Invertible Neural Networks for Graph Prediction

Chen Xu [1] Xiuyuan Cheng [2] Yao Xie [1]

[1]Georgia Institute of Technology [2]Duke University

Motivation and Problem Setup

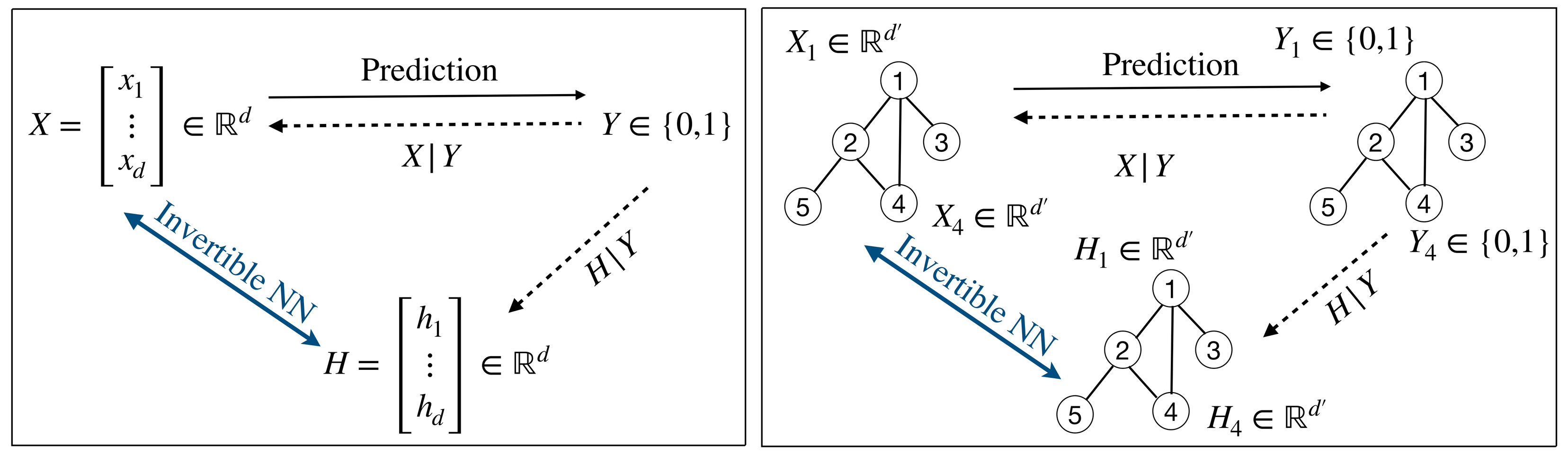


Figure 1. Illustration of the proposed iGNN applied to general data (left) and graph data (right). On graph data, for $v \in \{1, \cdots, 5\}$, X_v is the nodal feature, and Y_v is the nodal label. The inverse prediction is to generate the conditional distribution $X|Y$, which is a one-to-many mapping from Y to X in dash lines. In our approach, Y is first mapped to an intermediate feature $H|Y$ (one-to-many) and then through an invertible neural network to $X|Y$ (one-to-one).

This work focuses on both the nodal label (forward) prediction and the nodal feature (inverse) prediction, as illustrated in Fig. 1. In system monitoring, as an example, it is useful to predict the status of generators given historical observations (the forward prediction, indicated by solid black arrows), as well as to generate unobserved possible circumstances given generator status (the inverse, indicated by dashed black arrows) for cause analyses.

More precisely, the forward prediction problem is to predict Y from input X, which can be formulated as learning the conditional probability of $p(Y|X)$. The inverse prediction problem is to learn the conditional probability of $p(X|Y)$ and to *generate* samples X from it. Note that a discriminative task seeks to estimate the posterior $p(Y|X)$ at a given (test) point X, which can be done by a conventional classification model. The generative task is different in that one asks to sample X according to $p(X|Y)$, that is, given a label Y, to produce samples X which do not exist in the training data nor any provided test set. The problem is challenging when data X is in high dimensional space, where a grid of X can not be efficiently constructed.

Our approach

We propose *invertible graph neural network* (iGNN), which is a two-step procedure as illustrated in the right of Fig. 2. In particular, iGNN tackles the forward and inverse prediction problems at once through a single deep generative model. Given data-label pairs $\{X, Y\}, X \in \mathbb{R}^d, Y \in [K]$ and a residual network $F_\theta := F_{\theta_L} \circ \ldots \circ F_{\theta_1}$, the training objective of the proposed network is

$$\min_{\{\theta, \theta_c\}} \mathcal{L}_g + \mu \mathcal{L}_c + \gamma \mathcal{W}, \tag{1}$$

where $\mathcal{L}_g$, $\mathcal{L}_c$ and $\mathcal{W}$ are the generative loss, the K-class classification loss under softmax, and the Wasserstein-2 regularization. More specifically, in population versions:

$$\mathbb{E}[\mathcal{L}_g] = \mathbb{E}_{X,Y}[\log p_{H|Y}(F_\theta(X)) + \log|\det J_{F_\theta}(X)|], \tag{2}$$

$$\mathbb{E}[\mathcal{W}] = \mathbb{E}_{X,Y}\left[\sum_{l=1}^{L} \|X_l - X_{l-1}\|_2^2\right], \; X_l := X_{l-1} + F_{\theta_l}(X_{l-1}). \tag{3}$$

In other words, (2) is the conditional change-of-variable loss in which $p_{H|Y}$ denotes the density of a Gaussian mixture and (3) penalizes the movement of each residual block.

We explain invertibility of the ResNet F_θ. As $L \to \infty$, the proposed $\mathcal{W}$ in (3) serves to penalize the transport cost of the continuous flow, which is invertible under regularity conditions. Thus, we can expect the discrete ResNet to be invertible when it is a good approximation of the continuous flow. Meanwhile, comparing to [3], (3) allows free-form blocks and is computationally efficient.

Scalability to large graphs. To ensure scalability of iGNN, we introduce a factorized form of $H|Y$ and uses graph neural network (GNN) layers in ResNet blocks [4, 5]. Let $X = [X_1, \ldots, X_N]$ denote d-dimensional nodal features on N nodes. Suppose we have a K-component Gaussian mixture in $\mathbb{R}^{d'}$. We specify the graph $H|Y$ as

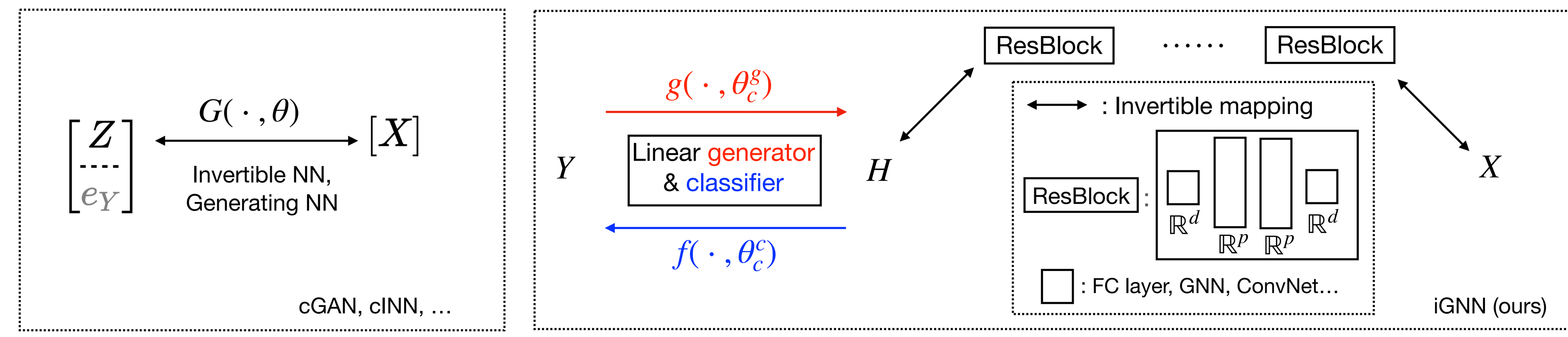


Figure 2. Comparison of existing conditional generative neural network models (left) and the proposed iGNN model (right). Most current approaches (e.g., [1, 2]) concatenate the encoded prediction label Y (e.g. one-hot encoding) as an additional input to the generative network G. Our model takes a two-step approach: a one-to-many mapping g from label Y to intermediate feature H by a Gaussian mixture model (which allows classification from H to Y by f), and a one-to-one mapping from H to input data X.

$$p(H|Y) = \prod_{v=1}^{N} p(H_v|Y_v), \quad H_v|Y_v \sim \mathcal{N}(\mu_{Y_v}, \sigma^2 I_{d'}), \quad \forall v = 1, \cdots, N, \tag{4}$$

that is, the joint distribution of $H|Y$ consists of independent and identical K-component Gaussian mixture distribution of $H_v|Y_v$ in $\mathbb{R}^{d'}$ across the graph. As a result, on the graph sample-label pair $\{X, Y\}$, the $\log p(H|Y)$ term in the generative loss (2) can be computed as

$$\log p_{H|Y}(F_\theta(X)) = \sum_{v=1}^{N} \log p_{H_1|Y_1}((F_\theta(X))_v). \tag{5}$$

Note that the factorized form of $H|Y$ reduces the complexity of modeling $H|Y$ in $\mathbb{R}^{d'N}$ to that of modeling a K-class mixture model in $\mathbb{R}^{d'}$.

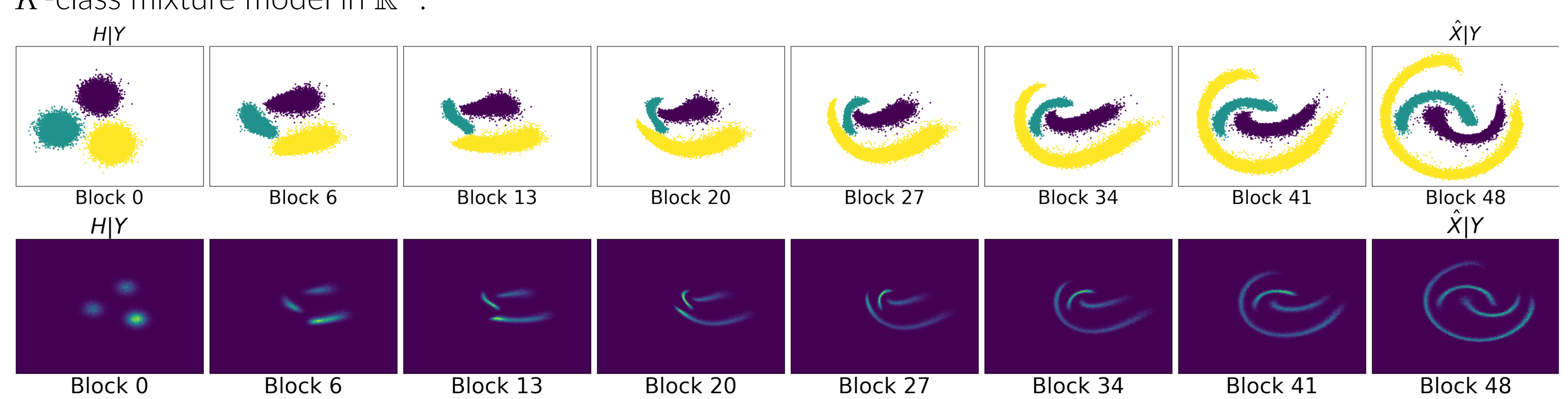
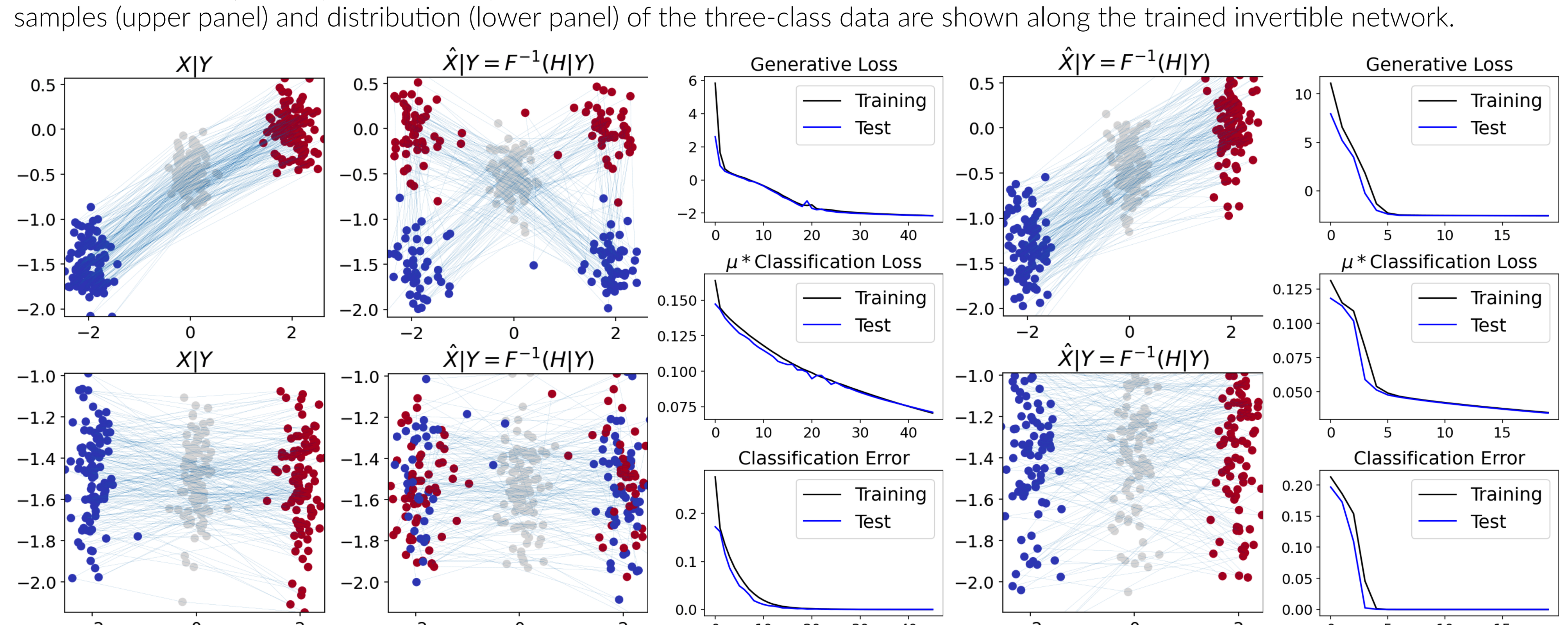


Figure 3. Flow map learned by iGNN model that transports three-class data in $\mathbb{R}^2$ to a three-component Gaussian mixture and back. The distribution $X|Y$ has $(0, 22, 0.22, 0.56)$ fractions in each classes respectively. The ResNet has 48 blocks. The transported data samples (upper panel) and distribution (lower panel) of the three-class data are shown along the trained invertible network.

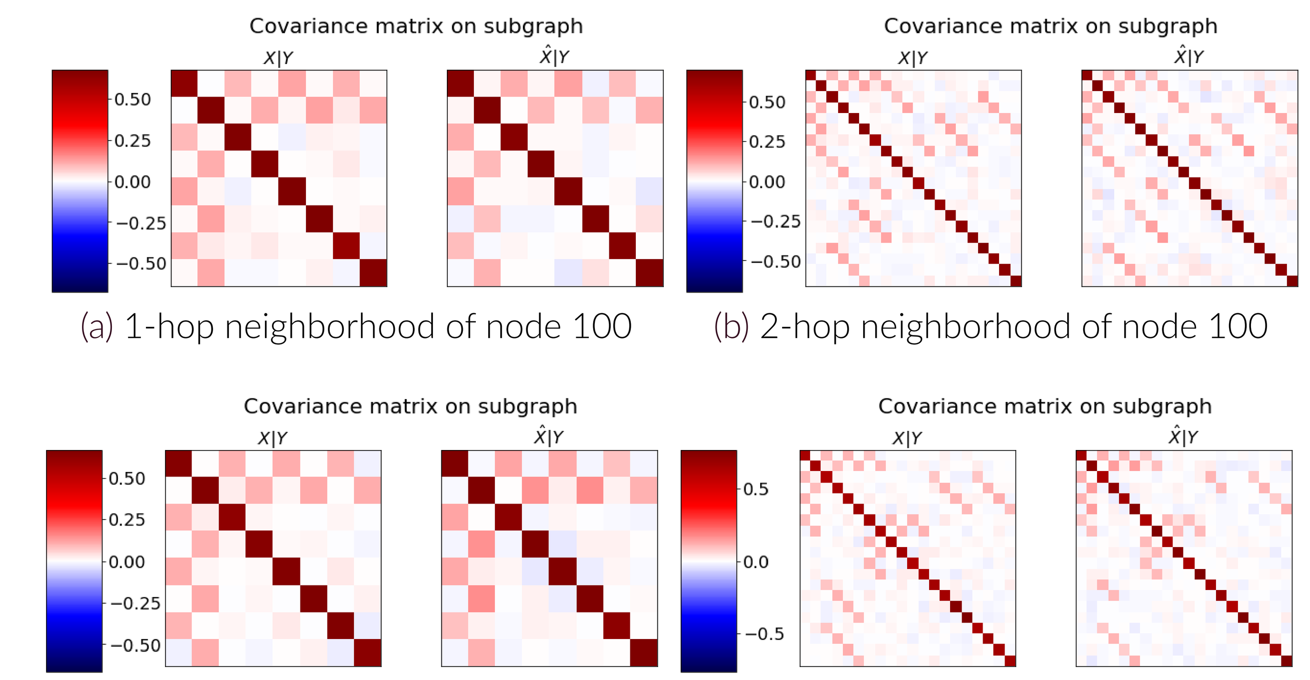


(a) True data (b) ChebNet (c) ChebNet losses (d) L3Net (e) L3Net losses

Figure 4. Comparison of using spectral and spatial GNN layer in iGNN model for conditional generation. Data in (a) are generated as two-dimensional graph node features lying on a three-node graph with binary node labels; colors indicate the node, and the gray lines connect three nodal features associated with the same data instance. The top row has $Y = [1, 1, 0]$ and the bottom row has $Y = [1, 1, 1]$. We visualize samples generated by iGNN using the spectral GNN layer (ChebNet) in (b) and spatial GNN layer (L3Net) in (d), as well as the losses over training epochs in (c) and (e).

Experiments

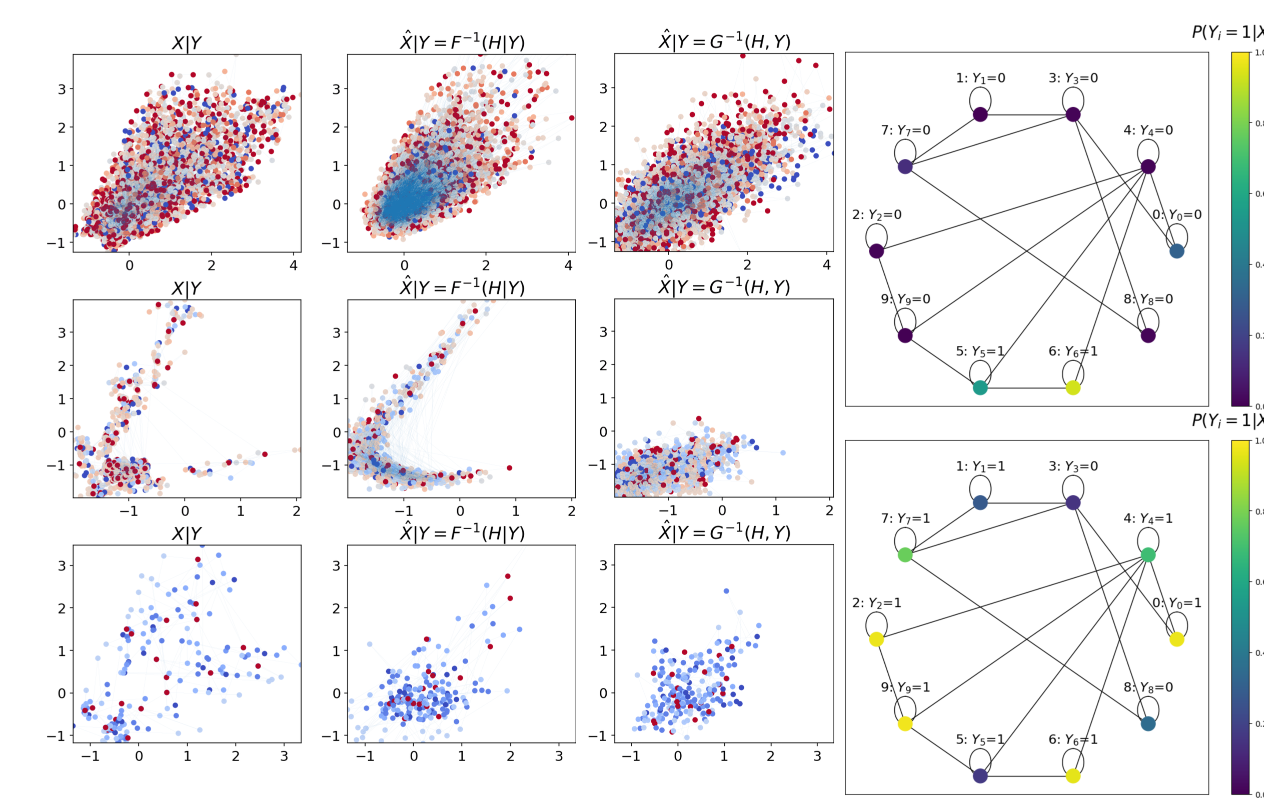
We show the generative and predictive performances of iGNN on simulated and real-data examples for both non-graph data and data on large graphs. Results are shown in Fig. 3-6.



(a) 1-hop neighborhood of node 100 (b) 2-hop neighborhood of node 100

(c) 1-hop neighborhood of node 500 (d) 2-hop neighborhood of node 500

Figure 5. Generating performance by iGNN of graph data $X|Y$ on a 503-node chordal cycle graph, where the node feature dimension $d' = 2$, and the per-node class number $K = 2$. To evaluate the conditional generation quality, we plot the covariance matrix of model-generated data $\hat{X}|Y$ (right plot in (a)-(d)) restricted to sub-graphs produced by 1 or 2-hop neighborhoods of a graph node in comparison with the ground truth (left plot in (a)-(d)).



(a) $X|Y$ (left) and iGNN $\hat{X}|Y$ (right) (b) cINN $\hat{X}|Y$ (c) $\mathbb{P}(Y_i = 1|X)$ by iGNN

Figure 6. Real-data generative and prediction results by iGNN. (a)-(b) show the model generated $\hat{X}|Y$ in comparison to the ground truth $X|Y$, where three rows in the plot represent three chosen representative Y (10-dimensional vector that contains nodal features over all nodes). (c) shows the predicted probabilities of graph labels Y where Y takes different values on graph nodes on test data. Given a node feature matrix X, we compute $\mathbb{P}(Y_i = 1|X)$ on each node using the linear classifier in the H-Y sub-network applied to the flow-mapped graph node feature $H = F_\theta(X)$.

References

[1] Lynton Ardizzone, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=rJed6j0cKX`.

[2] Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. *arXiv preprint arXiv:1907.02392*, 2019.

[3] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, pages 573–582. PMLR, 2019.

[4] Xiuyuan Cheng, Zichen Miao, and Qiang Qiu. Graph convolution with low-rank learnable local filters. In *International Conference on Learning Representations*, 2021.

[5] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.