HexaGAN: Generative Adversarial Nets for Real World Classification

Uiwon Hwang*, Dahuin Jung, and Sungroh Yoon

Seoul National University
Electrical and Computer Engineering

ICML | 2019

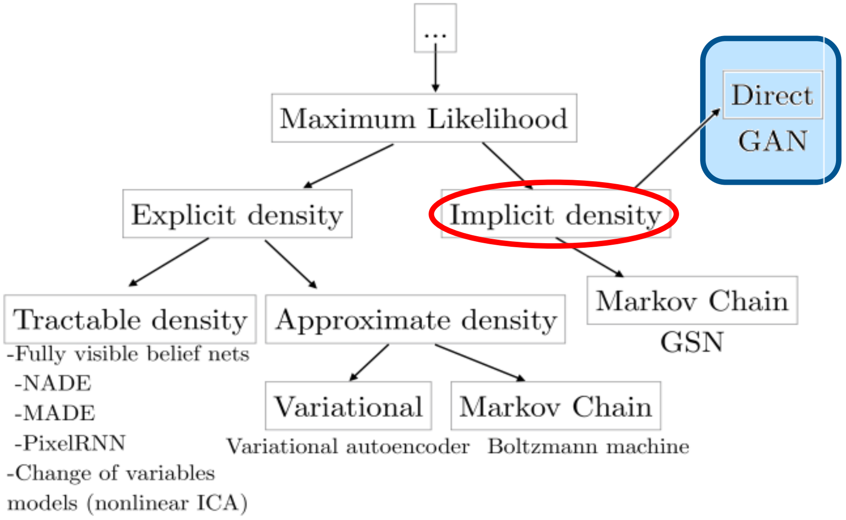
* speaker

Generative Adversarial Networks

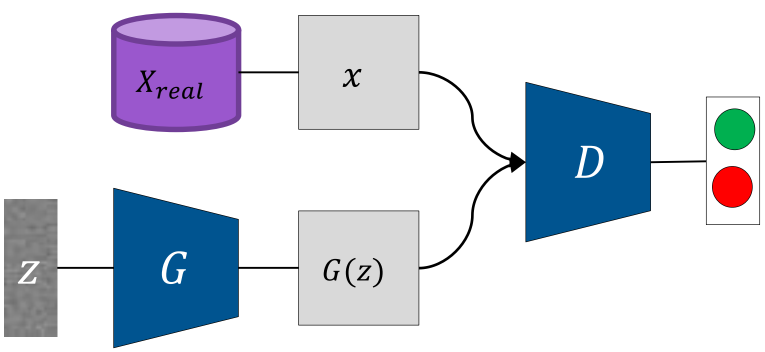
- Generative Adversarial Networks (GANs) (NIPS 2014)

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

  - **Discriminator (D)** learns to discriminate between real and fake data
  - **Generator (G)** learns to generate fake data that can fool the discriminator
  - The optimization of this objective is like finding a **Nash equilibrium** of a **minimax game** between the generator and the discriminator



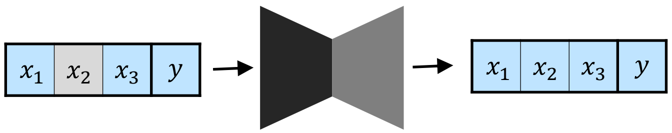
Taxonomy of deep generative models



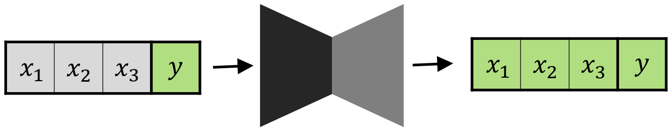
Generative Adversarial Networks

Problems to be Solved

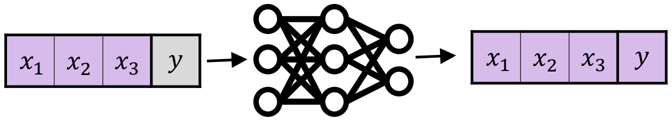
1. **Missing data problem**
   - Missing data imputation
   - Imputing missing elements in a data level



2. **Class imbalance problem**
   - Class-conditional generation
   - Imputing the entire elements of a sample conditioned on a label



3. **Missing label problem**
   - Semi-supervised learning
   - Generating a pseudo-label using a classifier

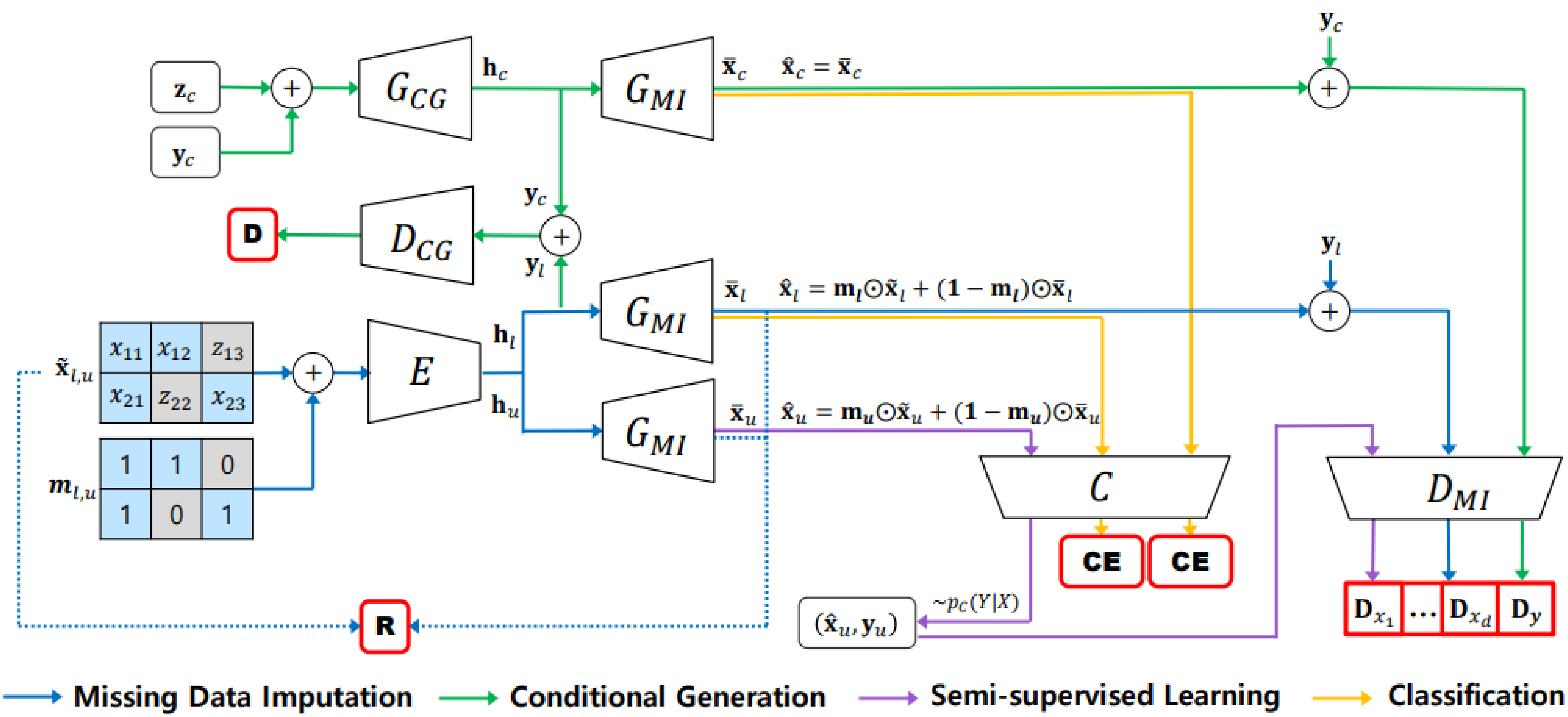


→ Keyword: **Imputation**

Is there a method to address these three problems **concurrently**? Not yet!

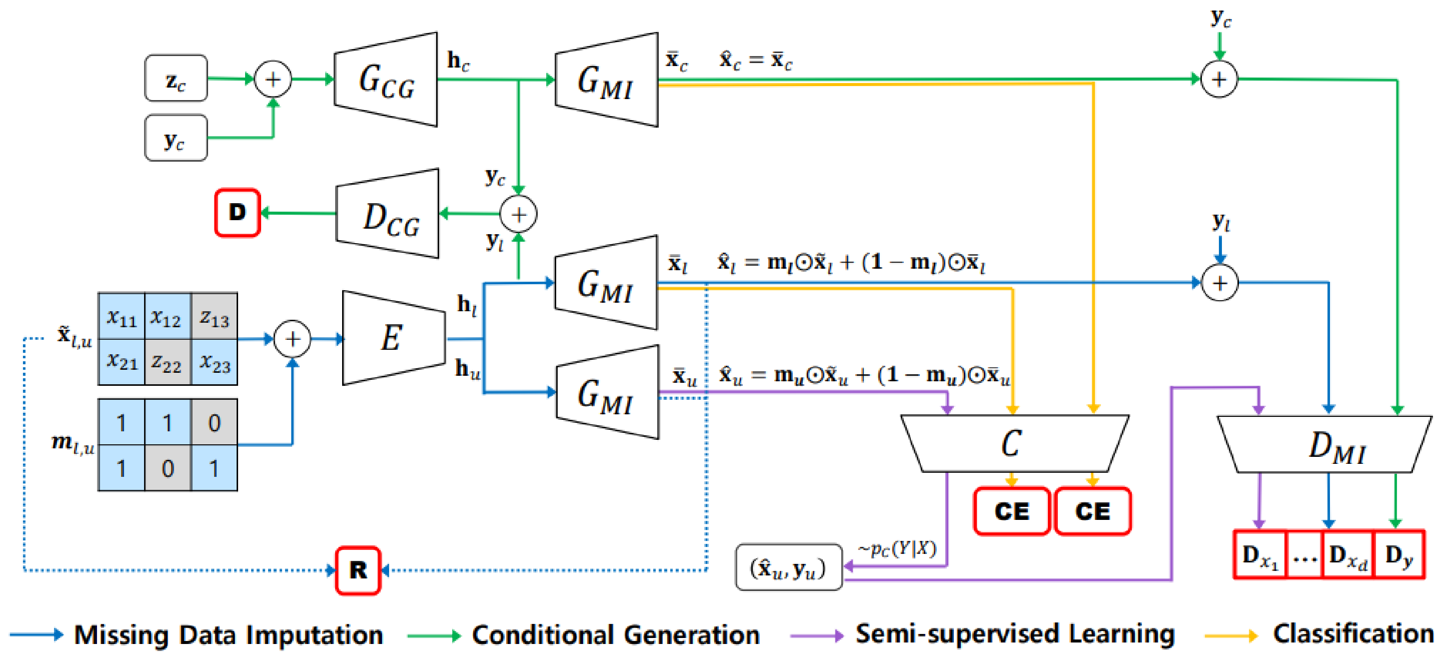
HexaGAN

- HexaGAN
  - We propose a generative adversarial network to solve the problems in real world classification **simultaneously**



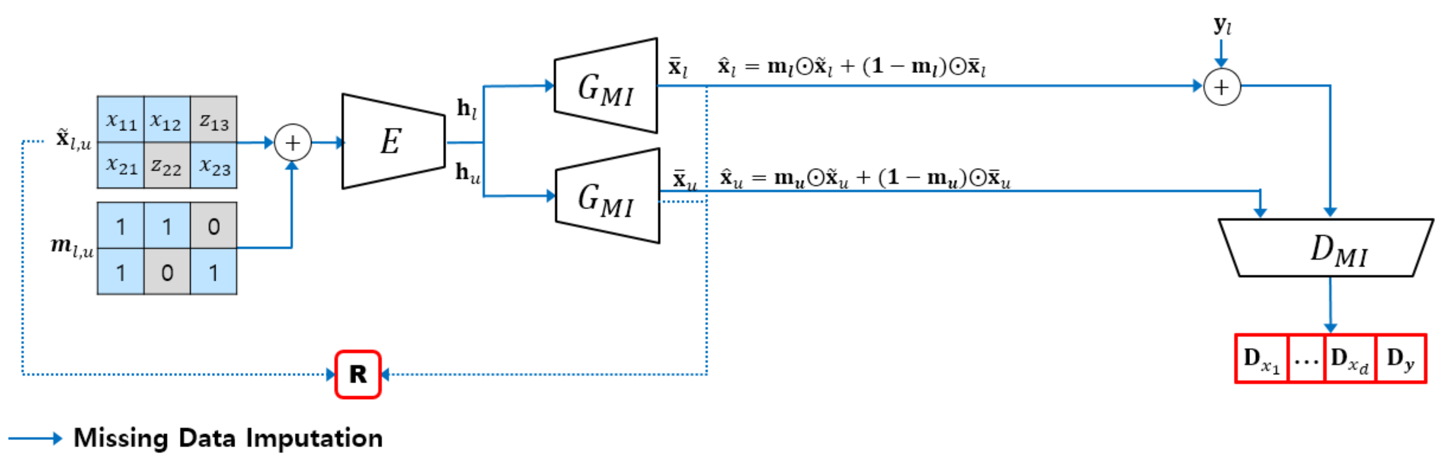
Addressing Three Problems

- **Missing data (element-wise) imputation** (to solve the missing data problem)
  - Components
    - E: transfers both labeled and unlabeled instances into the hidden space
    - G_{MI}: imputes missing data
    - $D_{MI}(\cdot)_{1:d}$: distinguishes b/w missing and non-missing elements



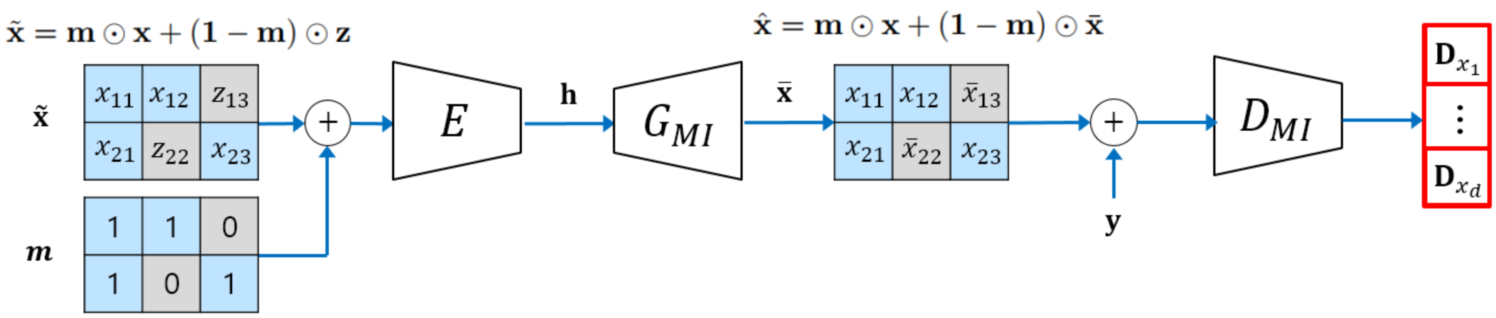
Addressing Three Problems

- **Missing data (element-wise) imputation** (to solve the missing data problem)
  - Components
    - E: transfers both labeled and unlabeled instances into the hidden space
    - G_{MI}: imputes missing data
    - $D_{MI}(\cdot)_{1:d}$: distinguishes b/w missing and non-missing elements



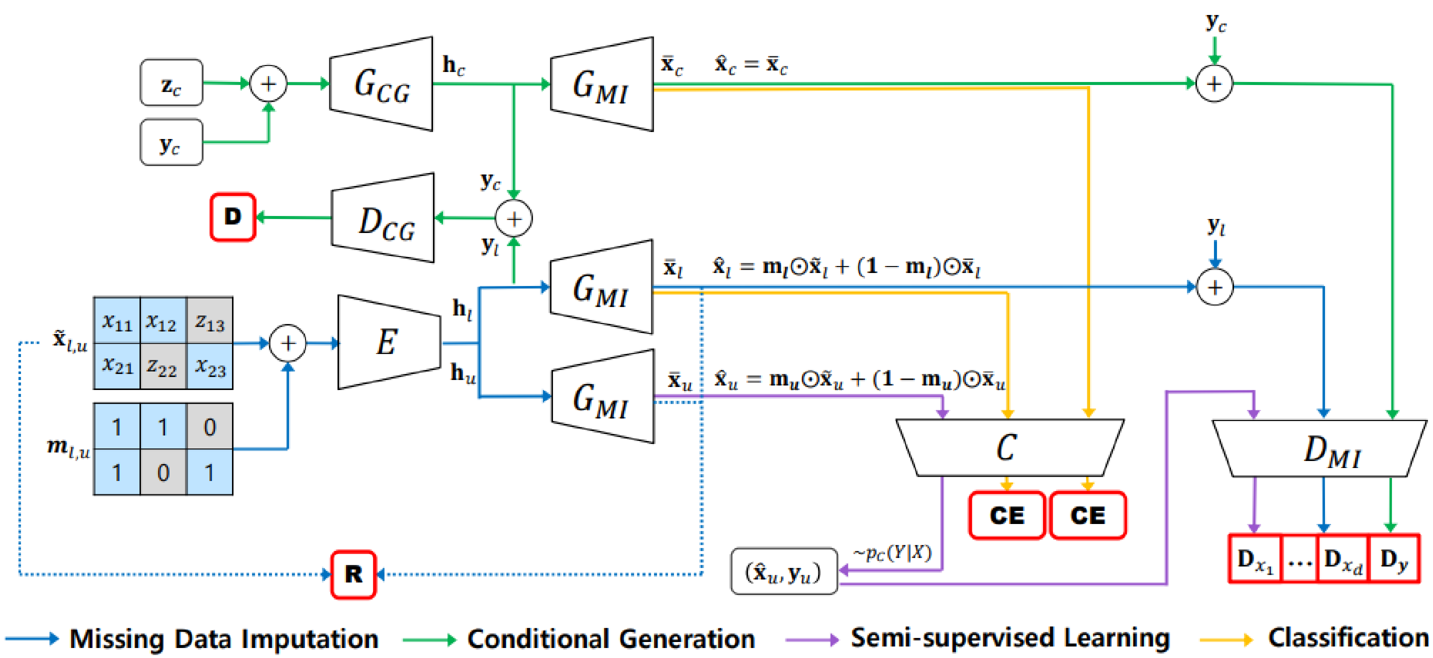
Addressing Three Problems

- **Missing data (element-wise) imputation** (to solve the missing data problem)
  - Components
    - E: transfers both labeled and unlabeled instances into the hidden space
    - G_{MI}: imputes missing data
    - $D_{MI}(\cdot)_{1:d}$: distinguishes b/w missing and non-missing elements
  - A novel element-wise adversarial loss function and gradient penalty
    - $\max_{G_{MI}} \min_{D_{MI}} \sum_{i=1}^{d} \mathbb{E}_{\hat{\mathbf{x}},\mathbf{y},\mathbf{m}}\left[(1-m_i)\cdot D_{MI}(\hat{\mathbf{x}},\mathbf{y})_i\right] - \mathbb{E}_{\hat{\mathbf{x}},\mathbf{y},\mathbf{m}}\left[m_i \cdot D_{MI}(\hat{\mathbf{x}},\mathbf{y})_i\right]$
    - $\mathcal{L}_{\mathrm{GP}_{MI}} = \sum_{i=1}^{d} \mathbb{E}_{p_{\mathcal{D}}(x_i)}\left[\|\nabla_{\hat{\mathbf{x}}} D_{MI}(\hat{\mathbf{x}})_i\|_2^2\right]$



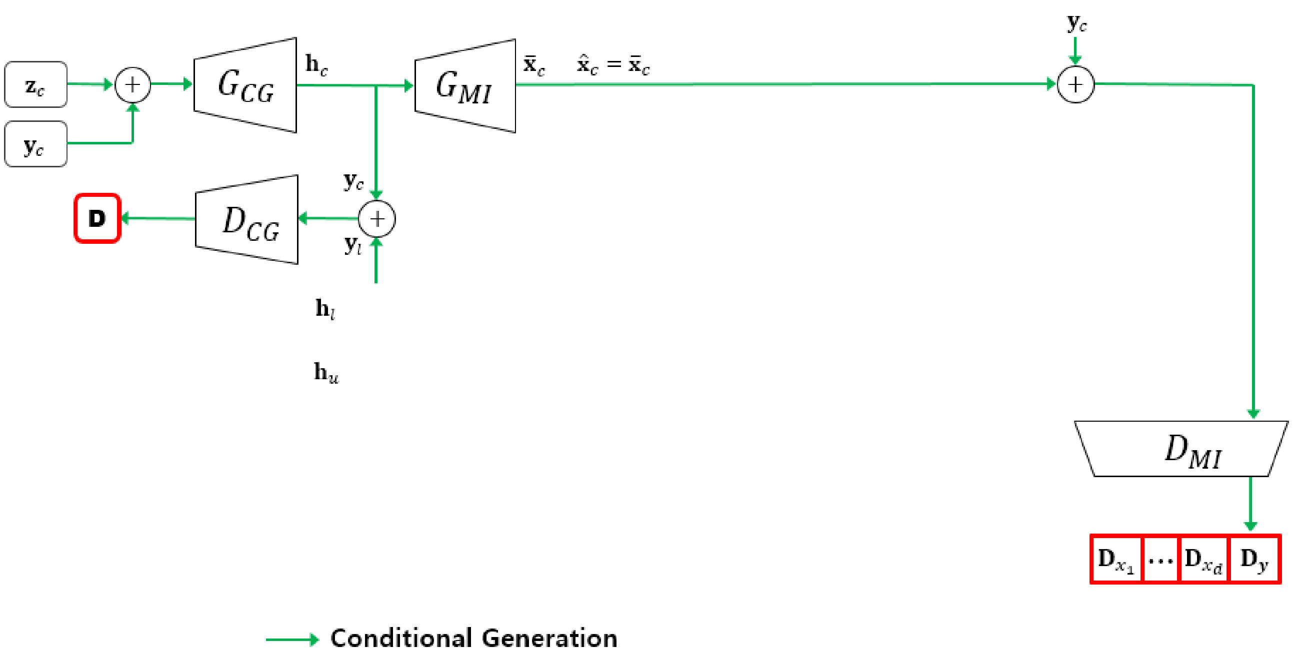
Addressing Three Problems (Cont'd)

- **Class conditional generation** (to solve the class imbalance problem)
  - Components
    - G_{CG}: creates conditional hidden vectors $\mathbf{h}_c$
    - D_{CG}: determines whether a hidden vector is from the dataset or has been created by G_{CG}
  - G_{MI} generates the entire elements conditioned on the minority class



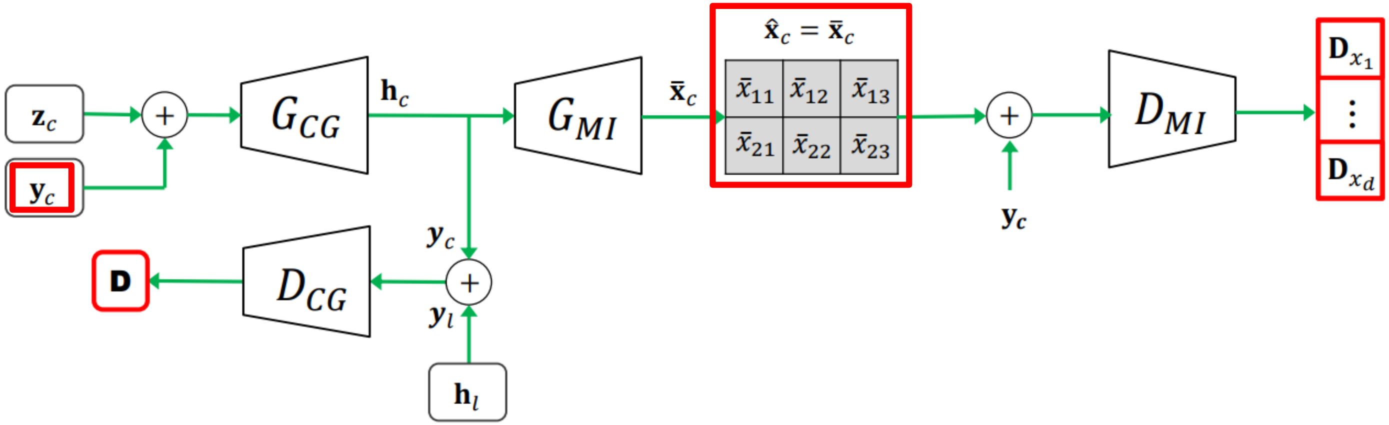
Addressing Three Problems (Cont'd)

- **Class conditional generation** (to solve the class imbalance problem)
  - Components
    - G_{CG}: creates conditional hidden vectors $\mathbf{h}_c$
    - D_{CG}: determines whether a hidden vector is from the dataset or has been created by G_{CG}
  - G_{MI} generates the entire elements conditioned on the minority class



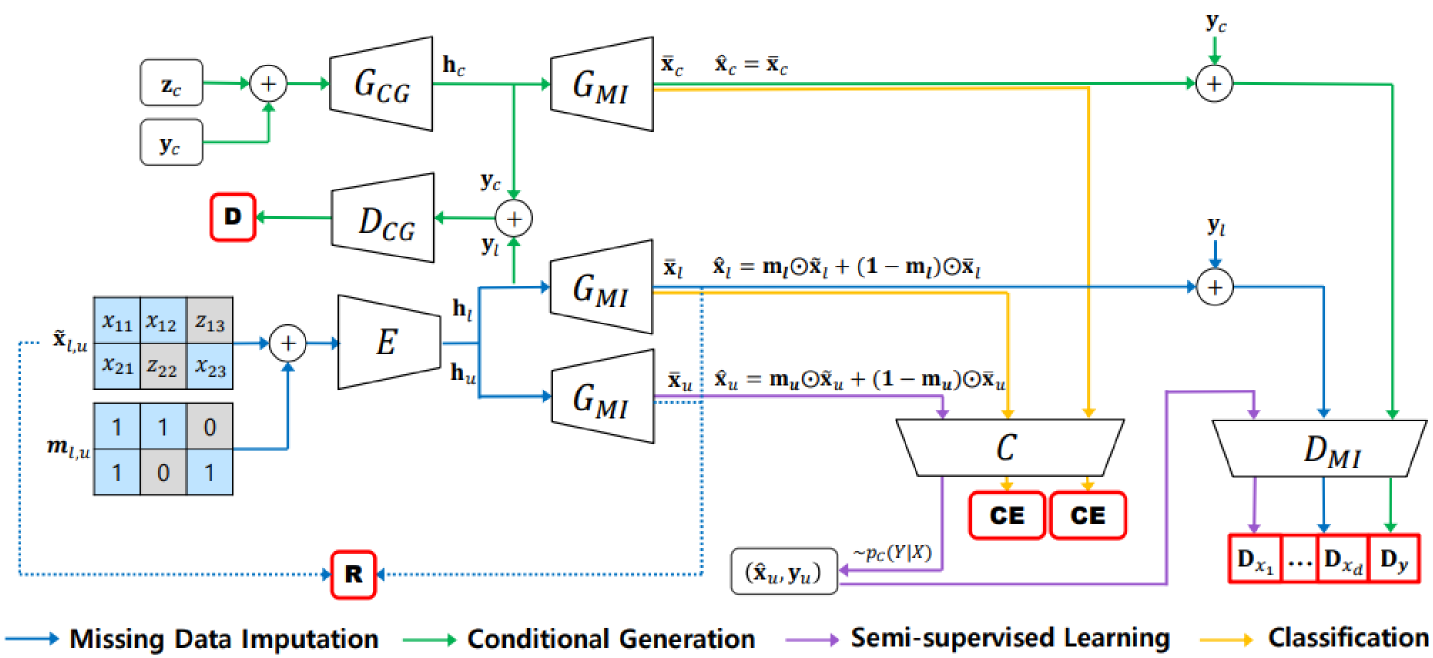
Addressing Three Problems (Cont'd)

- **Class conditional generation** (to solve the class imbalance problem)
  - Components
    - G_{CG}: creates conditional hidden vectors $\mathbf{h}_c$
    - D_{CG}: determines whether a hidden vector is from the dataset or has been created by G_{CG}
  - G_{MI} generates the entire elements conditioned on the minority class
  - Losses
    - WGAN loss + zero-centered gradient penalty
    - Add Loss of G_{MI} calculated from $\hat{\mathbf{x}}_c$ and the cross entropy of $(\hat{\mathbf{x}}, \mathbf{y}_c)$ to G_{CG}



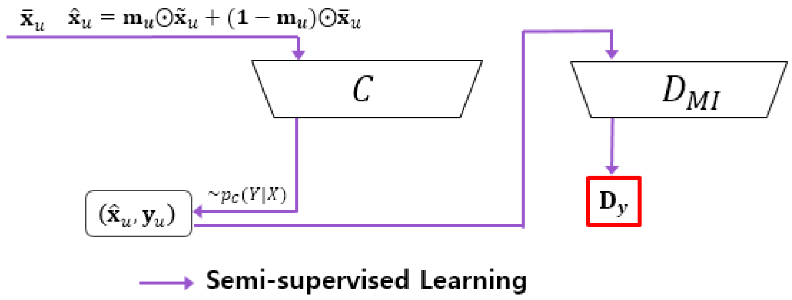
Addressing Three Problems (Cont'd)

- **Semi-supervised learning** (to solve the missing label problem)
  - Components
    - C: estimates class labels. This also works as the label generator
    - $D_{MI}(\cdot)_{d+1}$: distinguishes b/w real and pseudo (fake) labels
  - We adopt the pseudo-labeling technique of TripleGAN (NIPS 2017)



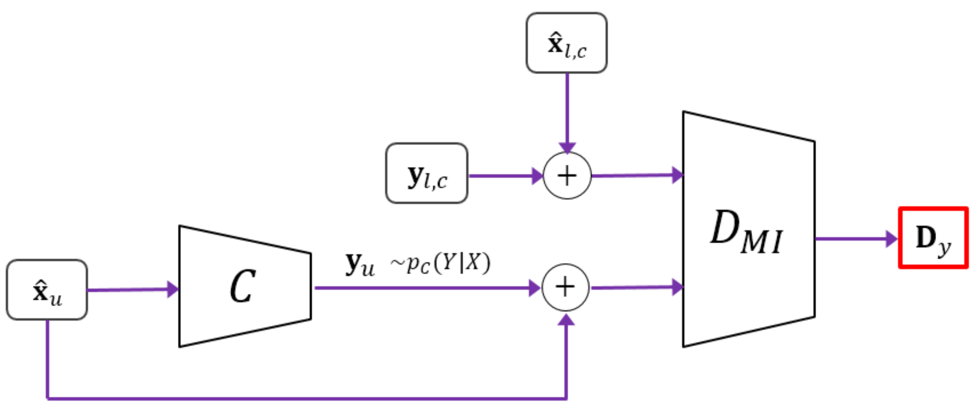
Addressing Three Problems (Cont'd)

- **Semi-supervised learning** (to solve the missing label problem)
  - Components
    - C: estimates class labels. This also works as the label generator
    - $D_{MI}(\cdot)_{d+1}$: distinguishes b/w real and pseudo (fake) labels
  - We adopt the pseudo-labeling technique of TripleGAN (NIPS 2017)



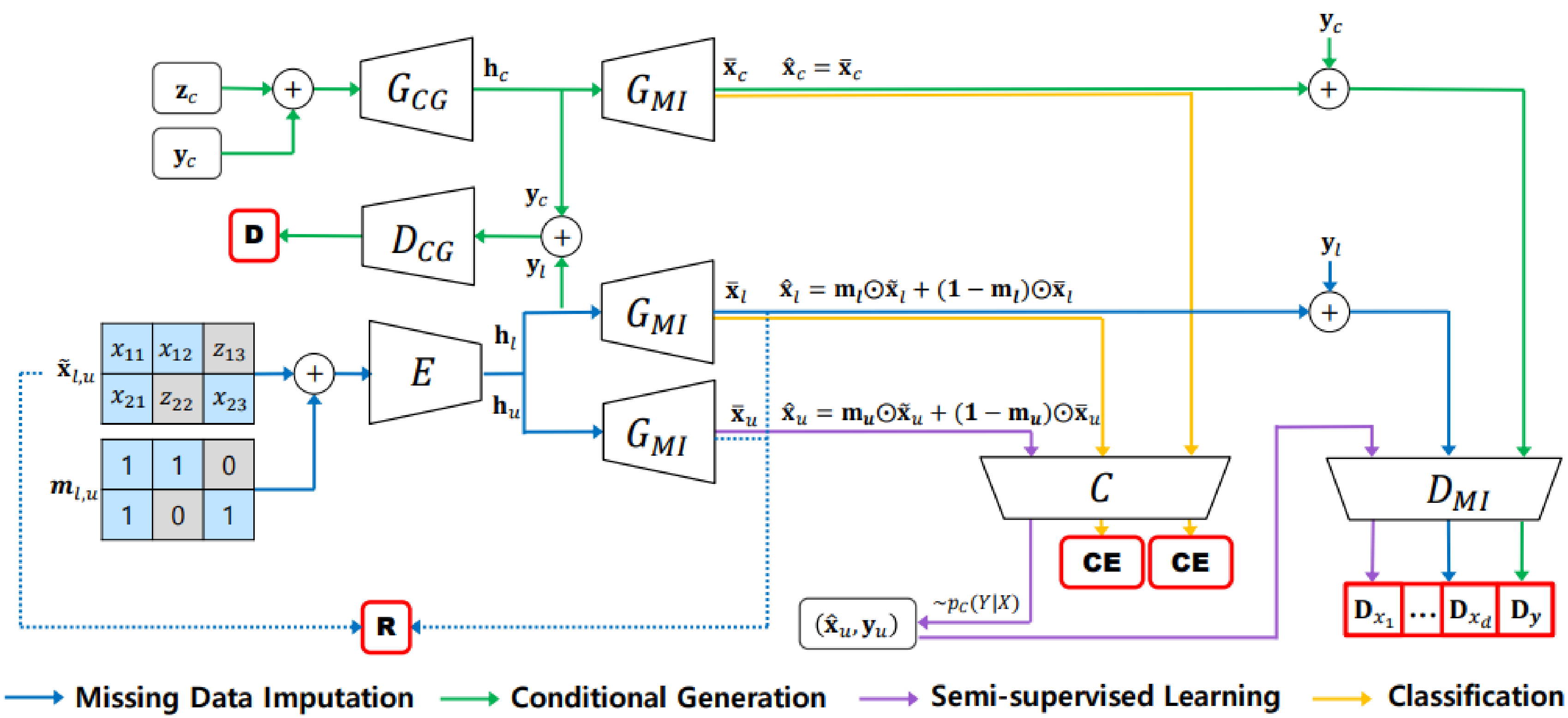
Addressing Three Problems (Cont'd)

- **Semi-supervised learning** (to solve the missing label problem)
  - Components
    - C: estimates class labels. This also works as the label generator
    - $D_{MI}(\cdot)_{d+1}$: distinguishes b/w real and pseudo (fake) labels
  - We adopt the pseudo-labeling technique of TripleGAN (NIPS 2017)
  - The two components are related adversarially
    - $L_C = -\mathbb{E}_{\mathbf{y}_u|\hat{\mathbf{x}}_u \sim p_C}\left[D_{MI}(\hat{\mathbf{x}}_u, \mathbf{y}_u)_{d+1}\right]$
    - $L_{D_{MI}}^{d+1} = \mathbb{E}_{\mathbf{y}_u|\hat{\mathbf{x}}_u \sim p_C}\left[D_{MI}(\hat{\mathbf{x}}_u, \mathbf{y}_u)_{d+1}\right] - \mathbb{E}_{\mathbf{y}|\hat{\mathbf{x}} \sim p_{data}}\left[D_{MI}(\hat{\mathbf{x}}, \mathbf{y})_{d+1}\right]$



Overview of HexaGAN (Revisited)

- Not three separate models, this is ONE model dubbed **HexaGAN**
- The six components of HexaGAN **interplay to solve the problems effectively**



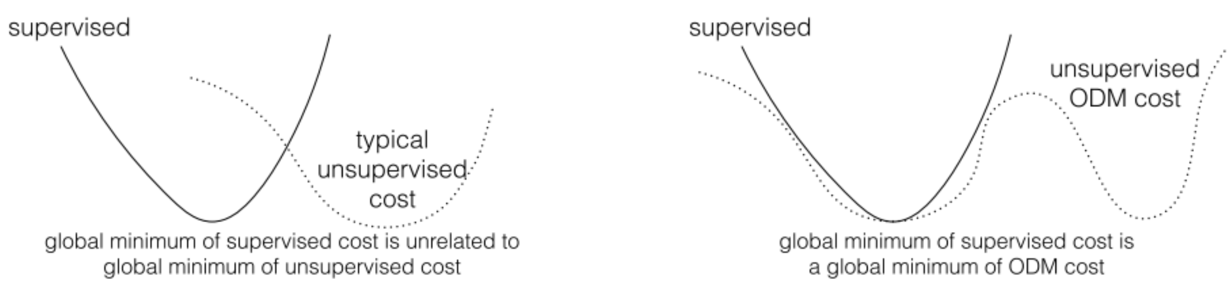
Theorems

- **Theorem 1**: Global optimality of $p(x|m_i = 1) = p(x|m_i = 0)$ for HexaGAN
  - *A generator distribution* $p(x|m_i = 0)$ *is **a global optimum for the min-max game of*** $\boldsymbol{G_{MI}}$ ***and*** $\boldsymbol{D_{MI}}$*, if and only if* $\boldsymbol{p(x|m_i = 1) = p(x|m_i = 0)}$ *for all* $\boldsymbol{x} \in \mathbb{R}^d$*, except possibly on a set of zero Lebesgue measure.*

- **Theorem 2**: The adversarial loss for semi-supervised learning is the ODM cost
  - Output distribution matching (ODM) cost function (ICLR workshop 2016)

$$\boxed{\mathrm{Distr}\left[F(x)\right] = \mathrm{Distr}\left[y\right]}$$

    - the global minimum of the supervised cost function is also a global minimum of the ODM cost function



  - *Optimizing the adversarial losses for C and* $D_{MI}(\cdot)_{d+1}$ *imposes an unsupervised constraint on C. Then, the adversarial losses for **semi-supervised learning in HexaGAN satisfy the definition of the ODM cost.***

$$W(\mathrm{Distr}[C(\hat{\mathbf{x}}_u)], \mathrm{Distr}[\mathbf{y}]) \to 0 \quad \Rightarrow \quad \boxed{\mathrm{Distr}[C(\hat{\mathbf{x}}_u)] = \mathrm{Distr}[\mathbf{y}]}$$

Training Procedure

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)
    $\nabla_{D_{MI}} \mathcal{L}_{D_{MI}} + \mathcal{L}^{d+1}_{D_{MI}} + \lambda_1 \mathcal{L}_{\mathrm{GP}_{MI}}$
  end for
  Update E using SGD
  $\nabla_E \mathcal{L}_{G_{MI}} + \alpha_1 \mathcal{L}_{\mathrm{recon}}$
  Update G_{MI} using SGD
  $\nabla_{G_{MI}} \mathcal{L}_{G_{MI}} + \alpha_1 \mathcal{L}_{\mathrm{recon}}$

  (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

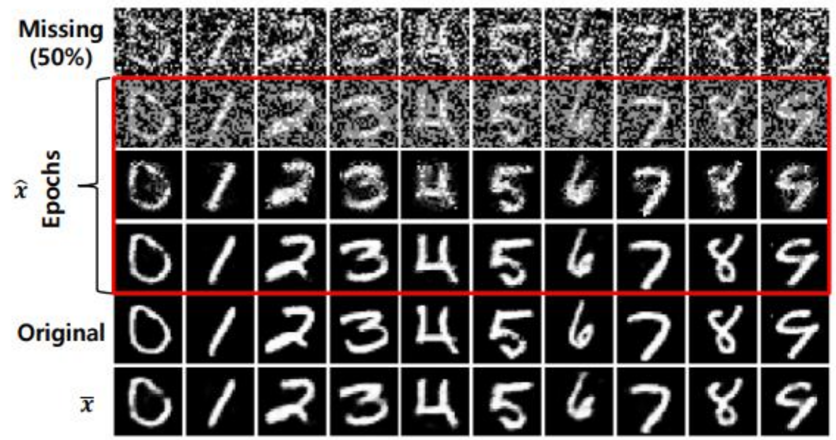
  (3) Semi-supervised classification
  Update C using SGD
  $\nabla_C \mathcal{L}_{\mathrm{CE}}(\hat{\mathbf{x}}_{l,c}, \mathbf{y}_{l,c}) + \alpha_4 \mathcal{L}_C$
end while

Experimental Results

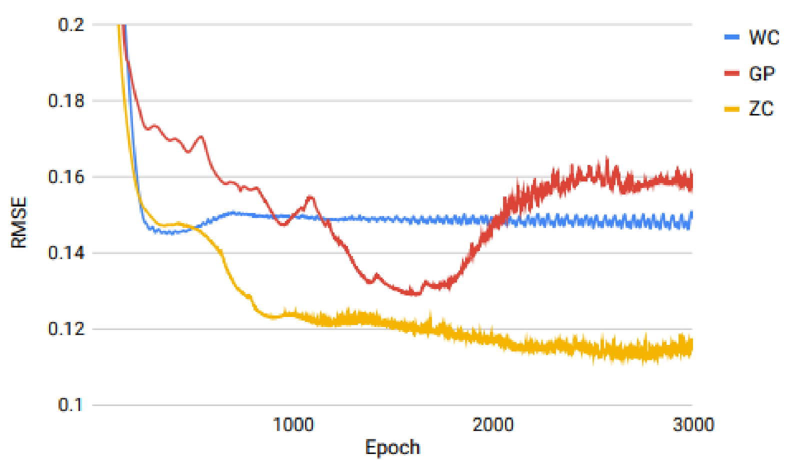
- **Missing data imputation**
  - HexaGAN shows good performances on various real world datasets
    - Medical, financial, vision, …

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**

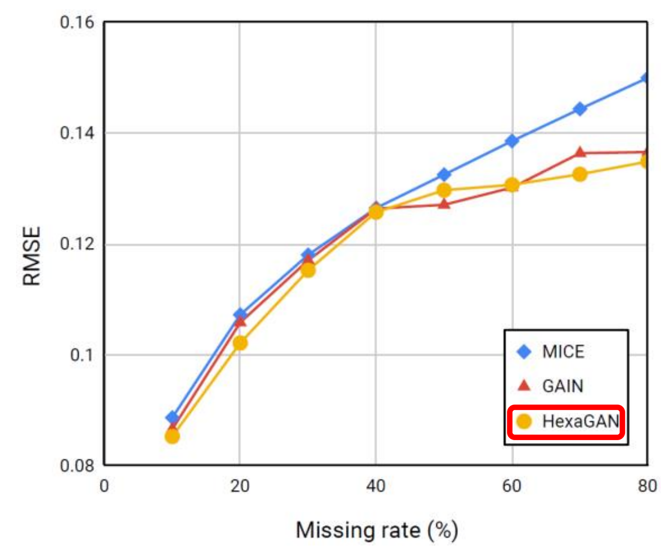
Imputation performance (RMSE)



Qualitative analysis (MNIST)

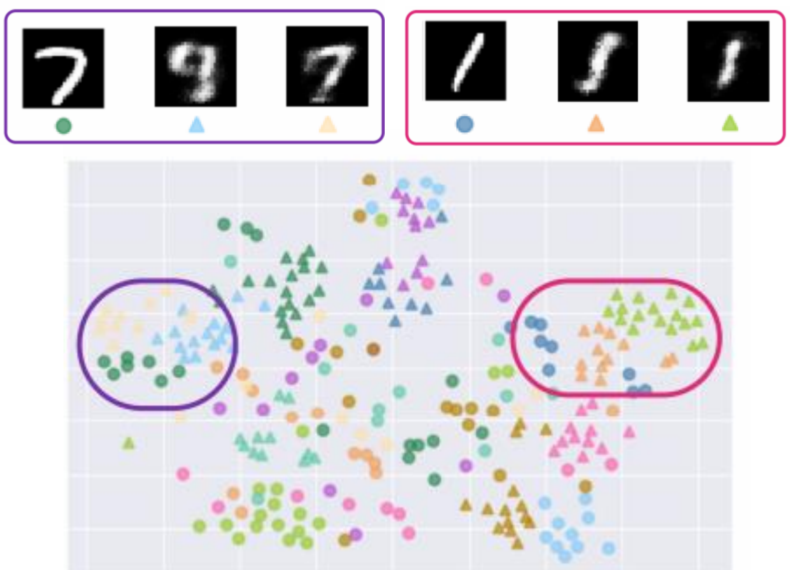


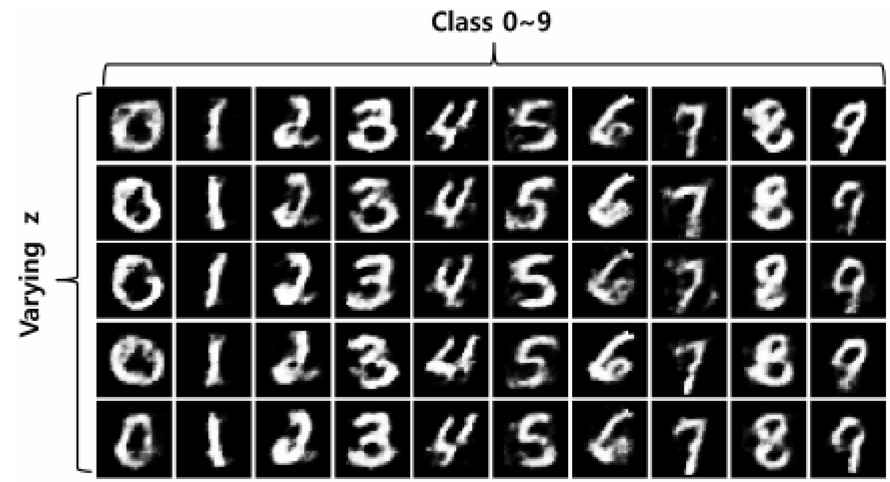
Comparison of the gradient penalty



Experimental Results (Cont'd)

- **Class conditional generation**

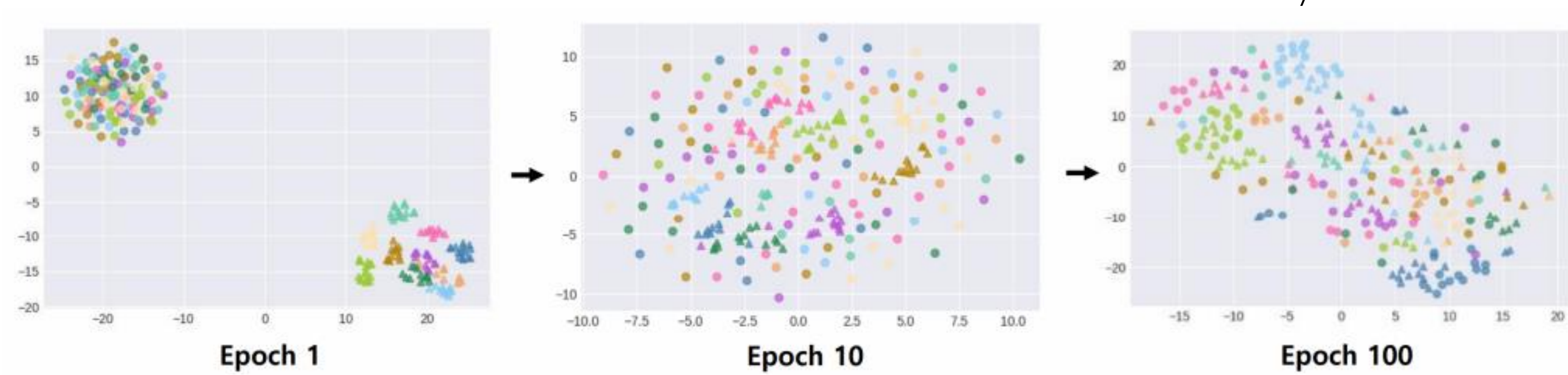




Qualitative analysis (MNIST)

Hyperparameter (Loss)	Setting	1	2	3	4
α_2 ($\mathcal{L}_{G_{MI}}$)	Value	0	**1**	10	100
	F1-score	0.4535	**0.4627**	0.4585	0.4523
α_3 ($\mathcal{L}_{\text{CE}}(\hat{\mathbf{x}}_c, \mathbf{y}_c)$)	Value	0	**0.01**	0.1	1
	F1-score	0.4535	**0.4627**	0.4585	0.4523

Ablation study



Analysis of the hidden space (tSNE)

Experimental Results (Cont'd)

- **Semi-supervised learning (classification)**

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	**0.9762 ±0.0021**	**0.4627 ±0.0040**	**0.9814 ±0.0059**	**0.6716 ±0.0019**

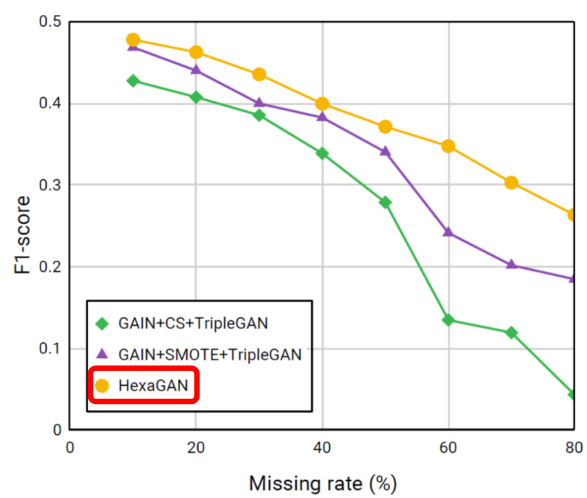
Ablation study (F1-score)

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	**0.9762 ±0.0021**	**0.4627 ±0.0040**	**0.9814 ±0.0059**	**0.6716 ±0.0019**

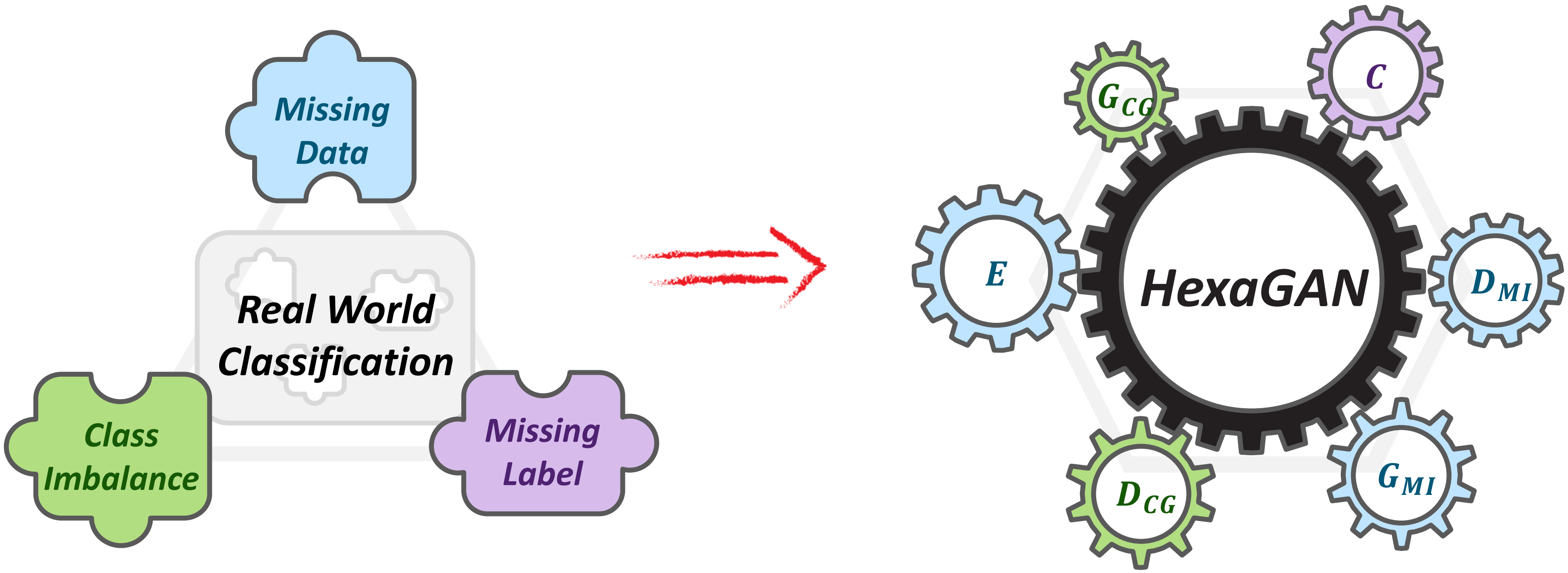
Classification performance (F1-score)

Attribute	Imb. ratio (1:x)	GAIN + TripleGAN	GAIN + CRL + TripleGAN	HexaGAN
Arched eyebrows	3	0.53	0.50	**0.55**
Attractive	1	**0.78**	0.74	0.74
Bags under eyes	4	0.30	0.44	**0.49**
⋮				
Mean	-	0.5152	0.5519	**0.5826**

Classification performance (CelebA)



Conclusion



Paper

TF Code (Github)


Thank You!