BUILDING MANUFACTURING DEEP LEARNING MODELS WITH MINIMAL AND IMBALANCED TRAINING DATA USING DOMAIN ADAPTATION AND DATA AUGMENTATION

Adrian Shuai Li (Purdue), Elisa Bertino (Purdue),

Rih-Teng Wu (NTU), Ting-Yan Wu (NTU)

ICIT 2023

PURDUE UNIVERSITY | Department of Computer Science

Using Deep Learning in Manufacturing

Goals

- Deep learning techniques have been used for tasks such as classification and prediction because of their ability to efficiently and effectively analyze different types of data, e.g., images, sounds, and vibrations

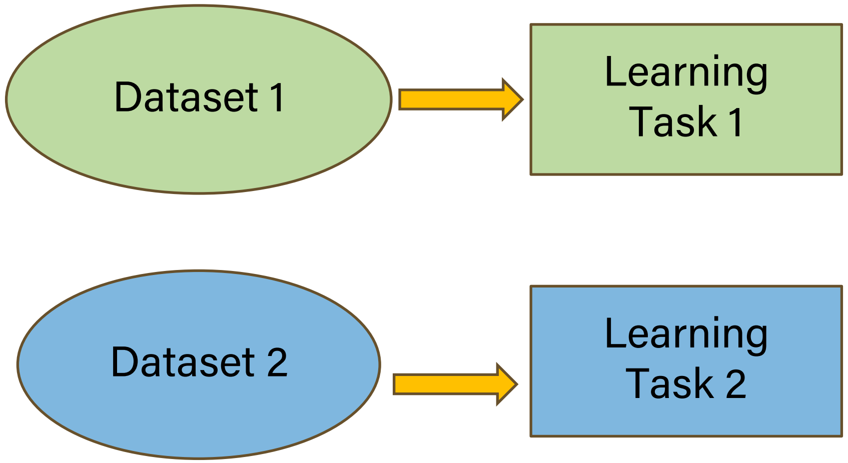
Challenges

- Deep learning models require **large** and **high-quality** labeled training data
- Training data may have low quality, such as lack of labels and imbalanced class distribution

Transfer Learning

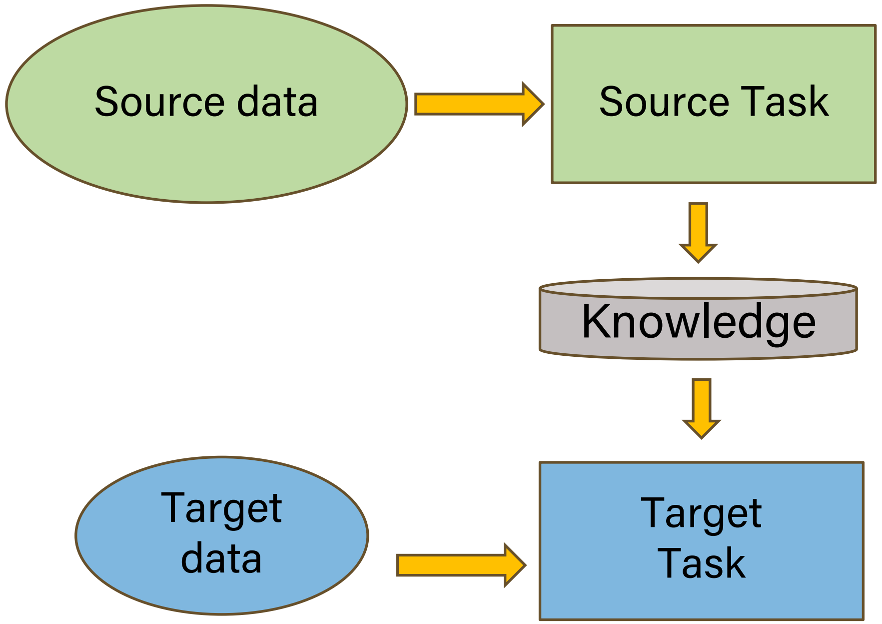
Traditional ML

Single task learning



Transfer Learning

Learning of a new task relies on the previous learned tasks



PURDUE UNIVERSITY | Department of Computer Science

Example: Picture vs Cartoon

Train

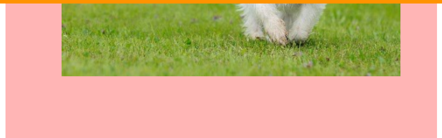
cat

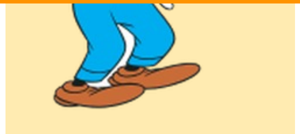
Test

cat

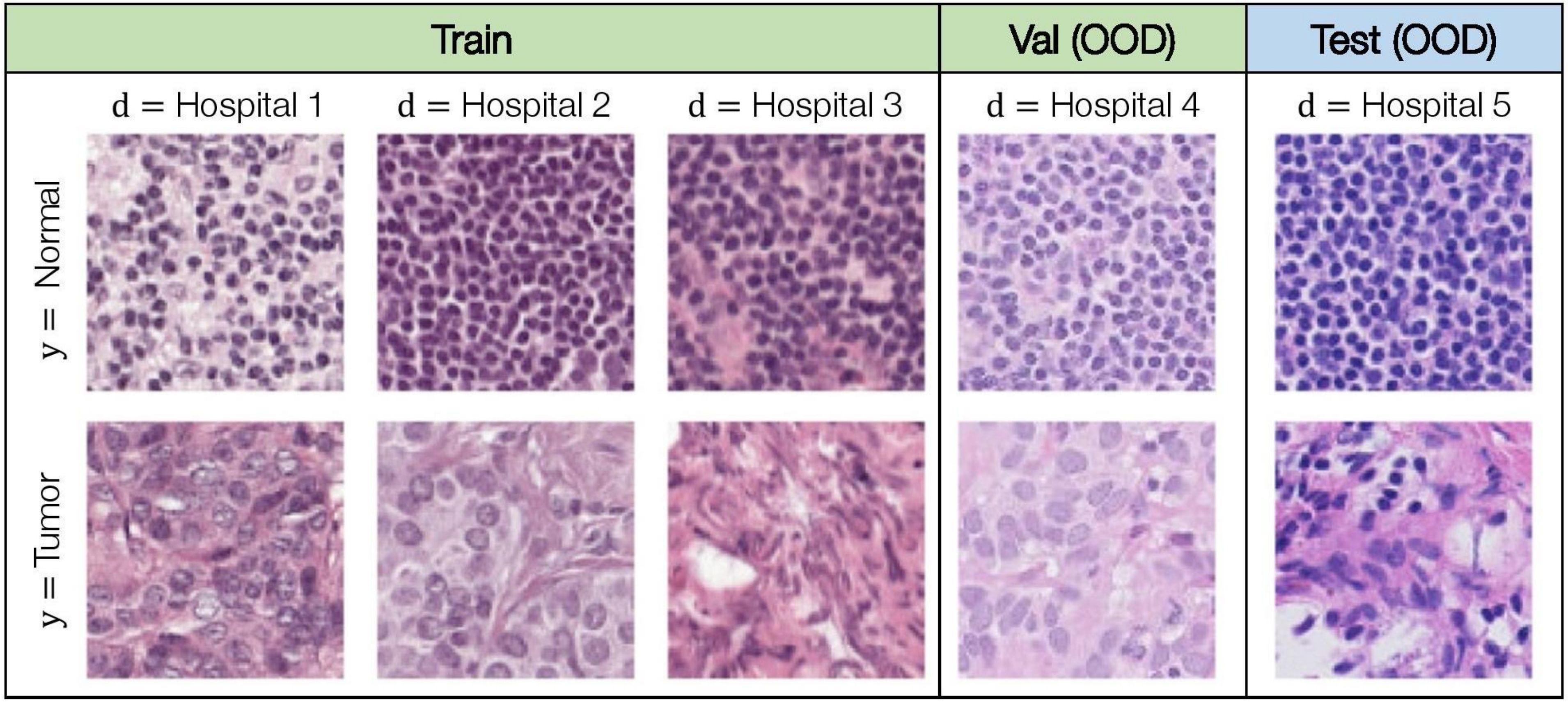
Target domain is ***related to*** the source domain

Different domains, same task





Example: Medical Data from Different Hospitals



Credit: Camelyon17 dataset. https://wilds.stanford.edu/datasets/

Related work - Transfer Learning

Underlying technique	Related work	Source and target can have different dimensions	Imbalanced target data
Fine Tuning	Deep Transfer Learning [Shao S, et al., 2018]	✓	✗
Adversarial Domain Adaptation	DANN [Ganin Y, et al., 2016]	✗	✗
	ADDA [Tzeng et al., 2017]	✗	✗
	GAN-based [Singla et al., 2020]	✓	✗
	Our approach	✓	✓

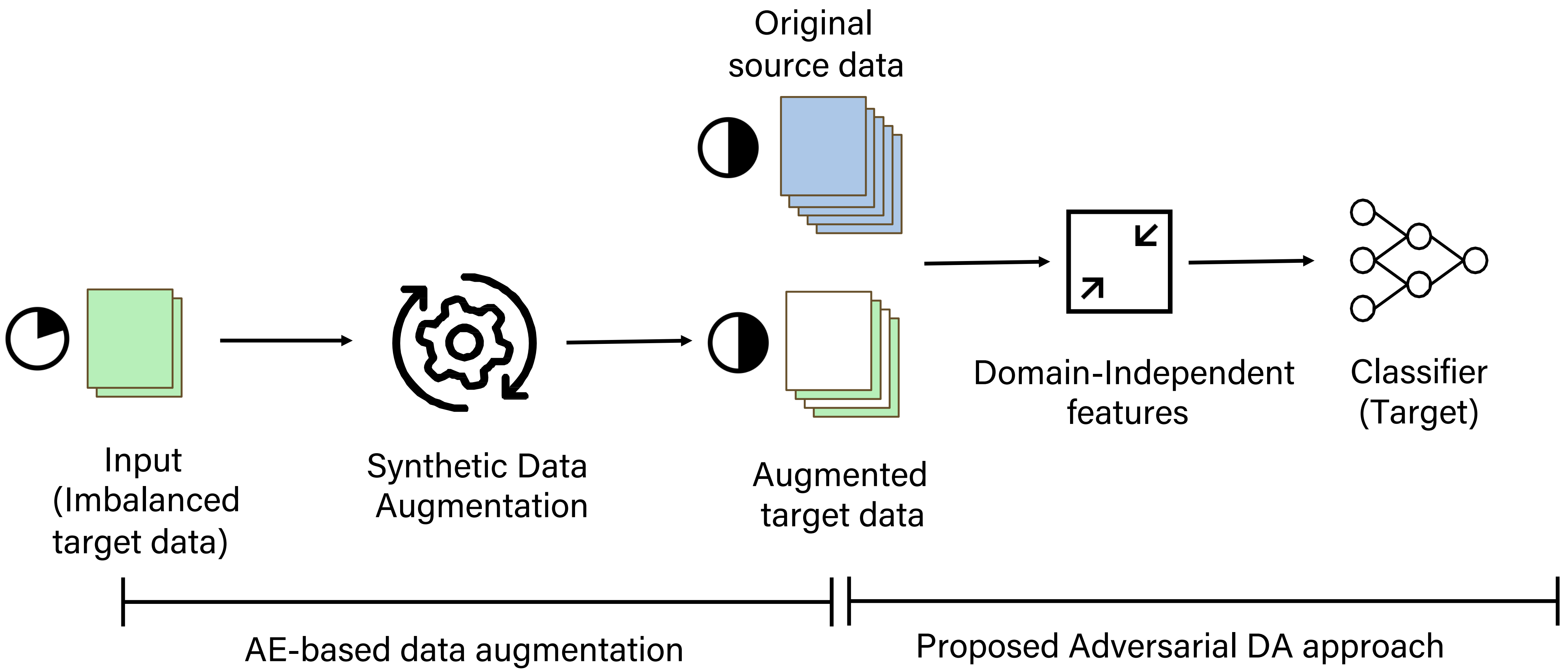
Related work - Dealing with Imbalanced Data

Generative models

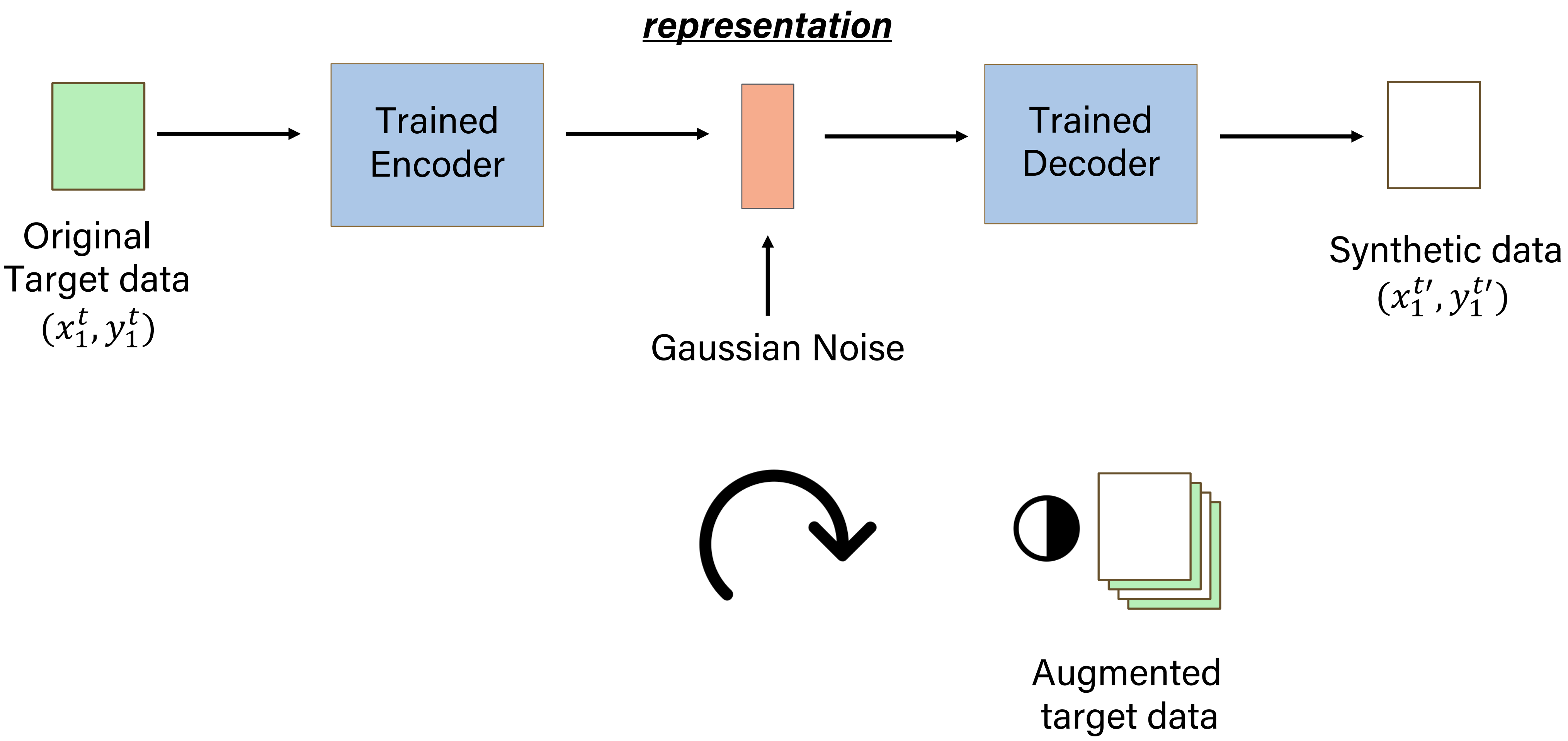
- Generate realistic synthetic data samples
    - Generative adversarial networks (GAN) [Goodfellow et al., 2020]
    - Autoencoders (AE) [Engel et al., 2017]
    - Diffusion models [Jonathan et al. 2022]

Problem:
Systems built using synthetic data sets often fail when deployed to the real world.

Overview of our pipeline

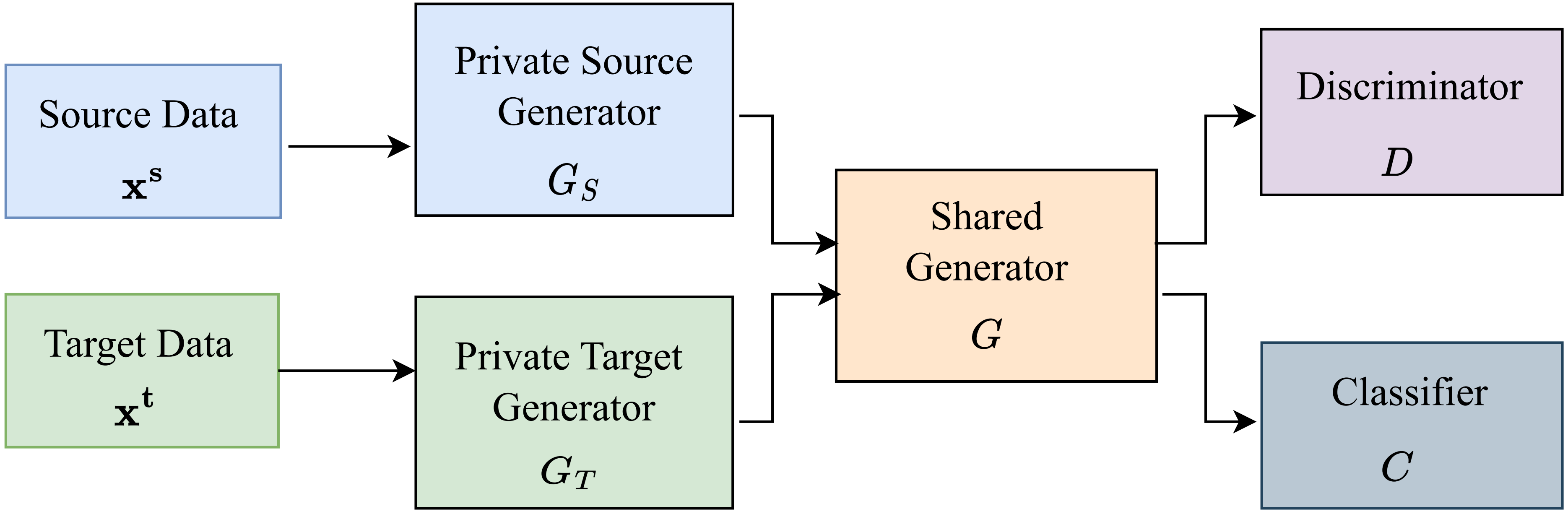


AE-based Data Augmentation



Proposed Adversarial DA approach

Maximize domain classification accuracy



Maximize label classification accuracy + minimize domain classification accuracy

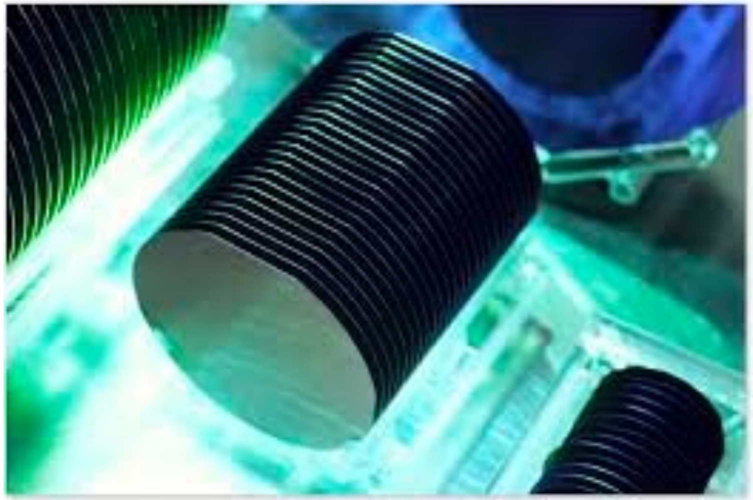
Maximize label classification accuracy

Application -Wafer defect prediction

- **Integrated circuits (IC)** are made by creating circuit structures on many layers of a single wafer and interconnecting the structures using wires.

- The wafer surface must be extremely clean
    - No particles (e.g., rock and ring shapes)
    - Flaws (e.g., spots and scratches)

- Human operator inspects the scanned microscope images of the wafer surface

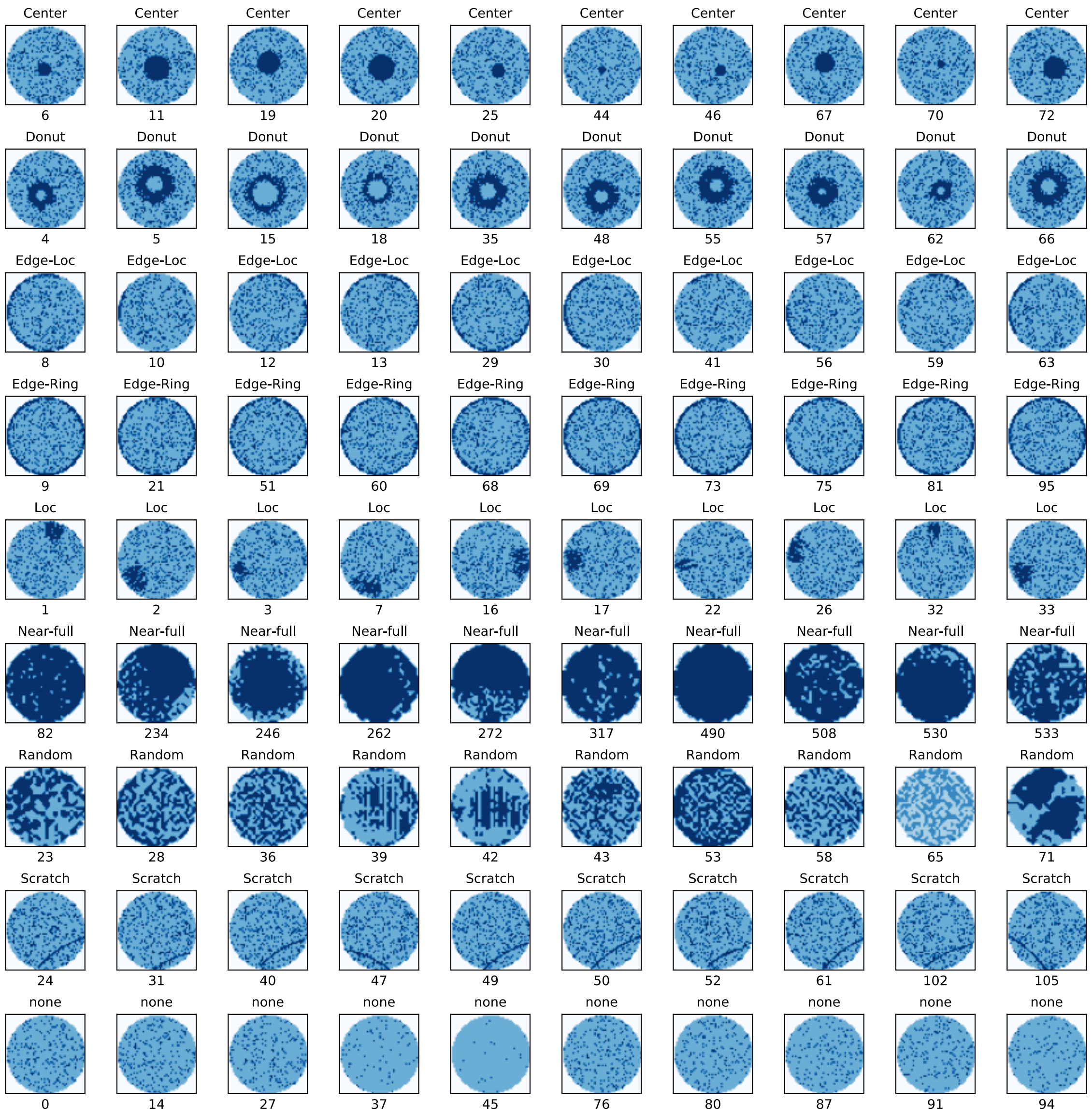


Application –Wafer defect prediction

Source data: MixedWM38 Dataset

- Total of 38,000 wafer maps
- 1 normal pattern, 8 single defect patterns
- 29 mixed defect patterns
- Each pattern has 1000 good quality samples/images

https://github.com/Junliangwangdhu/WaferMap



Application -Wafer defect prediction

Target data: WM - 811K Dataset

- 172,950 images with manual label
- 1 normal pattern, 8 single defect patterns

The dataset has several problems!

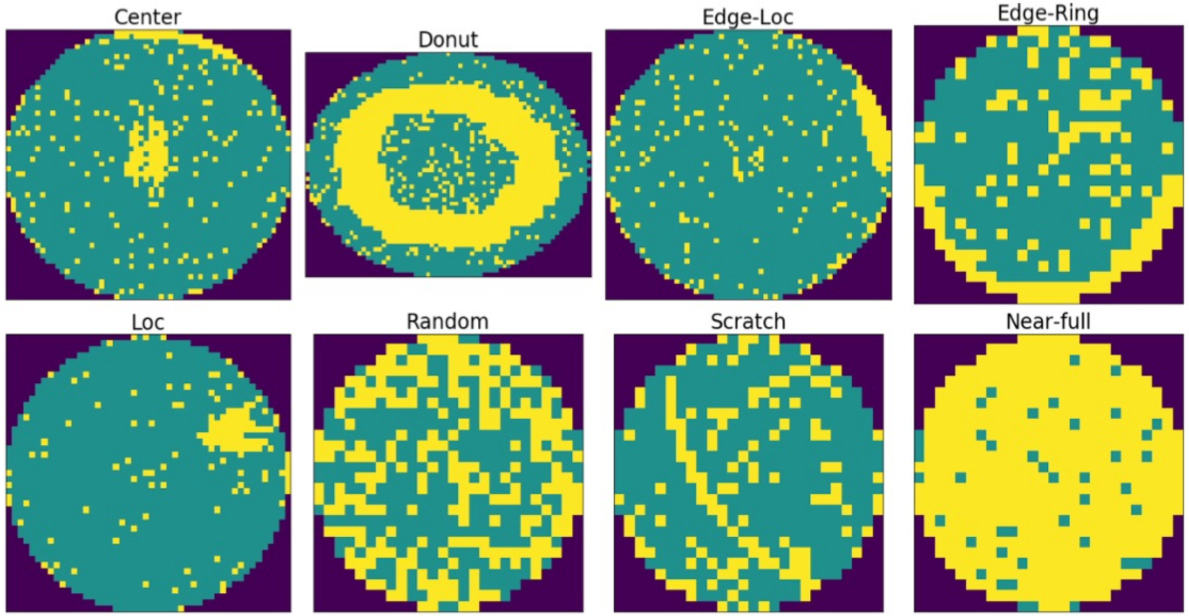


Figure 1: Wafer Defect map

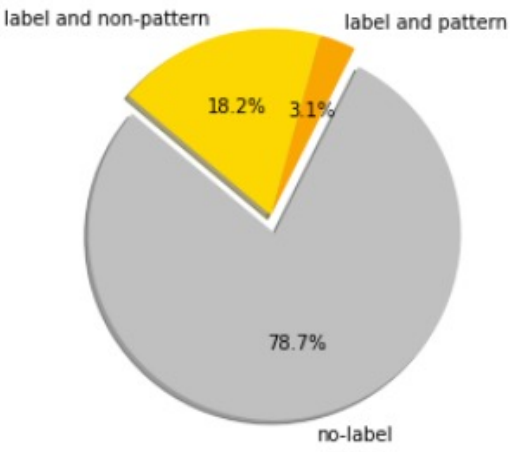
Application –Wafer defect prediction

WM-811K wafer map

Problem 1
A large amount of unlabeled samples

Problem 2
Varying sizes of input

Problem 3
Extremely imbalanced dataset



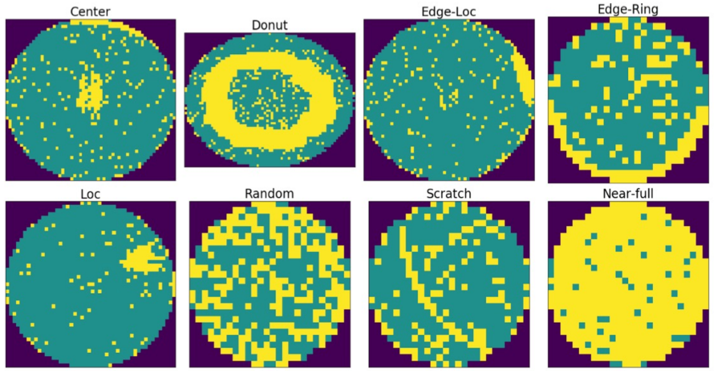
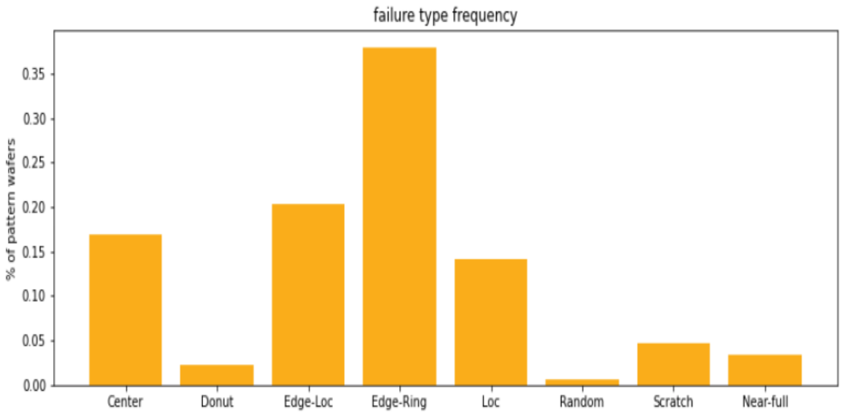


Figure 1: Wafer Defect map



https://www.kaggle.com/qingyi/wm811k-wafer-map

Description of Experiments

We compare our pipeline with different approaches. All the methods we consider are:

- Proposed adversarial DA
- Fine-tuning
    - Retrain the last two convolution blocks of a pre-trained VGG 16 model using the target data
- Vanilla classifier (No adaptation)

We evaluate each method under two settings

- Augmented target data
- Original (Imbalanced) target data

Results and Analysis

TABLE II

TRAINING AND TESTING TIME COMPARISON FOR THREE METHODS. THE RESULTS ARE OBTAINED WITH 1000 TARGET DATA SAMPLED FROM THE AUGMENTED TARGET TRAINING DATA. WE CAN TRAIN THE PROPOSED DA MODEL OFFLINE. THE PREDICTION TIME IS AS LOW AS A VANILLA CLASSIFIER.

Models	Training time (s)	Testing time (s)
Adversarial DA	3211.70	0.47
Fine-tuning	31.71	0.87
Vanilla classifier	86.33	0.45

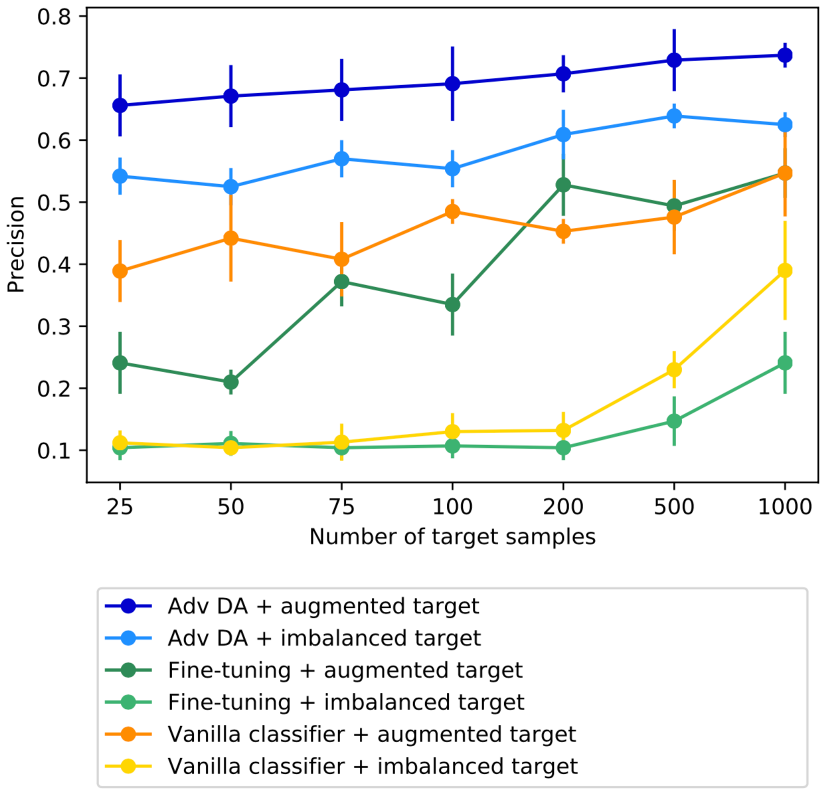


Fig. 2. Classification precision score achieved using augmented target data and imbalanced target data comparing six approaches: a vanilla deep CNN trained with the augmented/imbalanced target samples; a pretrained VGG 16 model fine-tuned with the augmented/imbalanced target data; our adversarial DA architecture trained with augmented/imbalanced target data.

Future work

- Handle the domain shift and accomplish effective knowledge transfer in the non-classification task such as optimization
- While we use available data from a single source domain to improve the generalization on a related target task, one may find data from many related domains useful.
- Another limitation of our approach is that it requires at least some labeled data from each class in the target domain.

References

- **[Singla *et al.*, 2020]** Ankush Singla, Elisa Bertino, and Di- 663 nesh Verma. Preparing network intrusion detection deep 664 learning models with minimal data using adversarial do- 665 main adaptation. In *Proceedings of the 15th ACM Asia* 666 *Conference on Computer and Communications Security*, 667 pages 127–140, 2020
- **[Tzeng *et al.*, 2017]** Eric Tzeng, Judy Hoffman, Kate 669 Saenko, and Trevor Darrell. Adversarial discriminative 670 domain adaptation. In *Proceedings of the IEEE confer-* 671 *ence on computer vision and pattern recognition*, pages 672 7167–7176, 2017.
- **[Goodfellow *et al.*, 2020]** Ian Goodfellow, Jean Pouget- Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Gen- erative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.
- **[Ganin *et al.*, 2016]** Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, Franc¸ois Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The jour- nal of machine learning research*, 17(1):2096–2030, 2016.
- **[Shao S, et al., 2018]** Shao, Siyu, Stephen McAleer, Ruqiang Yan, and Pierre Baldi. "Highly accurate machine fault diagnosis using deep transfer learning." *IEEE Transactions on Industrial Informatics* 15, no. 4 (2018): 2446-2455.
- **[Jonathan et al. 2022]** Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Moham- mad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23(47):1–33, 2022.
- **[Engel et al., 2017]** Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mo- hammad Norouzi, Douglas Eck, and Karen Simonyan. Neural audio synthesis of musical notes with wavenet autoencoders. In *International Conference on Machine Learning*, pages 1068–1077. PMLR, 2017.

THANK YOU

Loss functions

- The discriminator loss is calculated as

$$\mathcal{L}_d = -\sum_{i=1}^{N_s+N_t}\left\{d_i log\hat{d}_i + (1-d_i)log(1-\hat{d}_i)\right\}$$

- The generator loss is the above loss with inverted domain truth labels.

$$\mathcal{L}_g = -\sum_{i=1}^{N_s+N_t}\left\{(1-d_i)log\hat{d}_i + d_i log(1-\hat{d}_i)\right\}$$

- The classification loss L_c is calculated as

$$\mathcal{L}_c = -\sum_{i=1}^{N_s} y_i^s \cdot log\hat{y}_i^s - \lambda\sum_{i=1}^{N_t} y_i^t \cdot log\hat{y}_i^t$$

Learning Dynamics

$$\Delta_{G_S} = -\mu \left(\beta \frac{\partial \mathcal{L}_g}{\partial G_S} + \gamma \frac{\partial \mathcal{L}_c}{\partial G_S} \right)$$

$$\Delta_{G_T} = -\mu \left(\beta \frac{\partial \mathcal{L}_g}{\partial G_T} + \gamma \frac{\partial \mathcal{L}_c}{\partial G_T} \right)$$

$$\Delta_{G} = -\mu \left(\beta \frac{\partial \mathcal{L}_g}{\partial G} + \gamma \frac{\partial \mathcal{L}_c}{\partial G} \right)$$

$$\Delta_{D} = -\mu \frac{\partial \mathcal{L}_d}{\partial D}$$

$$\Delta_{C} = -\mu \frac{\partial \mathcal{L}_c}{\partial C}$$

where μ is the learning rate. The hyperparameters β, γ are the relative weights of the loss functions.

Results and Analysis

TABLE I
BALANCED CLASSIFICATION ACCURACY/AVERAGED RECALL COMPARISON ON THE WM-811K TESTING DATA

Models in IV-B	Number of samples in target dataset used for training						
	25	50	75	100	200	500	1000
Adversarial DA + augmented	70.3% $\pm$ 0.7%	71.7% $\pm$ 4.5%	71.2% $\pm$ 4.5%	72.5% $\pm$ 4.3%	72.5% $\pm$ 2.6%	72.3% $\pm$ 4.2%	73.7% $\pm$ 2.5%
Adversarial DA + imbalanced	54.7% $\pm$ 4.6%	57.9% $\pm$ 5.6%	56.3% $\pm$ 4.5%	65.2% $\pm$ 4.2%	67.2% $\pm$ 1.4%	65.9% $\pm$ 2.7%	65.8% $\pm$ 0.1%
Fine-tuning + augmented	28.9% $\pm$ 8.3%	28.3% $\pm$ 7.6%	45.5% $\pm$ 5.8%	49.1% $\pm$ 7.6%	56.1% $\pm$ 2.4%	65.3% $\pm$ 3.1%	67.4% $\pm$ 1.2%
Fine-tuning + imbalanced	11.1% $\pm$ 0%	13.3% $\pm$ 6.1%	11.1% $\pm$ 0.0%	13.2% $\pm$ 5.9%	11.1% $\pm$ 0.0%	15.2% $\pm$ 4.6%	24.8% $\pm$ 7.1%
Vanilla classifier + augmented	48.7% $\pm$ 3.8%	53.1% $\pm$ 7.3%	52.1% $\pm$ 1.7%	63.7% $\pm$ 7.3%	63.6% $\pm$ 7.4%	65.3% $\pm$ 7.0%	65.3% $\pm$ 7.7%
Vanilla classifier + imbalanced	11.4% $\pm$ 1.0%	11.1% $\pm$ 0.0%	11.4% $\pm$ 1.5%	11.7% $\pm$ 1.9%	13.2% $\pm$ 7.5%	27.3% $\pm$ 8.1%	35.5% $\pm$ 6.5%

Synthetic Data Augmentation

- Diffusion models
  - Diffusion models have high computational costs due to the iterative steps during training, making them unsuitable for tasks that are time-sensitive
- GAN
  - GANs are known to have training instability and being prone to mode collapse during training
  - GANs also require large amounts of training data

- Why we chose AE-based method?
  - The autoencoder-based data augmentation method requires less data for training, hence it aligns with the problem setting where the target has limited data
  - It is also faster than the more complicated diffusion model