HexaGAN: GANs for Real World Classification

Uiwon Hwang Dahuin Jung Sungroh Yoon

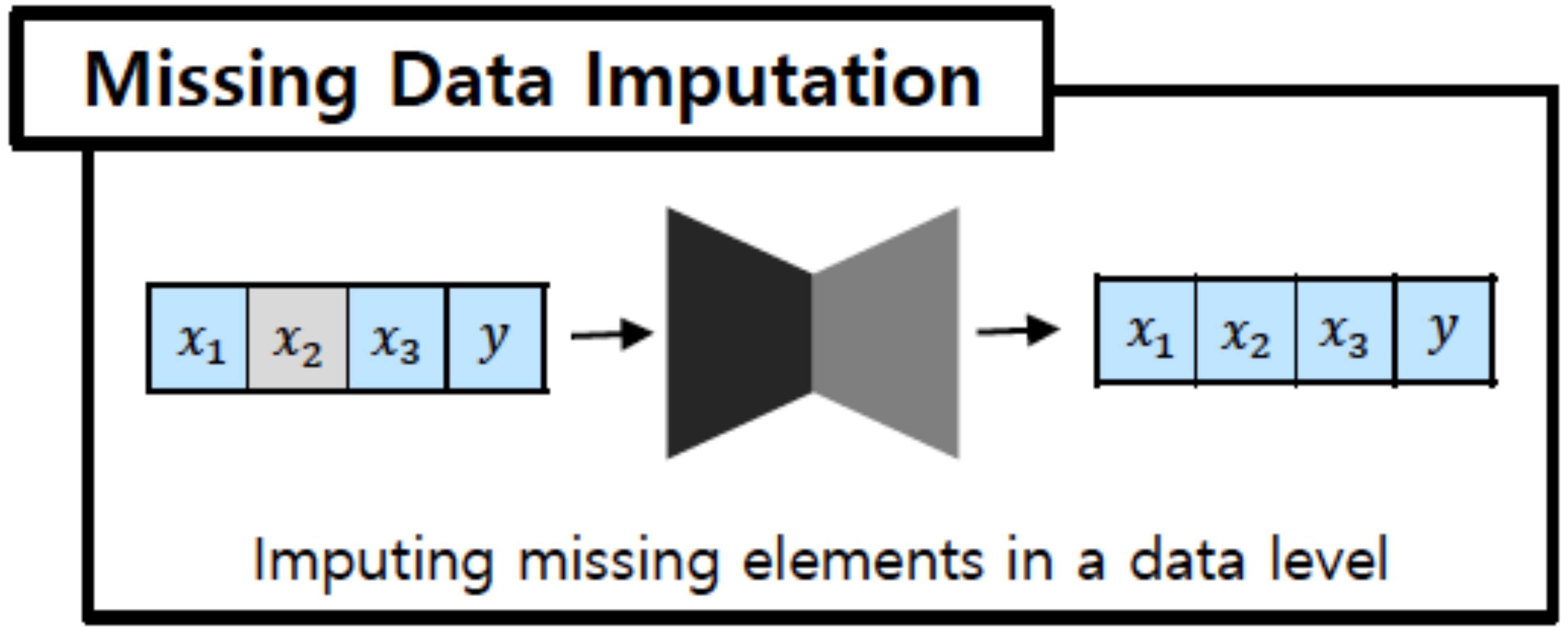
Compiled by Hongming Shan

Motivation

- Most deep learning classification studies assume data clean.
- When dealing with real world data, we encounter three problems:
    - **Missing data**
    - **Class imbalance**
    - **Missing label problem**
- Various preprocessing techniques have been proposed to mitigate one of these problems, but an algorithms that assumes and resolves all three problem has not been proposed yet.
- HexaGAN: a generative adversarial network framework showing promising classification performance for all three problems.

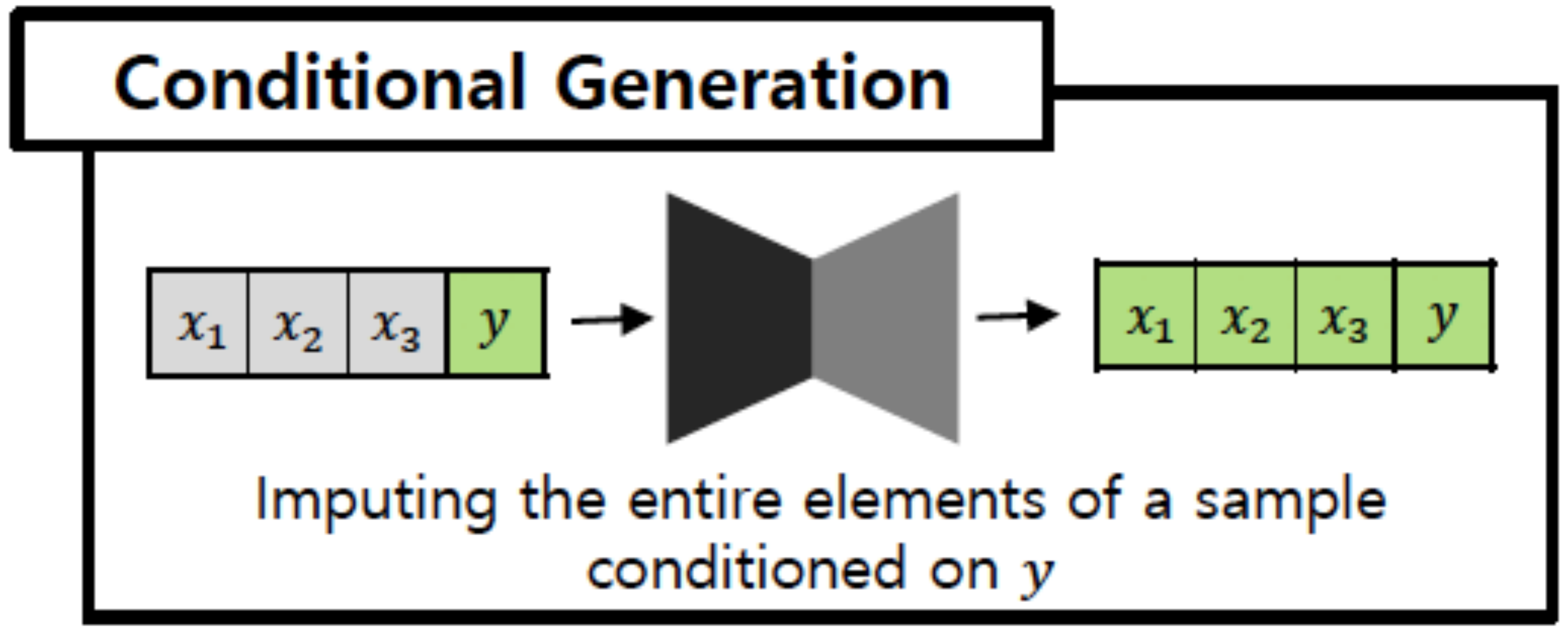
Imputation: The replacement of missing information within data

P1: Missing data imputation



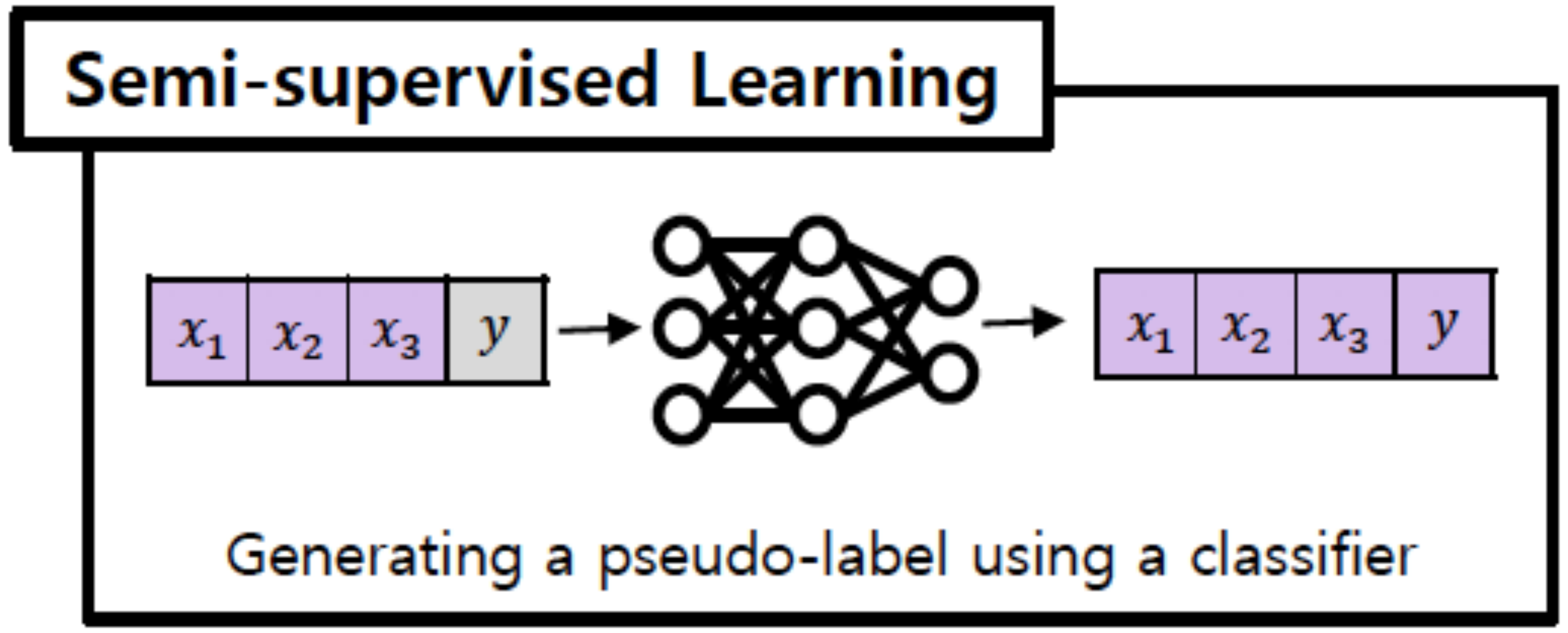
- Three main types of missing data
  - Data are missing completely at random (**MCAR**)
  - Data are missing at random (**MAR**)
  - Data are missing (but) not at random (**MNAR**)

P2: Conditional Generation



- Datasets such as anomaly detection and disease prediction involve poorly balanced classes.
  - Oversampling techniques
  - Cost sensitive loss

P3: Semi-supervised learning



- In deep learning, the amount of labeled training data has a significant impact on the performance.
- Insufficiency of labeled data is referred to as the missing label problem.

Contribution

- One of first studies that defines the three problems (missing data, class imbalance, and missing label) in terms of **imputation**.

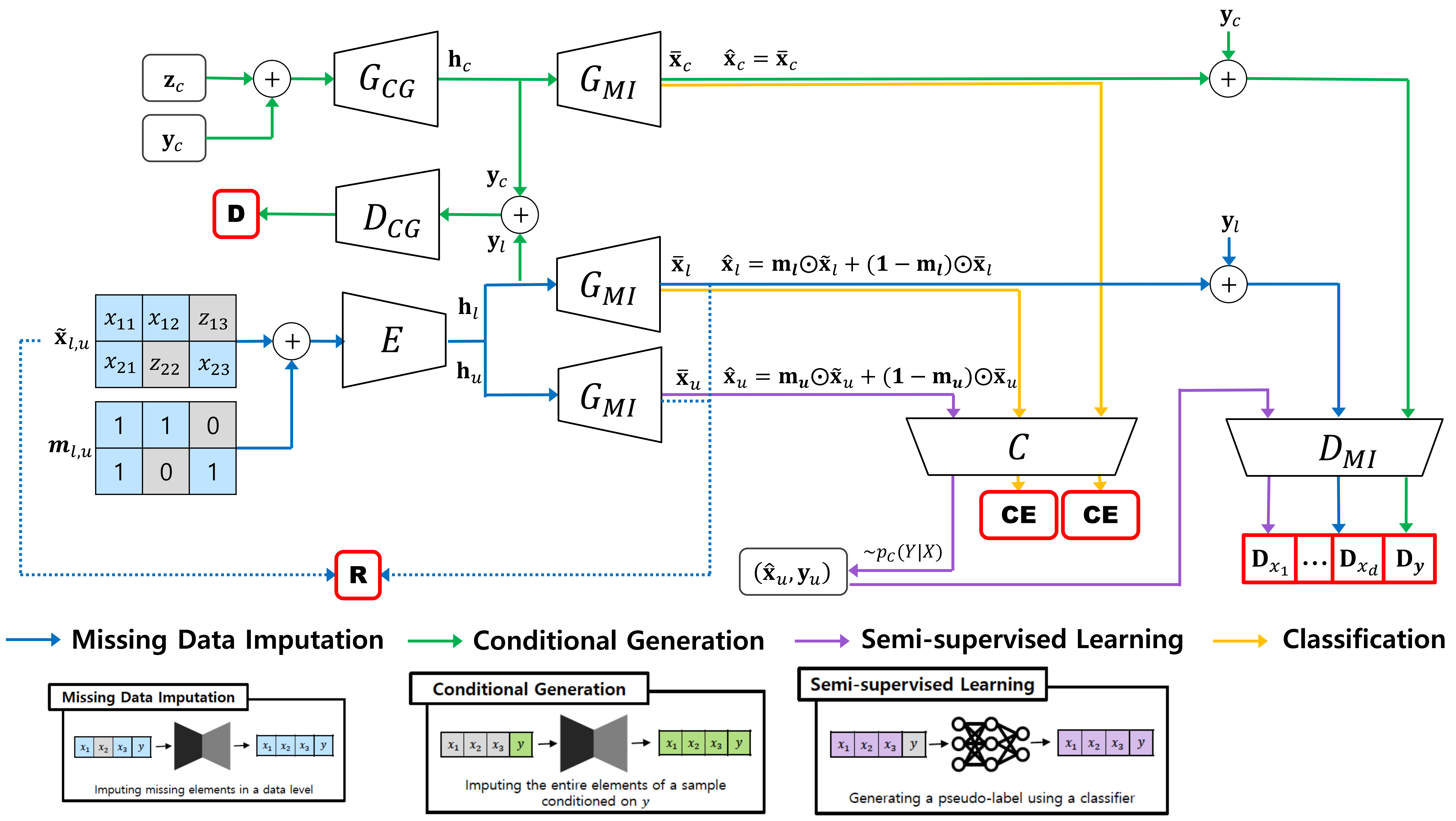
- HexaGAN is simple to **use and works automatically** when the absence of data elements and labels is indicated.

- Devise a combination of **six components** and the corresponding cost function

- The proposed method **significantly outperforms** cascading combination of the existing SOTA methods in real world classification

Notations

- E: the encoder, that transfers both labeled and unlabeled instances into the hidden space.
- G_{MI}: a generator that imputes missing data.
- D_{MI}: a discriminator for missing imputation, that distinguishes between missing and non-missing elements and labels.
- G_{CG}: a generator that creates conditional hidden vectors $\mathbf{h}_c$.
- D_{CG}: a discriminator for conditional generation, that determines whether a hidden vector is from the dataset or has been created by G_{CG}.
- C: the classifier, that estimates class labels. This also works as the label generator.



$\mathbf{z}_c$
$\mathbf{y}_c$
G_{CG}
$\mathbf{h}_c$
G_{MI}
$\bar{\mathbf{x}}_c$
$\hat{\mathbf{x}}_c = \bar{\mathbf{x}}_c$
D
D_{CG}
$\mathbf{y}_c$
$\mathbf{y}_l$
$\bar{\mathbf{x}}_l$
$\hat{\mathbf{x}}_l = \mathbf{m}_l \odot \tilde{\mathbf{x}}_l + (\mathbf{1} - \mathbf{m}_l) \odot \bar{\mathbf{x}}_l$
$\tilde{\mathbf{x}}_{l,u}$
x_{11} x_{12} z_{13}
x_{21} z_{22} x_{23}
E
$\mathbf{h}_l$
$\mathbf{h}_u$
$\bar{\mathbf{x}}_u$
$\hat{\mathbf{x}}_u = \mathbf{m}_u \odot \tilde{\mathbf{x}}_u + (\mathbf{1} - \mathbf{m}_u) \odot \bar{\mathbf{x}}_u$
$\boldsymbol{m}_{l,u}$
1 1 0
1 0 1
C
D_{MI}
CE
CE
$\sim p_C(Y|X)$
$(\hat{\mathbf{x}}_u, \mathbf{y}_u)$
$\mathbf{D}_{x_1}$ ... $\mathbf{D}_{x_d}$ $\mathbf{D}_y$
R
Missing Data Imputation
Conditional Generation
Semi-supervised Learning
Classification
Missing Data Imputation
x_1 x_2 x_3 y
Imputing missing elements in a data level
Conditional Generation
Imputing the entire elements of a sample conditioned on y
Semi-supervised Learning
Generating a pseudo-label using a classifier

Missing data imputation

$$\tilde{\mathbf{x}} = \mathbf{m} \odot \mathbf{x} + (\mathbf{1} - \mathbf{m}) \odot \mathbf{z} \tag{1}$$

$$\hat{\mathbf{x}} = \mathbf{m} \odot \mathbf{x} + (\mathbf{1} - \mathbf{m}) \odot \bar{\mathbf{x}} \tag{2}$$

$$\mathcal{L}_{G_{MI}} = -\sum_{i=1}^{d} \mathbb{E}_{\hat{\mathbf{x}},\mathbf{y},\mathbf{m}} \left[(1 - m_i) \cdot D_{MI}(\hat{\mathbf{x}}, \mathbf{y})_i\right] \tag{3}$$

$$\begin{aligned} \mathcal{L}_{D_{MI}} = \sum_{i=1}^{d} & \mathbb{E}_{\hat{\mathbf{x}},\mathbf{y},\mathbf{m}} \left[(1 - m_i) \cdot D_{MI}(\hat{\mathbf{x}}, \mathbf{y})_i\right] \\ & - \mathbb{E}_{\hat{\mathbf{x}},\mathbf{y},\mathbf{m}} \left[m_i \cdot D_{MI}(\hat{\mathbf{x}}, \mathbf{y})_i\right] \end{aligned} \tag{4}$$

$$\mathcal{L}_{\text{recon}} = \mathbb{E}_{\bar{\mathbf{x}}|\mathbf{x},\mathbf{m}} \left[\sum_{i=1}^{d} m_i (x_i - \bar{x}_i)^2\right] \tag{5}$$

$$\mathcal{L}_{\text{GP}_{MI}} = \sum_{i=1}^{d} \mathbb{E}_{p_{\mathcal{D}}(x_i)} \left[||\nabla_{\hat{\mathbf{x}}} D_{MI}(\hat{\mathbf{x}})_i||_2^2\right] \tag{6}$$

Missing data imputation

Algorithm 1 Missing data imputation

input : $\mathbf{x}$ - data with missing values sampled from D_l and D_u;
$\mathbf{m}$ - vector indicating whether elements are missing;
$\mathbf{z}$ - noise vector sampled from $U(0, 1)$

output : $\hat{\mathbf{x}}$ - imputed data

repeat

Sample a batch of pairs $(\mathbf{x}, \mathbf{m}, \mathbf{z})$

$\tilde{\mathbf{x}} \leftarrow \mathbf{m} \odot \mathbf{x} + (\mathbf{1} - \mathbf{m}) \odot \mathbf{z}$

$\mathbf{h} \leftarrow E(\tilde{\mathbf{x}}, \mathbf{m})$

$\bar{\mathbf{x}} \leftarrow G_{MI}(\mathbf{h})$

$\hat{\mathbf{x}} \leftarrow \mathbf{m} \odot \mathbf{x} + (\mathbf{1} - \mathbf{m}) \odot \bar{\mathbf{x}}$

Update D_{MI} using stochastic gradient descent (SGD)

$\nabla_{D_{MI}} \mathcal{L}_{D_{MI}} + \lambda_1 \mathcal{L}_{\mathrm{GP}_{MI}}$

Update E and G_{MI} using SGD

$\nabla_E \mathcal{L}_{G_{MI}} + \alpha_1 \mathcal{L}_{\mathrm{recon}}$

$\nabla_{G_{MI}} \mathcal{L}_{G_{MI}} + \alpha_1 \mathcal{L}_{\mathrm{recon}}$

until training loss is converged

Conditional generation

$$\mathcal{L}_{G_{CG}} = -\mathbb{E}_{\mathbf{h}_c \sim p_{G_{CG}}(\mathbf{h}_c|\mathbf{y}_c)}[D_{CG}(\mathbf{h}_c, \mathbf{y}_c)] \quad (7)$$

$$\begin{aligned}\mathcal{L}_{D_{CG}} =& \mathbb{E}_{\mathbf{h}_c \sim p_{G_{CG}}(\mathbf{h}_c|\mathbf{y}_c)}[D_{CG}(\mathbf{h}_c, \mathbf{y}_c)] \\ &- \mathbb{E}_{\mathbf{h}_l \sim p_E(\mathbf{h}_l|x_l)}[D_{CG}(\mathbf{h}_l, \mathbf{y}_l)]\end{aligned} \quad (8)$$

$$\mathcal{L}_{\mathrm{GP}_{CG}} = \mathbb{E}_{\mathbf{h}_l \sim p_E(\mathbf{h}_l|x_l)}\left[||\nabla_{\mathbf{h}_l} D_{CG}(\mathbf{h}_l, \mathbf{y}_l)||_2^2\right] \quad (9)$$

$$\mathcal{L}_{\mathrm{CE}}(\hat{\mathbf{x}}_c, \mathbf{y}_c) = -\mathbb{E}_{\hat{\mathbf{x}}_c|\mathbf{y}_c}\left[\sum_{k=1}^{n_c} \mathbf{y}_{c_k} \log(C(\hat{\mathbf{x}}_c)_k)\right] \quad (10)$$

$$\min_{D_{CG}} \mathcal{L}_{D_{CG}} + \lambda_2 \mathcal{L}_{\mathrm{GP}_{CG}} \quad (11)$$

$$\min_{G_{CG}} \mathcal{L}_{G_{CG}} + \alpha_2 \mathcal{L}_{G_{MI}} + \alpha_3 \mathcal{L}_{\mathrm{CE}}(\hat{\mathbf{x}}_c, y_c) \quad (12)$$

Semi-supervised classification

- Pseudo-labeling

$$\mathcal{L}_C = -\mathbb{E}_{\mathbf{y}_u|\hat{\mathbf{x}}_u \sim p_C}\left[D_{MI}(\hat{\mathbf{x}}_u, \mathbf{y}_u)_{d+1}\right] \quad (13)$$

$$\begin{aligned}\mathcal{L}_{D_{MI}}^{d+1} =& \mathbb{E}_{\mathbf{y}_u|\hat{\mathbf{x}}_u \sim p_C}\left[D_{MI}(\hat{\mathbf{x}}_u, \mathbf{y}_u)_{d+1}\right] \quad (14)\\ &- \mathbb{E}_{\mathbf{y}|\hat{\mathbf{x}} \sim p_{data}}\left[D_{MI}(\hat{\mathbf{x}}, \mathbf{y})_{d+1}\right]\end{aligned}$$

- Classification

$$\min_C \mathcal{L}_{\mathrm{CE}}(\hat{\mathbf{x}}_{l,c}, \mathbf{y}_{l,c}) + \alpha_4 \mathcal{L}_C \quad (15)$$

Algorithm 2 Training procedure of HexaGAN

Require : n_{CG} - the number of iterations for the conditional generation per an iteration for the other components;
n_{critic} - the number of iterations for discriminators per an iteration for generators

while training loss is not converged **do**
  (1) Missing data imputation
  for $k = 1, ..., n_{critic}$ **do**
    Update D_{MI} using stochastic gradient descent (SGD)
    $\nabla_{D_{MI}} \mathcal{L}_{D_{MI}} + \mathcal{L}_{D_{MI}}^{d+1} + \lambda_1 \mathcal{L}_{\mathrm{GP}_{MI}}$
  end for
  Update E using SGD
  $\nabla_E \mathcal{L}_{G_{MI}} + \alpha_1 \mathcal{L}_{\text{recon}}$
  Update G_{MI} using SGD
  $\nabla_{G_{MI}} \mathcal{L}_{G_{MI}} + \alpha_1 \mathcal{L}_{\text{recon}}$

  (2) Conditional generation
  for $i = 1, ..., n_{CG}$ **do**
    for $j = 1, ..., n_{critic}$ **do**
      Update D_{CG} using SGD
      $\nabla_{D_{CG}} \mathcal{L}_{D_{CG}} + \lambda_2 \mathcal{L}_{\mathrm{GP}_{CG}}$
    end for
    Update G_{CG} using SGD
    $\nabla_{G_{CG}} \mathcal{L}_{G_{CG}} + \alpha_2 \mathcal{L}_{G_{MI}} + \alpha_3 \mathcal{L}_{\mathrm{CE}}(\hat{\mathbf{x}}_c, \mathbf{y}_c)$
  end for

  (3) Semi-supervised classification
  Update C using SGD
  $\nabla_C \mathcal{L}_{\mathrm{CE}}(\hat{\mathbf{x}}_{l,c}, \mathbf{y}_{l,c}) + \alpha_4 \mathcal{L}_C$
end while

Experiments

- Data: real world datasets (breast, credit, wine), a synthetic dataset (madelon), and MNIST

Table 1. Dataset description. The imbalance ratio indicates the ratio of the number of instances in the majority class to the number of instances in the minority class.

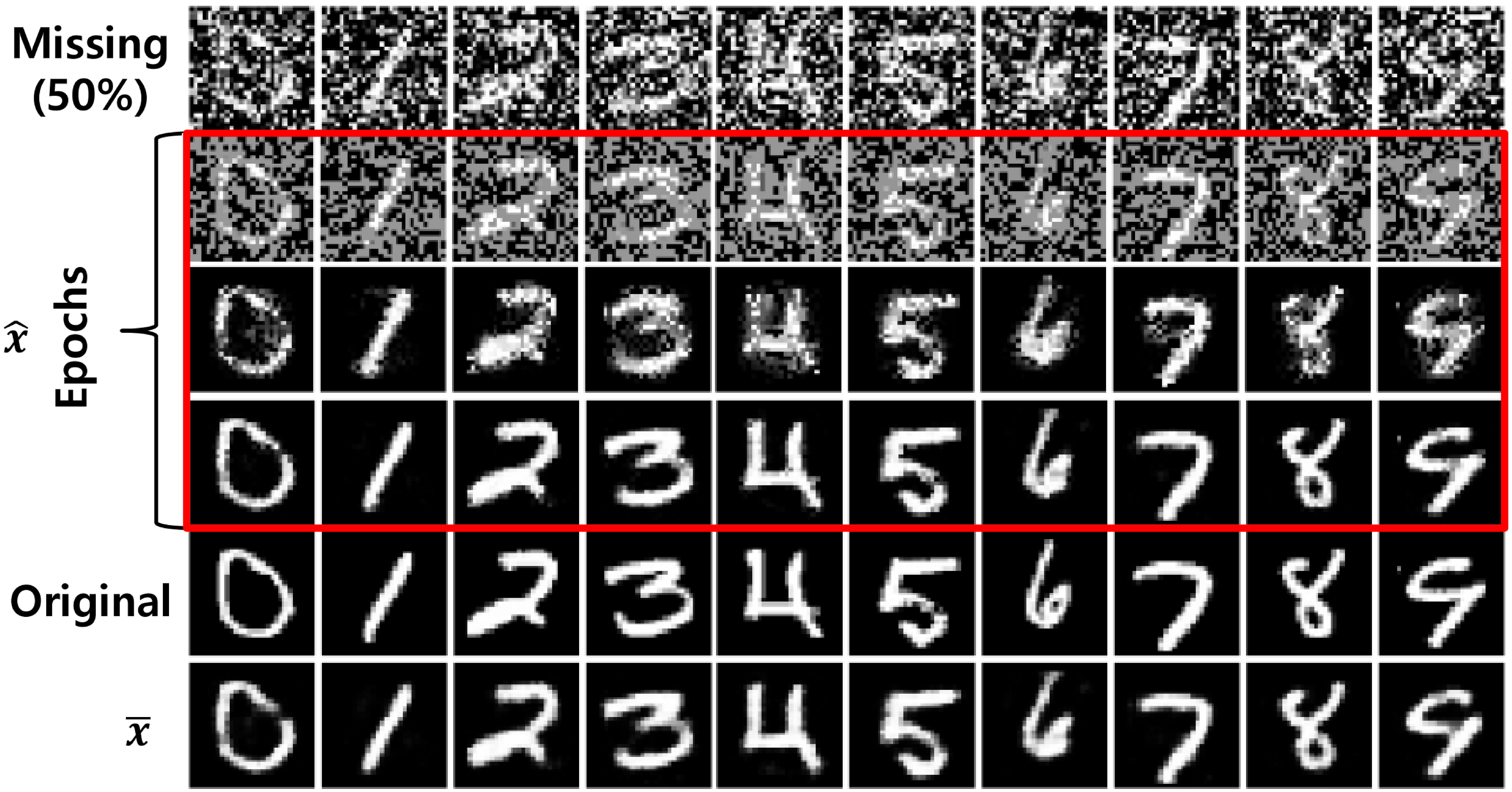
Dataset	# of features	# of instances	Imbalance ratio (1:x)
Breast	30	569	1.68
Credit	23	30,000	3.52
Wine (with binarized class)	13	178	2.02
Madelon	500	4,400	1.00

Imputation performance

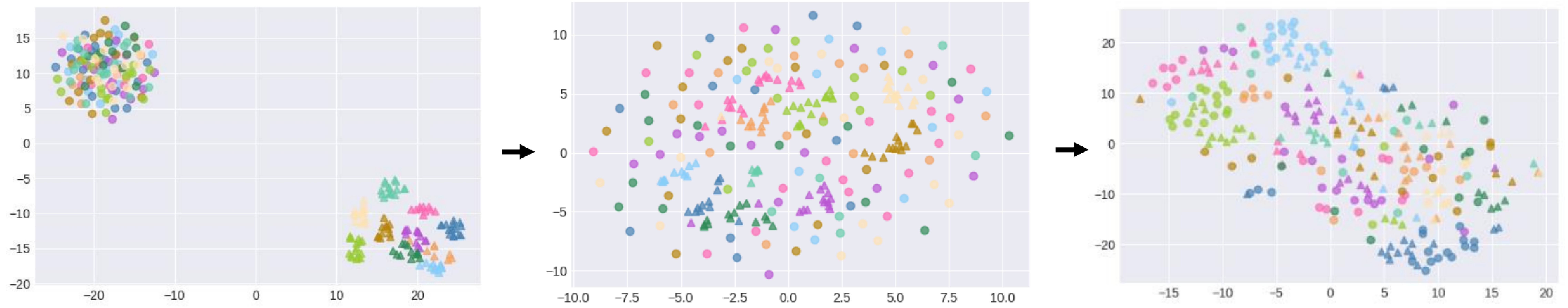
Table 1. Performance comparison with other imputation methods (RMSE)

Method	Breast	Credit	Wine	Madelon	MNIST
Zeros	0.2699	0.2283	0.4213	0.5156	0.3319
Matrix	0.0976	0.1277	0.1772	0.1456	0.2540
K-NN	0.0872	0.1128	0.1695	0.1530	0.2267
MICE	0.0842	0.1073	0.1708	0.1479	0.2576
Autoencoder	0.0875	0.1073	0.1481	0.1426	0.1506
GAIN	0.0878	0.1059	0.1406	0.1426	0.1481
HexaGAN	**0.0769**	**0.1022**	**0.1372**	**0.1418**	**0.1452**

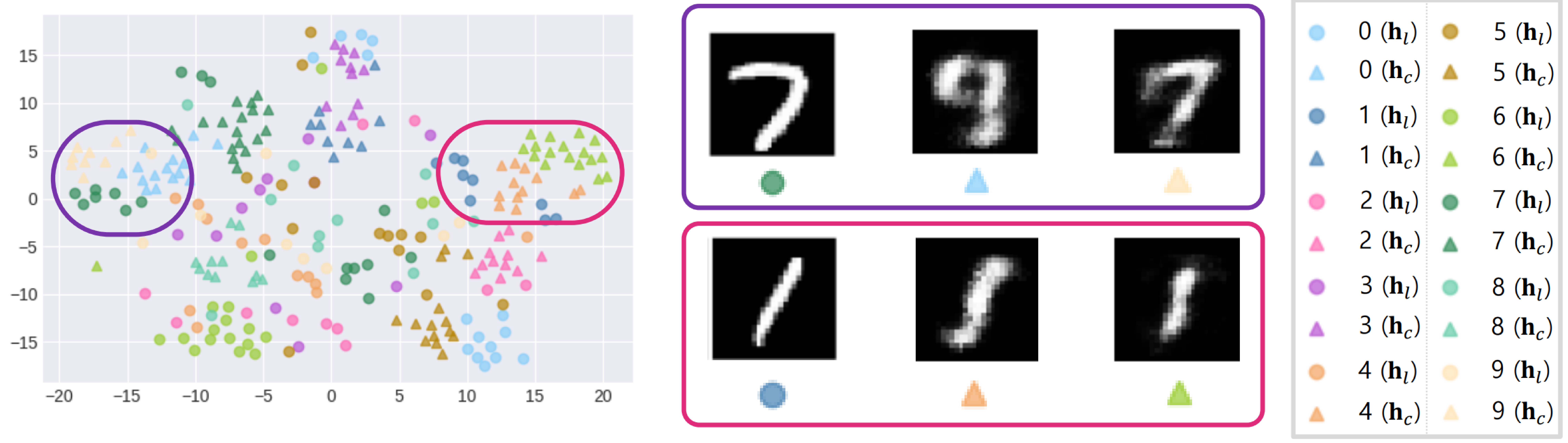
Visualization of imputation performance



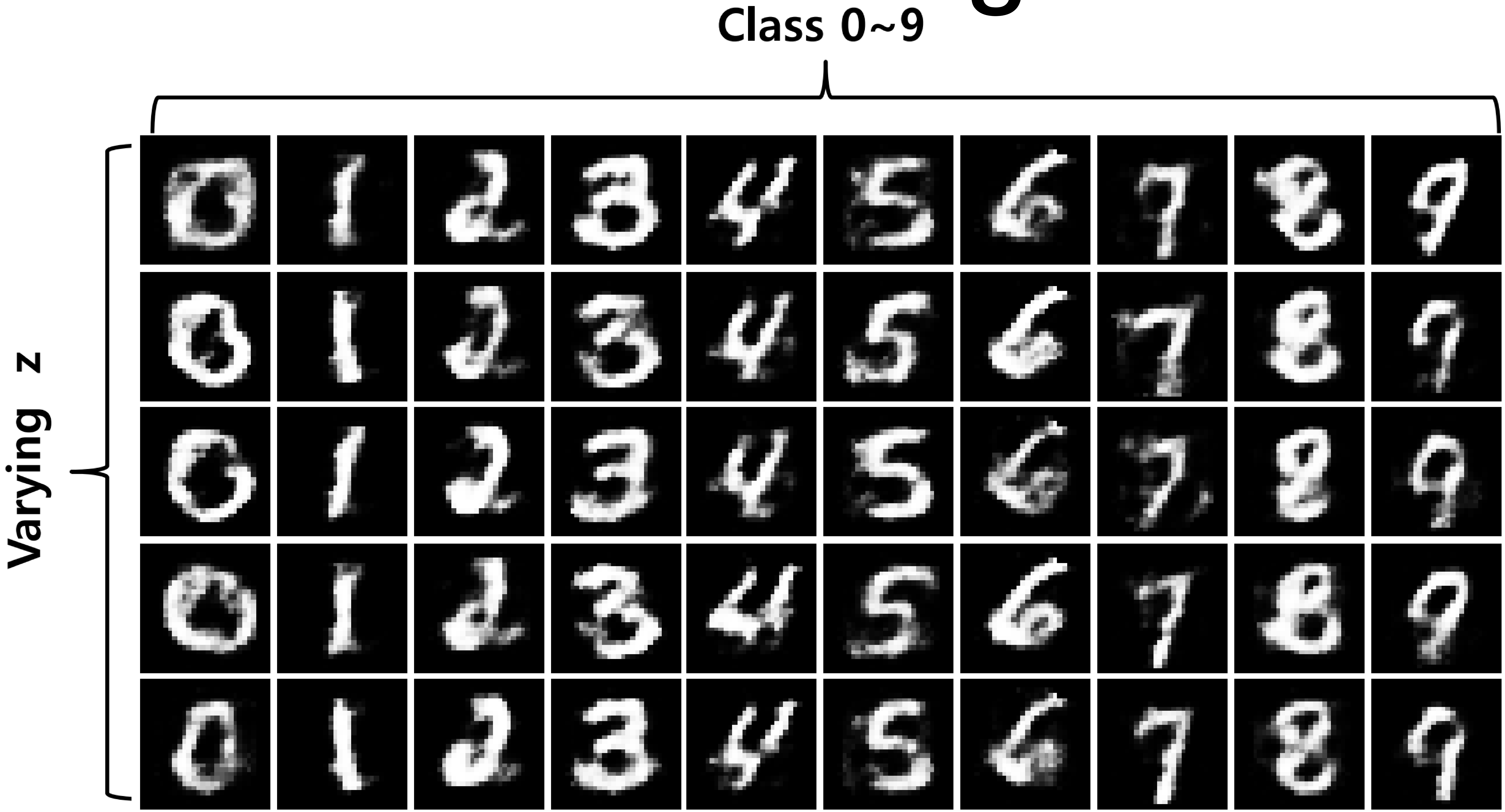
Conditional generation performance



Epoch 1 | Epoch 10 | Epoch 100



Class-conditional generation



Ablation study & Classification performance

- Ablation study

Table 2. Ablation study of HexaGAN (F1-score)

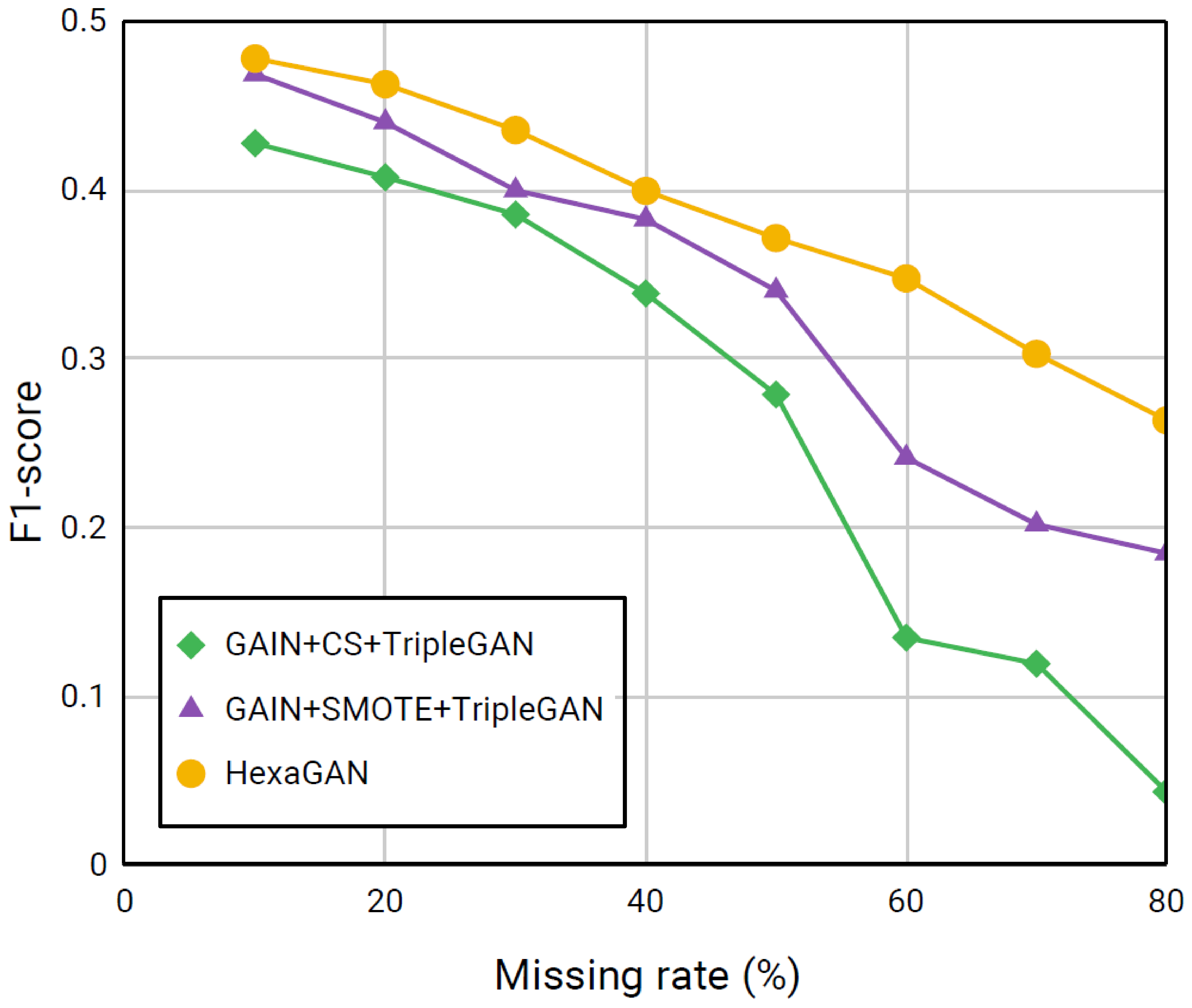
Method	Breast	Credit	Wine	Madelon
MLP (HexaGAN w/o G_{MI} & G_{CG} & $D_{MI_{d+1}}$)	0.9171 ± 0.0101	0.3404 ± 0.0080	0.9368 ± 0.0040	0.6619 ± 0.0017
HexaGAN w/o G_{CG} & $D_{MI_{d+1}}$	0.9725 ± 0.0042	0.4312 ± 0.0028	0.9724 ± 0.0065	0.6676 ± 0.0038
HexaGAN w/o G_{CG}	0.9729 ± 0.0007	0.4382 ± 0.0075	0.9738 ± 0.0135	0.6695 ± 0.0043
HexaGAN w/o $D_{MI_{d+1}}$	0.9750 ± 0.0030	0.4604 ± 0.0097	0.9770 ± 0.0037	0.6699 ± 0.0022
HexaGAN	$\mathbf{0.9762 \pm 0.0021}$	$\mathbf{0.4627 \pm 0.0040}$	$\mathbf{0.9814 \pm 0.0059}$	$\mathbf{0.6716 \pm 0.0019}$

- Classification performance

Table 3. Classification performance (F1-score) comparison with other combinations of state-of-the-art methods

Method	Breast	Credit	Wine	Madelon
MICE + CS + TripleGAN	0.9417 ± 0.0044	0.3836 ± 0.0052	0.9704 ± 0.0043	0.6681 ± 0.0028
GAIN + CS + TripleGAN	0.9684 ± 0.0102	0.4076 ± 0.0038	0.9727 ± 0.0046	0.6690 ± 0.0027
MICE + SMOTE + TripleGAN	0.9434 ± 0.0060	0.4163 ± 0.0029	0.9756 ± 0.0037	0.6712 ± 0.0008
GAIN + SMOTE + TripleGAN	0.9672 ± 0.0063	0.4401 ± 0.0031	0.9735 ± 0.0063	0.6703 ± 0.0032
HexaGAN	$\mathbf{0.9762 \pm 0.0021}$	$\mathbf{0.4627 \pm 0.0040}$	$\mathbf{0.9814 \pm 0.0059}$	$\mathbf{0.6716 \pm 0.0019}$

Classification performance (F1)



Conclusion

- To interactively overcome the three main problems in real world classification (missing data, class imbalance, and missing label), we define the three problems from the perspective of missing information.
- We propose a Hexa-GAN framework wherein six neural networks are actively correlated with others, and design several loss functions that maximize the utilization of any incomplete data
- Our proposed method encourages more powerful performance in both imputation and classification than existing state-of-the-art methods.
- HexaGAN is a one-stop solution that automatically solves the three problems commonly presented in real world classification.
- For future work, we plan to extend HexaGAN to time series datasets such as electronic health records.