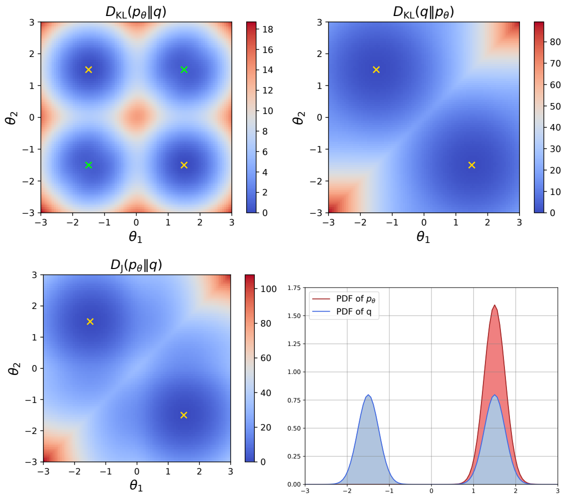

Figure 8. Landscape of $D_{\mathrm{KL}}(p_\theta\|q)$ (upper left), $D_{\mathrm{KL}}(q\|p_\theta)$ (upper right), and $D_{\mathrm{J}}(p_\theta\|q)$ (lower left), density function of p_θ and q when they reach one of the local minima (lower right). We mark the global minima (ground-truth) by golden cross, and other local minima by green cross.

Thus if p captures all modes in q but also contains some extra modes, described as "zero-avoiding" in (Nielsen & Nock, 2009), we could also observe a small value of $D_{\mathrm{KL}}(q\|p)$. Therefore, we choose to use the Jeffreys divergence as a robust learning objective to capture multi-modes.

C. Proof of Theorem 4.1

Theorem 4.1 *The Jeffreys divergence and its derivative to θ admit the following formulation which can be estimated by the Monte Carlo method without samples from q,*

$$D_{\mathrm{J}}(p_\theta\|q) = \mathbb{E}_{p_\theta}\left[\log\frac{p_\theta}{q}\right] + \mathbb{E}_{\tilde{q}}\left[\frac{q}{\tilde{q}}\log\frac{q}{p_\theta}\right]. \quad (10)$$

$$\frac{\partial}{\partial\theta}D_{\mathrm{J}}(p_\theta\|q) = \mathbb{E}_{p_\theta}\left[\left(1+\log\frac{p_\theta}{q}\right)\frac{\partial\log p_\theta}{\partial\theta}\right] - \mathbb{E}_{\tilde{q}}\left[\frac{q}{\tilde{q}}\frac{\partial\log p_\theta}{\partial\theta}\right]. \quad (11)$$

Furthermore, the Monte Carlo estimation doesn't need the normalizing constant Z in (1) as it can cancel itself.

Proof: Equation (10) can be seen from

$$\mathbb{E}_{\tilde{q}}\left[\frac{q}{\tilde{q}}\log\frac{q}{p_\theta}\right] = \int \tilde{q}(x)\frac{q(x)}{\tilde{q}(x)}\log\frac{q(x)}{p_\theta(x)}\mathrm{d}x$$
$$= \int q(x)\log\frac{q(x)}{p_\theta(x)}\mathrm{d}x = \mathbb{E}_q\left[\log\frac{q}{p_\theta}\right],$$

so the right hand side of (10) resumes the definition of Jeffreys divergence in (4).

For (11), we have, by definition

$$\frac{\partial}{\partial\theta}D_{\mathrm{J}}(p_\theta\|q) = \frac{\partial}{\partial\theta}\mathbb{E}_{p_\theta}\left[\log\frac{p_\theta}{q}\right] + \frac{\partial}{\partial\theta}\mathbb{E}_{\tilde{q}}\left[\frac{q}{\tilde{q}}\log\frac{q}{p_\theta}\right].$$

We compute

$$\frac{\partial}{\partial\theta}\mathbb{E}_{p_\theta}\left[\log\frac{p_\theta}{q}\right] = \frac{\partial}{\partial\theta}\int p_\theta(x)\log\frac{p_\theta(x)}{q(x)}\mathrm{d}x$$
$$= \int\left(\frac{\partial p_\theta(x)}{\partial\theta}\log\frac{p_\theta(x)}{q(x)} + p_\theta(x)\frac{\partial\log p_\theta(x)}{\partial\theta}\right)\mathrm{d}x,$$

and

$$\frac{\partial}{\partial\theta}\mathbb{E}_{\tilde{q}}\left[\frac{q}{\tilde{q}}\log\frac{q}{p_\theta}\right] = -\mathbb{E}_{\tilde{q}}\left[\frac{q}{\tilde{q}}\frac{\partial\log p_\theta}{\partial\theta}\right].$$

Now since $\frac{\partial}{\partial\theta}\log p_\theta(x) = \frac{1}{p_\theta(x)}\frac{\partial}{\partial\theta}p_\theta(x)$, we have

$$\frac{\partial}{\partial\theta}p_\theta(x) = p_\theta(x)\frac{\partial}{\partial\theta}\log p_\theta(x).$$

So the term $\frac{\partial}{\partial\theta}\mathbb{E}_{p_\theta}\left[\log\frac{p_\theta}{q}\right]$ further simplifies to

$$\frac{\partial}{\partial\theta}\mathbb{E}_{p_\theta}\left[\log\frac{p_\theta}{q}\right] = \int\left(p_\theta(x)\frac{\partial\log p_\theta(x)}{\partial\theta}\log\frac{p_\theta(x)}{q(x)}\right.$$
$$\left. + p_\theta(x)\frac{\partial\log p_\theta(x)}{\partial\theta}\right)\mathrm{d}x$$
$$= \mathbb{E}_{p_\theta}\left[\left(1+\log\frac{p_\theta}{q}\right)\frac{\partial\log p_\theta}{\partial\theta}\right].$$

So we can conclude (11).

Now instead of the normalized density q, suppose we only have its unnomralized version Zq, with Z unknown. When we replace q with Zq in (11), we get

$$\mathbb{E}_{p_\theta}\left[\left(1+\log\frac{p_\theta}{Zq}\right)\frac{\partial\log p_\theta}{\partial\theta}\right] - \mathbb{E}_{\tilde{q}}\left[\frac{q}{\tilde{q}}\frac{\partial\log p_\theta}{\partial\theta}\right]$$
$$= \mathbb{E}_{p_\theta}\left[\left(1+\log\frac{p_\theta}{q}\right)\frac{\partial\log p_\theta}{\partial\theta}\right] - \mathbb{E}_{\tilde{q}}\left[\frac{q}{\tilde{q}}\frac{\partial\log p_\theta}{\partial\theta}\right]$$
$$- \log Z\,\mathbb{E}_{p_\theta}\left[\frac{\partial\log p_\theta}{\partial\theta}\right]$$
$$= \frac{\partial}{\partial\theta}D_{\mathrm{J}}(p_\theta\|q) - \log Z\int p_\theta(x)\frac{\partial\log p_\theta(x)}{\partial\theta}\mathrm{d}x$$
$$= \frac{\partial}{\partial\theta}D_{\mathrm{J}}(p_\theta\|q) - \log Z\int\frac{\partial p_\theta(x)}{\partial\theta}\mathrm{d}x$$
$$= \frac{\partial}{\partial\theta}D_{\mathrm{J}}(p_\theta\|q) - \log Z\,\frac{\partial}{\partial\theta}\left(\int p_\theta(x)\mathrm{d}x\right)$$
$$= \frac{\partial}{\partial\theta}D_{\mathrm{J}}(p_\theta\|q),$$

as $\int p_\theta(x)\mathrm{d}x = 1$. We remark that we don't have the importance weight term like $Zq/\tilde{q}$ in this case, because we can use the self-normalized importance weight. In practice, if we have $\tilde{x}_i$ sampled i.i.d. from $\tilde{q}$ for $i = 1,\ldots,M$, the importance weight for $\tilde{x}_i$ is given by $w_i = \hat{w}_i/\sum_{j=1}^M \hat{w}_j$, where $\hat{w}_j = Zq(\tilde{x}_j)/\tilde{q}(\tilde{x}_j)$, for $j = 1,\ldots,M$. We can see that the weight w_i is independent from Z as it cancels itself. The similar argument goes for (10).

So we conclude that the Monte Carlo estimation of (10) and (11) doesn't need to know the normalizing constant Z in q as defined in (1). □

D. The Recursive Multiscale Structure

Here we detail the definitions and properties related to the multiscale structure. Recall the recursive design introduced in Section 3, and set L be the number of scales. At scale l ($1 \leq l \leq L$), the problem dimension is d_l, and d_l increases with l: $d_1 < d_2 < \ldots < d_L = d$.

For $2 \leq l \leq L$, the downsample operator A_l at scale l, introduced in Section 3, is a linear operator from $\mathbb{R}^{d_l}$ to $\mathbb{R}^{d_{l-1}}$. It links the variable x_l at scales l to the variable x_{l-1} at scales $l-1$ by $x_{l-1} = A_l x_l$. Similarly, the upsample operator B_l at scale l, introduced in Section 2, is a linear operator from $\mathbb{R}^{d_{l-1}}$ to $\mathbb{R}^{d_l}$, for $2 \leq l \leq L$.

The prior ρ_l at scale l is defined recursively: at the finest scale $l = L$, the prior $\rho_L = \rho$, and as for scale l ($1 \leq l < L$), ρ_l is the density of $A_{l+1}x_{l+1}$ if x_{l+1} follows the last scale prior ρ_{l+1}. In other words, ρ_l is the push-forward density of ρ_{l+1} by A_{l+1} for $1 \leq l < L$.

To define the posterior q_l at scale l, we first let $\hat{B}_l = B_L B_{L-1} \ldots B_{l+1}$ be the linear upsample operator from $\mathbb{R}^{d_l}$ to $\mathbb{R}^{d_L} = \mathbb{R}^d$, for $1 \leq l < L$. It maps $x_l \in \mathbb{R}^{d_l}$ to a valid input in $\mathbb{R}^d$ for $\mathcal{F}$. For consistency, we define $\hat{B}_L = I_{d_L}$, the identity map. Then we can introduce the likelihood $\mathcal{L}_l$ at scale l as, for $1 \leq l \leq L$,

$$\mathcal{L}_l(y|x_l) := \mathcal{L}(y|\hat{B}_l x_l) = \mathcal{N}(y - \mathcal{F}(\hat{B}_l x_l); 0, \Gamma) \,.$$

Now we define the posterior q_l at scale l, for $1 \leq l \leq L$, as

$$q_l(x_l) = \frac{1}{Z_l}\rho_l(x_l)\mathcal{L}_l(y|x_l) \,,$$

where Z_l is the normalizing constant.

The auxiliary distribution $\tilde{q}_l$ at scale l, for $2 \leq l \leq L$, introduced in Section 2, is defined as

$$\tilde{q}_l(x_l) = \frac{1}{\tilde{Z}_l}\rho_l(x_l)\mathcal{L}_{l-1}(y|A_l x_l) \,,$$

where $\tilde{Z}_l$ is the normalizing constant. To see why $\tilde{q}_l$ approximates q_l well, we notice that $A_l x_l$ is a coarse-scale version of x_l, and by the multiscale property, $\mathcal{F}(\hat{B}_l x_l) \approx \mathcal{F}(\hat{B}_{l-1} A_l x_l)$, so

$$\mathcal{L}_l(y|x_l) \approx \mathcal{L}_{l-1}(y|A_l x_l) \,,$$

which implies that $q_l \approx \tilde{q}_l$.

We also notice that, the hierarchical definition of ρ_l implies the following decoupling, for $x_{l-1} = A_l x_l$,

$$\rho_l(x_l) = \rho_{l-1}(x_{l-1})\rho_l(x_l|x_{l-1}) \,.$$

This decoupling is due to the conditional probability rule:

$$\rho_l(x_l|x_{l-1}) = \rho_l(x_l|Ax_l = x_{l-1}) = \rho_l(x_l)/\rho_{l-1}(x_{l-1}) \,.$$

Therefore, we arrive at an alternative formulation of $\tilde{q}_l$:

$$\begin{aligned}\tilde{q}_l(x_l) &:= \frac{1}{\tilde{Z}_l}\rho_l(x_l)\mathcal{L}_{l-1}(y|A_l x_l) \\ &= \frac{1}{\tilde{Z}_l}\rho_{l-1}(Ax_l)\rho_l(x_l|A_l x_l)\mathcal{L}_{l-1}(y|A_l x_l) \\ &= \frac{Z_{l-1}}{\tilde{Z}_l}\rho_l(x_l|Ax_l)q_{l-1}(Ax_l) \,,\end{aligned}$$

which suggests that a sample x_l of $\tilde{q}_l$ can be generated in the following way: (i) sample x_{l-1} from q_{l-1}, and (ii) then sample x_l from $\rho_l(x_l|x_{l-1})$. The relation of ρ_l, $\mathcal{L}_l$, q_l and $\tilde{q}_l$ is shown in Figure 9.

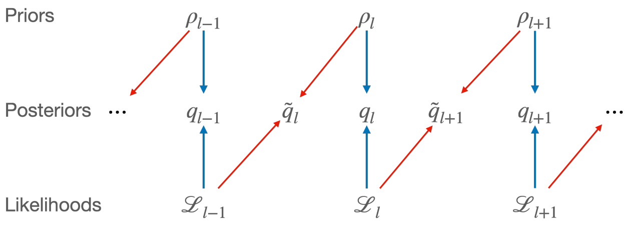

Figure 9. Conceptual diagram of the definitions. Arrows mean that "contribute to the definition of". We further remark that, (i) $\tilde{q}_l$ is the upsampling of q_{l-1} by $\rho_l(x_l|x_{l-1})$, because ρ_l is the upsampling of q_{l-1} by $\rho_l(x_l|x_{l-1})$, and (ii) q_l can be well approximated by $\tilde{q}_l$, because $\mathcal{L}_l(y|x_l)$ can be well approximated by $\mathcal{L}_{l-1}(y|Ax_l)$.

E. More Discussion about Related Work

In this section we provide more discussion and comparison of our approach to related works.

In (Parno et al., 2016), a similar notion of multiscale structure is developed as follows. A likelihood function has the *(Parno et al., 2016)-multiscale structure*, if there exists a coarse-scale *random variable* γ of dimension d_c ($d_c < d$) and a likelihood $\mathcal{L}_c$ such that

$$\mathcal{L}(y|x,\gamma) = \mathcal{L}_c(y|\gamma) \,. \tag{13}$$

Then the *joint posterior distribution* of the fine- and coarse-scale parameters (x,γ) can be decoupled as

$$\begin{aligned} q(x,\gamma) &\propto \rho(x,\gamma)\mathcal{L}(y|x,\gamma) \overset{(i)}{=} \rho(x,\gamma)\mathcal{L}_c(y|\gamma) \\ &\overset{(ii)}{=} \rho(x|\gamma)\rho(\gamma)\mathcal{L}_c(y|\gamma) \overset{(iii)}{=} \rho(x|\gamma)q_c(\gamma) \,, \end{aligned} \tag{14}$$

with normalizing constants omitted in the equivalence relations. We use the (Parno et al., 2016)-multiscale structure (13) in (i), and the conditional probability rule $\rho(x,\gamma) = \rho(x|\gamma)\rho(\gamma)$ in (ii). In (iii) we define $q_c(\gamma) := \rho(\gamma)\mathcal{L}_c(y|\gamma)$ as the (Parno et al., 2016)-posterior in coarse scale.

There are two important differences in these two definitions. First, our coarse-scale parameter x_c is a deterministic function of the fine-scale parameter x, while in (Parno et al., 2016), γ is a random variable that may contain extra randomness outside x (as demonstrated in numerical examples in (Parno et al., 2016)). This difference in definition results in significant difference in modeling: our invertible model has d-dimensional random noise z as input to approximate the target posterior $q(x)$, while models in (Parno et al., 2016) has $(d+d_c)$-dimensional random noise as input to approximate the joint-posterior $q(x,\gamma)$. Another consequence is that users need to define the joint prior $\rho(x,\gamma)$ in (Parno et al., 2016), while in our definition the prior of x_c is naturally induced by the prior of x.

Second, our multiscale structure (8) is an approximate relation and we use invertible flow F in MsIGN to model this approximation, while in (Parno et al., 2016) the multiscale structure (14) is an exact relation and authors treat the prior-upsampled solution $\rho(x|\gamma)q_c(\gamma)$ (right hand side of (14)) as the final solution. Our approximate multiscale relation and further treatment by transform F enables us to apply the method recursively in a multiscale fashion, while in (Parno et al., 2016) the proposed method is essentially a two-scale method and there is not further correction based on the prior-upsampled solution $\rho(x|\gamma)q_c(\gamma)$ at the fine-scale.

Finally, as we discussed in Section 5, the invertible model in (Parno et al., 2016) is polynomials, which suffer from the exponential growth of polynomial coefficients as dimension grows. In this work, the invertible model is deep generative networks, whose number of parameters are independent of the problem dimension.

We also observe that (Spantini et al., 2015; Chen et al., 2019a; Chen & Ghattas, 2020) seeks a best low-rank approximation of the posterior, and treat the approximation as the final solution with no extra modification. As we will see in Appendix F, the true posterior could still be far away from the prior-upsampled solution, especially in the first few coarse scales.

In addition, while in (Ardizzone et al., 2018) flow-based generative models are also used to in distribution capture in inverse problems, their definition of posterior is not equivalent to ours, as they assume no error in measurement. Furthermore, as their training strategy looks to capture the target distribution while simultaneously learning the forward map $\mathcal{F}$, they mainly focused on low-d Bayesian inference problems, in contrast with our high-d setting here.

F. Experimental Setting and Additional Results for BIPs in Section 6.1

F.1. Experimental Setting of BIPs

As introduced in Section 6.1, we don't distinguish between the vector representation of x as grid values on the 2-D 64×64 uniform lattice: $x \in \mathbb{R}^d$ with $d = 64 * 64 = 4096$, and the piece-wise constant function representation of x on the unit disk: $x(s)$ for $s \in \Omega = [0,1]^2$.

We place a Gaussian distribution $\mathcal{N}(0,\Sigma)$ with the covariance Σ as the discretization of $\beta^2(-\Delta)^{-1-\alpha}$ for both of our Bayesian inverse problem examples. Here the discretization of the Laplacian operator Δ can be understood as a graph Laplacian when we consider x gives grid values on a 2-D uniform lattice. We choose zero Dirichlet boundary condition for Δ. As for the distribution to model noise (error) as in (2), we set $\Gamma = \gamma^2 I$, where I is the identity matrix. We list the setting of (α,β,γ) for both BIPs in Table 3.

Table 3. Hyper-parameter (α,β,γ) setting in BIPs

PROBLEM NAME	α	β	γ
SYNTHETIC (SECTION 6.1.1)	0.1	2.0	0.2
ELLIPTIC (SECTION 6.1.2)	0.5	2.0	0.02

The synthetic BIP sets its ground-truth for x as $x(s) = \sin(\pi s_1)\sin(2\pi s_2)$, and defines its forward map as a non-linear measurement of x:

$$\mathcal{F}(x) = \langle \varphi, x\rangle^2 = \left(\int_\Omega \varphi(s)x(s)\mathrm{d}s\right)^2 ,$$

where $\varphi(s) = \sin(\pi s_1)\sin(2\pi s_2)$.

The elliptic BIP is a benchmark problem for high-d inference from geophysics and fluid dynamics (Iglesias et al., 2014; Cui et al., 2016). It also sets its ground-truth for x as $x(s) = \sin(\pi s_1)\sin(2\pi s_2)$. However, the forward map is defined as $\mathcal{F}(x) = \mathcal{O} \circ \mathcal{S}(x)$, where $u = \mathcal{S}(x)$ is the solution to an elliptic partial differential equation with zero Dirichlet boundary condition:

$$-\nabla \cdot \left(e^{x(s)}\nabla u(s)\right) = f(s) , \quad s \in \Omega ,$$

And $\mathcal{O}$ is linear measurements of the field function u:

$$\mathcal{O}(u) = \left[\int_\Omega \varphi_1(s)u(s)\mathrm{d}s \quad \dots \quad \int_\Omega \varphi_m(s)u(s)\mathrm{d}s\right]^T .$$

The force term f of the elliptic PDE is set as

$$\begin{aligned} f(s) = &\frac{50}{\pi}\left(2e^{-10\|s-f_1\|^2} + 2e^{-10\|s-f_2\|^2}\right. \\ &\left. -e^{-10\|s-f_3\|^2} - e^{-10\|s-f_4\|^2}\right) , \end{aligned}$$

where $f_1 = (0.25, 0.3)$, $f_2 = (0.25, 0.7)$, $f_3 = (0.7, 0.3)$, $f_4 = (0.7, 0.3)$, and $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^2$. f is mirror-symmetric along the s_2 direction: $f(s_1,s_2) = f(s_1, 1-s_2)$. As for the measurement functions φ_k $(1 \le k \le m)$, we set $m = 15$ and each φ_k gives local detection of u. They are also mirror-symmetry along the s_2 direction. See Figure 10 for the visualization of f and φ_k.

By the symmetric design, our posterior q has the property $q(x) = q(x')$, where, when considered in the representation of function, x and x' is linked by $x(s_1,s_2) = x'(s_1, 1-s_2)$ for $s = (s_1,s_2) \in \Omega$. We carefully choose our hyper-parameters (α,β,γ), as in Table 3, such that it ends up to be q not only mirror-symmetric, but also double-modal posterior distribution. To certify the multi-modality, we run multiple gradient ascent searching of maximum-a-posterior points, starting from different initial points. They all con-

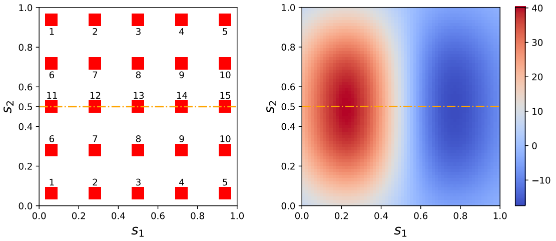

Figure 10. Left: 15 measurement functions in $\mathcal{O}$. Here we plotted non-zero patches of φ_k, $k = 1, \ldots, 15$, with k labeled next to them. φ_k has a constant non-zero value on its patch(es) and is zero else where. The constant value here is chosen so that we have $\|\varphi_k\|_{L^2} = 1$. Right: The force term f of the elliptic PDE in $\mathcal{S}$. We remark that both measurement functions and the force term are mirror-symmetric along the s_2 direction (the orange dash line).

verge to two mutually mirror-symmetric points x^* and $x^{*\prime}$: for $s = (s_1, s_2) \in \Omega$, $x^*(s_1, s_2) = x^{*\prime}(s_1, 1 - s_2)$. Visualization of the 1D landscape profile of the posterior q on the line passing through x^* and x'^* also shows a clear double-modal feature.

To simulate the forward process $\mathcal{F}$, we solve the PDE in map $\mathcal{S}$ by the Finite Element Method with mesh size $1/64$. We remark here this setting is independent of the scale l ($1 \leq l \leq L$) in our recursive strategy.

When counting the number of forward simulations (nFSs) as our indicator for computational cost, we notice that all SVGD-type methods: A-SVGD, SVGD and pSVGD, require not only the log posterior $\log q(x)$ but also its gradient: $\partial_x \log q(x)$. Thanks to the adjoint method, the gradient can be computed with only one extra forward simulation.

In Table 4 we report our hyperparameter of network setting in BIPs. To initialize our multi-stage training as in line 2 of Algorithm 1, we still try to minimize the Jeffreys divergence $D_J(p_\theta \| q) = \mathbb{E}_{p_\theta}[\log(p_\theta/q)] + \mathbb{E}_q[\log(q/p_\theta)]$, but this time it is directly estimated by the Monte Carlo method with samples from distribution p_θ and q. p_θ samples come from the model itself and q samples come from an HMC chain. We remark that at $l = 1$, the posterior lies in 4-D space, which is relatively a low-d problem, so an HMC run can approximate the target distribution q_1 well. Our present solution of seeking help from HMC can be replaced by some other strategies, like other MCMC methods, and deep generative networks.

For A-SVGD, we choose Glow (Kingma & Dhariwal, 2018) as its network design, with the same network hyperparameter in Table 4. Due to the fact that MsIGN is more parameter-saving than Glow with the same hyperparameter, A-SVGD model has more trainable parameters than our MsIGN model, reducing the possibility that that its network is not expressive enough to capture the modes.

Table 4. Hyperparameter setting for MsIGN in Section 6.1. The meaning of terms can be found in (Kingma & Dhariwal, 2018).

Problem Name	Synthetic	Elliptic
Minibatch Size	100	100
Scales (L)	6	6
♯ of Glow Blocks (K)	16	32
♯ of Hidden Channels	32	64

As for our training of HMC, we grid search its hyperparameters, and use curves of acceptance rate and autocorrelation as evidence of mixing. We consider our HMC chain mixing successfully if the acceptance rate stabilizes and falls between $30\% - 75\%$, as suggested by (Neal et al., 2011), and the autocorrelation decays fast with respect to lag.

As for the ablation study shown in Section 6.1.3, all models involved Glow or MsIGN adopt network hyperparameters as shown in Table 4.We remark that it is not straightforward to design multi-stage strategy for Glow models, because their channel size increases with l. So for models with different number of scales L, there is no direct way to initialize one model with another. Therefore for methods using Glow, we don't consider multi-stage training.

Also, as will be seen in Appendix F.2.2, the elliptic problem at $l = 1$ is ill-posed, its posterior is highly rough, and MsIGN variants (like MsIGN trained by the KL divergence) can hardly capture its two modes, see Table 5 and Figure 13. We report that in general it is unlikely for multi-stage training to pick up the missing mode. Therefore, to make more convincing comparison, for models with multi-stage training, we use pretrained MsIGN model at $l = 1$ (who captures q_1 well) as their initialization for $l = 2$.

F.2. Additional Results of BIPs

In this section we provide more results on the Bayesian inverse problems examples in Section 6.1.

F.2.1. Synthetic Bayesian Inverse Problem

In Figure 11 we provide comparison of the marginal distribution in the critical direction w^* at intermediate scales $l = 1, \ldots, 5$. For the final scale $l = 6$ please refer to Figure 3(a). We can see that as the dimension increases, A-SVGD and SVGD become less robust in mode capture and collapse to one mode. Besides, HMC becomes imbalanced between modes, and pSVGD is a bit biased for q_6 in Figure 3(a). We remark here that in q_1, A-SVGD failed to capture both modes as it did to q_2. This phenomenon might be caused by the aliasing effect. Very rough resolution at this scale pushes the prior to penalize the smoothness much, and also adds the sensitivity to likelihood because entries of x can easily influence its global behavior. Therefore, there is a larger log density gap between modes in the posterior q_1 than other scales, which adds up to the difficulty of multi-mode capture. A similar effect is observed in the elliptic example as in the next section.

The learning curve in Figure 12 shows the effectiveness of our multi-stage training of MsIGN. As we can see, the training process at $l = 6$ did improve the model, with the Jeffreys divergence dropped from 252 to 56.8. Rather than simply refining the resolution, our multi-stage training strategy does improve our approximation to the distribution when entering the next scale. We will show more evidence about this in the next section.

F.2.2. Elliptic Bayesian Inverse Problem

In Figure 11 we provide comparison of marginal comparison in the critical direction w^* at intermediate scales $l = 1, \dots, 5$. For $l = 6$ please refer to Figure 4(a). Again, for this complicated posterior we observe that all methods except MsIGN and HMC failed in detecting all modes, and could even get stuck in the middle. In this testbed, HMC seems to capture both modes well. However we will point out that its samples can't be treated like a reference solution. The failure of HMC at q_1 is due to the aliasing effect: the prior penalizes fluctuation in spatial directions heavily, and the likelihood is also very strong. As a consequence, the posterior q_1 is highly twisted, and the log density gap between two modes becomes significant.

In Figure 12, we also show the necessity of training after prior conditioning. In other words, q_l is not the same as the prior-conditioned surrogate $\tilde{q}_{l-1}$, though they are similar. We plot one of the modes we detected by our models for $l = 4, 5, 6$. Comparing figures of Figure 12, we can see the location, shape and scale of bumps and caves are different, which means the learned q_l is different from the prior-conditioned surrogate $\tilde{q}_{l-1}$, who serves as its initialization. Our multi-stage training does learn more information at each scale, rather than simply scale up the resolution.

F.2.3. Ablation Study of Bayesian Inverse Problem

In Figure 5 we compared different variants of MsIGN and its training strategy at scale $l = 6$. In Figure 13 we plot the same comparison at intermediate scales $l = 1, \dots, 5$. Since the curves overlap each other heavily in Figure 13, we conclude their results of mode capturing (together with Figure 5) in Table 5.

We can see from Table 5 that our framework and strategy outperforms all its variants in these two Bayesian inverse problems, which proved the necessity of our prior conditioning layer, network design, multi-stage training strategy, and Jeffreys divergence. In particular, the experiment of MsIGN-SNN supports our prior conditioning layer design, the experiment of MsIGN-KL supports our use of the Jeffreys divergence and MsIGN-KL-S supports our use of multi-stage training strategy.

Besides that, we can also see that multi-stage training also benefits other models like MsIGN with KL divergence objective or A-SVGD with MsIGN. By carefully comparing the marginals plotted in Figure 13, we can also conclude that Jeffreys divergence can help capture more balanced modes than KL divergence.

G. Experimental Setting and Additional Results for Image Synthesis in Section 6.2

G.1. Experimental Setting of Image Synthesis

Although there is no posterior for natural images, we can still use MsIGN to capture the distribution of natural images. We still feed Gaussian noises to MsIGN, and hope to get high-quality images from it as in (3). The training of MsIGN is now governed by the Maximal Likelihood Estimation due to the lack of the posterior density. In other words, we train our MsIGN by maximizing $\mathbb{E}_{x\sim q}[\log p_\theta(x)]$, which is equivalent to minimizing $D_{\mathrm{KL}}(q\|p_\theta)$, where q is the empirical distribution of natural images given by the data set. As for the multiscale strategy, we naturally take q_l to be the distribution of (downsampled) images at resolution d_l.

We use the invertible block introduced in (Kingma & Dhariwal, 2018) as our model for the invertible flow. For our numbers in Table 2, we report our hyperparameter settings in Table 6. Samples from those data sets are treated as 8-bit images. For all experiments we use Adam (Kingma & Ba, 2014) optimizer with $\alpha = 0.001$ and default choice of β_1, β_2. For models here that requires mutli-stage training in Algorithm 1, non-final stages ($l < L$) will only be trained for 125 epochs.

To establish the prior conditioning layer PC in this image application, we let the downsample operator A_l from scale l to scale $l-1$ be the average pooling operator with kernel size 2 and stride 2. We further assume the covariance Σ_l at each scale be a scalar matrix, i.e. a diagonal matrix with equal diagonal elements.

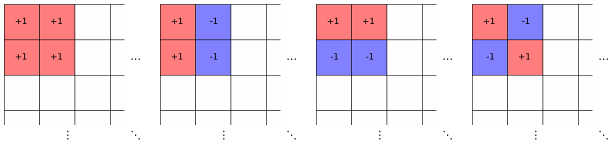

Figure 14. Left most: an example row of A_l, plotted as a matrix; The rest: example rows of $\tilde{A}_l$ correspond to the former row of A_l. They (with some unplotted ones) form the Haar basis, and can be expressed as local convolution operation.

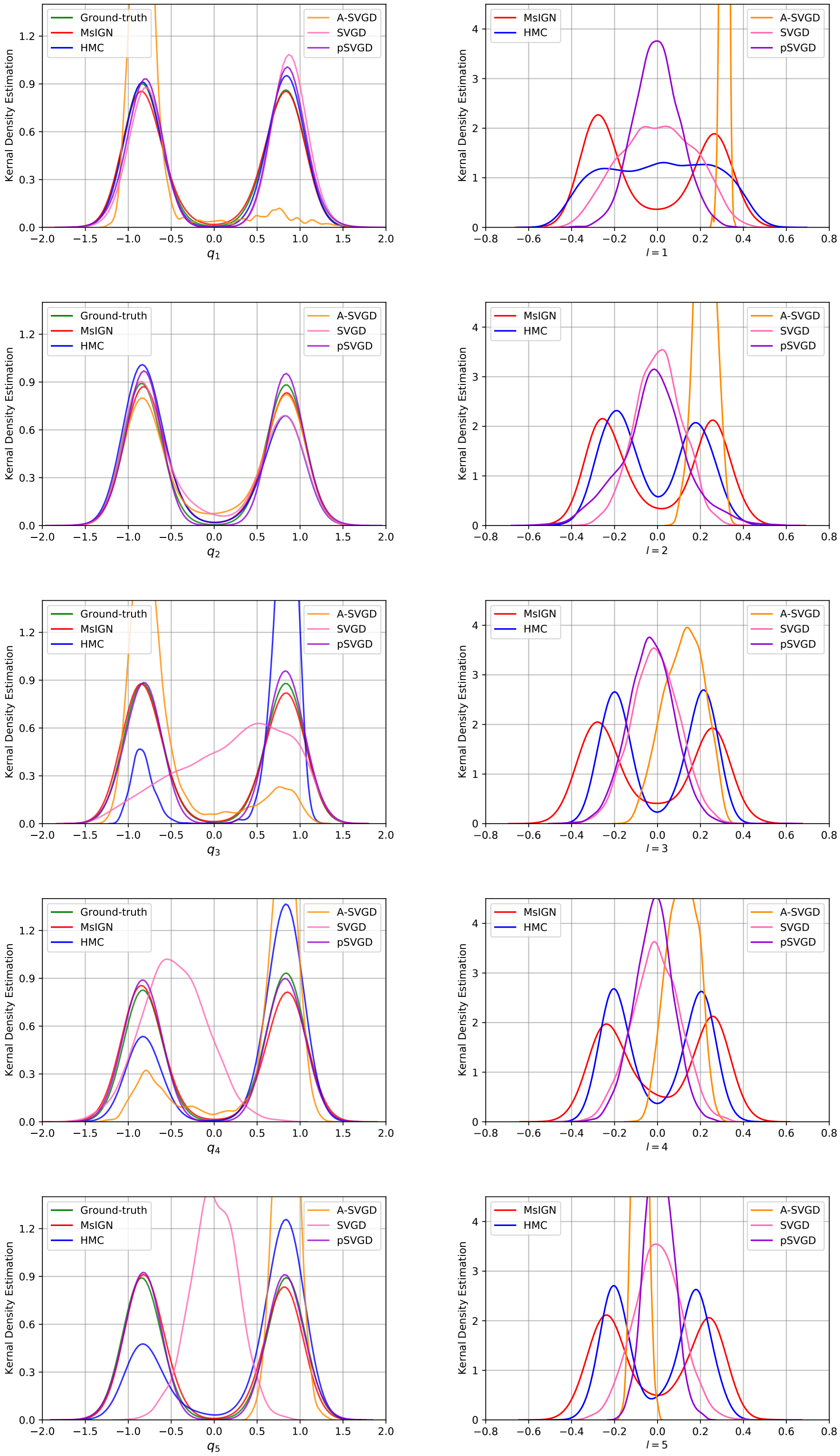

Figure 11. Marginal comparison at the intermediate scales $l = 1, \dots, 5$. Left: Synthetic BIP; Right: Elliptic BIP. In the synthetic example, as the dimension increases, SVGD and A-SVGD failed in mode capture. Besides, HMC becomes imbalanced between modes, and pSVGD is a bit biased for q_6 in Figure 3(a). In the elliptic example, all methods except MsIGN and HMC failed in detecting all modes, and could even get stuck in the middle. HMC has acceptable performance, but still suffers from imbalanced modes at some scales.

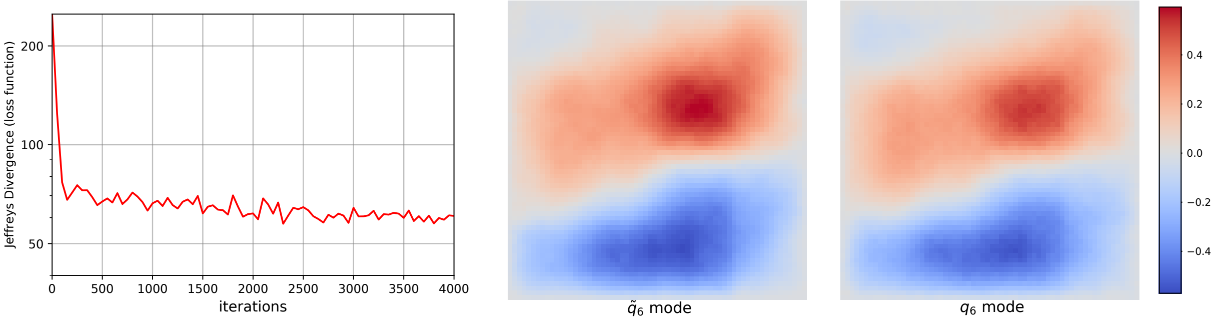

Figure 12. Necessity of training after prior conditioning. Left: learning curve of multi-stage MsIGN at $l = 6$ in the synthetic BIP example; Middle and Right: comparison of the modes captured by the prior conditioned untrained model and the trained model in the elliptic BIP example. The learning curve shows that the model distribution is constantly getting closer to the target distribution in the last stage of training, supporting the necessity of training after prior conditioning. The mode comparison shows that bumps and caves in the left images are different from the right ones, especially in scale, as shown by color shade. Therefore, prior conditioning provides a good initial guess, but training is still necessary.

Table 5. Table for mode capturing results by eye ball norm. Upper: synthetic Bayesian inverse problem; Lower: elliptic Bayesian inverse problem. “T” demotes the successful capturing of two modes, “F” denotes mode collapse, while “I” denotes biased, not well-separated modes capturing. For results marked with “I”, we refer readers to Figure 13 for detail information. *: we initialize the $l = 2$ model by our MsIGN $l = 1$ pretrained model, see Appendix F.1.

SCALE	$l = 1$	$l = 2$	$l = 3$	$l = 4$	$l = 5$	$l = 6$
GLOW	T	F	F	F	F	F
MSIGN-SNN	T	T	T	T	I	I
MSIGN-KL-S	T	F	F	F	I	F
MSIGN-KL*	T	T	T	T	T	T
MSIGN-AS-S	T	F	F	F	F	F
MSIGN-AS*	T	T	T	I	I	I
MSIGN	**T**	**T**	**T**	**T**	**T**	**T**
SCALE	$l = 1$	$l = 2$	$l = 3$	$l = 4$	$l = 5$	$l = 6$
GLOW	F	F	F	F	F	F
MSIGN-SNN	F	F	F	F	F	F
MSIGN-KL-S	F	F	F	F	F	F
MSIGN-KL*	F	I	I	I	I	I
MSIGN-AS-S	F	F	F	F	F	F
MSIGN-AS*	F	I	I	T	I	I
MSIGN	**T**	**T**	**T**	**T**	**T**	**T**

Table 6. Hyperparameter setting for results in Table 2. Here the meaning of terms can be found in (Kingma & Dhariwal, 2018).

DATA SET	MNIST	CIFAR-10	CELEBA	IMAGENET 32	IMAGENET 64
MINIBATCH SIZE	400	400	200	400	200
SCALES (L)	2	3	3	3	3
♯ OF GLOW BLOCKS (K)	32	32	32	32	32
♯ OF HIDDEN CHANNELS	512	512	512	512	512
♯ OF EPOCHS	2000	2000	1000	400	200

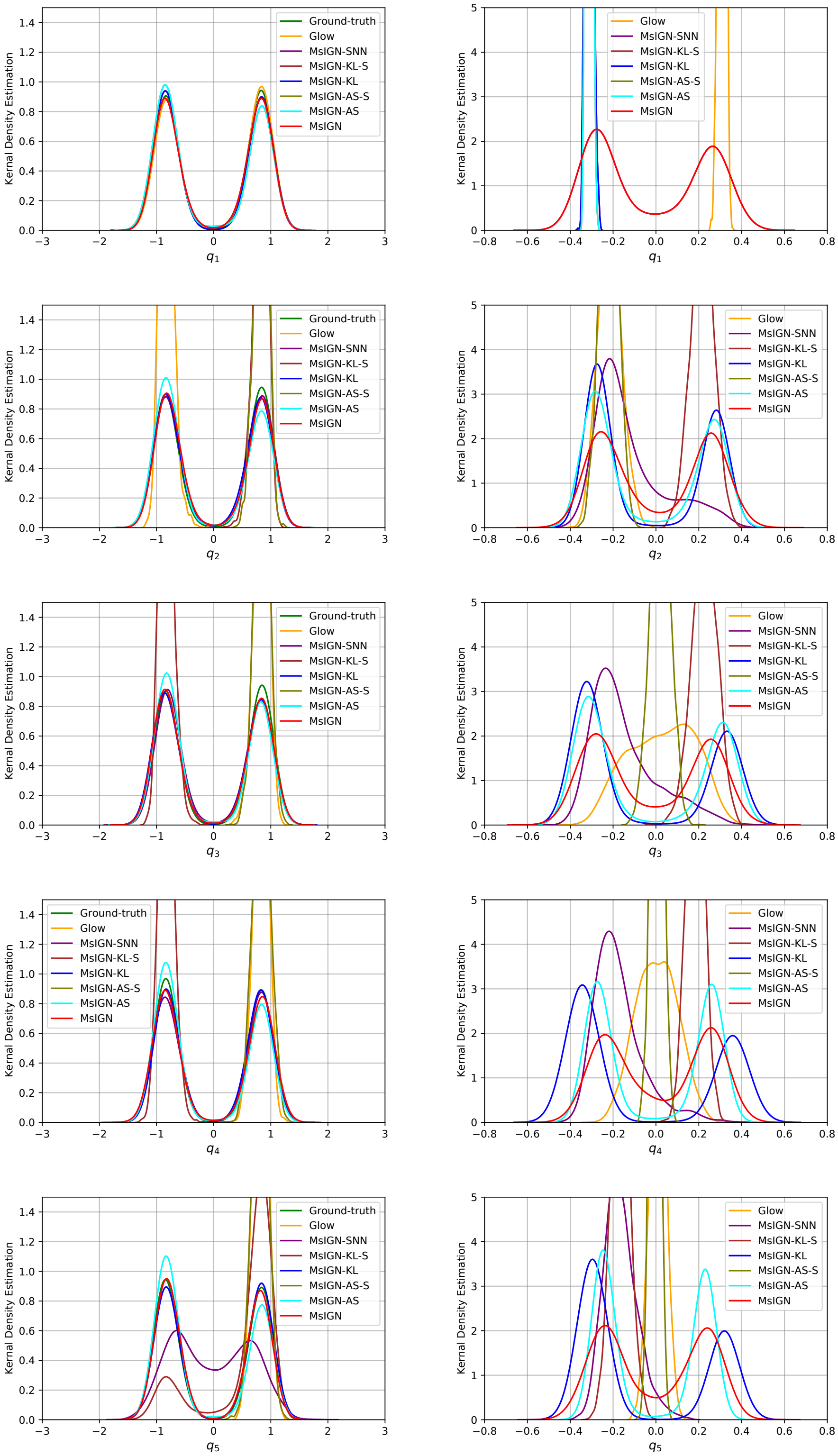

Figure 13. Ablation study at intermediate scales $l = 1, \ldots, 5$. Left: Synthetic Bayesian inverse problem; Right: Elliptic Bayesian inverse problem. For MsIGN-AS and MsIGN-KL, we initialize their $l = 2$ models by our MsIGN $l = 1$ pretrained model, see Appendix F.1.

Since $A_l \in \mathbb{R}^{d_{l-1} \times d_l}$ is the average pooling operator, its rows, which give averages of each local patch, is a subset of the Haar basis, see Figure 14. We can collect the rest Haar basis as $\tilde{A}_l \in \mathbb{R}^{(d_l - d_{l-1}) \times d_l}$. Due to the orthogonality of the Haar basis, there exists a constant $\lambda_l > 0$ such that

$$\begin{aligned}\begin{bmatrix} A_l A_l^T & \\ & \tilde{A}_l \tilde{A}_l^T \end{bmatrix} &= \begin{bmatrix} A_l \\ \tilde{A}_l \end{bmatrix} \begin{bmatrix} A_l \\ \tilde{A}_l \end{bmatrix}^T = \lambda_l I_{d_l} \\ &= \begin{bmatrix} A_l \\ \tilde{A}_l \end{bmatrix}^T \begin{bmatrix} A_l \\ \tilde{A}_l \end{bmatrix} = A_l^T A_l + \tilde{A}_l^T \tilde{A}_l \,.\end{aligned}$$

As a by-product we see $A_l A_l^T = \lambda_l I_{d_{l-1}}$ and $\tilde{A}_l \tilde{A}_l^T = \lambda_l I_{d_l - d_{l-1}}$. In our case, as A_l is the average pooling operator, we actually have $\lambda_l = 1/4$.

Since we assume the covariance Σ_l is a scalar matrix, we can find a scalar $c_l > 0$ such that $\Sigma_l = c_l I_{d_l}$. Now following Theorem 3.1, we can find an explicit form for $\Sigma_{l|l-1}$, $l \geq 2$, which is the Σ^c at scale l:

$$\begin{aligned}\Sigma_{l|l-1} &= \Sigma_l - \Sigma_l A_l^T (A_l \Sigma_l A_l^T)^{-1} A_l \Sigma_l \\ &= c_l I_{d_l} - c_l A_l^T (\lambda_l I_{d_{l-1}})^{-1} A_l \\ &= \frac{c_l}{\lambda_l} (A_l^T A_l + \tilde{A}_l^T \tilde{A}_l) - \frac{c_l}{\lambda_l} A_l^T A_l \\ &= \frac{c_l}{\lambda_l} \tilde{A}_l^T \tilde{A}_l \,.\end{aligned}$$

Therefore, we obtain the decomposition of $\Sigma_{l|l-1} = W_l W_l^T$ in Theorem 3.1 for free, where now W_l is the original W at scale l. One apparent choice is $W_l = \mu_l \tilde{A}_l^T$ with $\mu_l = \sqrt{\frac{c_l}{\lambda_l}}$. Finally, as suggested by Theorem 3.1 we are now only left to estimate the scalar μ_l for each $l \geq 2$ to establish PC_l.

The constant μ_l is estimated numerically on data sets. In fact, we have accessible to different resolutions of images from the data set when we perform pooling operation. We take x_l to be the pooling of images from data set to its resolution, and estimate μ_l according to Theorem 3.1:

$$x_l = U_{l-1} x_{l-1} + W_l z_l = U_{l-1} x_{l-1} + \mu_l \tilde{A}_l^T z_l \,,$$

where $z_l \sim \mathcal{N}(0, I_{d_l - d_{l-1}})$ are the random noise at scale l, and U_{l-1} by definition is

$$\begin{aligned}U_{l-1} &= \Sigma_l A_l^T (A_l \Sigma_l A_l^T)^{-1} \\ &= c_l A_l^T (c_l A_l A_l^T)^{-1} = A_l^T (A_l A_l^T)^{-1} \\ &= A_l^T (\lambda_l I_{d_{l-1}})^{-1} = \frac{1}{\lambda_l} A_l^T \,.\end{aligned}$$

Plug it back, we have

$$x_l = \frac{1}{\lambda_l} A_l^T x_{l-1} + \mu_l \tilde{A}_l^T z_l \,.$$

Now multiply both sides with $\tilde{A}_l$, noticing that $\tilde{A}_l \tilde{A}_l^T = \lambda_l I_{d_l - d_{l-1}}$ and $\tilde{A}_l A_l^T = 0$, we arrive at

$$\tilde{A}_l x_l = \lambda_l \mu_l z_l \,.$$

Since μ_l is a scalar, it can be estimated by moment matching of both sides, as λ_l and $\tilde{A}_l$ is known. Here x_l is the natural images at resolution d_l. For example, we use 10000 randomly sampled images from each data set and estimate μ_l by matching the variance of both sides, we report our estimates of μ_l in Table 7.

Table 7. Estimate of μ_l for different data sets and scale l.

DATA SET	μ_2	μ_3
MNIST	0.67	–
CIFAR-10	0.48	0.46
CELEBA 64	0.22	0.30
IMAGENET 32	0.32	0.42
IMAGENET 64	0.28	0.36

G.2. Additional Results of Image Synthesis

We attach more synthesized images by MsIGN from MNIST and CIFAR-10 in Figure 15, 16. For the CelebA data set, we made use of our multiscale design and trained our MsDGN for a higher resolution 128. In this case, the number of scales $L = 4$, and we set the hyperparameters for the first 3 scales the same as we use for the $64 * 64$ resolution model. For the last scale $l = 4$, due to memory limitation, we set $K = 32$ and hidden channels 128. We show our synthesized $128 * 128$ resolution results in Figure 17.

We also use this 4-scale model to show the interpret-ability of our internal neurons in Figure 18. We snapshot internal neurons for 4 times every scale, resulting a snapshot chain of length $4 * 4 = 16$ for every generated image. We can see our MsIGN generates global features at the beginning scales and starts to add more local details at higher scales.

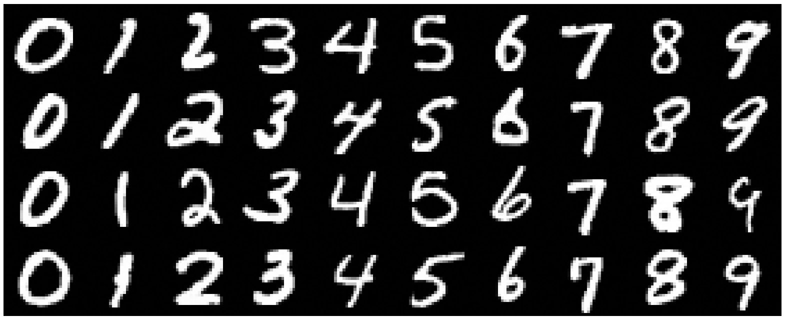

Figure 15. Synthesized 28×28-resolution images from MsIGN on the MNIST data set, temperature $= 1.0$. We show 4 samples per digit.

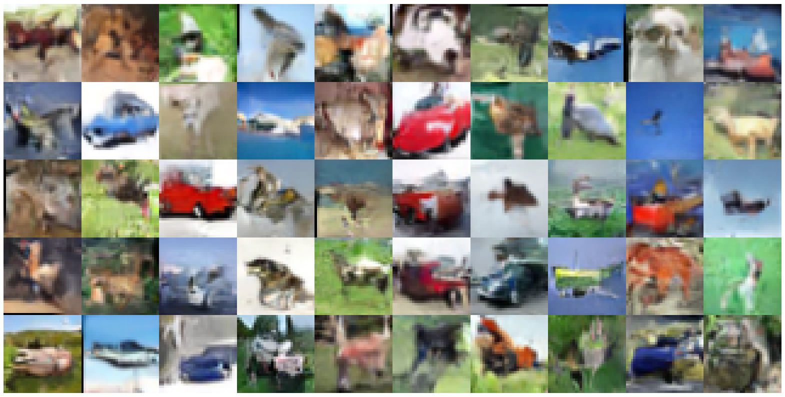

Figure 16. Synthesized images of resolution 32×32 from MsIGN on the CIFAR-10 data set, temperature $= 1.0$.

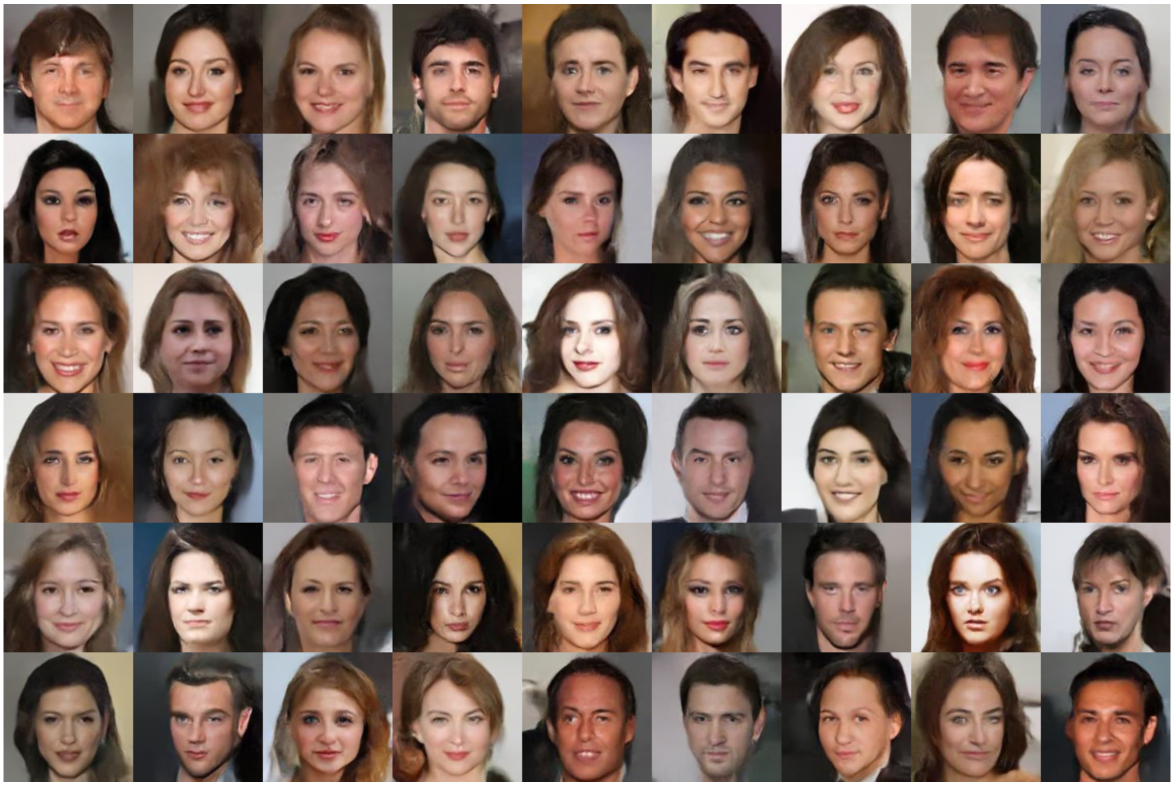

Figure 17. Synthesized images of resolution 128×128 from MsIGN on the CelebA data set, temperature $= 0.8$.

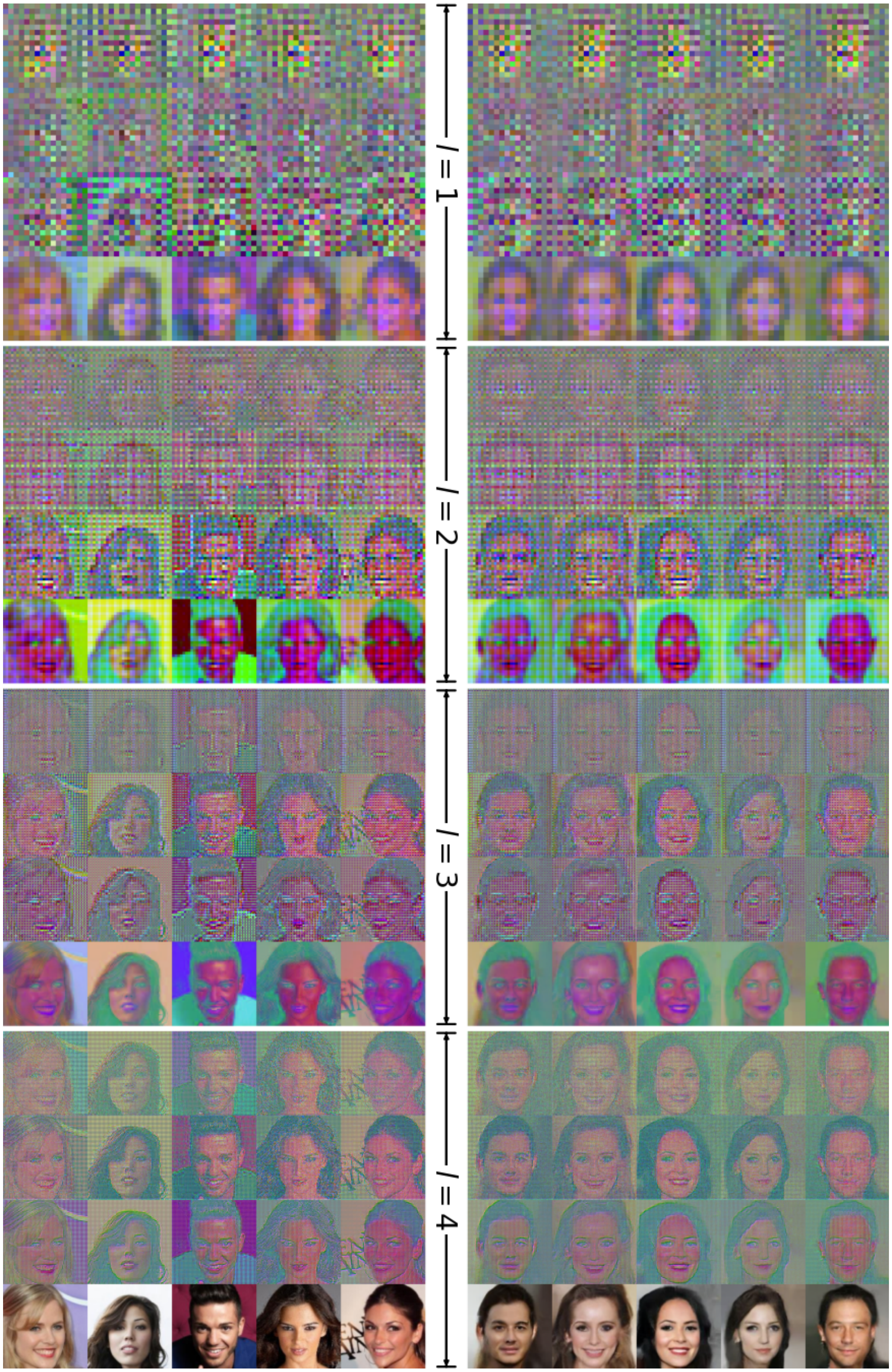

Figure 18. Visualization of internal neurons of MsIGN in synthesizing or recovering 128×128-resolution images on CelebA data set. Snapshots (from top to bottom) are taken 4 times every scale, resulting $4 * 4 = 16$ checkpoints for every image generated. At scale l ($1 \le l \le 4$), where the resolution is $2^{3+l} \times 2^{3+l}$, we take 4 snapshots at the head, two trisection points and tail of the invertible flow F_l. Left: when recovering images from the data set; Right: when synthesizing new images from random noise.