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

Problem Definition

- **Missing data problem**

- **Class imbalance problem**

- **Missing label problem**

Problem Definition

- **Missing data problem**
  - Missing data imputation
  - filling missing elements in a data level

- Class imbalance problem

- Missing label problem

Problem Definition

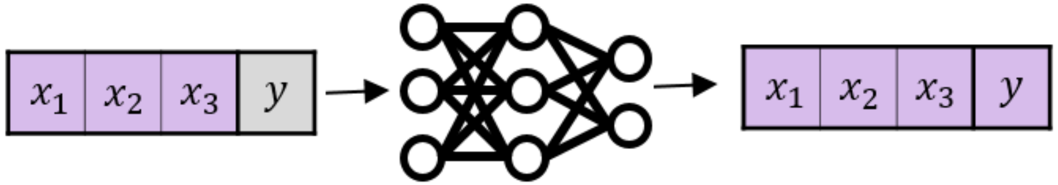
- Missing data problem
  - Missing data imputation
  - filling missing elements in a data level
- **Class imbalance problem**
  - Class-conditional generation
  - Imputing the entire elements of a sample conditioned on a label
- Missing label problem

Problem Definition

- Missing data problem
  - Missing data imputation
  - filling missing elements in a data level

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

- **Missing label problem**
  - Semi-supervised learning
  - Imputing missing class labels using a classifier

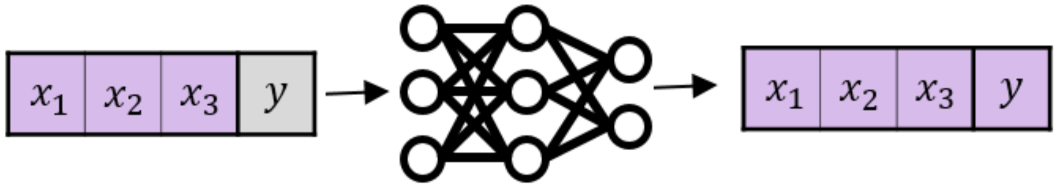


Problem Definition

- **Missing data problem**
  - Missing data imputation
  - filling missing elements in a data level

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

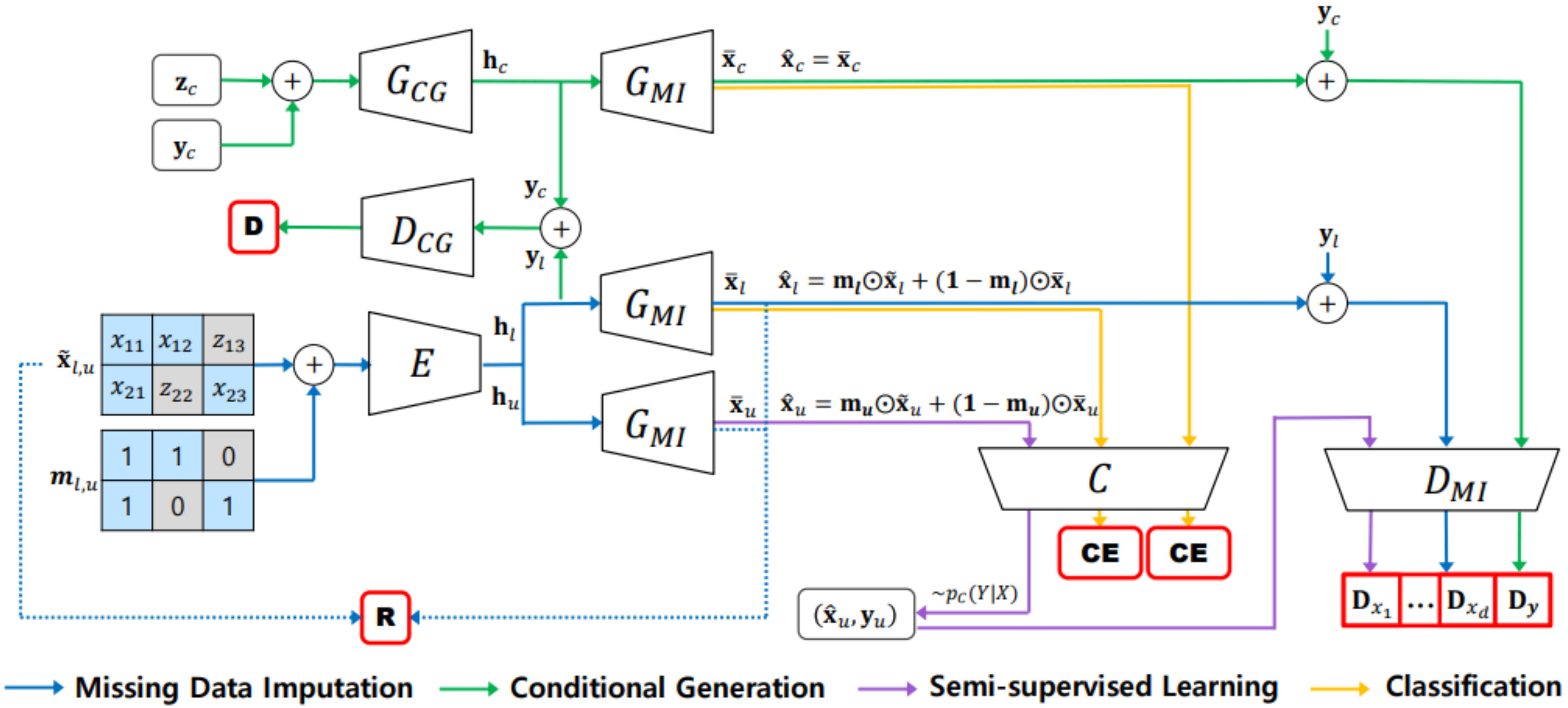
- **Missing label problem**
  - Semi-supervised learning
  - Imputing missing class labels using a classifier



→ **Keyword: Imputation**

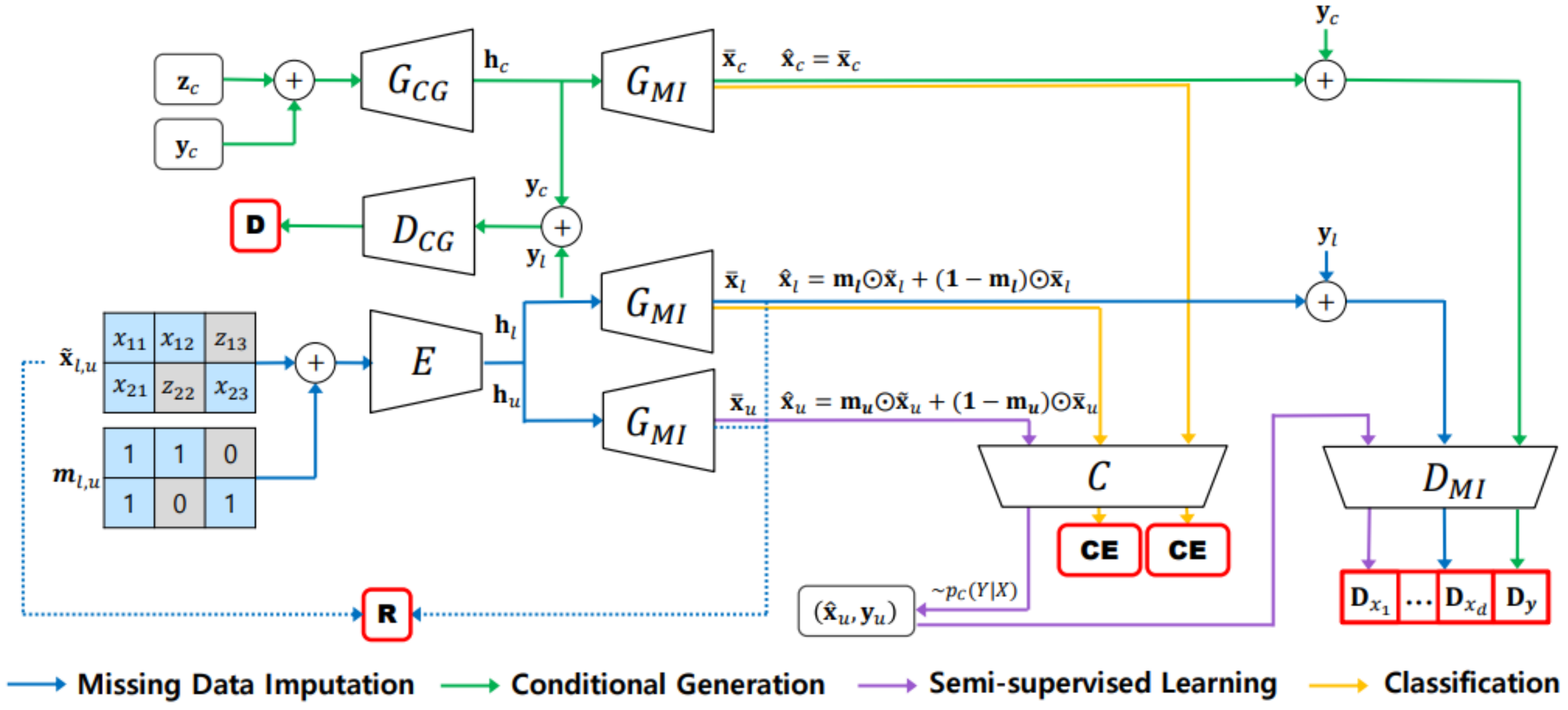
Overview of HexaGAN

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



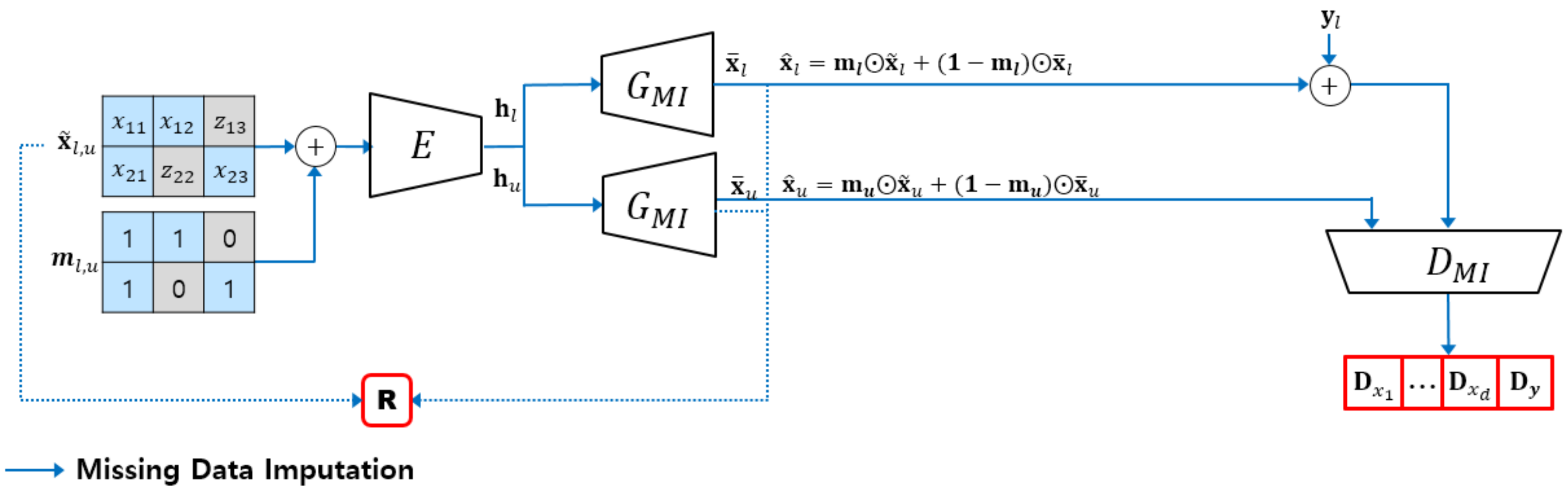
Addressing Three Problems

- **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



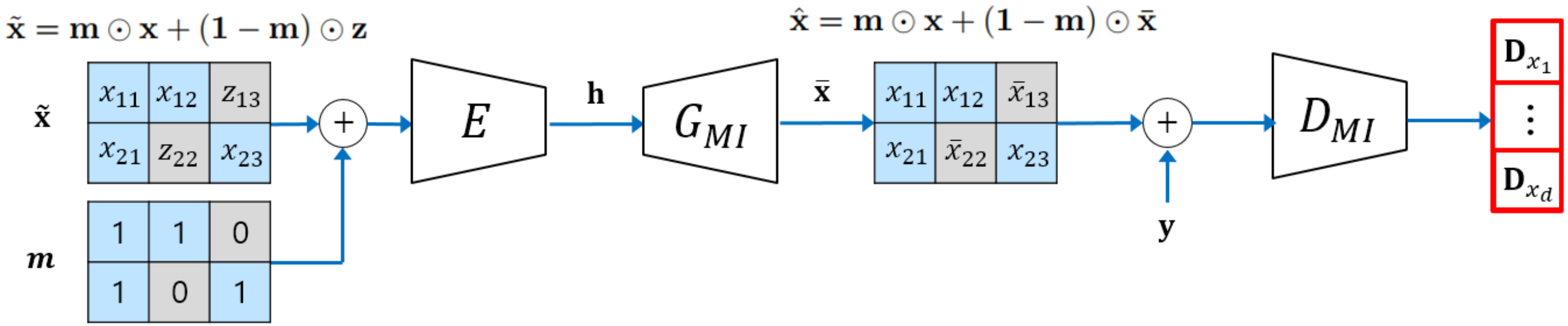
Addressing Three Problems

- **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

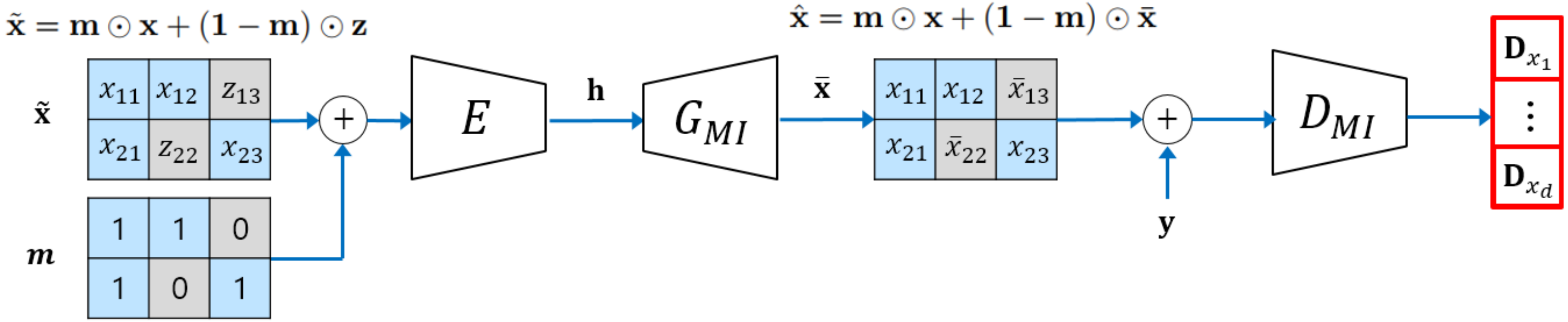
- **Missing data (element-wise) imputation** (to solve the missing data problem)
  - Components
    - E: transfers both labeled and unlabeled instances into the hidden space
    - G_{MI}: imputes missing data
    - $D_{MI}(\cdot)_{1:d}$: distinguishes b/w missing and non-missing elements



Addressing Three Problems

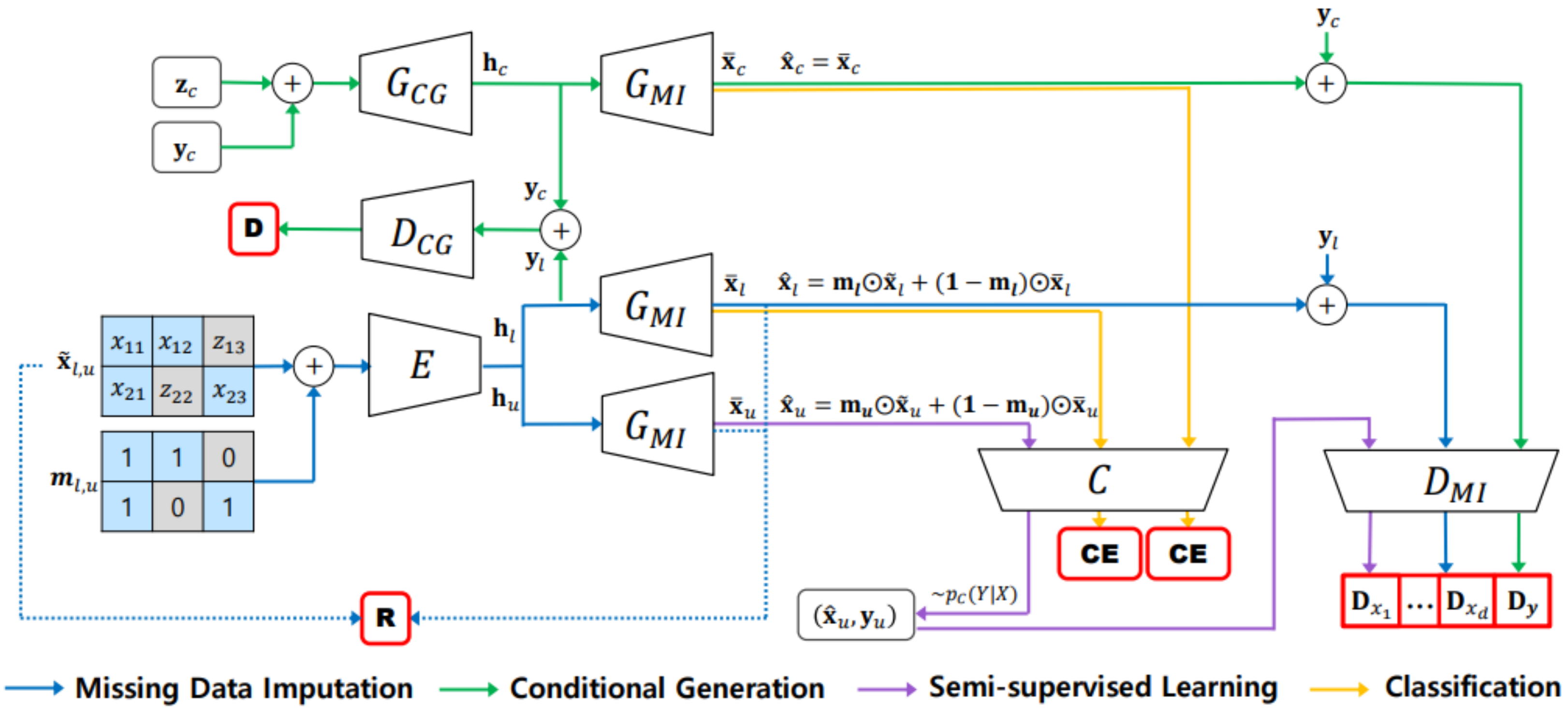
DSAIL Data Science & AI Laboratory

- **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
    - $\max_{G_{MI}} \min_{D_{MI}} \sum_{i=1}^{d} \mathbb{E}_{\hat{\mathbf{x}},\mathbf{y},\mathbf{m}} [(1-m_i) \cdot D_{MI}(\hat{\mathbf{x}}, \mathbf{y})_i] - \mathbb{E}_{\hat{\mathbf{x}},\mathbf{y},\mathbf{m}} [m_i \cdot D_{MI}(\hat{\mathbf{x}}, \mathbf{y})_i]$
    - $\mathcal{L}_{\mathrm{GP}_{MI}} = \sum_{i=1}^{d} \mathbb{E}_{p_{\mathcal{D}}(x_i)} [||\nabla_{\hat{\mathbf{x}}} D_{MI}(\hat{\mathbf{x}})_i||_2^2]$

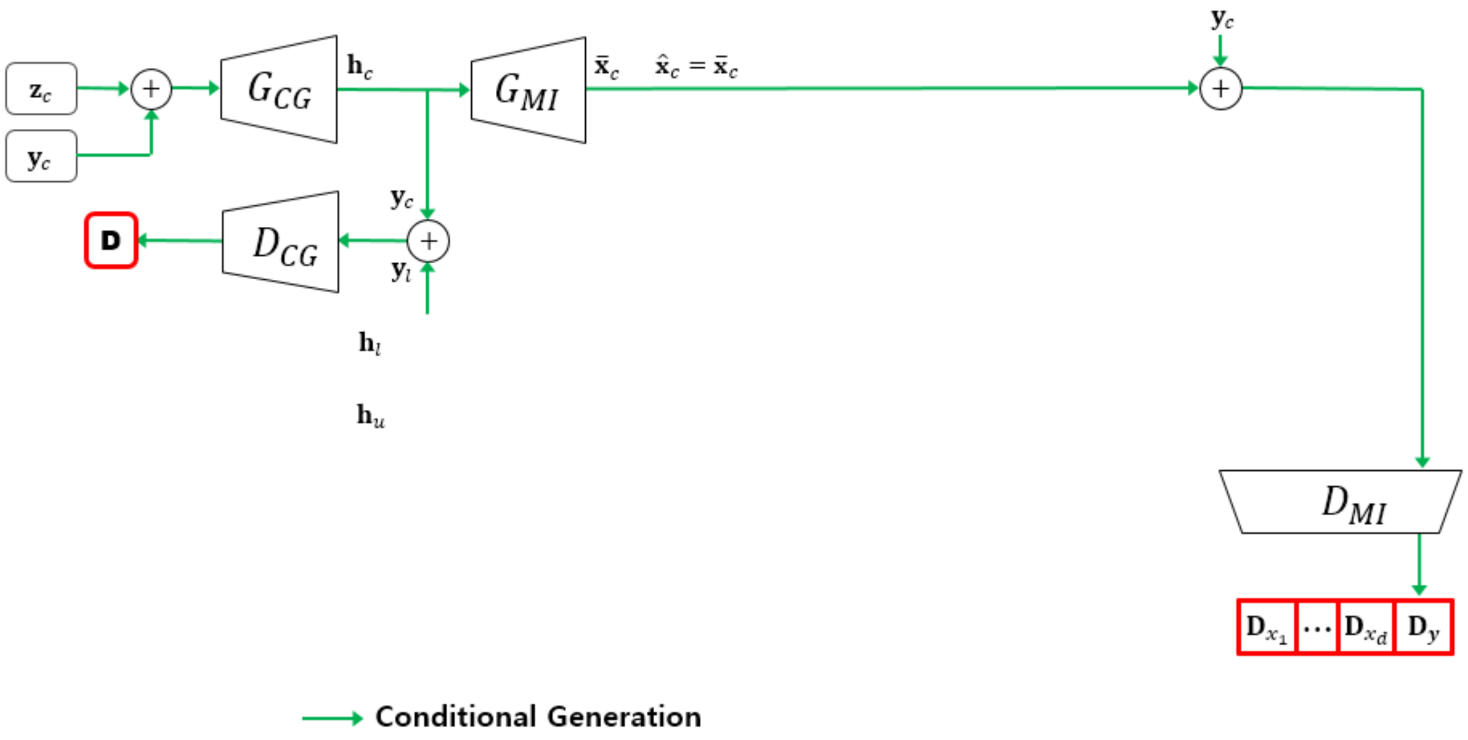


Addressing Three Problems

- **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

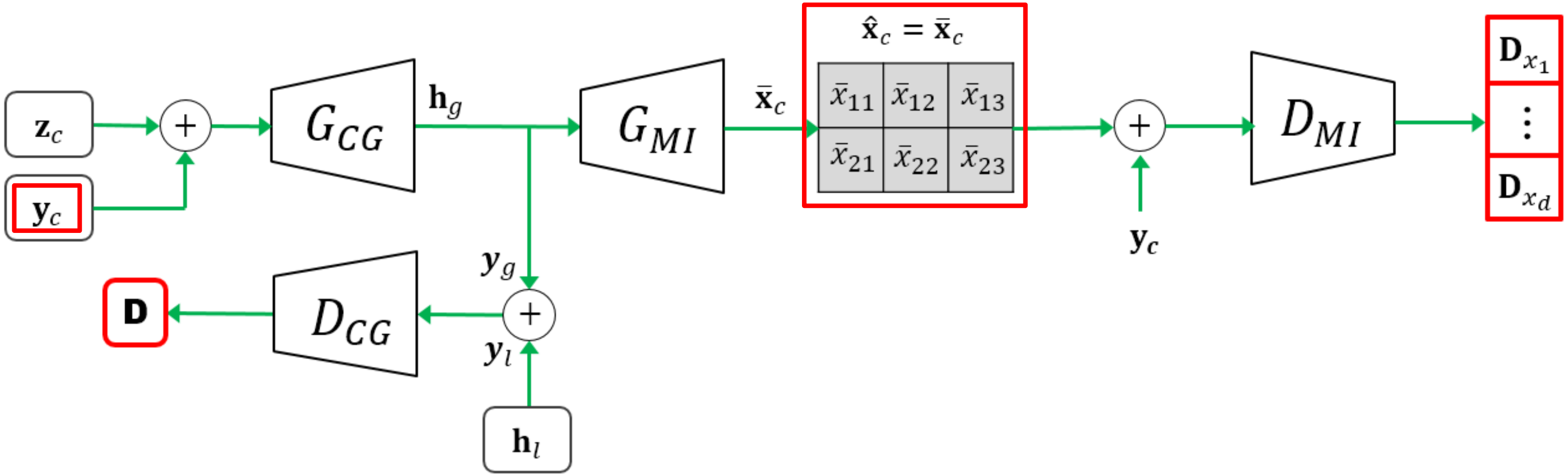


- **Class conditional generation** (to solve the class imbalance problem)
  - Components
    - G_{CG}: creates conditional hidden vectors $\mathbf{h}_c$
    - D_{CG}: determines whether a hidden vector is from the dataset or has been created by G_{CG}
  - G_{MI} generates the entire elements conditioned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

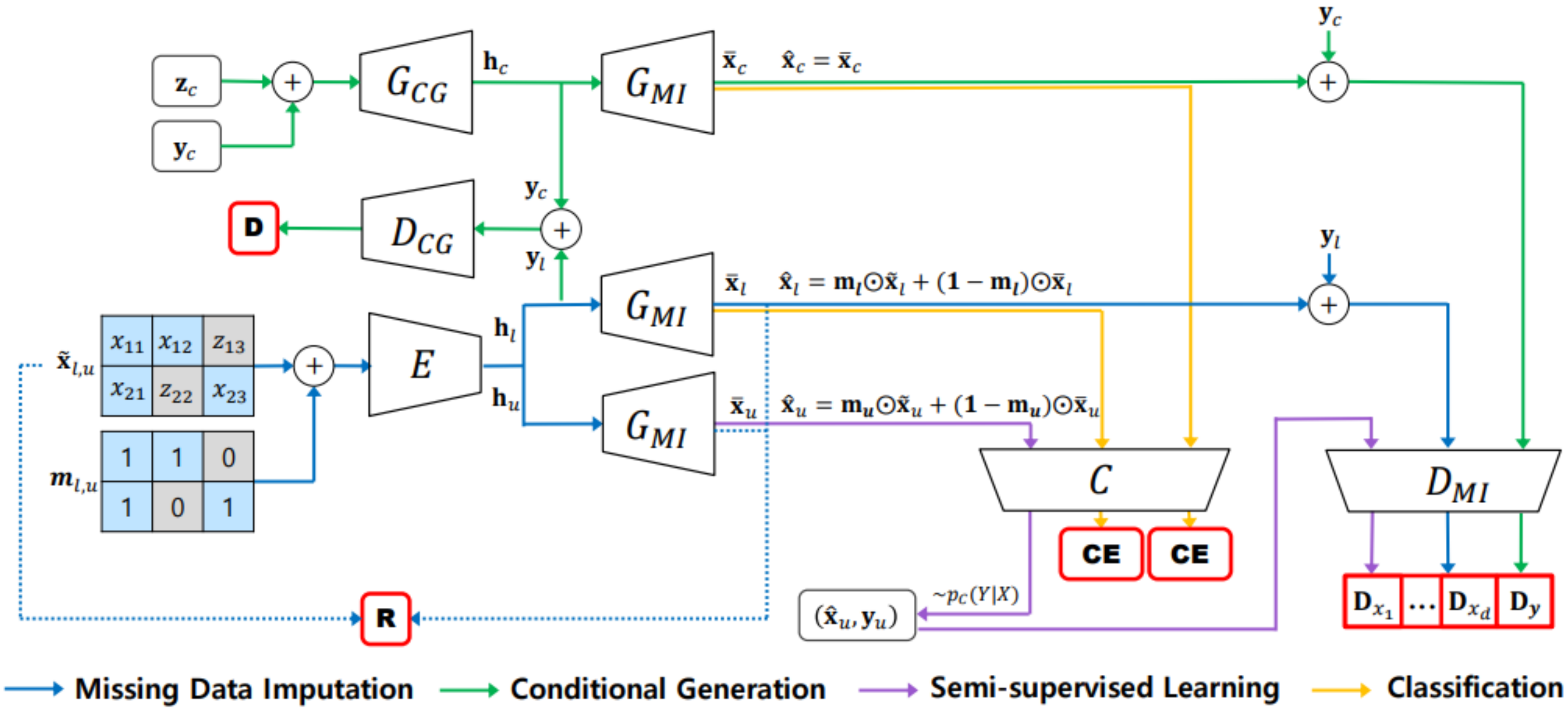


Addressing Three Problems

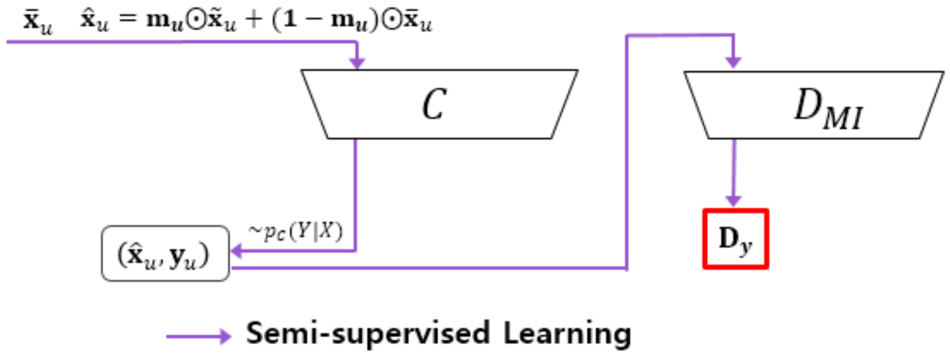
- **Class conditional generation** (to solve the class imbalance problem)
  - Components
    - G_{CG}: creates conditional hidden vectors $\mathbf{h}_c$
    - D_{CG}: determines whether a hidden vector is from the dataset or has been created by G_{CG}
  - G_{MI} generates the entire elements conditioned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



- **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

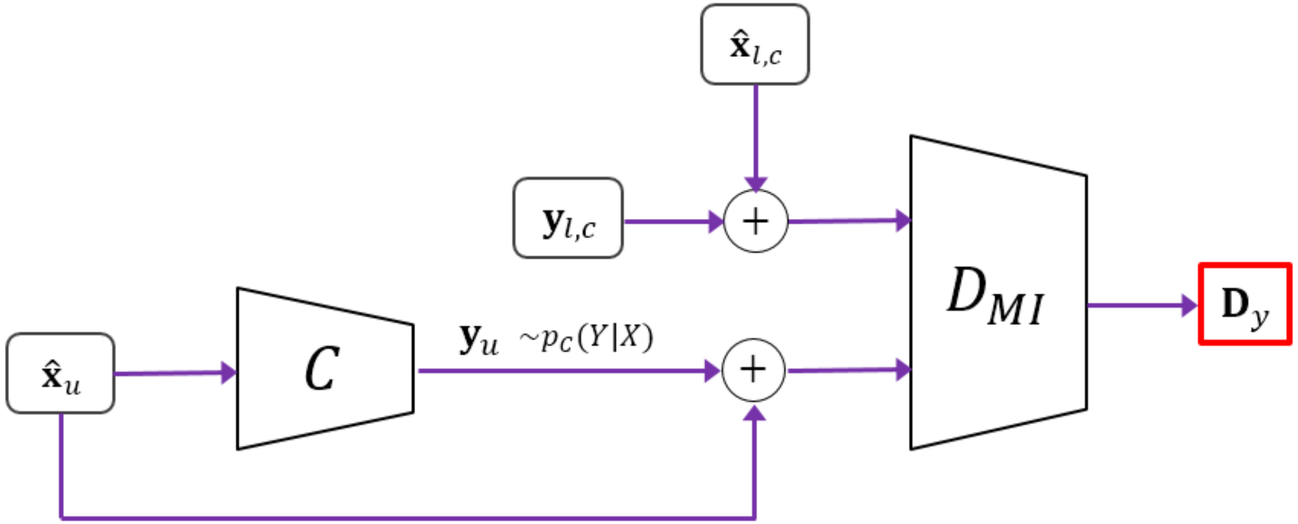


- **Semi-supervised learning** (to solve the missing label problem)
  - Components
    - C: estimates class labels. This also works as the label generator
    - $D_{MI}(\cdot)_{d+1}$: distinguishes b/w real and pseudo (fake) labels
  - We adopt the pseudo-labeling technique of TripleGAN (Li et al., NIPS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



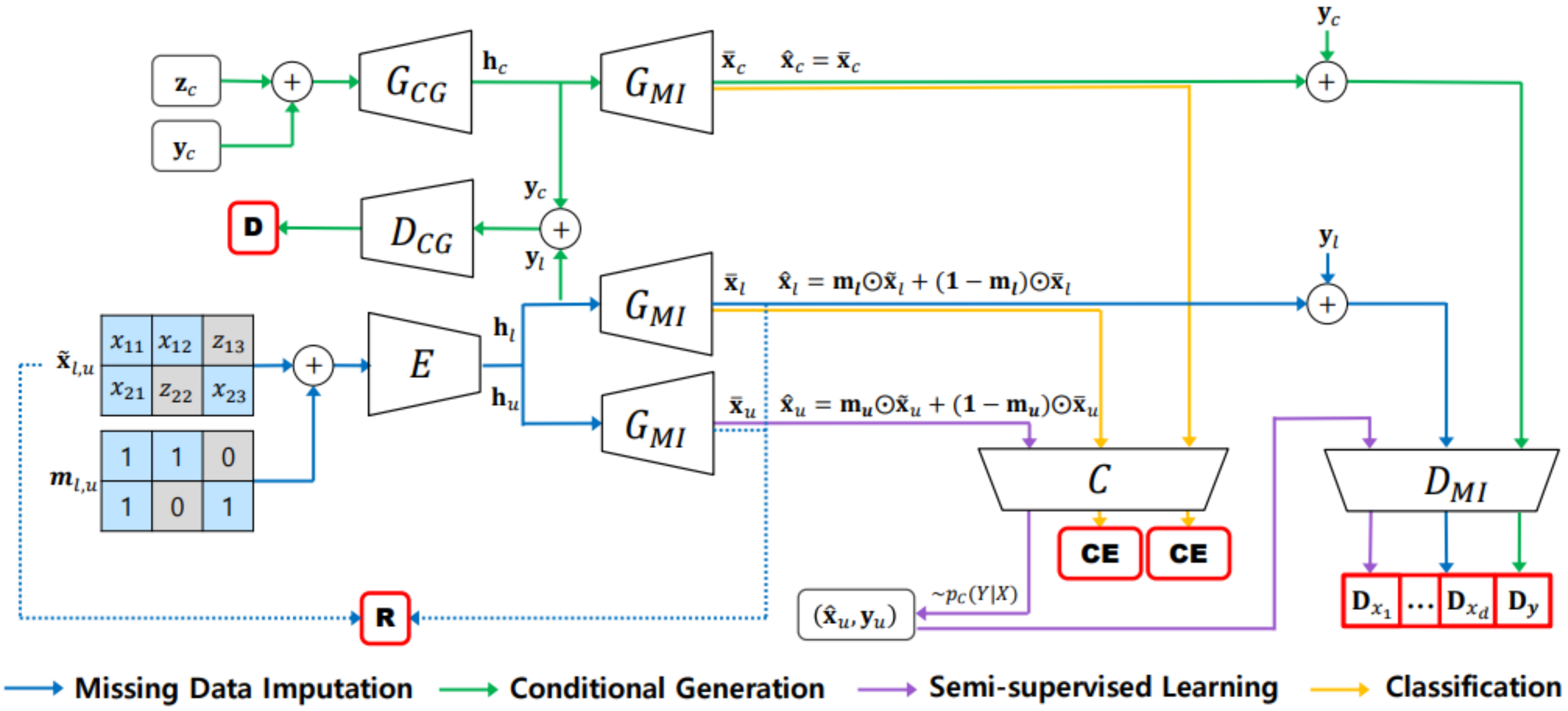
Addressing Three Problems

- **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 (Li et al., NIPS 2017)
  - The two components are related adversarially
    - $L_C = -\mathbb{E}_{\mathbf{y}_u|\hat{\mathbf{x}}_u \sim p_C}[D_{MI}(\hat{\mathbf{x}}_u, \mathbf{y}_u)_{d+1}]$
    - $L_{D_{MI}}^{d+1} = \mathbb{E}_{\mathbf{y}_u|\hat{\mathbf{x}}_u \sim p_C}[D_{MI}(\hat{\mathbf{x}}_u, \mathbf{y}_u)_{d+1}] - \mathbb{E}_{\mathbf{y}|\hat{\mathbf{x}} \sim p_{data}}[D_{MI}(\hat{\mathbf{x}}, \mathbf{y})_{d+1}]$



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.*

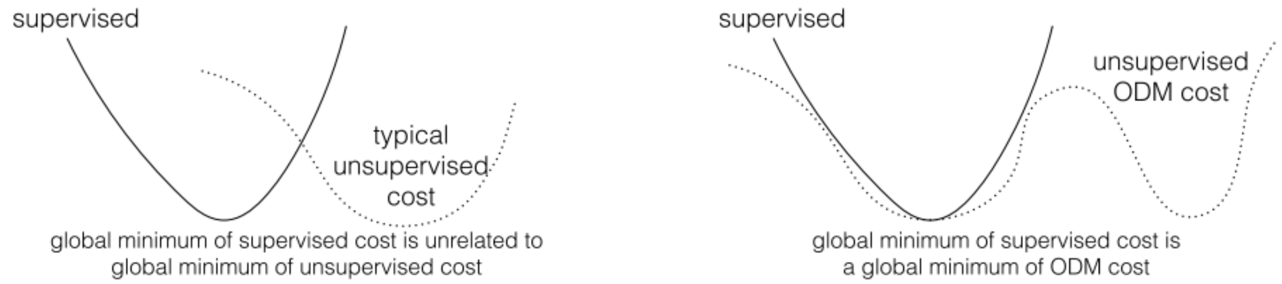
Theorems

- 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 G_{MI} and 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 (Sutskever et al., 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.***

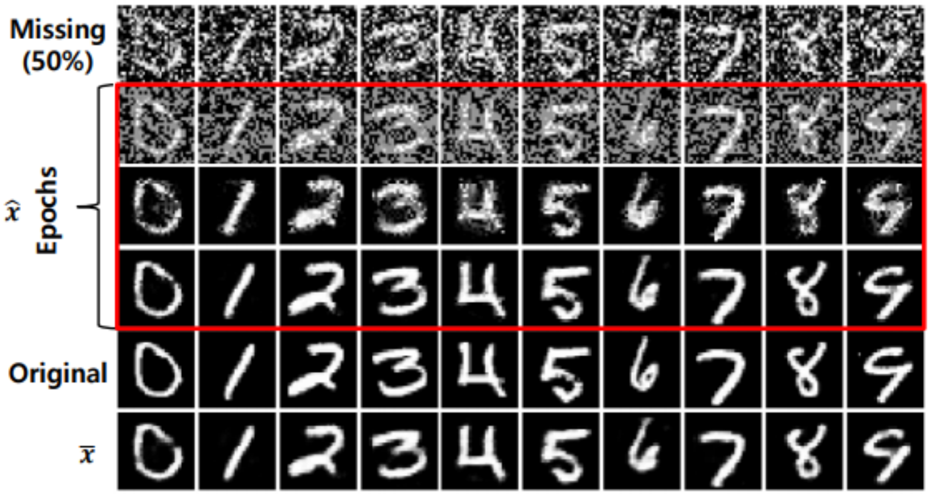
$$W(\mathrm{Distr}[C(\hat{\mathbf{x}}_u)], \mathrm{Distr}[\mathbf{y}]) \rightarrow 0 \quad \Rightarrow \boxed{\mathrm{Distr}[C(\hat{\mathbf{x}}_u)] = \mathrm{Distr}[\mathbf{y}]}$$

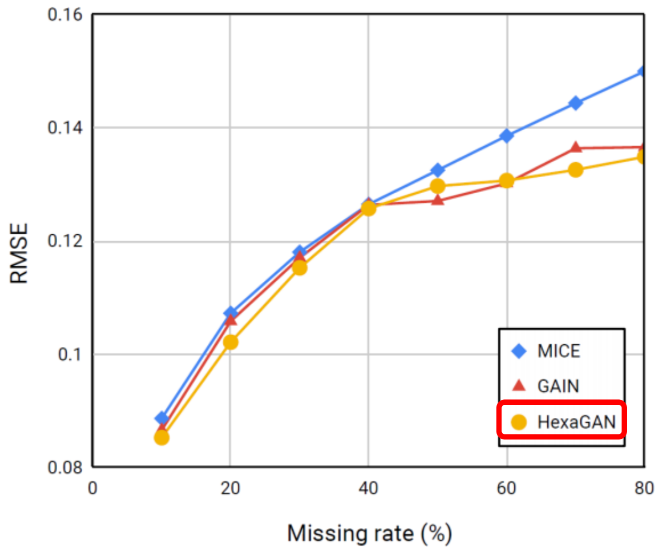
Experimental Results

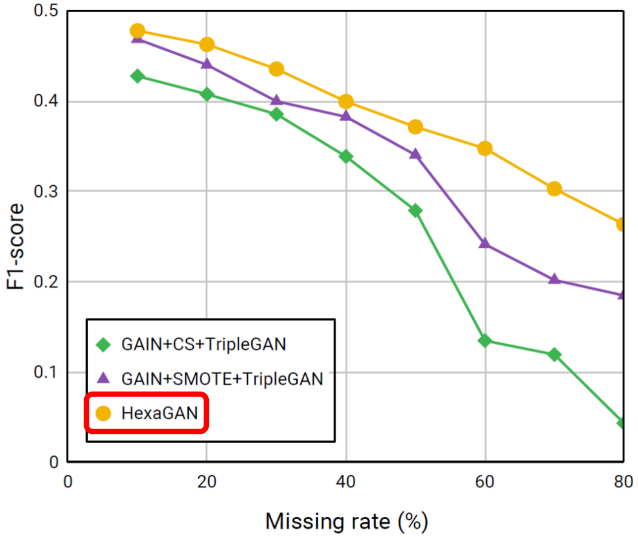
- Missing data imputation
  - HexaGAN shows good performances on various real world datasets
    - Medical, financial, vision, …

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**

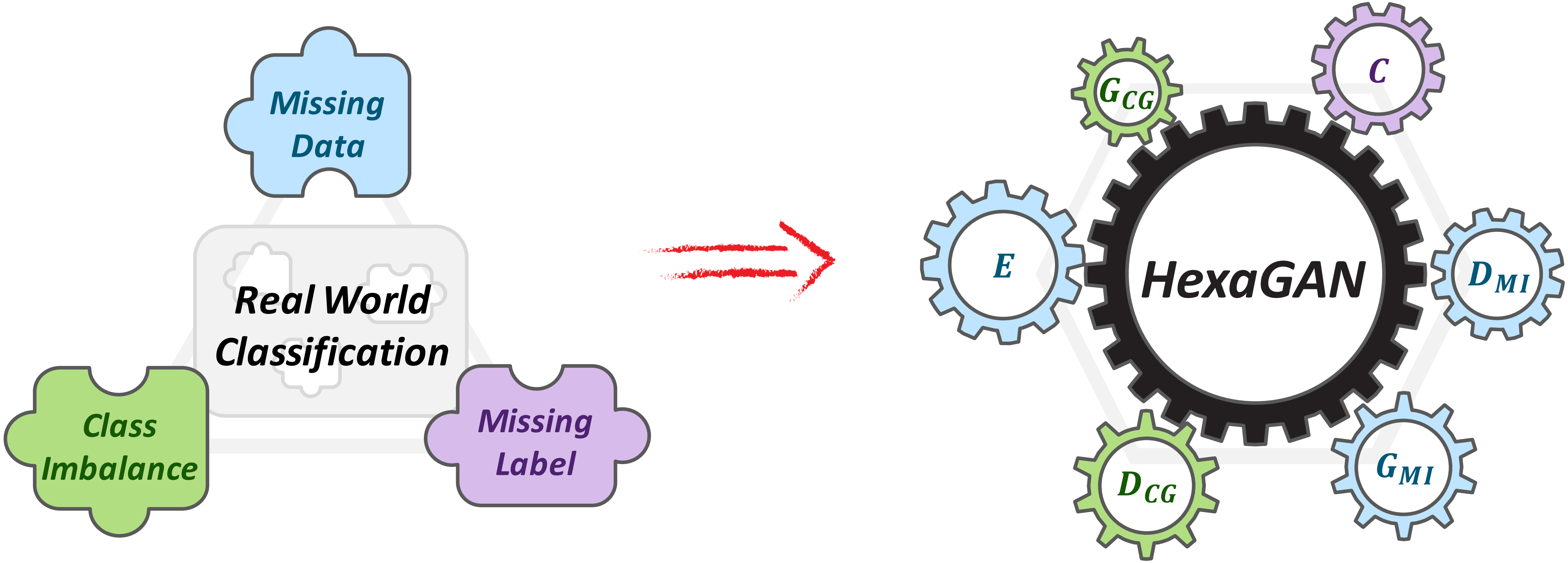
Method	Breast	Credit	Wine	Madelon
MICE + CS + TripleGAN	0.9417 ±0.0044	0.3836 ±0.0052	0.9704 ±0.0043	0.6681 ±0.0028
GAIN + CS + TripleGAN	0.9684 ±0.0102	0.4076 ±0.0038	0.9727 ±0.0046	0.6690 ±0.0027
MICE + SMOTE + TripleGAN	0.9434 ±0.0060	0.4163 ±0.0029	0.9756 ±0.0037	0.6712 ±0.0008
GAIN + SMOTE + TripleGAN	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**





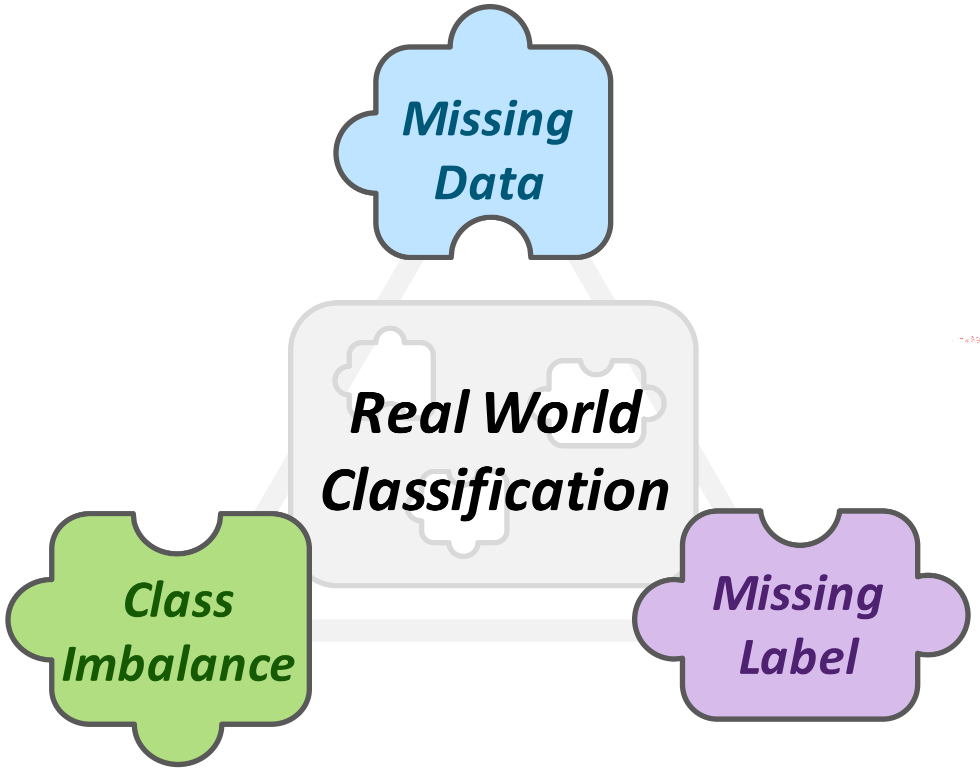


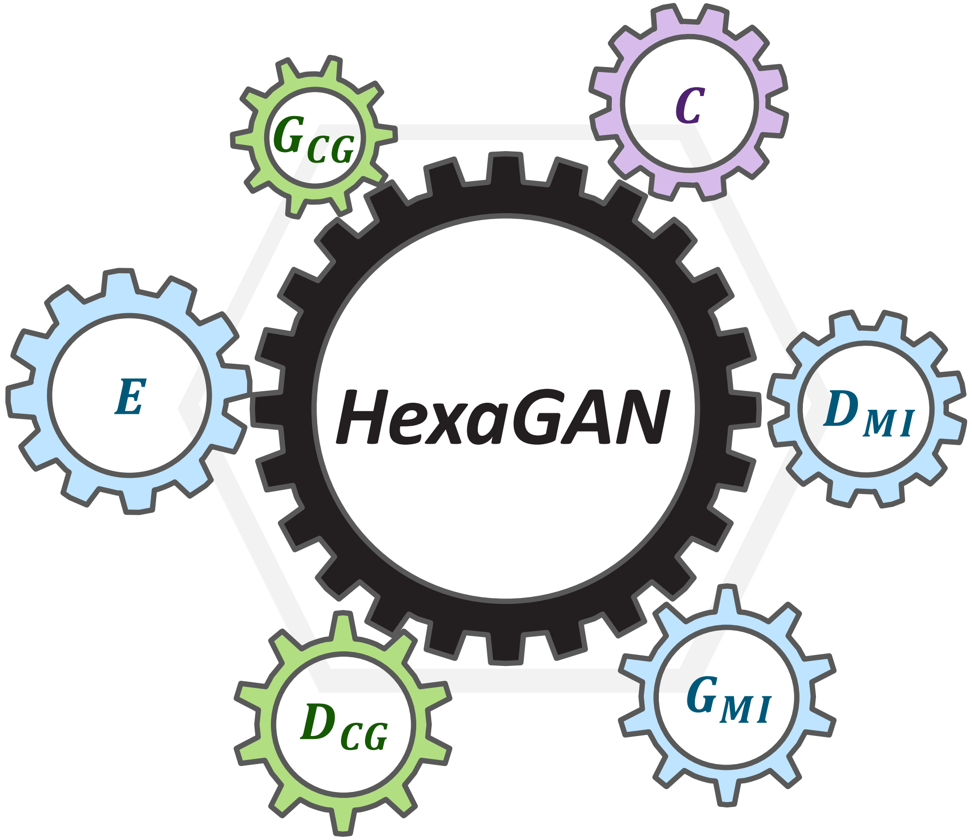
Conclusions



- **For more details, please visit our poster session!**
  - June 11th **(Today)**, 6:30 – 9:00 pm, Pacific Ballroom #20

Conclusions





- **For more details, please visit our poster session!**
  - June 11th **(Today)**, 6:30 – 9:00 pm, Pacific Ballroom #20