Multiscale Invertible Generative Networks
for High-Dimensional Bayesian Inference

Shumao Zhang [1] **Pengchuan Zhang** [2] **Thomas Y. Hou** [1]

Abstract

We propose a Multiscale Invertible Generative Network (MsIGN) and associated training algorithm that leverages multiscale structure to solve high-dimensional Bayesian inference. To address the curse of dimensionality, MsIGN exploits the low-dimensional nature of the posterior, and generates samples from coarse to fine scale (low to high dimension) by iteratively upsampling and refining samples. MsIGN is trained in a multi-stage manner to minimize the Jeffreys divergence, which avoids mode dropping in high-dimensional cases. On two high-dimensional Bayesian inverse problems, we show superior performance of MsIGN over previous approaches in posterior approximation and multiple mode capture. On the natural image synthesis task, MsIGN achieves superior performance in bits-per-dimension over baseline models and yields great interpret-ability of its neurons in intermediate layers.

1. Introduction

To infer about hidden system states $x \in \mathbb{R}^d$ from observed system data $y \in \mathbb{R}^s$, Bayesian inference blends some prior knowledge, given as a distribution ρ, with data y into a powerful posterior. Since direct measurement of x can be inaccessible, the data y is generated through $y = \mathcal{F}(x) + \varepsilon$, where $\mathcal{F}$ is a forward map that can be highly nonlinear and complicated, $\varepsilon \in \mathbb{R}^s$ is random noise modelled by some distribution. For illustration simplicity, we assume a Gaussian $\mathcal{N}(0, \Gamma)$ for ε. The posterior is characterized as

$$q(x|y) = \frac{1}{Z}\rho(x)\mathcal{L}(y|x)\,, \tag{1}$$

[1]Department of Computational & Mathematical Sciences, Caltech, Pasadena, California, USA [2]MSR AI Lab, Redmond, Washington, USA. Correspondence to: Shumao Zhang <shumaoz@caltech.edu>.

where $\mathcal{L}$ is the likelihood given as

$$\mathcal{L}(y|x) = \mathcal{N}(y - \mathcal{F}(x); 0, \Gamma)\,, \tag{2}$$

which is the density of $\varepsilon = y - \mathcal{F}(x)$, and Z is some normalizing constant that is usually intractable in practice. For simplicity reason, in the following context we abbreviate $q(x|y)$ in (1) as $q(x)$, because the data y *only* plays the role of defining the target distribution $q(x)$ in our framework.

A key and long-standing challenge in Bayesian inference is to approximate, or draw samples from the posterior q, especially in high-dimensional (high-d) cases. An arbitrary distribution can concentrate its density anywhere in the space, and these concentrations (also called "modes") become less connected as d increases. As a result, detecting these modes requires computational cost that grows exponentially with d. This intrinsic difficulty of mode collapse is a consequence of the curse of dimensionality, which all existing Bayesian inference methods suffer from, e.g., MCMC-based methods (Neal et al., 2011; Welling & Teh, 2011; Cui et al., 2016), SVGD-type methods (Liu & Wang, 2016; Chen et al., 2018; 2019a), and generative modeling (Morzfeld et al., 2012; Parno et al., 2016; Hou et al., 2019).

In this paper, we exploit the multiscale structure to deal with the high-dimensional Bayesian inference problems. The multiscale structure means that the forward map $\mathcal{F}$ depends mostly on some low-d structure of x, referred as coarse scale, instead of the high-d structure, referred as fine scale. For example, the terrain shape x, given as the discretization of 2-D elevation map on a 2-D lattice grid, is a quantity with dimension equal to the number of grid points. Simulating the 2-D precipitation distribution y using terrain shape x at the scale of kilometer is a reasonable approximation to itself at the scale of meter. The former one is a coarse-scale version of the latter, and has 10^6-times fewer problem dimension (grid points). Such multiscale structure is very common in high-d problems, especially when x is some spatial or temporal quantity. The coarse-scale approximation to the original fine-scale problem is low-d and computationally attractive, and can help divide-and-conquer the high-d challenge. The multiscale property is discussed in detail in Section 2.

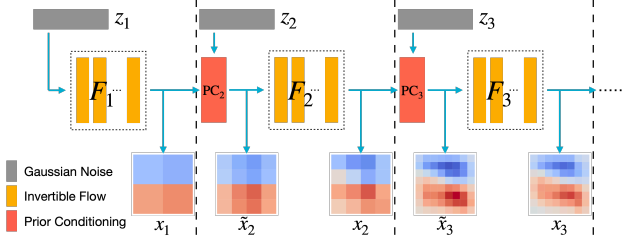We approximate the target q by a parametric family of distri-

Figure 1. MsIGN generates samples from coarse to fine scale, as depicted by (3). Each scale, separated by dash lines, takes in x_{l-1} from the coarser scale with random seed z_l, and outputs a sample x_l of the finer scale. MsIGN iteratively upsamples (by PC_l) and refines (by F_l) samples to the target scale.

Figure 2. An example of multiscale problem from Section 6.1.2. The coarse-scale $x_1, \ldots, x_5$ are downsampling of the original x_6 from resolution 64×64. As the resolution gets refined, the relative error to $\mathcal{F}(x_6)$ significantly drops. Typically, it suggests a good approximation in (5) when, for example, setting $Ax = x_5$.

bution p_θ, and look for an optimal choice of θ. The working distribution p_θ is the density of $T(z; \theta)$, where z is random seed which we assume to be Gaussian noise here, T is a transport map parameterized by θ that drives z to the sample of p_θ. The optimality of θ is determined by the match of p_θ to q, measured by the Jeffreys divergence $D_{\mathrm{J}}(p_\theta \| q)$.

We propose a Multiscale Invertible Generative Network (MsIGN) as the map T, with a novel training strategy to minimize the Jeffreys divergence. Specifically, T maps z to the sample $x = x_L$ of p_θ in a coarse-to-fine manner:

$$x_1 = F_1(z_1),$$
$$\tilde{x}_l = PC_l(x_{l-1}, z_l), \quad x_l = F_l(\tilde{x}_l), \quad 2 \leq l \leq L. \quad (3)$$

Here we split z into $(z_1, z_2, \ldots, z_L)$. At scale l, the prior conditioning layer PC_l upsamples the coarse-scale $x_{l-1} \in \mathbb{R}^{d_{l-1}}$ to a finer scale $\tilde{x}_l \in \mathbb{R}^{d_l}$, which is the "best guess" of x_l given its coarse scale version x_{l-1} and the prior ρ. The invertible flow F_l then modifies $\tilde{x}_l$ to $x_l \in \mathbb{R}^{d_l}$, which again can be considered as a coarse scale version of x_{l+1}. The final sample $x = x_L$ is constructed iteratively, as the dimension $d_1 < d_2 < \ldots < d_L = d$ grows up, see Figure 1. The overall map T is invertible from z to x.

We train MsIGN by minimizing the Jeffreys divergence $D_{\mathrm{J}}(p_\theta \| q)$, defined by (Jeffreys et al., 1973) as

$$D_{\mathrm{J}}(p_\theta \| q) = D_{\mathrm{KL}}(p_\theta \| q) + D_{\mathrm{KL}}(q \| p_\theta)$$
$$= \mathbb{E}_{p_\theta}[\log(p_\theta/q)] + \mathbb{E}_q[\log(q/p_\theta)]. \quad (4)$$

Jeffreys divergence removes bad local minima of single-sided Kullback-Leibler (KL) divergence to avoid mode missing. We build its unbiased estimation by importance sampling, with the output of the prior conditioning layer as proposal distribution. Furthermore, MsIGN is trained in a multi-stage manner, from coarse to fine scale. At stage l, we train $\{F_{l'} : l' \leq l\}$ so that x_l approximates the posterior at its scale, while PC_l are pre-computed and fixed. Each stage provides a good proposal distribution for the importance sampling at the next stage thanks to the multiscale property.
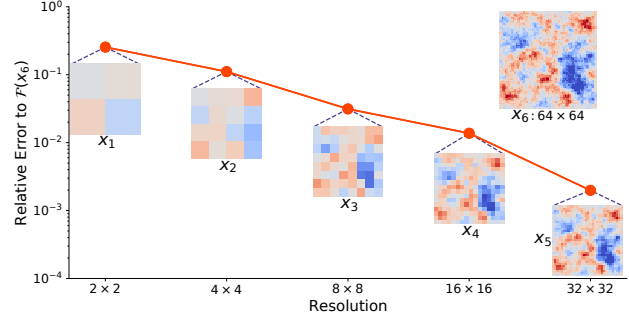
Contribution We claim four contributions in this work. First, we propose a Multiscale Invertible Generative Network (MsIGN) with a novel prior conditioning layer that can generate samples from a coarse-to-fine manner. Second, MsIGN allows multi-stage training to minimize the Jeffreys divergence, which helps avoid mode collapse in high-d problems. Third, when applied to two Bayesian inverse problems, MsIGN clearly captures multiple modes in the high-d posterior and approximates the posterior accurately, demonstrating its superior performance over previous methods. Fourth, we also apply MsIGN to image synthesis tasks, where it achieves superior performance in bits-per-dimension among baseline models. MsIGN also yields great interpret-ability of its neurons in intermediate layers.

We introduce the theoretical motivation in Section 2, and give detailed introduction of the network structure of our MsIGN in Section 3, while its training strategy is described in Section 4. Then we review related work, and provide numerical studies in Section 5 and 6 respectively.

2. Theoretical Motivation

Let $A \in \mathbb{R}^{d_c \times d}$ be a linear operator that downsamples x to its coarse-scale low-d version $x_c = Ax \in \mathbb{R}^{d_c}$ with $d_c < d$. For example, A can be the average pooling operator with kernel size 2 and stride 2 which downsamples x to $1/4$ of its original dimensions.

Multiscale structure In many high-d Bayesian inference problems, the observation y relies more on global, coarse-scale structure than local, fine-scale structure of x. This multiscale structure can be described as

$$\mathcal{F}(x) \approx \mathcal{F}(BAx), \quad \forall x \in \mathbb{R}^d, \quad (5)$$

where $A \in \mathbb{R}^{d_c \times d}$ is the downsample operator that com-

press x to a coarse-scale version Ax, and $B \in \mathbb{R}^{d \times d_c}$ transforms the coarse-scale low-d Ax to a valid system input for $\mathcal{F}$. For example, B can be the nearest-neighbor upsample operator such that BAx has the same size as x, but only contains its coarse-scale information. The relation (5) arises frequently when x has some spatial or temporal structure, see an example in Figure 2.

Scale decoupling   Let $x_c = Ax$ be the coarse-scale variable. Like in (2), the coarse-scale likelihood is defined as

$$\mathcal{L}_c(y|x_c) = \mathcal{N}(y - \mathcal{F}(Bx_c); 0, \Gamma), \qquad (6)$$

and we expect $\mathcal{L}_c(y|Ax) \approx \mathcal{L}(y|x)$ due to (5). On the other hand, let ρ_c be the probability density of $x_c = Ax$ when $x \sim \rho$, which is the coarse-scale prior, the conditional probability rule suggests that $\rho(x|x_c) = \rho(x|Ax = x_c) = \rho(x)/\rho_c(x_c)$, which is equivalent to $\rho(x) = \rho_c(x_c)\rho(x|x_c)$.

With the likelihood approximation and the prior decoupling, the posterior q admits the following scale decoupling:

$$\begin{aligned} q(x) &= \frac{1}{Z}\rho(x)\mathcal{L}(y|x) \approx \frac{1}{Z}\rho(x)\mathcal{L}_c(y|x_c) \\ &= \frac{1}{Z}\rho_c(x_c)\rho(x|x_c)\mathcal{L}_c(y|x_c) \qquad (7) \\ &= \frac{Z_c}{Z}\rho(x|x_c)q_c(x_c) \propto \frac{1}{\tilde{Z}}\rho(x|x_c)q_c(x_c) := \tilde{q}(x), \end{aligned}$$

where $q_c(x_c) := \frac{1}{Z_c}\rho_c(x_c)\mathcal{L}_c(y|x_c)$ is the coarse-scale posterior analog to (1), and $\tilde{q}(x) := \frac{1}{\tilde{Z}}\rho(x|x_c)q_c(x_c)$ is a distribution to approximate q, with normalizing constants $Z_c, \tilde{Z}$.

3. Network Architecture

The key observation (7) is essentially

$$\underbrace{q(x) \approx \tilde{q}(x)}_{(iii)} = \underbrace{q_c(x_c)}_{(i)} \underbrace{\rho(x|x_c)}_{(ii)}, \qquad (8)$$

where $\approx$ and $=$ are up to some multiplicative constant. It suggests a three-step way to sample from q:

(i) generate a sample x_c from q_c;
(ii) sample $\tilde{x}$ from $\rho(\cdot|x_c)$;
(iii) further modify $\tilde{x}$ to x to better approximate q.

We design a prior conditioning layer PC to sample $\tilde{x}$ from $\rho(\cdot|x_c)$ for (ii), and an invertible flow F that modifies $\tilde{x}$ for (iii). To obtain x_c from q_c in (i), we apply the above procedure recursively until the dimension of the coarsest scale is small enough so that q_c can be easily sampled by a standard method. As an example of this three-step sampling strategy, in the image synthesis task, a high-resolution image x can be *approximated* by $\tilde{x}$, the upsampled image of its low-resolution version x_c superimposed with random noise according to the prior ρ, which will be specified in Section 6.2 for this task. Needless to say, $\tilde{x}$ needs further modification to achieve good quality in high resolution.

Prior conditioning   We feed the coarse-scale sample x_c together with some random seed $z \in \mathbb{R}^{d - d_c}$ to the prior conditioning layer PC to sample from the conditional distribution $\rho(\cdot|x_c)$: $x = PC(x_c, z)$. The conditional sample x should satisfy the constrain $Ax = x_c$. We further require the layer PC to be invertible between x and (x_c, z) to maintain the invertiblity of our overall network. Since PC depends only on the prior distribution ρ and downsampling operator A, it can be pre-computed *regardless* of the likelihood $\mathcal{L}$. In fact, when the prior is a Gaussian, the prior conditional distribution is still a Gaussian and the prior conditioning layer PC admits a closed form:

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).

We leave the proof in Appendix A. When the prior is non-Gaussian, the prior conditioning layer PC still exists with invertibility guarantee, but it is now *nonlinear*. In this case, we can pre-train an invertible network to approximate the conditional sampling process. Once PC is pre-computed, its parameters are fixed in the training stage.

Invertible flow   The invertible flow F is a parametric invertible map that modifies the sample $\tilde{x}$ from the prior conditioning layer to a sample of the target q, in other words, it modifies the distribution $\tilde{q}$ in (8) to the target q. In our experiments in Section 6, we utilize the invertible block of Glow (Kingma & Dhariwal, 2018), which consists of actnorm, invertible 1×1 convolution, and affine coupling layer, and stack several such blocks as the inverse flow F in MsIGN. The approximation (8) also suggests that F be initialized as an identity map in training, see Section 4.

Recursive design   To initialize our sampling strategy (8) with a sample x_c from the coarse-scale posterior q_c, we recursively apply our strategy until the dimension of the coarsest-scale is small enough. Let L be the number of recursion, also called scales in the following context. Let $x_l \in \mathbb{R}^{d_l}$ be the variable at scale l ($1 \leq l \leq L$), whose distribution is the l-th scale posterior q_l analog to the q_c in Section 2 and $q_L = q$. The problem dimension keeps increasing as l goes up: $d_1 < d_2 < \ldots < d_L = d$. Details of constructions at scale l can be found in Appendix D.

Our network structure is shown in (3) and Figure 1, with $z_1 \in \mathbb{R}^{d_1}$ and $z_l \in \mathbb{R}^{d_l - d_{l-1}}$ ($2 \leq l \leq L$) be the random

seed drawn from standard Gaussian at each scale. At scale l ($2 \leq l \leq L$), a prior conditioning layer PC_l randomly upsamples $x_{l-1} \in \mathbb{R}^{d_{l-1}}$, taken from q_{l-1} approximately, to $\tilde{x}_l \in \mathbb{R}^{d_l}$, and an invertible flow F_l modifies $\tilde{x}_l$ to x_l to approximate q_l. At scale $l = 1$, we directly learn an invertible flow F_1 that transports $z_1 \sim \mathcal{N}(0, I)$ to $x_1 \sim q_1$ since the problem dimension is small enough to allow efficient application of standard methods.

Write the overall random seed $z \in \mathbb{R}^d$ as a concatenation of $(z_1, z_2, \ldots, z_L)$, and write θ as the parameters in MsIGN. The overall network of MsIGN parameterizes a map $T(\cdot; \theta)$ such that samples are generated by $x = T(z; \theta)$. Let p_z, p_θ be the density of z, x respectively. Our design also allows the invertible mapping $z = T^{-1}(x; \theta)$, so by the change-of-variable formula the density of p_θ is given by

$$p_\theta(x) = p_z(T^{-1}(x; \theta)) |\det \mathrm{J}_x T^{-1}(x; \theta)|, \qquad (9)$$

where $\mathrm{J}_x T^{-1}$ is the Jacobian of T^{-1} with respect to x.

We also remark that when certain bound needs to be enforced on the output, we can append element-wise output activations at the end of MsIGN. For example, image synthesis can use the sigmoid function so that pixel values lie in $[0, 1]$. Such activations should be bijective to keep the invertible relation between random seed z to the sample x.

4. Training Strategy

We learn network parameter θ by solving the optimization $\min_\theta D_\mathrm{J}(p_\theta \| q)$. Since prior conditioning layers PC_l, for $2 \leq l \leq L$, are pre-computed and fixed, trainable parameter θ only comes from the invertible flows F_l, for $1 \leq l \leq L$.

Jeffreys divergence While the KL divergence is widely used as the training objective for its easiness to compute, its landscape could admit local minima that don't favor the optimization. In fact, (Nielsen & Nock, 2009) suggests that $D_\mathrm{KL}(p_\theta \| q)$ is zero-forcing, meaning that it enforces p_θ be small whenever q is small. As a consequence, mode missing can still be a local minimum, see Appendix B. Therefore, we turn to the Jeffreys divergence (4) which significantly penalizes mode missing and can remove such local minima.

Estimating the Jeffreys divergence requires computing an expectation with respect to the target q, which is normally prohibited. Since MsIGN constructs a good approximation $\tilde{q}$ to q, we do importance sampling with $\tilde{q}$ as the proposal distribution for the Jeffreys divergence and its derivative:

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,*

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

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

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

Detailed derivation is left in Appendix C. With the derivative given above, we optimize the Jeffreys divergence by stochastic gradient descent. We remark that $\partial \log p_\theta / \partial \theta$ is available by the backward propagation of MsIGN, and $\tilde{q}$ comes from coarser scale model in multi-stage training.

Multi-stage training The multiscale design of MsIGN enables a coarse-to-fine multi-stage training. At stage l, we target at capturing the posterior q_l at scale l, and only train invertible flows before or at this scale: $F_{l'}$, with $l' \leq l$.

Additionally, at stage l, we initialize F_l as the identity map, and $F_{l'}$, with $l' < l$, as the trained model at stage $l - 1$. The reason is implied by (8), where now q, q_c represents q_l, q_{l-1} respectively. The stage $l - 1$ model provides good approximation to q_{l-1}, and together with PC_l it provides a good approximation $\tilde{q}_l$ to q_l. Thus, setting F_l as the identity map will give a good initialization to MsIGN in training. Our experiment shows such multi-stage strategy significantly stabilizes training and improves final performance.

We conclude the training of MsIGN in Algorithm 1.

Algorithm 1 Train MsIGN by optimizing the Jeffreys divergence in a multi-stage manner

---

Output: $\theta = (\theta_1, \ldots, \theta_L)$, θ_l are parameters in F_l.
 1: Pre-compute and fix all prior conditioning layers PC_l.
 2: Learn θ_1 by standard methods such that sample $x_1 = F_1(z_1)$ approximates q_1.
 3: **for** $l = 2$ **to** L **do**
 4:     Initialize θ_l so that F_l is an identity map.
 5:     Concatenate last-stage model with PC_l as $\tilde{q}_l$.
 6:     With q_l as the target q, $\tilde{q}_l$ as the proposal $\tilde{q}$ in (11), compute the gradient using Monte Carlo.
 7:     Learn θ_l by stochastic gradient descent.
 8: **end for**

---

5. Related Work

Invertible generative models (Deco & Brauer, 1995) are powerful exact likelihood models with efficient sampling and inference. They have achieved great success in natural image synthesis, see, e.g., (Dinh et al., 2016; Kingma & Dhariwal, 2018), and variational inference in providing a tight evidence lower bound, see, e.g, (Rezende & Mohamed, 2015). In this paper, our proposed MsIGN utilizes the invertible block in Glow (Kingma & Dhariwal, 2018) as building piece for the invertible flow at each scale. The Glow block

can be replaced by any other invertible blocks, without any algorithmic changes. Different from Glow, MsIGN adopts a novel multiscale structure such that different scales can be trained separately, making training much more stable. Besides, the multiscale idea enables better explain-ability of its hidden neurons. Invertible generative models like (Dinh et al., 2016; Kingma & Dhariwal, 2018; Ardizzone et al., 2019) adopted a similar multiscale idea, but their multiscale strategy is not in a "spatial" sense: the intermediate neurons are not semantically interpret-able as shown in Figure 7. The multiscale idea is also used in generative adversarial networks (GANs), as in (Denton et al., 2015; Odena et al., 2017; Karras et al., 2017; Xu et al., 2018). But the lack of invertibility in these models makes it difficult for them to apply to Bayesian inference problems.

Different from the image synthesis task where large amount of samples from target distribution are available, in Bayesian inference problems only an unnormalized density is available and i.i.d. samples from the posterior are the target. This main goal of this paper is to train MsIGN to approximate certain high-d Bayesian posteriors. Various kinds of parametric distributions have been proposed to approximate posteriors before, such as polynomials (El Moselhy & Marzouk, 2012; Parno et al., 2016; Matthies et al., 2016; Spantini et al., 2018), non-invertible generative networks (Feng et al., 2017; Hou et al., 2019), invertible networks (Rezende & Mohamed, 2015; Ardizzone et al., 2018; Kruse et al., 2019) and certain implicit maps (Chorin & Tu, 2009; Morzfeld et al., 2012). Generative modeling approach has the advantage that i.i.d. samples can be efficiently obtained by evaluating the model in the inference stage. However, due to the tricky non-convex optimization problem, this approach for both invertible (Chorin & Tu, 2009; Kruse et al., 2019) and non-invertible (Hou et al., 2019) generative models becomes increasingly challenging as the dimension grows. To overcome this difficulty, we propose to minimize the Jeffreys divergence, which has fewer local minima and better landscape compared with the commonly-used KL divergence, and to train MsIGN in a coarse-to-fine manner.

Other than the generative modeling, various Markov Chain Monte Carlo (MCMC) methods have been the most popular in Bayesian inference, see, e.g., (Beskos et al., 2008; Neal et al., 2011; Welling & Teh, 2011; Chen et al., 2014; 2015; Cui et al., 2016). Particle-optimization-based sampling is a recently developed effective sampling technique with Stein variational gradient descent (SVGD) (Liu & Wang, 2016)) and many related works, e.g., (Liu, 2017; Liu & Zhu, 2018; Chen et al., 2018; 2019a; Chen & Ghattas, 2020). The intrinsic difficulty of Bayesian inference displays itself as highly correlated samples, leading to undesired low sample efficiency, especially in high-d cases. The multiscale structure and multi-stage strategy proposed in this paper can also benefit these particle-based methods, as we can observe that

they benefit the amortized-SVGD (Feng et al., 2017; Hou et al., 2019) in Section 6.1.3. We leave more discussion about the related work in Appendix E.

6. Experiment

We study two high-d Bayesian inverse problems (BIPs) in Section 6.1 as test beds for distribution approximation and multi-mode capture. We also apply MsIGN to the image synthesis task to benchmark with flow-based generative models and demonstrate its interpret-ability in Section 6.2.

In both experiments, we utilize average pooling with kernel size 2 and stride 2 as the operator A, and stack several of the invertible block in Glow (Kingma & Dhariwal, 2018) to build our invertible flow F, as mentioned in Section 3.

6.1. Bayesian Inverse Problems

We study two nonlinear and high-d BIPs known to have at least two equally important modes in this section: one with true samples available as reference in Section 6.1.1; one without true samples but close to real-world applications of subsurface flow in fluid dynamics in Section 6.1.2. In both problems, sample x of the target posterior q is a vector on a 2-D uniform 64×64 lattice, which means the problem dimension d is 4096. Every x is equivalent to a piece-wise constant function on the unit disk: $x(s)$ for $s \in \Omega = [0, 1]^2$, and we don't distinguish between them thereafter. We equip x with a Gaussian prior $\mathcal{N}(0, \Sigma)$ with Σ as the discretization of $\beta^2 (-\Delta)^{-1-\alpha}$, where α, β are parameters.

To make the high-d inference more challenging, the target q is built to be multi-modal by leveraging spatial symmetry. Combining properties of the prior defined above and the likelihood defined afterwards, the posterior is innately mirror-symmetric: $q(x) = q(x')$ if $x(s_1, s_2) = x'(s_1, 1 - s_2)$ for any $s = (s_1, s_2) \in \Omega$. Furthermore, we carefully select the prior and the likelihood so that q has at least two modes. They are mirror-symmetric to each other and possess equal importance, see discussion in Appendix F.

We train MsIGN following Algorithm 1 with $L = 6$ scales. The problem dimension at scale l is $d_l = 2^l * 2^l = 4^l$. We compare MsIGN with representative approaches for high-d BIPs: Hamiltonian Monte Carlo (short as HMC) (Neal et al., 2011), SVGD (Liu & Wang, 2016), amortized-SVGD (short as A-SVGD) (Feng et al., 2017), and projected SVGD (short as pSVGD) (Chen & Ghattas, 2020). Since simulating the forward map $\mathcal{F}$ dominates the training time cost, especially in Section 6.1.2 (more than 75% of the wall clock time), we set a budget for the **n**umber of **f**orward **s**imulations (nFSs) for all methods for fair comparison in computational cost. For both problems, we aim at generating 2500 samples from the target. More details of experimental setting and additional numerical results can be found in Appendix F.
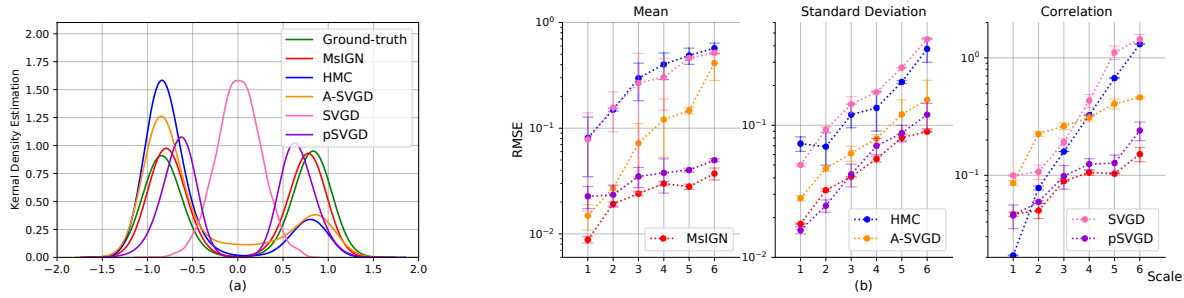
Figure 3. Results in the synthetic BIP. (a): Sample marginal distribution along the critical direction w_{k*}. MsIGN is more robust in capturing both modes and close to ground-truth. (b): Root mean square error (RMSE) and its 95% confidence interval of three independent experiments. MsIGN is more accurate in distribution approximation, especially at finer scale when the problem dimension is high. The margin is statistical significant as shown by the confidence interval.
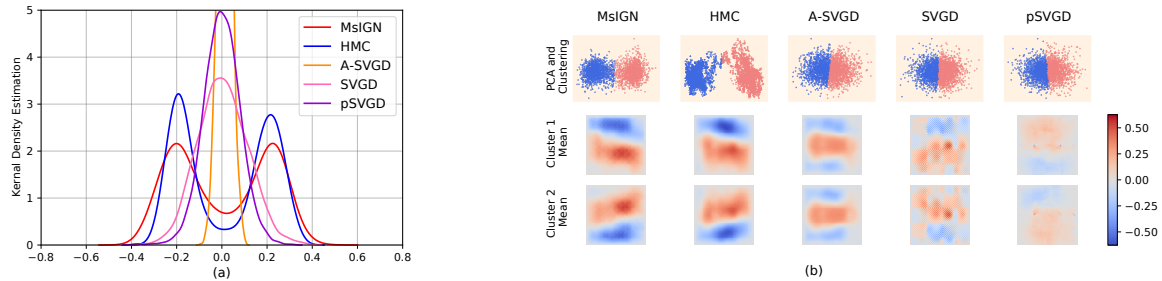

Figure 4. Results in the elliptic BIP. (a): Sample marginal distribution along the critical direction. MsIGN and HMC capture two modes in this marginal distribution, but the others fail. (b): Clustering result of samples. Samples of MsIGN are more balanced between two modes. The similarity of the cluster means of MsIGN and HMC implies that they both are likely to capture the correct modes.
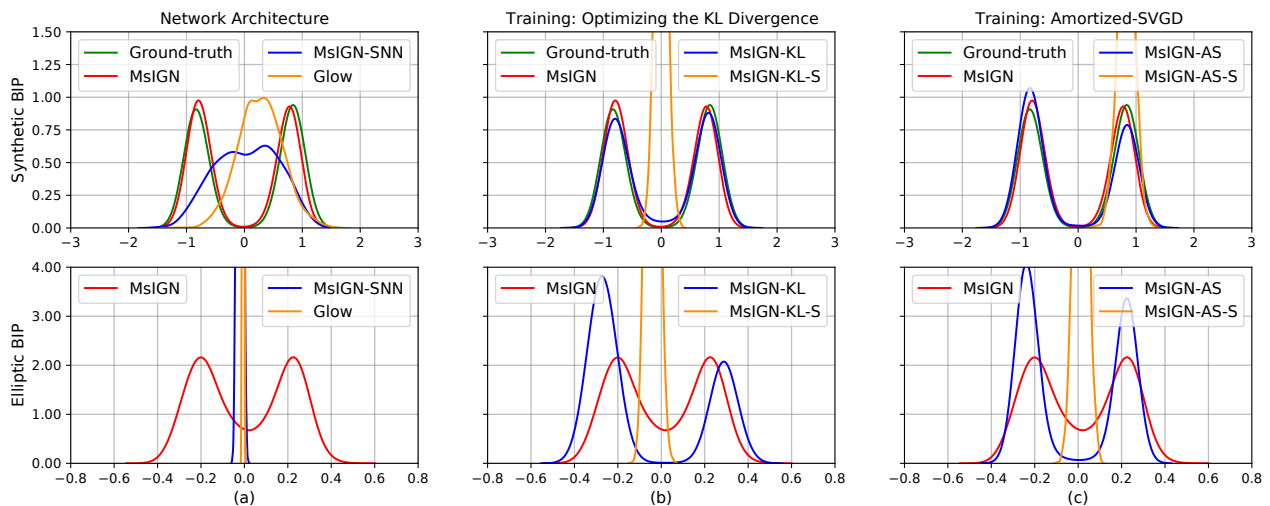

Figure 5. Ablation study of the network architecture and training strategy. "MsIGN" means our default setting: training MsIGN with Jeffreys divergence and multi-stage strategy. Other models are named by a base model (MsIGN or Glow), followed by strings indicating its difference from the default setting. For example, "MsIGN-KL" refers to training MsIGN by optimizing the KL divergence in a multi-stage way, while "MsIGN-AS-S" means training MsIGN using the amortizied-SVGD algorithm in a single-stage way. See Section 6.1.3 and Appendix F for thorough discussion.

6.1.1. SYNTHETIC BAYESIAN INVERSE PROBLEMS

This problem allows access to ground-truth samples so the comparison is clear and solid. We set $\mathcal{F}(x) = \langle \varphi, x \rangle^2 = \left(\int_\Omega \varphi(s) x(s) \mathrm{d}s \right)^2$, where $\varphi(s) = \sin(\pi s_1) \sin(2\pi s_2)$. Together with the prior, our posterior can be factorized into 1-D sub-distributions, namely $q(x) = \prod_{k=1}^d q_k(\langle w_k, x \rangle)$ for some orthonormal basis $\{w_k\}_{k=1}^d$ of $\mathbb{R}^d$. This property gives us access to true samples via inversion cumulative function sampling along each direction w_k. Furthermore, these 1-D sub-distributions are all single modal except that there's one, which is the marginal distribution q_{k^*} along direction w_{k^*}, with two symmetric modes. This confirms our construction of two equally important modes. The computation budget is fixed at 8×10^5 nFSs.

Multi-mode capture　To visualize mode capture, we plot the marginal distribution of generated samples along the critical direction w_{k^*}, which is the source of double-modality. From Figure 3(a), MsIGN gives the best mode capture among our baselines in this $d = 4096$ problem.

Distribution approximation　We use the root mean square errors (RMSE) of sample mean, standard deviation, and correlation, with the Jeffreys divergence to measure distribution approximation. We compare the sample mean, variance and correlation with theoretical ground-truths, and report the averaged RMSE of all sub-distributions at all scales in Figure 3(b). Additionally, since MsIGN and A-SVGD also gives density estimation, we report the Monte Carlo estimates of the Jeffreys divergence (4) with the target posterior in Table 1. We can see that MsIGN has superior accuracy in approximating the target distribution.

Table 1. Jeffreys divergence $D_\mathrm{J}(p_\theta | q)$ in three independent runs.

MODEL	MsIGN	A-SVGD
ERROR	56.77±0.15	3372±21

6.1.2. ELLIPTIC BAYESIAN INVERSE PROBLEMS

This problem is a benchmark problem for high-d inference from geophysics and fluid dynamics (Iglesias et al., 2014; Cui et al., 2016). The forward map $\mathcal{F}(x) = \mathcal{O} \circ \mathcal{S}(x)$, 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:

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

where f and φ_k are given and fixed. The map $\mathcal{S}$ is solved by the finite element method with mesh size $1/64$. Unfortunately, there is no known access to true samples of q. But

the trick of symmetry introduced in Section 6.1 guarantees at least two equally important modes in the posterior. We put a 5×10^5-nFS budget on our computation cost.

Multi-mode capture　Due to the lack of true samples, we check the marginal distribution of the posterior along eigenvectors of the prior, and pick a particular one to show if we capture double modes in Figure 4(a). We also confirm the capture of multiple modes by embedding samples by Principle Component Analysis (PCA) to a 2-D space. We report the clustering (by K-means) result and means of each cluster in Figure 4(b), where we can see that MsIGN has a more balanced capture of the symmetric posterior than HMC, while others fail to detect two modes. We refer readers to Appendix F for more comprehensive study of mode capture ability of different methods.

6.1.3. ABLATION STUDY

We run extensive experiments to study the effectiveness of the network architecture and training strategy of MsIGN. Detailed setting and extra results are left in Appendix F.

Network architecture　We replace the prior conditioning layer PC by two direct alternatives: a stochastic nearest-neighbor upsample layer independent of the prior (model named "MsIGN-SNN"), or the split and squeeze layer in Glow design (it resumes Glow model, so we call it "Glow"). Figure 5(a) shows that the prior conditioning layer design is crucial to the performance of MsIGN on both problems, because neither alternatives has a successful mode capture.

Training strategy　We study the effectiveness of the Jeffreys divergence objective and multi-stage training. We substitute our optimizing of the Jeffreys divergence in Algorithm 1 by optimizing the KL divergence (marked with a suffix "-KL"), or using the amortized-SVGD (A-SVGD) method (marked with a suffix "-AS"). We also switch between multi-stage (the default, no extra suffix) or single-stage training (marked with an additional suffix "-S"). We remark that single-stage training using Jeffreys divergence is infeasible because of the difficulty to estimate $D_\mathrm{KL}(q\|p_\theta)$. Figure 5(b) and (c) show that, all models trained in the single-stage manner ("MsIGN-KL-S", "MsIGN-AS-S") will face mode collapse. We observe that our multi-stage training strategy can benefit training with other objectives, see "MsIGN-KL" and "MsIGN-AS". We also notice that the Jeffreys divergence leads to a more balanced samples for these symmetric problems, especially for the complicated elliptic BIP.

6.2. Image Synthesis

The transport map approach to Bayesian inference has two critical difficulties: the model capacity and the training effectiveness. Since the distribution of images is complicated

Table 2. Bits-per-dimension value comparison with baseline models of flow-based generative networks. All models in this table do not use the "variational dequantization" technique in (Ho et al., 2019). *: Score obtained by our own reproducing experiment.

MODEL	MNIST	CIFAR-10	CELEBA 64	IMAGENET 32	IMAGENET 64
REAL NVP(DINH ET AL., 2016)	1.06	3.49	3.02	4.28	3.98
GLOW(KINGMA & DHARIWAL, 2018)	1.05	3.35	2.20*	4.09	3.81
FFJORD(GRATHWOHL ET AL., 2018)	0.99	3.40	–	–	–
FLOW++(HO ET AL., 2019)	–	3.29	–	–	–
I-RESNET(BEHRMANN ET AL., 2019)	1.05	3.45	–	–	–
RESIDUAL FLOW(CHEN ET AL., 2019B)	0.97	**3.28**	–	**4.01**	3.76
MsIGN (OURS)	**0.93**	**3.28**	**2.15**	4.03	**3.73**

Figure 6. Left: Synthesized CelebA images of resolution 64×64 with temperature 0.9. Right: Linear interpolation in latent space shows MsIGN's parameterization of natural image manifold is semantically meaningful. For images x_1, x_2 at the left and right ends, we retrieve their latent feature by $z_i = T^{-1}(x_i; \theta), i = 1, 2$, and then interpolate between them by $T((1 - \lambda)z_1 + \lambda z_2; \theta)$ for $\lambda = 0.2, 0.4, 0.6, 0.8$.
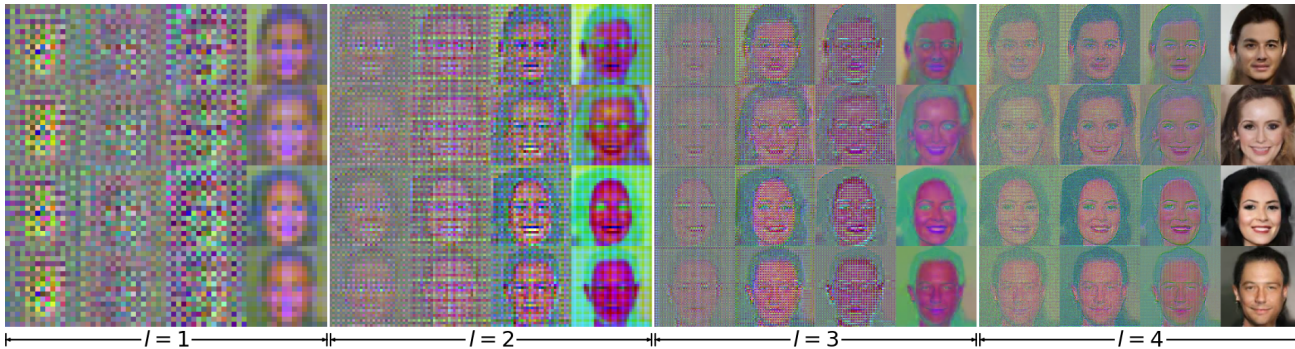

Figure 7. Visualization of internal activation shows the interpret-ability of MsIGN hidden neurons. This MsIGN model has $L = 4$ scales. From left to right, we take 4 snapshots (head, two trisection points, and tail) in each invertible flow F_l for $l = 1, 2, 3, 4$, to show how MsIGN progressively generates new samples from low to high resolution.

and multi-modal, we present the image synthesis task result to show case the model capacity of the MsIGN. It also provides a good test bed for our MsIGN to benchmark with other flow-based generative networks.

We train MsIGN by maximum likelihood estimation. We assume a simple Gaussian prior ρ for natural images, whose covariance is a scalar matrix learned from the data. See Appendix G for experimental details and additional results.

We report the bits-per-dimension value comparison with baseline models in Table 2. Our MsIGN is superior in number and also is more efficient in parameter size: for example, MsIGN uses 24.4% fewer parameters than Glow for CelebA 64, and uses 37.4% fewer parameters than Residual Flow for ImageNet 64.

Figure 6 shows synthesized images of MsIGN from CelebA data set, and linear interpolation of real images in the latent feature space. In Figure 7, we visualize internal activations

at checkpoints in the invertible flow at different scales which demonstrate the interpret-ability of MsIGN.

7. Conclusion

For high-dimensional Bayesian inference problems with multiscale structure, we propose Multiscale Invertible Generative Networks (MsIGN) and associated training algorithms to approximate the posterior. We demonstrate the potential of this approach in high-dimensional (up to 4096) Bayesian inference problems, leaving several important directions as future work. The network architecture also achieves superior performance over benchmarks in various image synthesis tasks. We plan to apply this methodology to other Bayesian inference problems, e.g., Bayesian deep learning with multiscale structure in model width or depth (e.g., (Chang et al., 2017; Haber et al., 2018)) and data assimilation problem with multiscale structure in the temporal variation (e.g., (Giles, 2008)).

Acknowledgements

References

Ardizzone, L., Kruse, J., Wirkert, S., Rahner, D., Pellegrini, E. W., Klessen, R. S., Maier-Hein, L., Rother, C., and Köthe, U. Analyzing inverse problems with invertible neural networks. *arXiv preprint arXiv:1808.04730*, 2018.

Ardizzone, L., Lüth, C., Kruse, J., Rother, C., and Köthe, U. Guided image generation with conditional invertible neural networks. *arXiv preprint arXiv:1907.02392*, 2019.

Behrmann, J., Grathwohl, W., Chen, R. T., Duvenaud, D., and Jacobsen, J.-H. Invertible residual networks. In *International Conference on Machine Learning*, pp. 573–582, 2019.

Beskos, A., Roberts, G., Stuart, A., and Voss, J. Mcmc methods for diffusion bridges. *Stochastics and Dynamics*, 8(03):319–350, 2008.

Chang, B., Meng, L., Haber, E., Tung, F., and Begert, D. Multi-level residual networks from dynamical systems view. *arXiv preprint arXiv:1710.10348*, 2017.

Chen, C., Ding, N., and Carin, L. On the convergence of stochastic gradient mcmc algorithms with high-order integrators. In *Advances in Neural Information Processing Systems*, pp. 2278–2286, 2015.

Chen, C., Zhang, R., Wang, W., Li, B., and Chen, L. A unified particle-optimization framework for scalable bayesian sampling. *arXiv preprint arXiv:1805.11659*, 2018.

Chen, P. and Ghattas, O. Projected stein variational gradient descent. *arXiv preprint arXiv:2002.03469*, 2020.

Chen, P., Wu, K., Chen, J., O'Leary-Roseberry, T., and Ghattas, O. Projected stein variational newton: A fast and scalable bayesian inference method in high dimensions. In *Advances in Neural Information Processing Systems*, pp. 15104–15113, 2019a.

Chen, T., Fox, E., and Guestrin, C. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pp. 1683–1691, 2014.

Chen, T. Q., Behrmann, J., Duvenaud, D. K., and Jacobsen, J.-H. Residual flows for invertible generative modeling. In *Advances in Neural Information Processing Systems*, pp. 9913–9923, 2019b.

Chorin, A. J. and Tu, X. Implicit sampling for particle filters. *Proceedings of the National Academy of Sciences*, 106 (41):17249–17254, 2009.

Cui, T., Law, K. J., and Marzouk, Y. M. Dimension-independent likelihood-informed mcmc. *Journal of Computational Physics*, 304:109–137, 2016.

Deco, G. and Brauer, W. Nonlinear higher-order statistical decorrelation by volume-conserving neural architectures. *Neural Networks*, 8(4):525–535, 1995.

Denton, E. L., Chintala, S., szlam, a., and Fergus, R. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems*, volume 28, pp. 1486–1494, 2015.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

El Moselhy, T. A. and Marzouk, Y. M. Bayesian inference with optimal maps. *Journal of Computational Physics*, 231(23):7815–7850, 2012.

Feng, Y., Wang, D., and Liu, Q. Learning to draw samples with amortized stein variational gradient descent. *arXiv preprint arXiv:1707.06626*, 2017.

Giles, M. B. Multilevel monte carlo path simulation. *Operations research*, 56(3):607–617, 2008.

Grathwohl, W., Chen, R. T., Bettencourt, J., Sutskever, I., and Duvenaud, D. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

Haber, E., Ruthotto, L., Holtham, E., and Jun, S.-H. Learning across scales—multiscale methods for convolution neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Ho, J., Chen, X., Srinivas, A., Duan, Y., and Abbeel, P. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International Conference on Machine Learning*, pp. 2722–2730. PMLR, 2019.

Hou, T. Y., Lam, K. C., Zhang, P., and Zhang, S. Solving bayesian inverse problems from the perspective of deep generative networks. *Computational Mechanics*, 64(2): 395–408, 2019.

Iglesias, M. A., Lin, K., and Stuart, A. M. Well-posed bayesian geometric inverse problems arising in subsurface flow. *Inverse Problems*, 30(11):114001, 2014.

Jeffreys, H. et al. *Scientific inference*. Cambridge University Press, 1973.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Kruse, J., Ardizzone, L., Rother, C., and Köthe, U. Benchmarking invertible architectures on inverse problems. In *Thirty-sixth International Conference on Machine Learning*, 2019.

Liu, C. and Zhu, J. Riemannian stein variational gradient descent for bayesian inference. In *Thirty-second aaai conference on artificial intelligence*, 2018.

Liu, Q. Stein variational gradient descent as gradient flow. In *Advances in neural information processing systems*, pp. 3115–3123, 2017.

Liu, Q. and Wang, D. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pp. 2378–2386, 2016.

Matthies, H. G., Zander, E., Rosić, B. V., Litvinenko, A., and Pajonk, O. Inverse problems in a bayesian setting. In *Computational Methods for Solids and Fluids*, pp. 245–286. Springer, 2016.

Morzfeld, M., Tu, X., Atkins, E., and Chorin, A. J. A random map implementation of implicit filters. *Journal of Computational Physics*, 231(4):2049–2066, 2012.

Neal, R. M. et al. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11), 2011.

Nielsen, F. and Nock, R. Sided and symmetrized bregman centroids. *IEEE transactions on Information Theory*, 55 (6):2882–2904, 2009.

Odena, A., Olah, C., and Shlens, J. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pp. 2642–2651, 2017.

Parno, M., Moselhy, T., and Marzouk, Y. A multiscale strategy for bayesian inference using transport maps. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1): 1160–1190, 2016.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows, 2015.

Spantini, A., Solonen, A., Cui, T., Martin, J., Tenorio, L., and Marzouk, Y. Optimal low-rank approximations of bayesian linear inverse problems. *SIAM Journal on Scientific Computing*, 37(6):A2451–A2487, 2015.

Spantini, A., Bigoni, D., and Marzouk, Y. Inference via low-dimensional couplings. *The Journal of Machine Learning Research*, 19(1):2639–2709, 2018.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., and He, X. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1316–1324, 2018.

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.* $\mathrm{rank}(A) = d_c$, *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

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

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. $\square$

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

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

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, $x - U^c A x = (I_d - U^c A)x$ is independent from Ax.

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

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

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

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

Plug in the definition of U^c, we find

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

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

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

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

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

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,

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

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

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

The first part holds true because

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

as shown in (12). The second part holds because, when plug

in $x = U^c x_c + Wz$, we notice that

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

and similarly

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

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. $\square$

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 $D_{\mathrm{KL}}(p\|q) = \mathbb{E}_p[\log(p/q)] = \int p(x)(\log p(x) - \log q(x))\,\mathrm{d}x$ 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, $D_{\mathrm{KL}}(q\|p) = \mathbb{E}_q[\log(q/p)] = \int q(x)(\log q(x) - \log p(x))\,\mathrm{d}x$ becomes small as long as q is close to zero wherever p is close to zero.
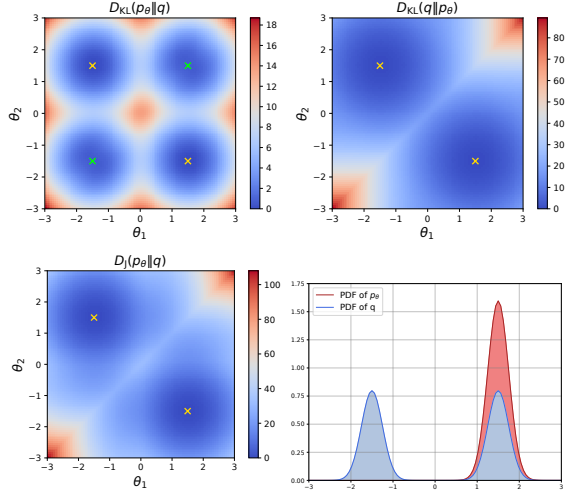
Figure 8. Landscape of $D_{\mathrm{KL}}(p_\theta\|q)$ (upper left), $D_{\mathrm{KL}}(q\|p_\theta)$ (upper right), and $D_{\mathrm{J}}(p_\theta\|q)$ (lower left), density function of p_θ and q when they reach one of the local minima (lower right). We mark the global minima (ground-truth) by golden cross, 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). $\square$

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, $\mathcal{F}(\hat{B}_l x_l) \approx$
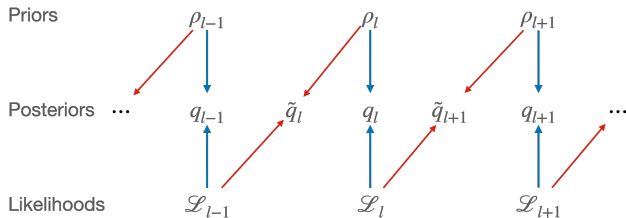

Figure 9. Conceptual diagram of the definitions. Arrows mean that "contribute to the definition of". We further remark that, (i) $\tilde{q}_l$ is the upsampling of q_{l-1} by $\rho_l(x_l|x_{l-1})$, because ρ_l is the upsampling of q_{l-1} by $\rho_l(x_l|x_{l-1})$, and (ii) q_l can be well approximated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

$\mathcal{F}(\hat{B}_{l-1} A_l x_l)$, 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.

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

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

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 ap-

proximate 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

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

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.

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:

$$\mathcal{O}(u) = \begin{bmatrix} \int_\Omega \varphi_1(s)u(s)\mathrm{d}s & \dots & \int_\Omega \varphi_m(s)u(s)\mathrm{d}s \end{bmatrix}^T.$$

The force term f of the elliptic PDE is set as

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

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-
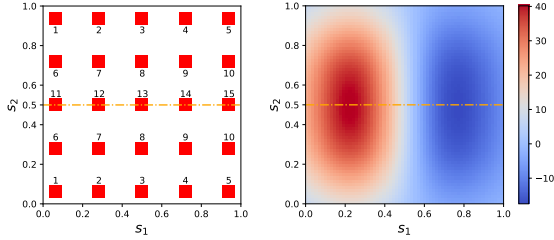
Figure 10. Left: 15 measurement functions in $\mathcal{O}$. Here we plotted non-zero patches of φ_k, $k = 1, \ldots, 15$, with k labeled next to them. φ_k has a constant non-zero value on its patch(es) and is zero else where. The constant value here is chosen so that we have $\|\varphi_k\|_{L^2} = 1$. 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 ($1 \le l \le L$) 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 $D_{\mathrm{J}}(p_\theta \| q) = \mathbb{E}_{p_\theta}[\log(p_\theta/q)] + \mathbb{E}_q[\log(q/p_\theta)]$, 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 multi-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.
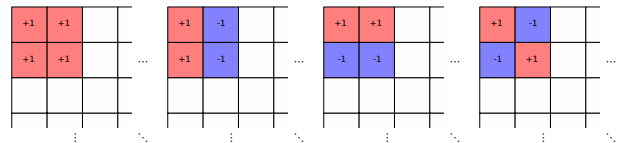

Figure 14. Left most: an example row of A_l, plotted as a matrix; The rest: example rows of $\tilde{A}_l$ correspond to the former row of A_l. They (with some unplotted ones) form the Haar basis, and can be expressed as local convolution operation.
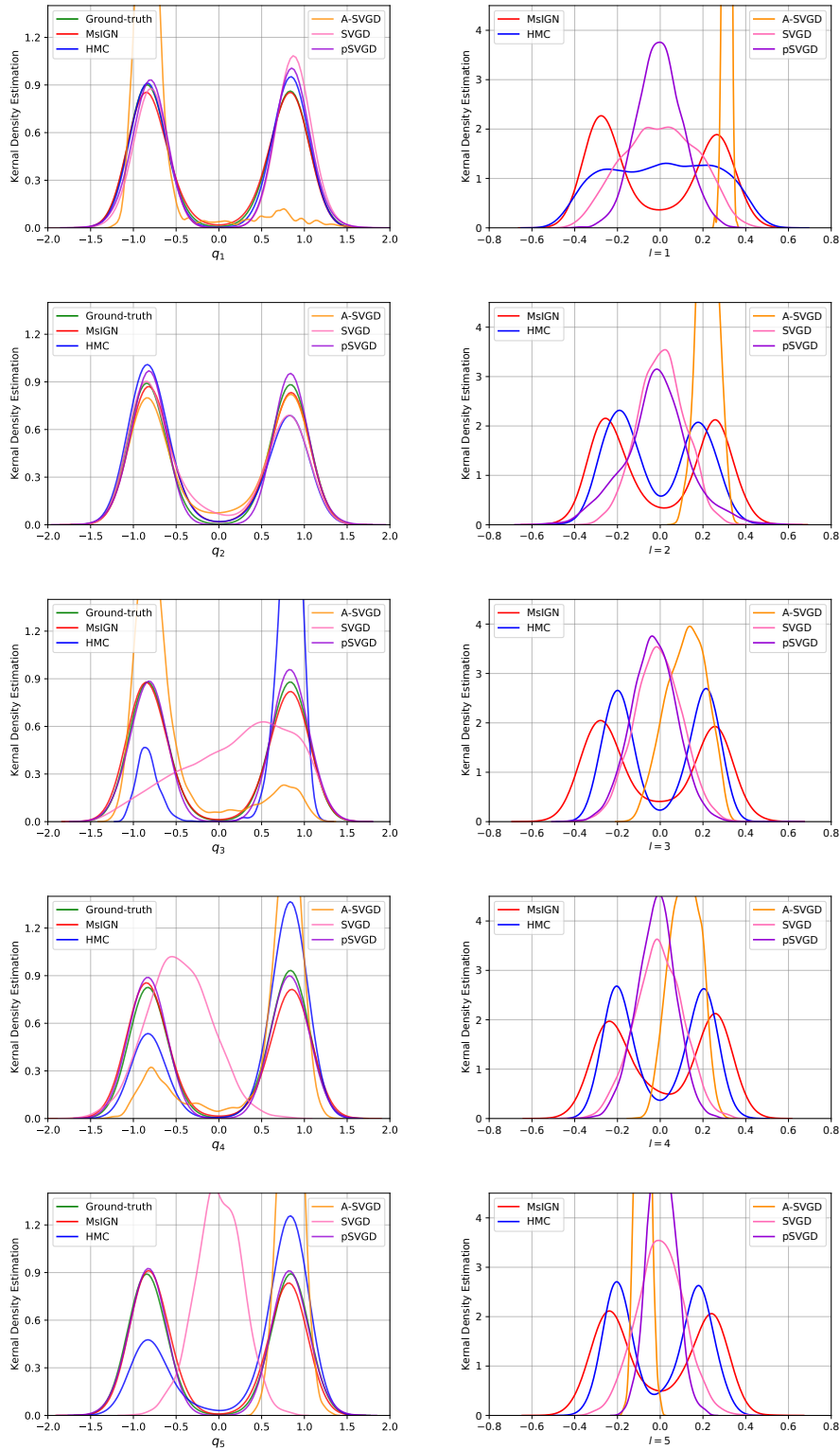
Figure 11. Marginal comparison at the intermediate scales $l = 1, \ldots, 5$. Left: Synthetic BIP; Right: Elliptic BIP. In the synthetic example, as the dimension increases, SVGD and A-SVGD failed in mode capture. Besides, HMC becomes imbalanced between modes, and pSVGD is a bit biased for q_6 in Figure 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.
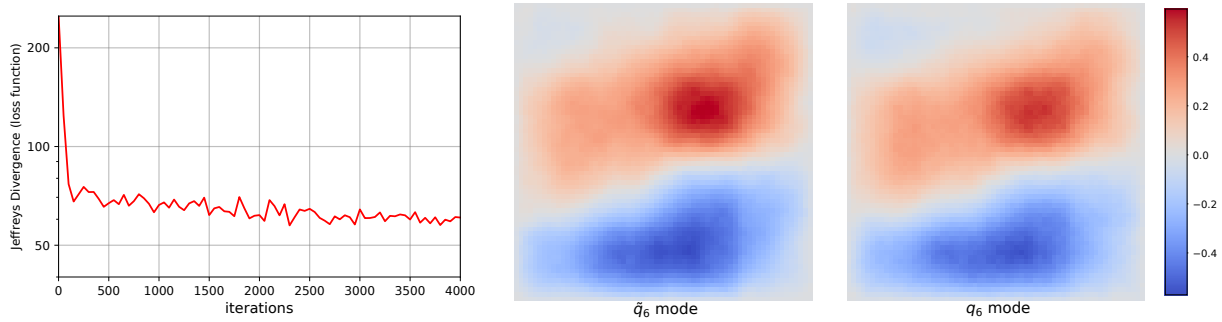
Figure 12. Necessity of training after prior conditioning. Left: learning curve of multi-stage MsIGN at $l = 6$ in the synthetic BIP example; Middle and Right: comparison of the modes captured by the prior conditioned untrained model and the trained model in the elliptic BIP example. The learning curve 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).

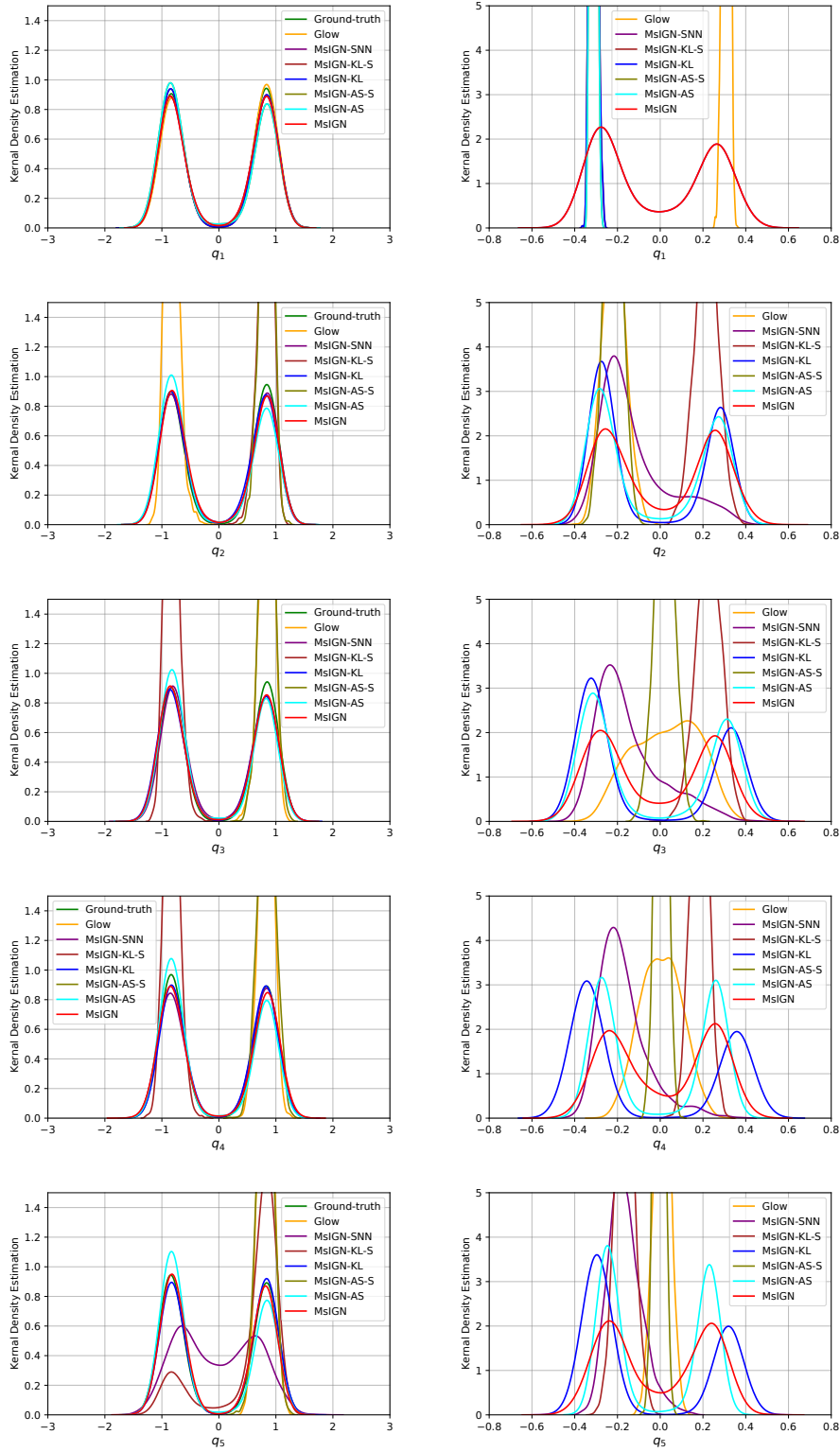| DATA SET	MNIST	CIFAR-10	CELEBA	IMAGENET 32	IMAGENET 64
MINIBATCH SIZE	400	400	200	400	200
SCALES (L)	2	3	3	3	3
♯ OF GLOW BLOCKS (K)	32	32	32	32	32
♯ OF HIDDEN CHANNELS	512	512	512	512	512
♯ OF EPOCHS	2000	2000	1000	400	200

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

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

$$= \begin{bmatrix} A_l \\ \tilde{A}_l \end{bmatrix}^T \begin{bmatrix} A_l \\ \tilde{A}_l \end{bmatrix} = A_l^T A_l + \tilde{A}_l^T \tilde{A}_l .$$

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 match-

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

ing 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.

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.
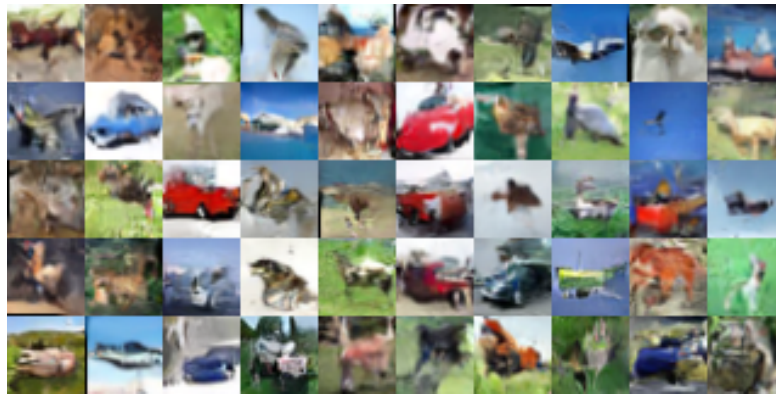

Figure 16. Synthesized images of resolution 32×32 from MsIGN on the CIFAR-10 data set, temperature $= 1.0$.
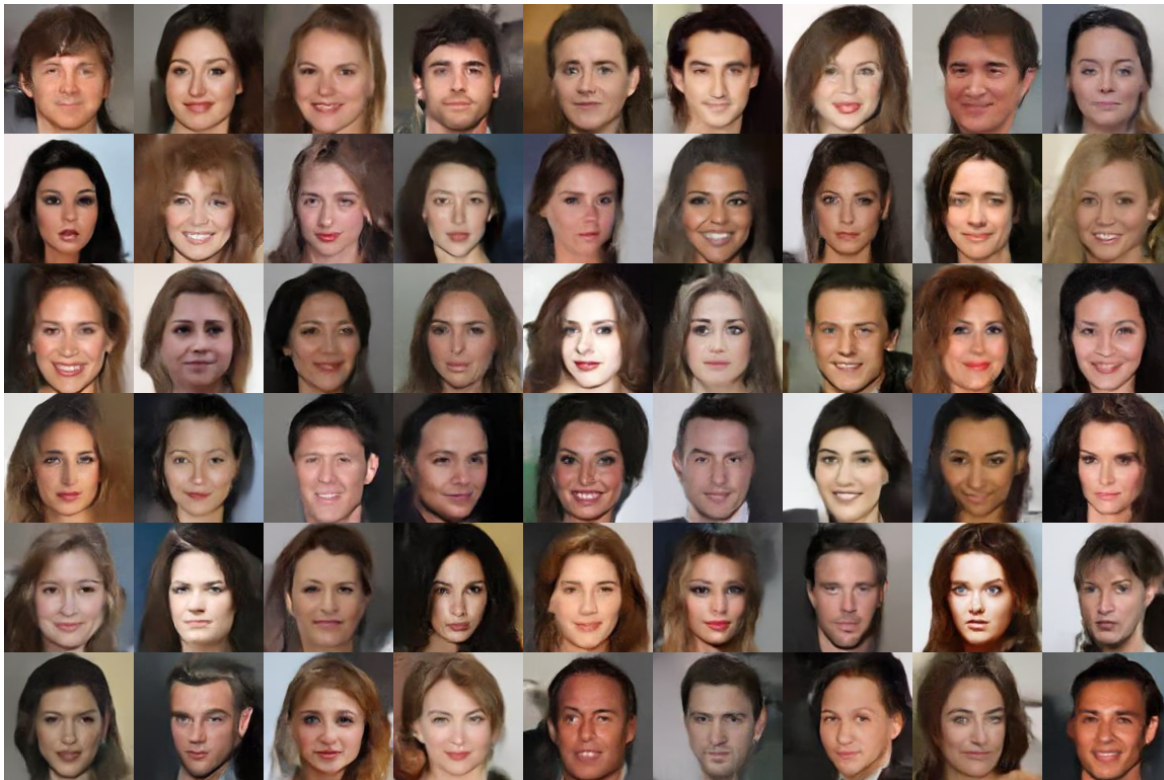

Figure 17. Synthesized images of resolution 128×128 from MsIGN on the CelebA data set, temperature $= 0.8$.
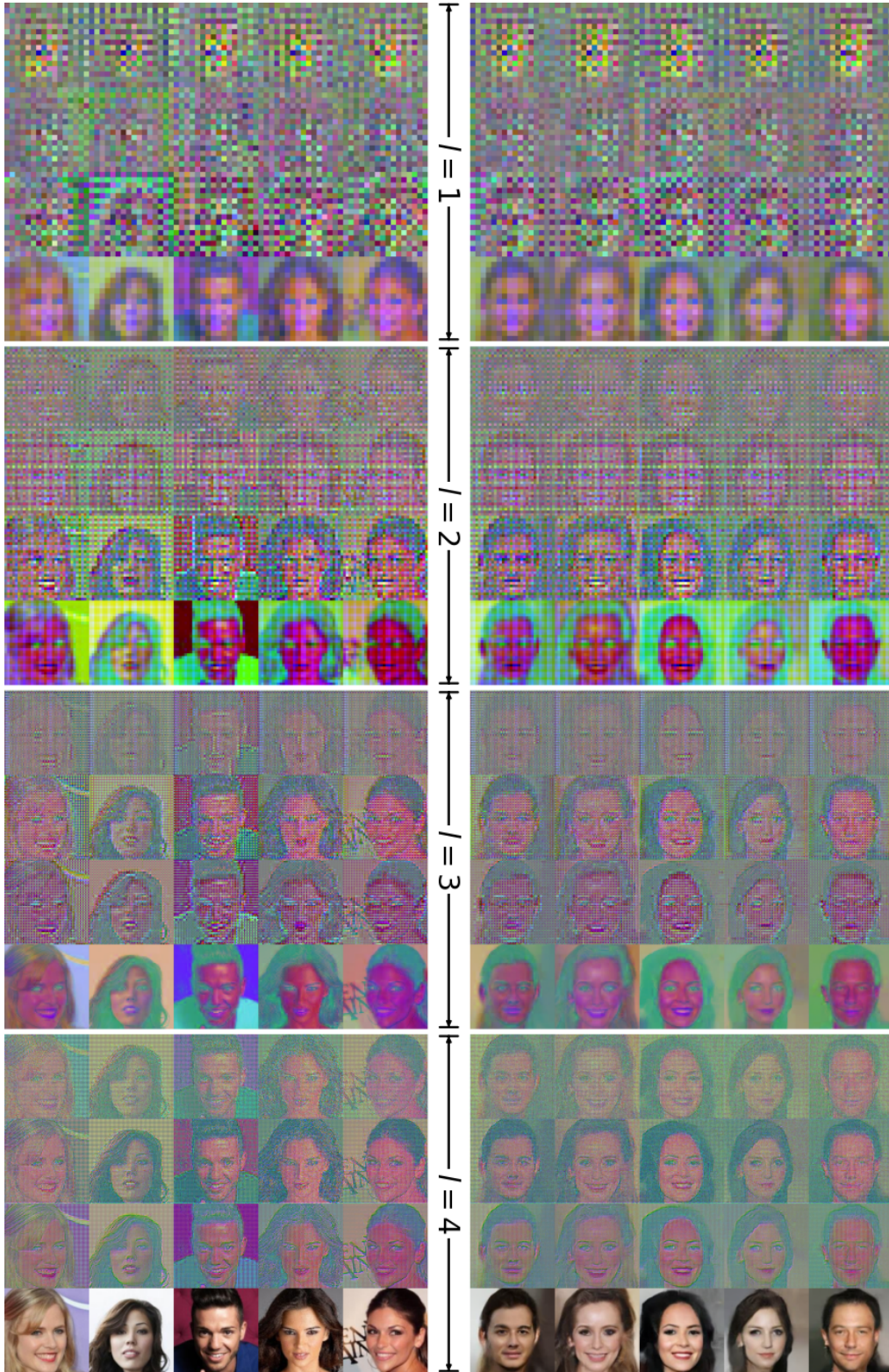
Figure 18. Visualization of internal neurons of MsIGN in synthesizing or recovering 128×128-resolution images on CelebA data set. Snapshots (from top to bottom) are taken 4 times every scale, resulting $4 * 4 = 16$ checkpoints for every image generated. At scale l ($1 \leq l \leq 4$), where the resolution 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.