Sliced Iterative Normalizing Flows

SUPPLEMENTARY DOCUMENT

S1. Proofs

Proposition S1. *Let $P_p(\Omega)$ be the set of Borel probability measures with finite p'th moment on metric space (Ω, d). The maximum K-sliced p-Wasserstein distance is a metric over $P_p(\Omega)$.*

Proof. We firstly prove the triangle inequality. Let μ_1, μ_2 and μ_3 be probability measures in $P_p(\Omega)$ with probability density function p_1, p_2 and p_3, respectively. Let $\{\theta_1^*, \cdots, \theta_K^*\} = \arg\max_{\{\theta_1,\cdots,\theta_K\} \text{ orthonormal}} \left(\frac{1}{K}\sum_{k=1}^K W_p^p((\mathcal{R}p_1)(\cdot,\theta_k), (\mathcal{R}p_3)(\cdot,\theta_k))\right)^{\frac{1}{p}}$; then

$$
\begin{aligned}
&\max\text{-}K\text{-}SW_p(p_1,p_3)\\
=&\max_{\{\theta_1,\cdots,\theta_K\} \text{ orthonormal}}\\
&\left(\frac{1}{K}\sum_{k=1}^K W_p^p((\mathcal{R}p_1)(\cdot,\theta_k), (\mathcal{R}p_3)(\cdot,\theta_k))\right)^{\frac{1}{p}}\\
=&\left(\frac{1}{K}\sum_{k=1}^K W_p^p((\mathcal{R}p_1)(\cdot,\theta_k^*), (\mathcal{R}p_3)(\cdot,\theta_k^*))\right)^{\frac{1}{p}}\\
\le&\left(\frac{1}{K}\sum_{k=1}^K [W_p((\mathcal{R}p_1)(\cdot,\theta_k^*), (\mathcal{R}p_2)(\cdot,\theta_k^*))\right.\\
&\left.+W_p((\mathcal{R}p_2)(\cdot,\theta_k^*), (\mathcal{R}p_3)(\cdot,\theta_k^*))]^p\right)^{\frac{1}{p}}\\
\le&\left(\frac{1}{K}\sum_{k=1}^K W_p^p((\mathcal{R}p_1)(\cdot,\theta_k^*), (\mathcal{R}p_2)(\cdot,\theta_k^*))\right)^{\frac{1}{p}} \qquad \text{(S1)}\\
&+\left(\frac{1}{K}\sum_{k=1}^K W_p^p((\mathcal{R}p_2)(\cdot,\theta_k^*), (\mathcal{R}p_3)(\cdot,\theta_k^*))\right)^{\frac{1}{p}}\\
\le&\max_{\{\theta_1,\cdots,\theta_K\} \text{ orthonormal}}\\
&\left(\frac{1}{K}\sum_{k=1}^K W_p^p((\mathcal{R}p_1)(\cdot,\theta_k), (\mathcal{R}p_2)(\cdot,\theta_k))\right)^{\frac{1}{p}}\\
&+\max_{\{\theta_1,\cdots,\theta_K\} \text{ orthonormal}}\\
&\left(\frac{1}{K}\sum_{k=1}^K W_p^p((\mathcal{R}p_2)(\cdot,\theta_k), (\mathcal{R}p_3)(\cdot,\theta_k))\right)^{\frac{1}{p}}\\
=&\max\text{-}K\text{-}SW_p(p_1,p_2)+\max\text{-}K\text{-}SW_p(p_2,p_3),
\end{aligned}
$$

where the first inequality comes from the triangle inequality of Wasserstein distance, and the second inequality follows Minkowski inequality. Therefore $\max\text{-}K\text{-}SW_p$ satisfies the triangle inequality.

Now we prove the identity of indiscernibles. For any probability measures μ_1 and μ_2 in $P_p(\Omega)$ with probability density function p_1 and p_2, let $\hat{\theta} = \arg\max_{\theta\in\mathbb{S}^{d-1}} W_p((\mathcal{R}p_1)(\cdot,\theta), (\mathcal{R}p_2)(\cdot,\theta))$, and $\{\theta_1^*, \cdots, \theta_K^*\} = \arg\max_{\{\theta_1,\cdots,\theta_K\} \text{ orthonormal}} \left(\frac{1}{K}\sum_{k=1}^K W_p^p((\mathcal{R}p_1)(\cdot,\theta_k), (\mathcal{R}p_2)(\cdot,\theta_k))\right)^{\frac{1}{p}}$, we have

$$
\begin{aligned}
&\max\text{-}K\text{-}SW_p(p_1,p_2)\\
=&\left(\frac{1}{K}\sum_{k=1}^K W_p^p((\mathcal{R}p_1)(\cdot,\theta_k^*), (\mathcal{R}p_2)(\cdot,\theta_k^*))\right)^{\frac{1}{p}}\\
\le&\left(\frac{1}{K}\sum_{k=1}^K W_p^p((\mathcal{R}p_1)(\cdot,\hat{\theta}), (\mathcal{R}p_2)(\cdot,\hat{\theta}))\right)^{\frac{1}{p}} \qquad \text{(S2)}\\
=&W_p((\mathcal{R}p_1)(\cdot,\hat{\theta}), (\mathcal{R}p_2)(\cdot,\hat{\theta}))\\
=&\max\text{-}SW_p(p_1,p_2).
\end{aligned}
$$

On the other hand, let $\{\hat{\theta}, \tilde{\theta}_2, \cdots, \tilde{\theta}_K\}$ be a set of orthonormal vectors in $\mathbb{S}^{d-1}$ where the first element is $\hat{\theta}$, we have

$$
\begin{aligned}
&\max\text{-}K\text{-}SW_p(p_1,p_2)\\
=&\left(\frac{1}{K}\sum_{k=1}^K W_p^p((\mathcal{R}p_1)(\cdot,\theta_k^*), (\mathcal{R}p_2)(\cdot,\theta_k^*))\right)^{\frac{1}{p}}\\
\ge&\left(\frac{1}{K}W_p^p((\mathcal{R}p_1)(\cdot,\hat{\theta}), (\mathcal{R}p_2)(\cdot,\hat{\theta}))\right.\\
&\left.+\frac{1}{K}\sum_{k=2}^K W_p^p((\mathcal{R}p_1)(\cdot,\tilde{\theta}_k), (\mathcal{R}p_2)(\cdot,\tilde{\theta}_k))\right)^{\frac{1}{p}} \qquad \text{(S3)}\\
\ge&\left(\frac{1}{K}W_p^p((\mathcal{R}p_1)(\cdot,\hat{\theta}), (\mathcal{R}p_2)(\cdot,\hat{\theta}))\right)^{\frac{1}{p}}\\
=&(\frac{1}{K})^{\frac{1}{p}}\max\text{-}SW_p(p_1,p_2).
\end{aligned}
$$

Therefore we have $(\frac{1}{K})^{\frac{1}{p}}\max\text{-}SW_p(p_1,p_2) \le \max\text{-}K\text{-}SW_p(p_1,p_2) \le \max\text{-}SW_p(p_1,p_2)$. Thus $\max\text{-}K\text{-}SW_p(p_1,p_2) = 0 \Leftrightarrow \max\text{-}SW_p(p_1,p_2) = 0 \Leftrightarrow$

$\mu_1 = \mu_2$, where we use the non-negativity and identity of indiscernibles of $\max\text{-}SW_p$.

Finally, the symmetry of $\max\text{-}K\text{-}SW_p$ can be proven using the fact that p-Wasserstein distance is symmetric:

$$\begin{aligned}
&\max\text{-}K\text{-}SW_p(p_1, p_2) \\
=&\left(\frac{1}{K}\sum_{k=1}^{K} W_p^p((\mathcal{R}p_1)(\cdot,\theta_k^*),(\mathcal{R}p_2)(\cdot,\theta_k^*))\right)^{\frac{1}{p}} \\
=&\left(\frac{1}{K}\sum_{k=1}^{K} W_p^p((\mathcal{R}p_2)(\cdot,\theta_k^*),(\mathcal{R}p_1)(\cdot,\theta_k^*))\right)^{\frac{1}{p}} \\
=&\max\text{-}K\text{-}SW_p(p_2, p_1).
\end{aligned} \tag{S4}$$

□

Proof of Equation 10. Let $\{\theta_1, \cdots, \theta_K, \cdots, \theta_d\}$ be a set of orthonormal basis in $\mathcal{R}^d$ where the first K vectors are $\theta_1, \cdots, \theta_K$, respectively. Let $R_l = [\theta_1, \cdots, \theta_d]$ be an orthogonal matrix whose i-th column vector is θ_i, $U_l = [\theta_{K+1}, \cdots, \theta_d]$. Since $A_l = [\theta_1, \cdots, \theta_K]$, we have $R_l = [A_l, U_l]$ (the concatenation of columns of A and U). Let $\mathbf{I}^{d-K} = [\mathrm{id}_1, \cdots, \mathrm{id}_{d-K}]^T$ be a marginal transformation that consists of $d-K$ 1D identity transformation, $\hat{\mathbf{\Psi}}_l = \begin{bmatrix} \mathbf{\Psi}_l \\ \mathbf{I}^{d-K} \end{bmatrix}$, we have

$$\begin{aligned}
X_{l+1} =& A_l \mathbf{\Psi}_l(A_l^T X_l) + X_l - A_l A_l^T X_l \\
=& A_l \mathbf{\Psi}_l(A_l^T X_l) + R_l R_l^T X_l - A_l A_l^T X_l \\
=& A_l \mathbf{\Psi}_l(A_l^T X_l) + [A_l, U_l] \begin{bmatrix} A_l^T \\ U_l^T \end{bmatrix} X_l - A_l A_l^T X_l \\
=& A_l \mathbf{\Psi}_l(A_l^T X_l) + U_l U_l^T X_l \\
=& A_l \mathbf{\Psi}_l(A_l^T X_l) + U_l \mathbf{I}^{d-K}(U_l^T X_l) \\
=& [A_l, U_l] \begin{bmatrix} \mathbf{\Psi}_l \\ I^{d-K} \end{bmatrix} ([A_l, U_l]^T X_l) \\
=& R_l \hat{\mathbf{\Psi}}_l(R_l^T X_l).
\end{aligned} \tag{S5}$$

Since R_l is an orthogonal matrix with determinant ± 1, and the Jacobian of the marginal transformation $\hat{\mathbf{\Psi}}_l$ is diagonal, the Jacobian determinant of the above equation can be written as

$$\begin{aligned}
\det\left(\frac{\partial X_{l+1}}{\partial X_l}\right) &= \prod_{k=1}^{K} \frac{d\Psi_{lk}(x)}{dx} \cdot \prod_{k=1}^{d-K} \frac{d(\mathrm{id}_k(x))}{dx} \\
&= \prod_{k=1}^{K} \frac{d\Psi_{lk}(x)}{dx}.
\end{aligned} \tag{S6}$$

□

S2. Monotonic Rational Quadratic Spline

Monotonic Rational Quadratic Splines (Gregory & Delbourgo, 1982; Durkan et al., 2019) approximate the function in each bin with the quotient of two quadratic polynomials. They are monotonic, contineously differentiable, and can be inverted analytically. The splines are parametrized by the coordinates and derivatives of M knots: $\{(x_m, y_m, y'_m)\}_{m=1}^{M}$, with $x_{m+1} > x_m$, $y_{m+1} > y_m$ and $y'_m > 0$. Given these parameters, the function in bin m can be written as (Durkan et al., 2019)

$$y = y_m + (y_{m+1} - y_m)\frac{s_m \xi^2 + y'_m \xi(1-\xi)}{s_m + \sigma_m \xi(1-\xi)}, \tag{S7}$$

where $s_m = (y_{m+1} - y_m)/(x_{m+1} - x_m)$, $\sigma_m = y'_{m+1} + y'_m - 2s_m$ and $\xi = (x - x_m)/(x_{m+1} - x_m)$. The derivative is given by

$$\frac{dy}{dx} = \frac{s_m^2[y'_{m+1}\xi^2 + 2s_m\xi(1-\xi) + y'_m(1-\xi)^2]}{[s_m + \sigma_m \xi(1-\xi)]^2}. \tag{S8}$$

Finally, the inverse can be calculated with

$$x = x_m + (x_{m+1} - x_m)\frac{2c}{-b - \sqrt{b^2 - 4ac}}, \tag{S9}$$

where $a = (s_m - y'_m) + \zeta\sigma_m$, $b = y'_m - \zeta\sigma_m$, $c = -s_m\zeta$ and $\zeta = (y - y_m)/(y_{m+1} - y_m)$. The derivation of these formula can be found in Appendix A of Durkan et al. (2019).

In our algorithm the coordinates of the knots are determined by the quantiles of the marginalized PDF (see Algorithm 2). The derivative y'_m $(1 < m < M)$ is determined by fitting a local quadratic polynomial to the neighboring knots (x_{m-1}, y_{m-1}), (x_m, y_m), and (x_{m+1}, y_{m+1}):

$$y'_m = \frac{s_{m-1}(x_{m+1} - x_m) + s_m(x_m - x_{m-1})}{x_{m+1} - x_{m-1}}. \tag{S10}$$

The function outside $[x_1, x_M]$ is linearly extrapolated with slopes y'_1 and y'_M. In SIG, y'_1 and y'_M are fixed to 1, while in GIS they are fitted to the samples that fall outside $[x_1, x_M]$.

We use $M = 400$ knots in SIG to interpolate each $\Psi_{l,k}$, while in GIS we allow M to vary between $[50, 200]$, depending on the dataset size $M = \sqrt{N_{\text{train}}}$. The performance is insensitive to these choices, as long as M is large enough to fully characterize the 1D transformation $\Psi_{l,k}$.

S3. Optimization on the Stiefel Manifold

The calculation of max K-SWD (Equation 7) requires optimization under the constraints that $\{\theta_1, \cdots, \theta_K\}$ are orthonormal vectors, or equivalently, $A^T A = I_K$ where $A = [\theta_1, \cdots, \theta_K]$ is the matrix whose i-th column vector is θ_i. As suggested by Tagare (2011), the optimization

of matrix A can be performed by doing gradient ascent on the Stiefel Manifold:

$$A_{(j+1)} = \left(I_d + \frac{\tau}{2}B_{(j)}\right)^{-1}\left(I_d - \frac{\tau}{2}B_{(j)}\right)A_{(j)}, \quad \text{(S11)}$$

where $A_{(j)}$ is the weight matrix at gradient descent iteration j (which is different from the iteration l of the algorithm), τ is the learning rate, which is determined by backtracking line search, $B = GA^T - AG^T$, and G is the negative gradient matrix $G = [-\frac{\partial \mathcal{F}}{\partial A_{p,q}}] \in \mathbb{R}^{d\times K}$. Equation S11 has the properties that $A_{(j+1)} \in V_K(\mathbb{R}^d)$, and that the tangent vector $\frac{dA_{(j+1)}}{d\tau}|_{\tau=0}$ is the projection of gradient $[\frac{\partial \mathcal{F}}{\partial A_{p,q}}]$ onto $T_{A_{(j)}}(V_K(\mathbb{R}^d))$ (the tangent space of $V_K(\mathbb{R}^d)$ at $A_{(j)}$) under the canonical inner product (Tagare, 2011).

However, Equation S11 requires the inverse of a $d\times d$ matrix, which is computationally expensive in high dimensions. The matrix inverse can be simplified using the Sherman-Morrison-Woodbury formula, which results in the following equation (Tagare, 2011):

$$A_{(j+1)} = A_{(j)} - \tau U_{(j)}(I_{2K} + \frac{\tau}{2}V_{(j)}^T U_{(j)})^{-1}V_{(j)}^T A_{(j)}, \quad \text{(S12)}$$

where $U = [G, A]$ (the concatenation of columns of G and A) and $V = [A, -G]$. Equation S12 only involves the inverse of a $2K \times 2K$ matrix. For high dimensional data (e.g. images), we use a relatively small K to avoid the inverse of large matrices. A large K leads to faster training, but one would converge to similar results with a small K using more iterations. In Appendix S4 we show that the convergence is insensitive to the choice of K.

S4. Hyperparameter Study and Ablation Analysis

Here we study the sensitivity of SINF to hyperparameters and perform ablation analyses.

S4.1. Hyperparameter K, Objective Function, and Patch-Based Approach

We firstly test the convergence of SIG on MNIST dataset with different K choices. We measure the SWD (Equation 5) and max SWD (Equation 6) between the test data and model samples for different iterations (without patch based hierarchical modeling). The results are presented in Figure S1. The SWD is measured with 10000 Monte Carlo samples and averaged over 10 times. The max SWD is measured with Algorithm 1 ($K = 1$) using different starting points in order to find the global maximum. We also measure the SWD and max SWD between the training data and test data, which gives an estimate of the noise level arising from the finite number of test data. For the range of K we consider ($1 \leq K \leq 128$), all tests we perform converges

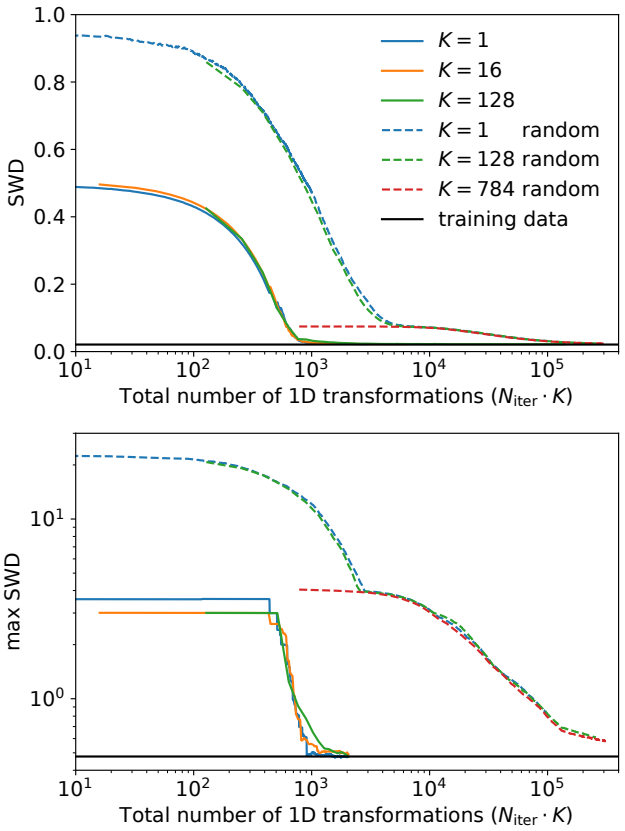

Figure S1. Sliced Wasserstein Distance (SWD, top panel) and Max-Sliced Wasserstein Distance (max SWD, bottom panel) between the MNIST test data and model samples as a function of total number of marginal transformations. The legend in the top panel also applies to the bottom panel. The SWD and max SWD between the training data and test data is shown in the horizontal solid black lines. The lines with "random" indicate that the axes are randomly chosen (like RBIG) instead of using the axes of max K-SWD. We also test $K = 2,\ 4,\ 8,\ 32$, and 64. Their curves overlap with $K = 1,\ 16$ and 128 and are not shown in the plot.

to the noise level, and the convergence is insensitive to the choice of K, but mostly depends on the total number of 1D transformations ($N_{\text{iter}} \cdot K$). As a comparison, we also try running SIG with random orthogonal axes per iteration, and for MNIST, our greedy algorithm converges with two orders of magnitude fewer marginal transformations than random orthogonal axes (Figure S1).

For $K = 1$, the objective function (Equation 11) is the same as max SWD, so one would expect that the max SWD between the data and the model distribution keep decreasing as the iteration number increases. For $K > 1$, the max K-SWD is bounded by max SWD (Equation S2 and S3) so one would also expect similar behavior. However, from Figure S1 we find that max SWD stays constant in the first 400 iterations. This is because SIG fails to find the global maximum of the objective function in those iterations, i.e.,

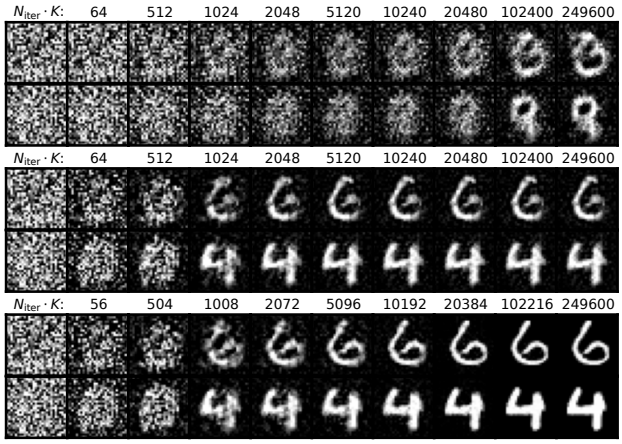

Figure S2. Top panel: SIG samples with random axes ($K = 64$). Middle panel: SIG samples with optimized axes ($K = 64$). Bottom panel: SIG samples with optimized axes and patch based hierarchical approach. The numbers above each panel indicate the number of marginal transformations.

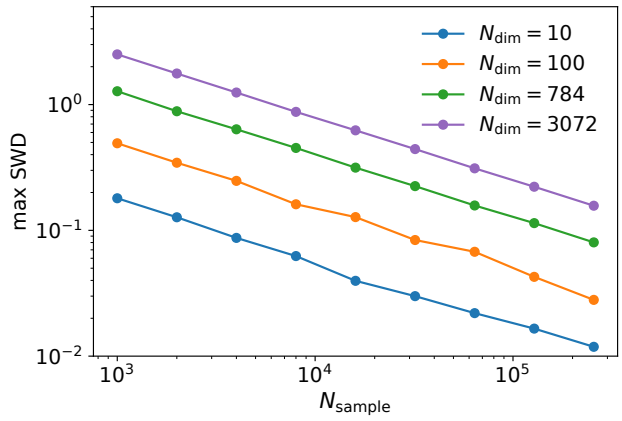

Figure S3. The measured maximum sliced Wasserstein distance between two Gaussian datasets as a function of number of samples. 10 different starting points are used to find the global maximum.

the algorithm converges at some local maximum that is almost perpendicular to the global maximum in the high dimensional space, and therefore the max SWD is almost unchanged. This suggests that our algorithm does not require global optimization of A at each iteration: even if we find only a local maximum, it can be compensated with subsequent iterations. Therefore our model is insensitive to the initialization and random seeds. This is very different from the standard non-convex loss function optimization in deep learning with a fixed number of layers, where the random seeds often make a big difference (Lucic et al., 2018).

In Figure S2 we show the samples of SIG of random axes, optimized axes and hierarchical approach. On the one hand, the sample quality of SIG with optimized axes is better than that of random axes, suggesting that our proposed objective max K-SWD improves both the efficiency and the accuracy of the modeling. On the other hand, SIG with optimized axes has reached the noise level on both SWD and max SWD at around 2000 marginal transformations (Figure S1), but the samples are not good at that point, and further increasing the number of 1D transformations from 2000 to 200000 does not significantly improve the sample quality. At this stage the objective function of Equation 11 is dominated by the noise from finite sample size, and the optimized axes are nearly random, which significantly limits the efficiency of our algorithm. To better understand this noise, we do a simple experiment by sampling two sets of samples from the standard normal distribution $\mathcal{N}(0, I)$ and measuring the max SWD using the samples. The true distance should be zero, and any nonzero value is caused by the finite number of samples. In Figure S3 we show the measured max SWD as a function of sample size and dimensionality. For small number of samples and high dimensionality, the measured max SWD is quite large, suggesting that we can easily find an axis where the marginalized PDF of the two sets of samples are significantly different, while their underlying distribution are actually the same. Because of this sample noise, once the generated and the target distribution are close to each other (the max K-SWD reached the noise level), the optimized axes becomes random and the algorithm becomes inefficient. To reduce the noise level, one needs to either increase the size of training data or decrease the dimensionality of the problem. The former can be achieved with data augmentation. In this study we adopt the second approach, i.e., we effectively reduce the dimensionality of the modeling with a patch based hierarchical approach. The corresponding samples are shown in the bottom panel of Figure S2. We see that the sample quality keeps improving after 2000 marginal transformations, because the patch based approach reduces the effective noise level.

S4.2. Effects of Regularization Parameter α in Density Estimation

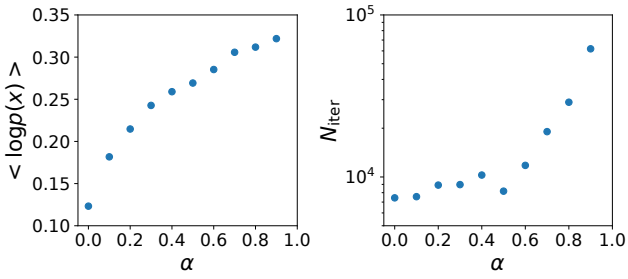

Figure S4. Test log-likelihood (left panel) and number of iterations (right panel) as a function of regularization parameter α on POWER dataset.

To explore the effect of regularization parameter α, we train

GIS on POWER dataset with different α. We keep adding iterations until the log-likelihood of validation set stops improving. The final test $\log p$ and the number of iterations are shown in Figure S4. We see that with a larger α, the algorithm gets better density estimation performance, at the cost of taking more iterations to converge. Setting the regularization parameter α is a trade-off between performance and computational cost.

S5. Experimental Details

The hyperparameters of GIS include the number of axes per iteration K, the regularization α, and the KDE kernel width factor b. We have two different α values: $\alpha = (\alpha_1, \alpha_2)$, where α_1 regularizes the rational quadratic splines, and α_2 regularizes the linear extrapolations. The KDE kernel width σ is determined by the Scott's rule (Scott, 2015):

$$\sigma = bN^{-0.2}\sigma_{\text{data}}, \tag{S13}$$

where N is the number of training data, and σ_{data} is the standard deviation of the data marginalized distribution.

The hyperparameters for density-estimation results in Table 2 are shown in Table S1. K is determined by $K = \min(8, d)$. For BSDS300 we first whiten the data before applying GIS. For high dimensional image datasets MNIST and Fashion-MNIST, we add patch-based iterations with patch size $q = 4$ and $q = 2$ alternately. Logit transformation is used as data preprocessing. For all of the datasets, we keep adding iterations until the validation $\log p$ stops improving.

For density estimation of small datasets, we use the following hyperparameter choices for large regularization setting: $b = 1$, $K = \min(8, d)$, $\alpha = (1 - 0.02\log_{10}(N_{\text{train}}), 1 - 0.001\log_{10}(N_{\text{train}}))$. While for low regularization setting we use $b = 2$ and $\alpha = (0, 1 - 0.01\log_{10}(N_{\text{train}}))$. The size of the validation set is 30% of the training set size. All results are averaged over 5 different realizations.

The hyperparameters of SIG include the number of axes per iteration K, and the patch size for each iteration, if the patch-based approach is adopted. We show the SIG hyperparameters for modeling image datasets in Table S2. As discussed in Section 3.5, the basic idea of setting the architecture is to start from the entire image, and then gradually decrease the patch size until $q = 2$. An illustration of the patch-based hierarchical approach is shown in Figure S5. We set $K = q$ or $K = 2q$, depending on the datasets and the depth of the patch. For each patch size we add 100 or 200 iterations.

For OOD results in Section 5.4, we train SIG and GIS on Fashion-MNIST with $K = 56$. GIS is trained with $b = 1$ and $\alpha = 0.9$ (the results are insensitive to all these hyperparameter choices). We do not use logit transformation preprocessing, as it overamplifies the importance of pixels with low variance. The number of iterations are determined by optimizing the validation $\log p$. For SIG, which cannot produce good $\log p$, the results shown in Table 5 use 100 iterations, but we verify they do not depend on this choice and are stable up to thousands of iterations.

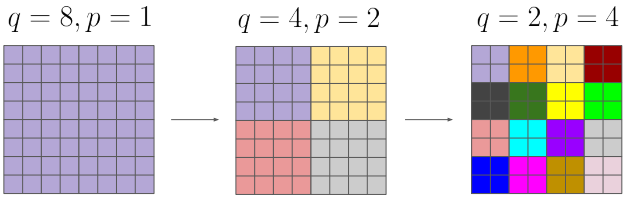

Figure S5. Illustration of the hierarchical modeling of an $S = 8$ image. The patch size starts from $q = 8$ and gradually decreases to $q = 2$.

References

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural spline flows. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 7509–7520, 2019.

Gregory, J. and Delbourgo, R. Piecewise rational quadratic interpolation to monotonic data. *IMA Journal of Numerical Analysis*, 2(2):123–130, 1982.

Lucic, M., Kurach, K., Michalski, M., Gelly, S., and Bousquet, O. Are gans created equal? A large-scale study. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 698–707, 2018.

Scott, D. W. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

Tagare, H. D. Notes on optimization on stiefel manifolds. In *Technical report, Technical report*. Yale University, 2011.

Table S1. GIS hyperparameters for density-estimation results in Table 2.

Hyperparameter	POWER	GAS	HEPMASS	MINIBOONE	BSDS300	MNIST	Fashion
K	6	8	8	8	8	8 ($q = 4$) 4 ($q = 2$)	8 ($q = 4$) 4 ($q = 2$)
$\alpha = (\alpha_1, \alpha_2)$	(0.9,0.9)	(0.9,0.9)	(0.95, 0.99)	(0.95, 0.999)	(0.95, 0.95)	(0.9, 0.99)	(0.9, 0.99)
b	2	1	1	2	5	1	1

Table S2. The architectures of SIG for modeling different image datasets in Section 5.2. The architecture is reported in the format of $(q^2 \cdot c, K) \times L$, where q is the side length of the patch, c is the depth of the patch, K is the number of marginal transformations per patch, and L is the number of iterations for that patch size. MNIST and Fashion-MNIST share the same architecture.

	MNIST / Fashion-MNIST	CIFAR-10	CelebA
architecture	$(28^2 \cdot 1, 56) \times 100$ $(14^2 \cdot 1, 28) \times 100$ $(7^2 \cdot 1, 14) \times 100$ $(6^2 \cdot 1, 12) \times 100$ $(5^2 \cdot 1, 10) \times 100$ $(4^2 \cdot 1, 8) \times 100$ $(3^2 \cdot 1, 6) \times 100$ $(2^2 \cdot 1, 4) \times 100$	$(32^2 \cdot 3, 64) \times 200$ $(16^2 \cdot 3, 32) \times 200$ $(8^2 \cdot 3, 16) \times 200$ $(8^2 \cdot 1, 8) \times 100$ $(7^2 \cdot 3, 14) \times 200$ $(7^2 \cdot 1, 7) \times 100$ $(6^2 \cdot 3, 12) \times 200$ $(6^2 \cdot 1, 6) \times 100$ $(5^2 \cdot 3, 10) \times 200$ $(5^2 \cdot 1, 5) \times 100$ $(4^2 \cdot 3, 8) \times 200$ $(4^2 \cdot 1, 4) \times 100$ $(3^2 \cdot 3, 6) \times 200$ $(3^2 \cdot 1, 3) \times 100$ $(2^2 \cdot 3, 4) \times 200$ $(2^2 \cdot 1, 2) \times 100$	$(64^2 \cdot 3, 128) \times 200$ $(32^2 \cdot 3, 64) \times 200$ $(16^2 \cdot 3, 32) \times 200$ $(8^2 \cdot 3, 16) \times 200$ $(8^2 \cdot 1, 8) \times 100$ $(7^2 \cdot 3, 14) \times 200$ $(7^2 \cdot 1, 7) \times 100$ $(6^2 \cdot 3, 12) \times 200$ $(6^2 \cdot 1, 6) \times 100$ $(5^2 \cdot 3, 10) \times 200$ $(5^2 \cdot 1, 5) \times 100$ $(4^2 \cdot 3, 8) \times 200$ $(4^2 \cdot 1, 4) \times 100$ $(3^2 \cdot 3, 6) \times 200$ $(3^2 \cdot 1, 3) \times 100$ $(2^2 \cdot 3, 6) \times 100$
Total number of iterations L_{iter}	800	2500	2500