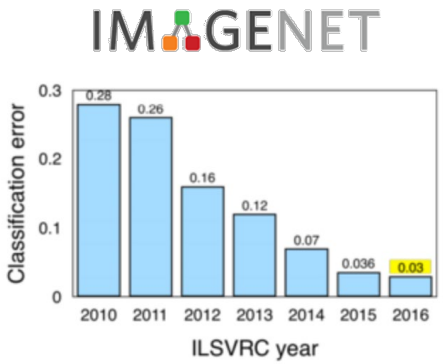
Domain Adaptation for Data-Driven Fault Diagnosis

Qin Wang

Success of Data-driven Methods on General Tasks

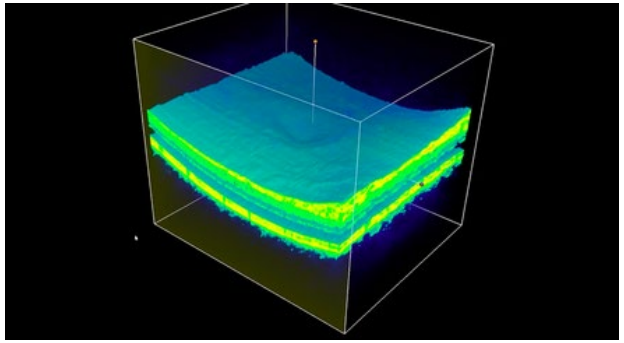
Image classification



Detection and segmentation

Machine Translation

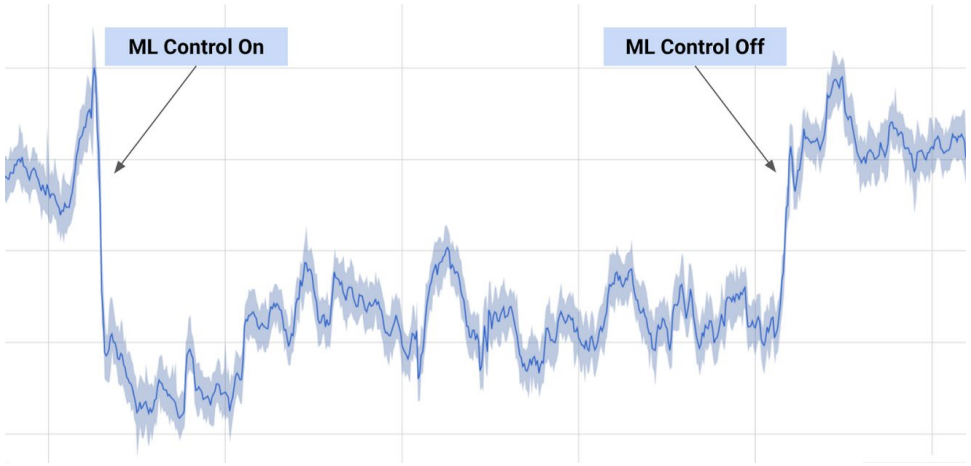
Eye disease diagnosis



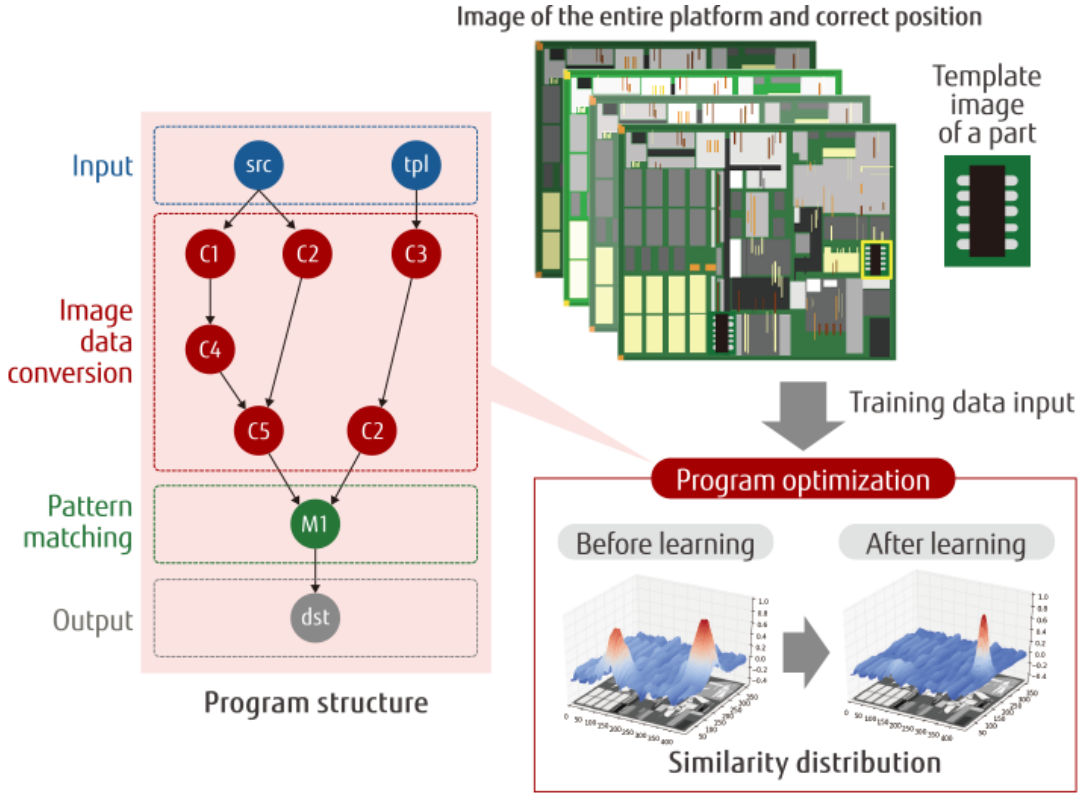
VQA: Goyal, Yash, et al. "Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.
Detection: https://github.com/facebookresearch/detectron2
Eye: https://deepmind.com/blog/article/moorfields-major-milestone

Success of Data-driven Methods on Industrial Applications

DeepMind Reduces Google Data Centre's Cooling Bill by 40%



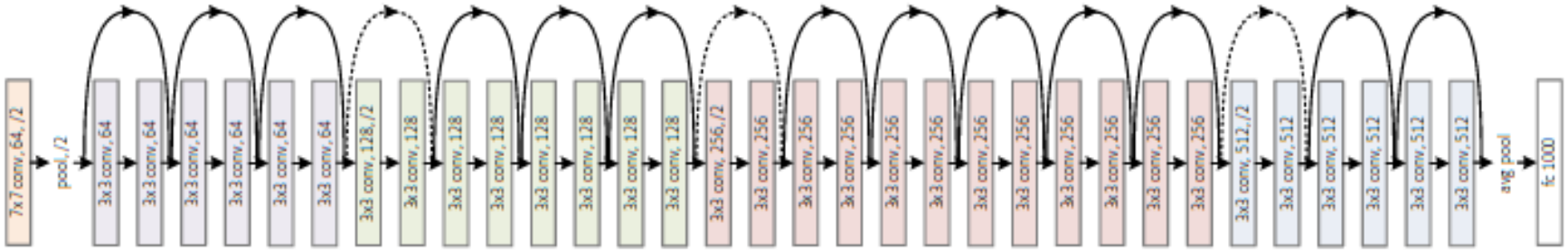
Fujitsu reduces development time of parts assembly machines by 80% while maintaining recognition rates of 97%+



https://deepmind.com/blog/deepmind-ai-reduces-google-data-centre-cooling-bill-40/
http://journal.jp.fujitsu.com/en/2017/04/19/01/

Potential Problem

- Data hungry models
  - Requires a **huge** amount of labeled data
  - Popular ResNet-50: ~25M parameters
  - **Representativeness** matters a lot



He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

What will happen if training data are

not representative?

What will happen if training data is not representative?

- Significant Performance **Drop**

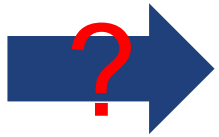
amazon.com

SVM accuracy: **54%**

webcam

SVM accuracy: **20%**

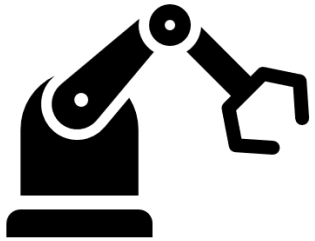
Training **Source** Domain



Target Domain

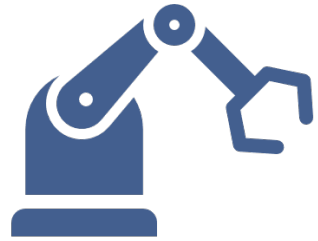
Hoffman, Judy, et al. "Discovering latent domains for multisource domain adaptation." European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2012.
Torralba, Antonio, and Alexei A. Efros. "Unbiased look at dataset bias." CVPR 2011. IEEE, 2011.
BDD100K: A Large-scale Diverse Driving Video Database https://bair.berkeley.edu/blog/2018/05/30/bdd/

More Problems for Fault Diagnosis in the Wild



Source Machine

- Well studied
- Well labeled
- We trained a model !



Target Machine

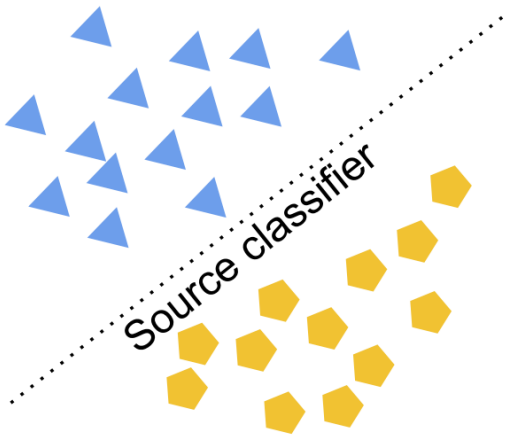
- Newly setup, Not yet labeled
- **Different** operating conditions
- **Non-identical** component

Source data **cannot represent** target machine well,

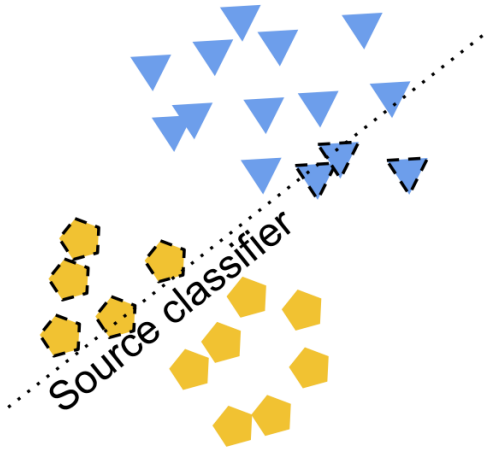
the trained model from source **suffers performance deterioration**

Domain Shift - Target domain different from Source domain

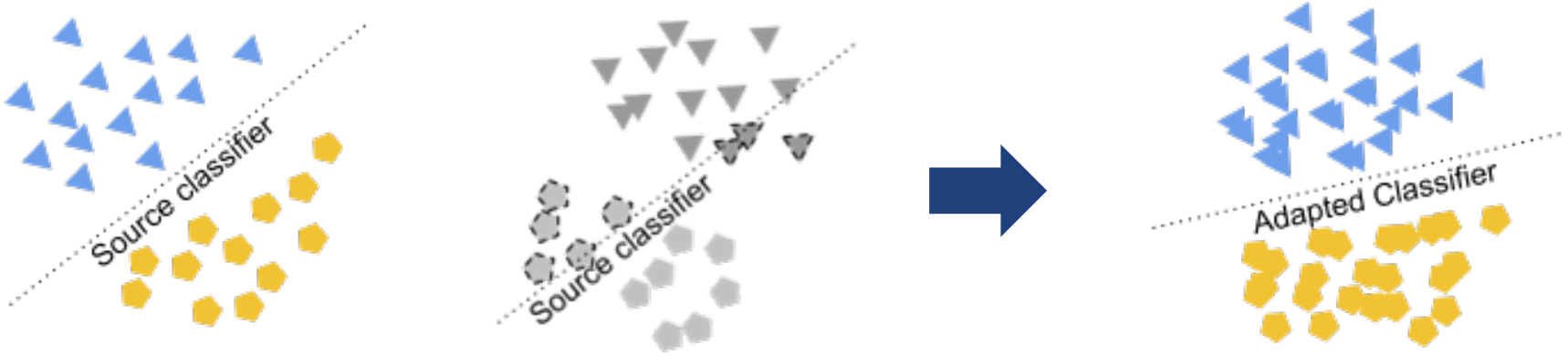
Source



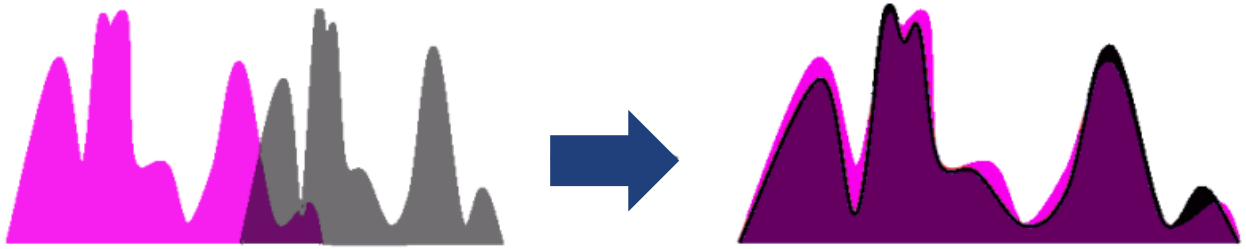
Target



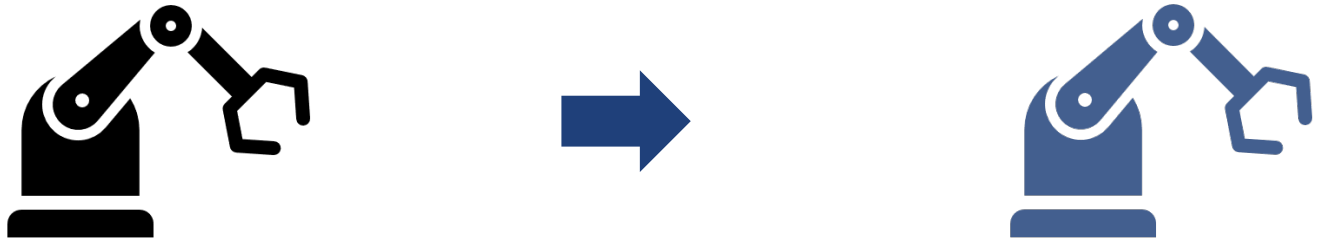
Domain Adaptation (DA)



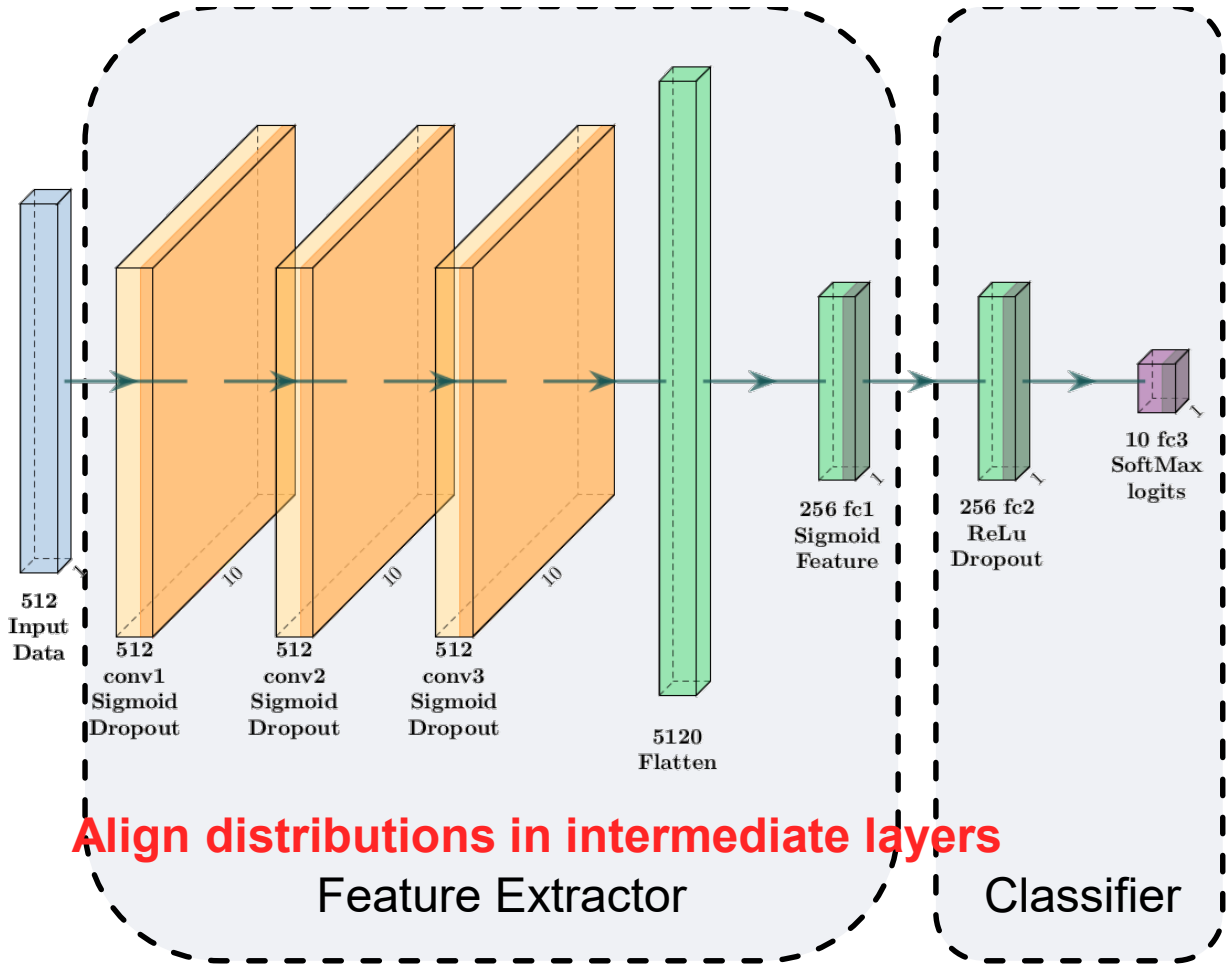
Labeled Source Domain **Unlabeled Target Domain** **Adapted Domain**



How can we achieve this adaptation?



Start from a Basic Network[14]



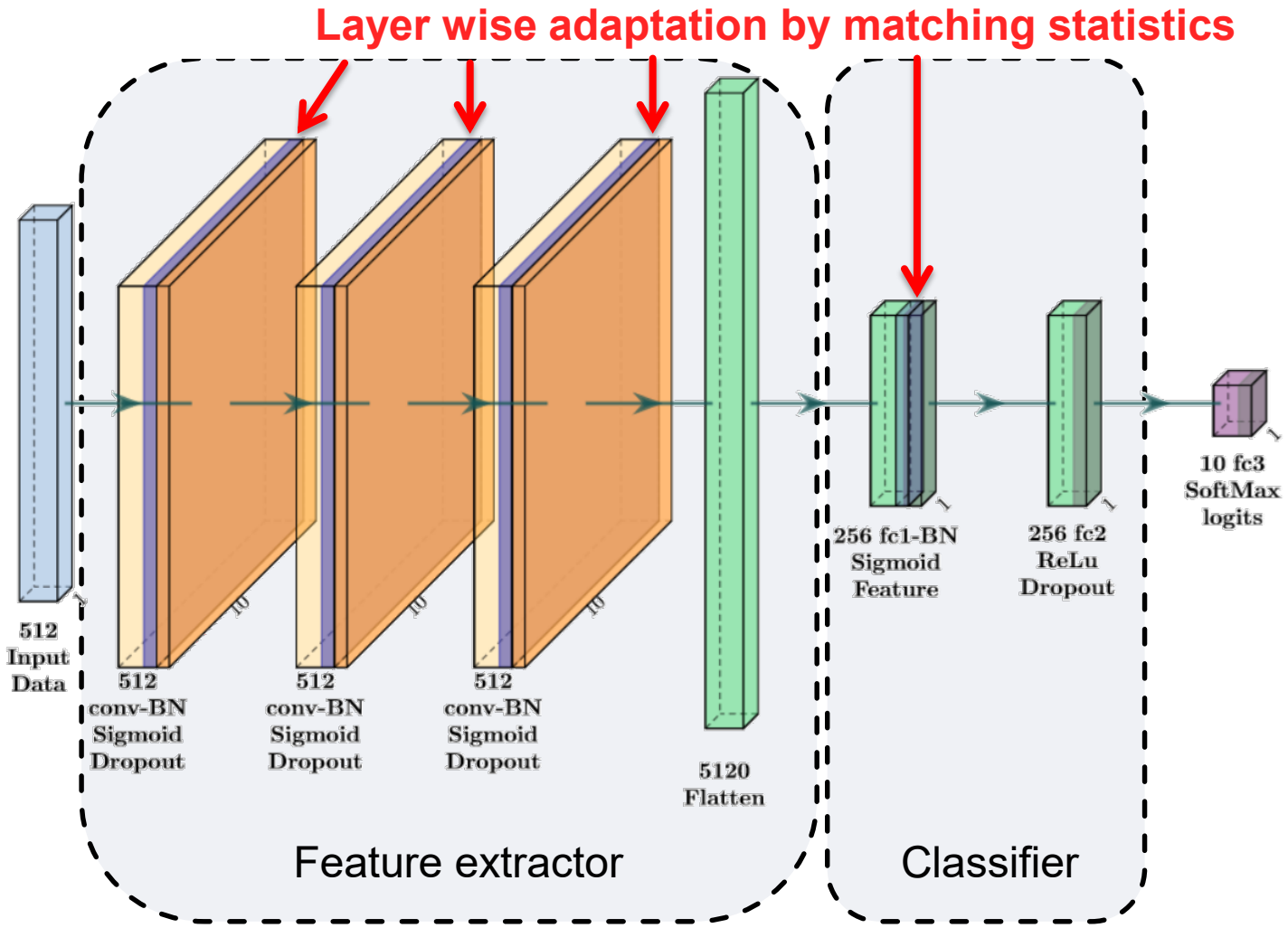
How should we add

domain adaptation ability

to this network?

[14] X. Li, W. Zhang, and Q. Ding, "Cross-domain fault diagnosis of rolling element bearings using deep generative neural networks," IEEE Transactions on Industrial Electronics, 2018.

Adaptive Batch Normalization (AdaBN)



$$\hat{x_j} = \frac{x_j - \mathbb{E}[x_{.j}]}{\sqrt{Var[x_{.j}]}}$$

$$y_j = \gamma_j \hat{x_j} + \beta_j,$$

Keep different batch norm statistics for source and target.

- Pros
  Simplicity.
- Cons
  Performance not optimized

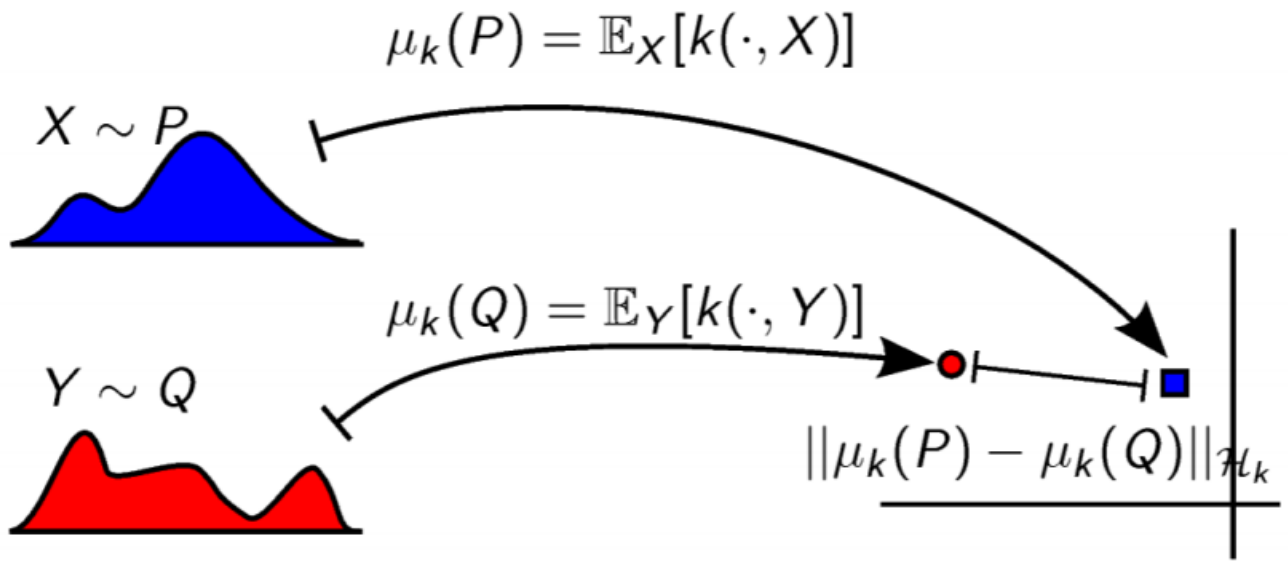
Li, Yanghao, et al. "Adaptive batch normalization for practical domain adaptation." *Pattern Recognition* 80 (2018): 109-117.

AdaBN adapts via statistics,

can we **adapt** between domains via

Backpropagation?

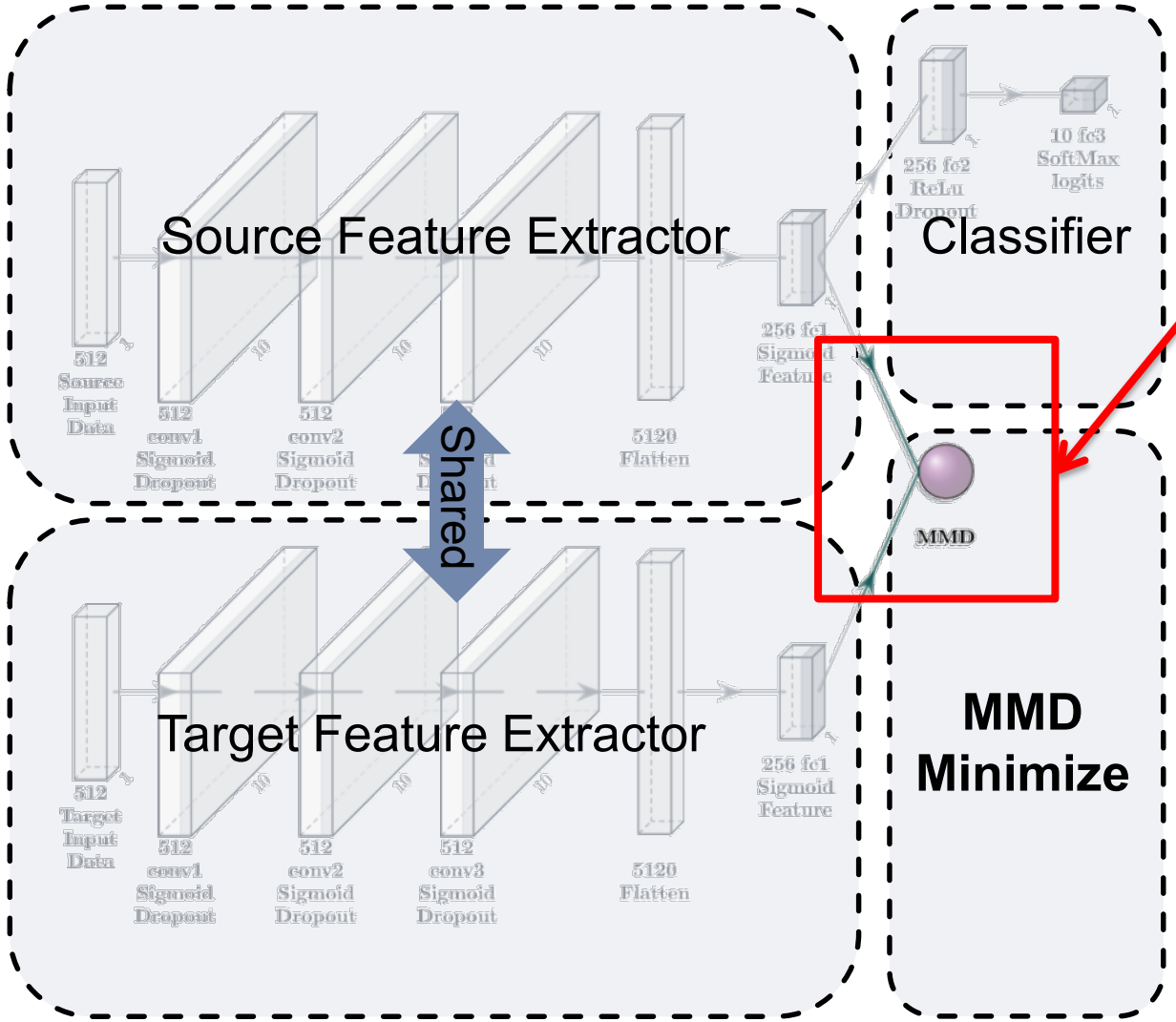
Explicitly model the source-target discrepancy!



Off-the-shelf solution: Maximum Mean Discrepancy (MMD)

Gretton, Arthur, et al. "A kernel two-sample test." *Journal of Machine Learning Research* 13.Mar (2012): 723-773.
Dino Sejdinovic, Inference with Kernel Embeddings, RSS Conference, Manchester

Maximum Mean Discrepancy Minimization



MMD as an additional loss to minimize.

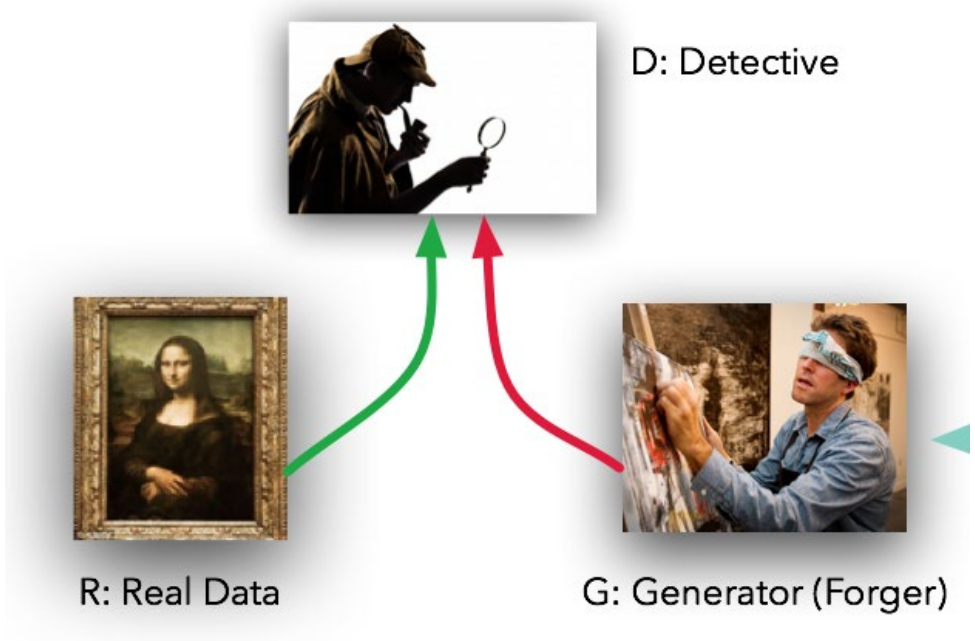
Directly minimize the discrepancy between source and target features.

- Pros
  Finer alignment.
- Cons
  Multiple kernels needed
  model complexity O(n^2)

Explicit discrepancy modelling is **expensive**, $O(n^2)$

alternative **cheaper** solution?

Adversarial Training



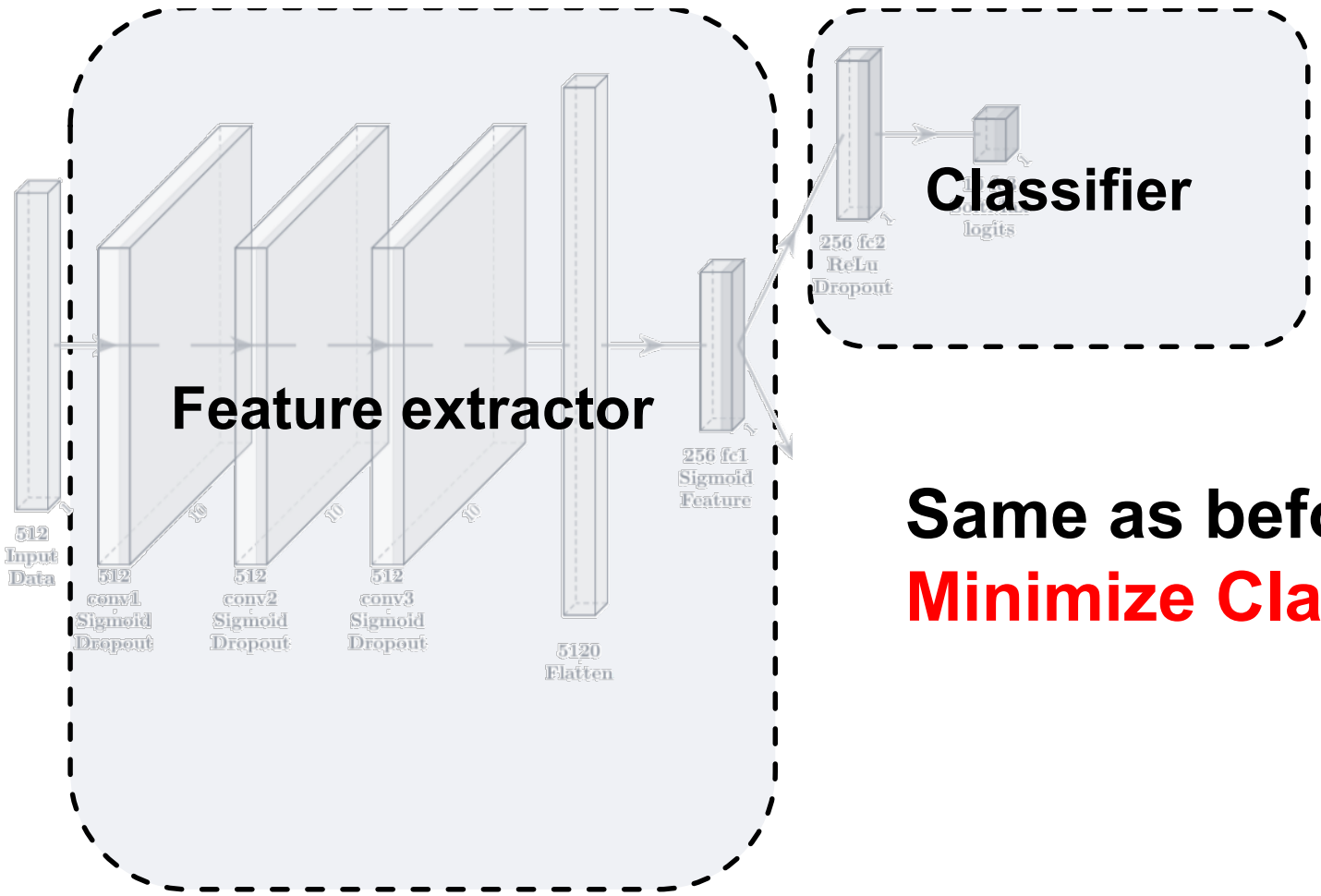
GAN

StyleGAN

https://github.com/devnag/pytorch-generative-adversarial-networks

Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." *CVPR*. 2019.

Domain-Adversarial Training (DANN)

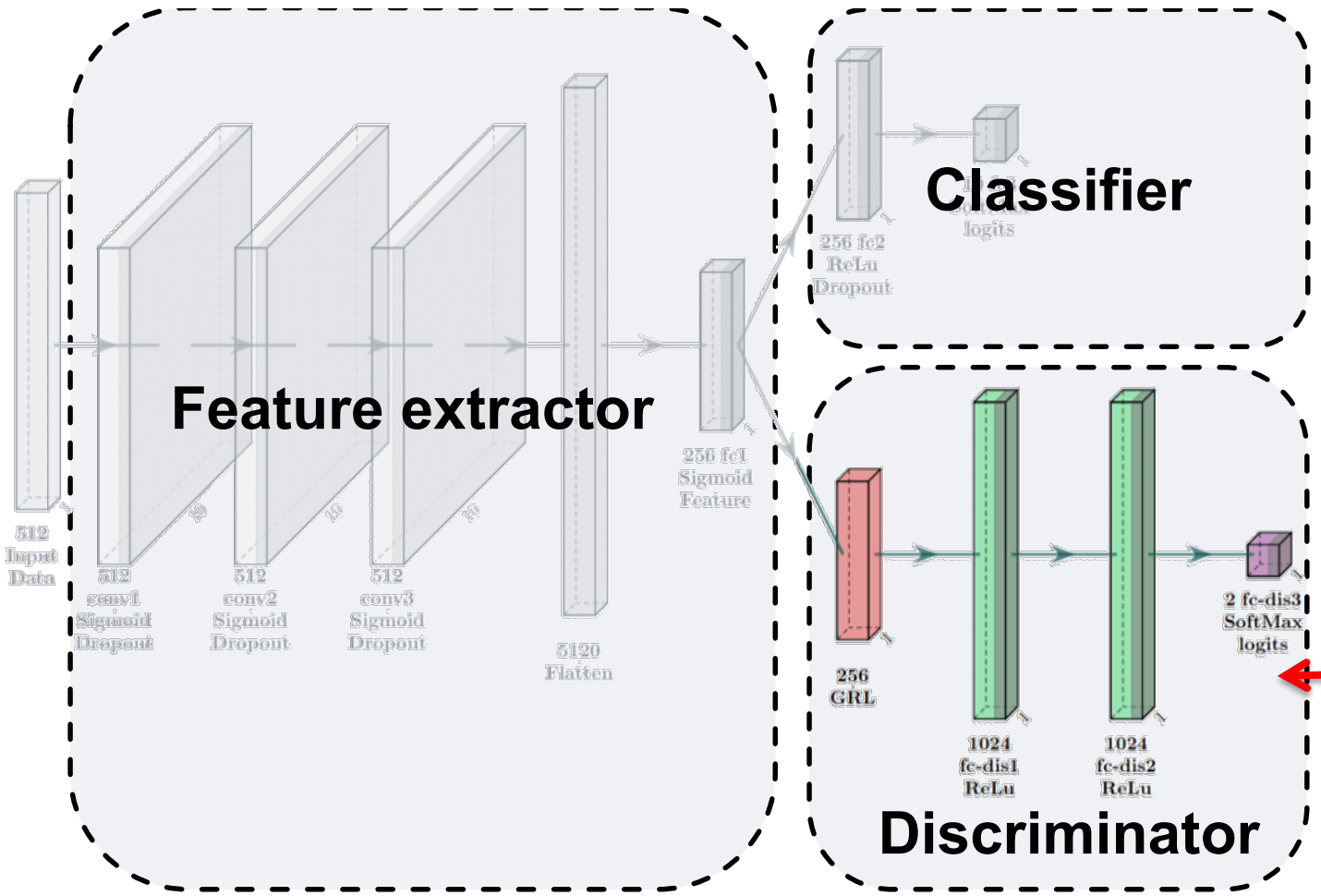


Same as before

Minimize Classification Loss

Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” arXiv preprint arXiv:1409.7495, 2014

Domain-Adversarial Training (DANN)



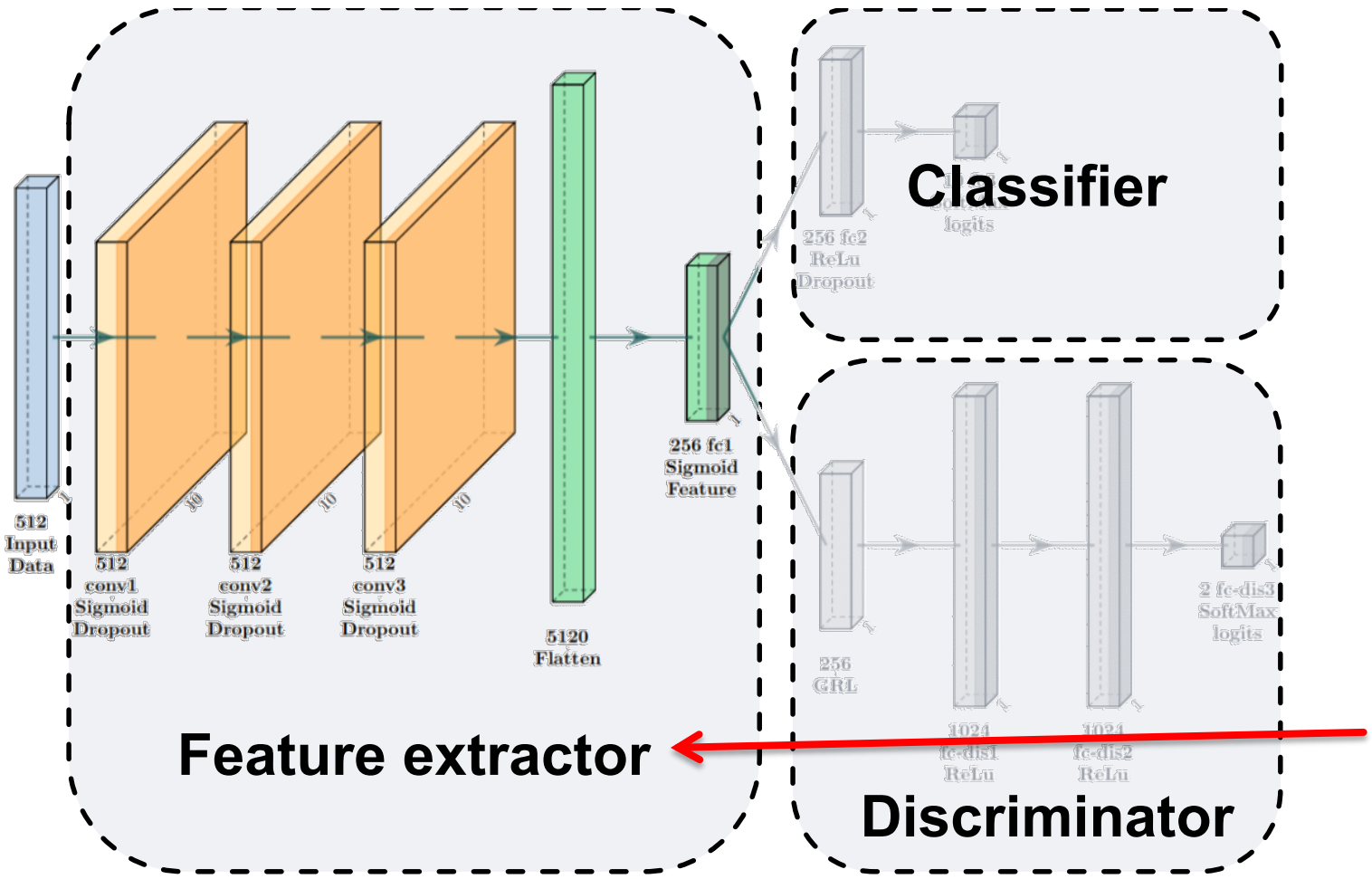
D: Detective

Train a **discriminator** to tell target from source

Minimize Discriminator loss w.r.t discriminator weights

Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," arXiv preprint arXiv:1409.7495, 2014 | 14.09.2020 | 19

Domain-Adversarial Training (DANN)

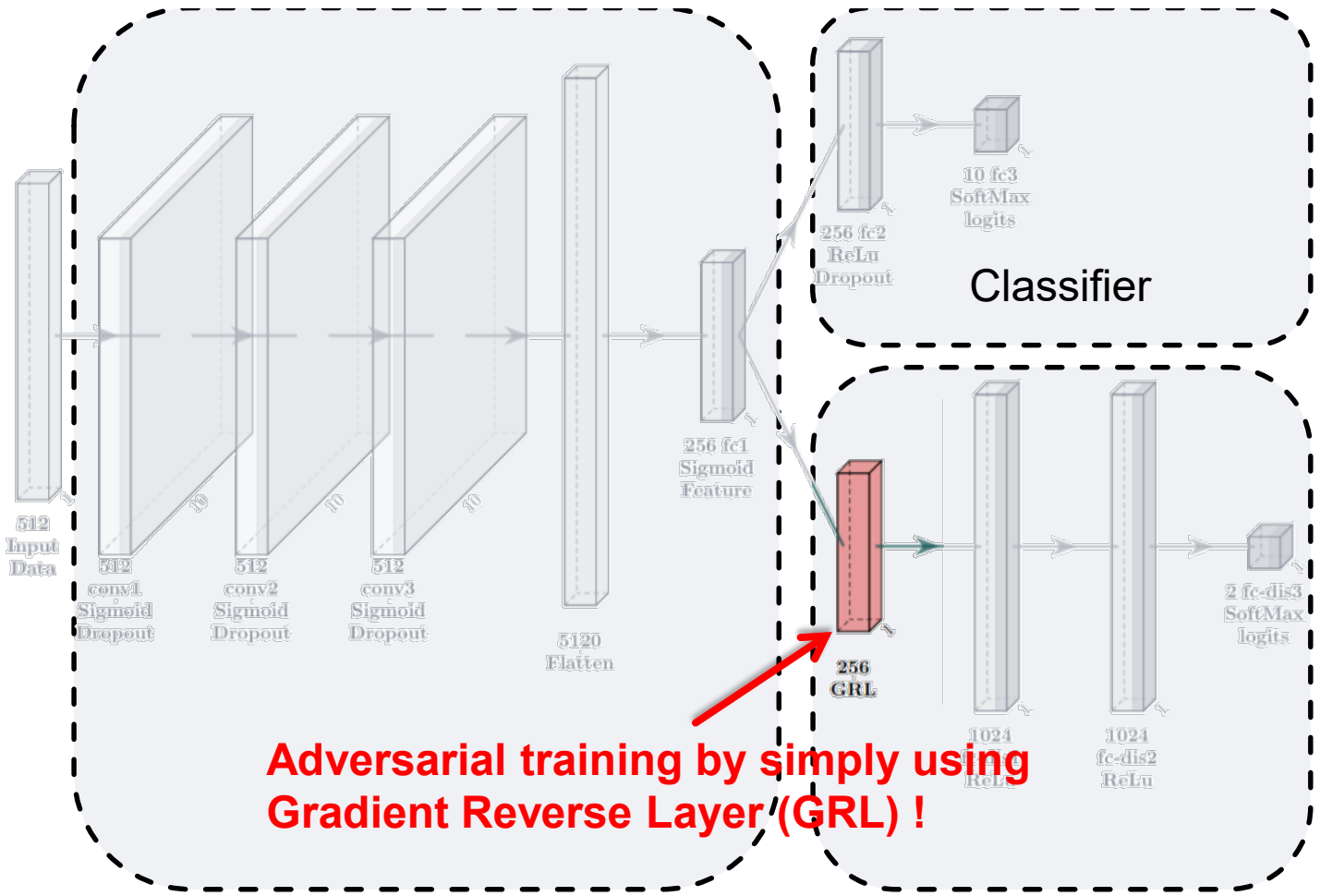


Encourage the feature extractor to **fool the discriminator**, and generate **unbiased** features

Maximize Discriminator loss w.r.t feature extractor weights

Adversarial Training!

Domain-Adversarial Training (DANN)

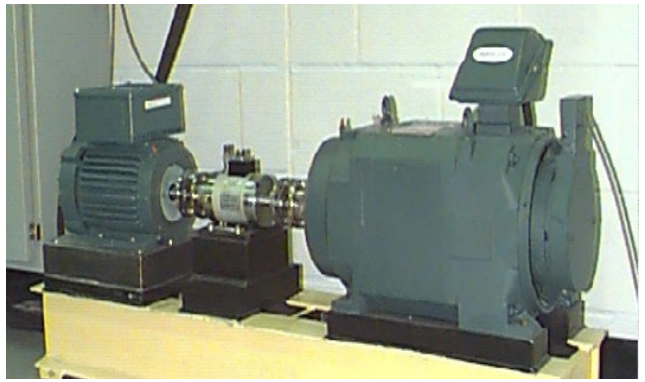


Proved to be able to implicitly **reduce H-Divergence** between source and target distributions.

O(n^1) instead of O(n^2)

Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," arXiv preprint arXiv:1409.7495, 2014

Case Study on CWRU(Case Western Reserve University) Bearing Dataset



- Dataset information
  - Ten Classes classification

Fault	Class Label									
	0	1	2	3	4	5	6	7	8	9
Loc	NA[1]	IF	IF	IF	BF	BF	BF	OF	OF	OF
Size	0	7	14	21	7	14	21	7	14	21

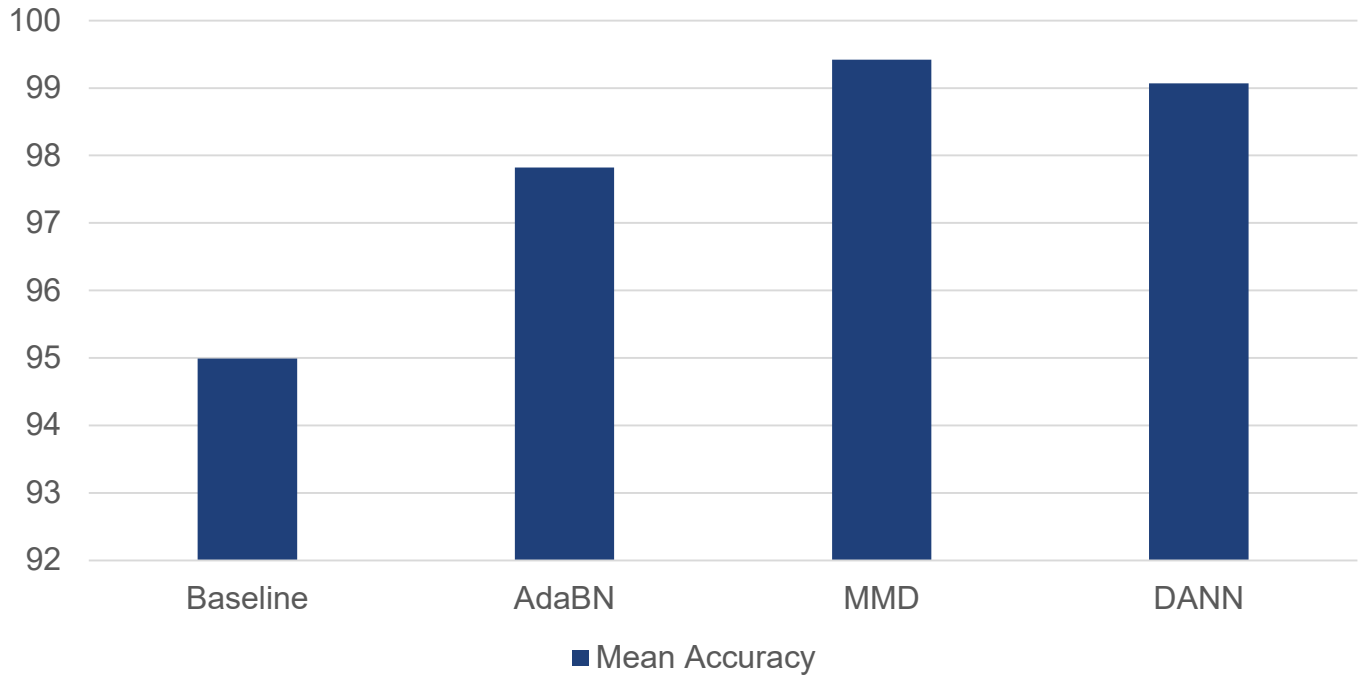
  - Domain adaptation to a new operating condition

    Diagnosing faults on the unlabelled target bearing under a different load.

Motor Load (HP)	Approx. Motor Speed (rpm)
0	1797
1	1772
2	1750
3	1730

Model Performance on CWRU Dataset

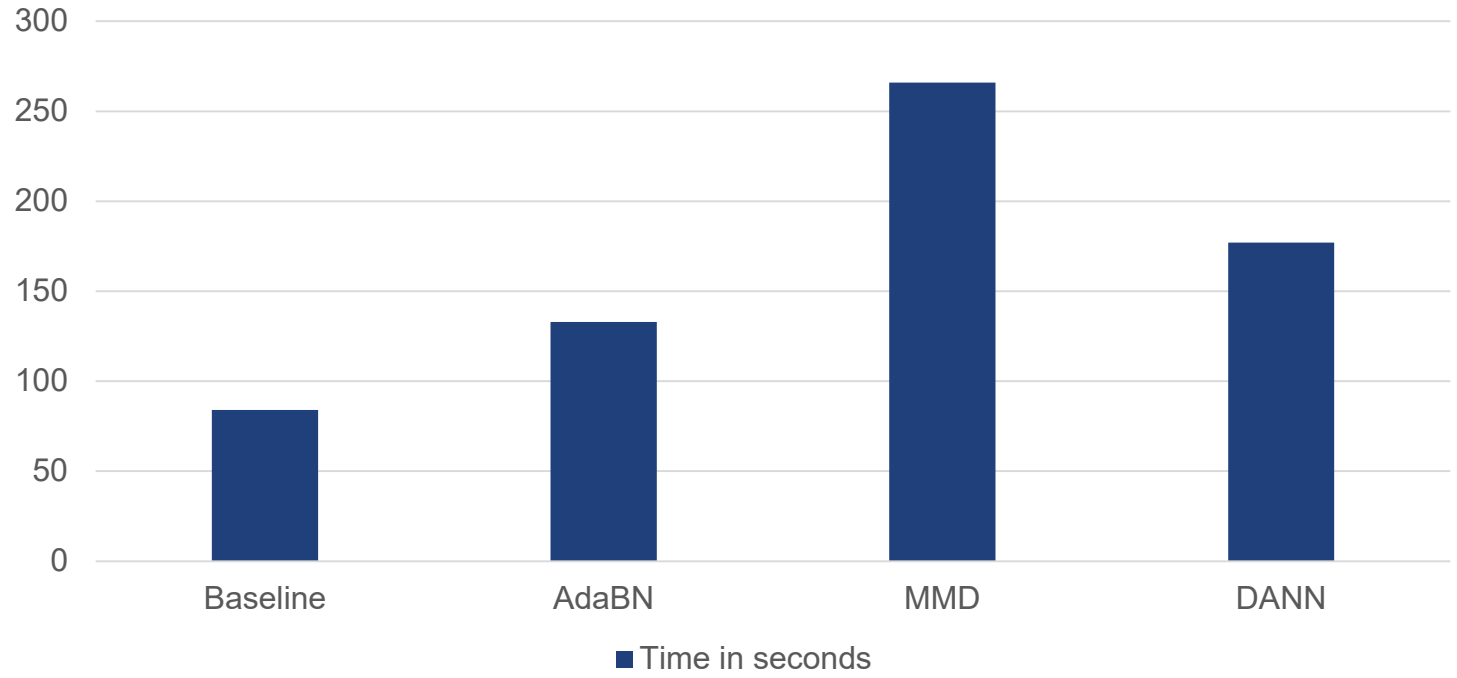
- Same basic architecture
- Same budget for hyper-parameter tuning
- Same optimizer and learning rate



Average Accuracy over **5** runs, trained on a NVIDIA GTX 1080

Wang, Qin, Gabriel Michau, and Olga Fink. "Domain adaptive transfer learning for fault diagnosis." *2019 Prognostics and System Health Management Conference (PHM-Paris)*. IEEE, 2019.

Model Efficiency: Training Time



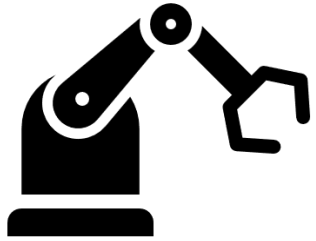
Linear Linear Quadratic Linear

w.r.t. # training samples

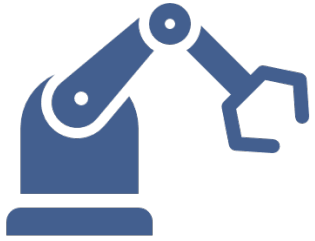
Quick Summary

- All domain adaptation methods managed to **improve** the model **performance**.
- The adversarial method provided competitive results using **significantly less** training time.
- AdaBN is fast but performs worse than other two.

Domain Adaptation for Fault Diagnosis



Source Machine



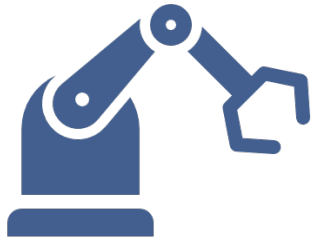
Target Machine

- Newly setup, Not yet labeled
- **Different** operating conditions
- **Non-identical** component

Have we solved the problem?

Until now, **full access to target faults is assumed**,

but some **faults are rare.**



DA for Fault Diagnosis

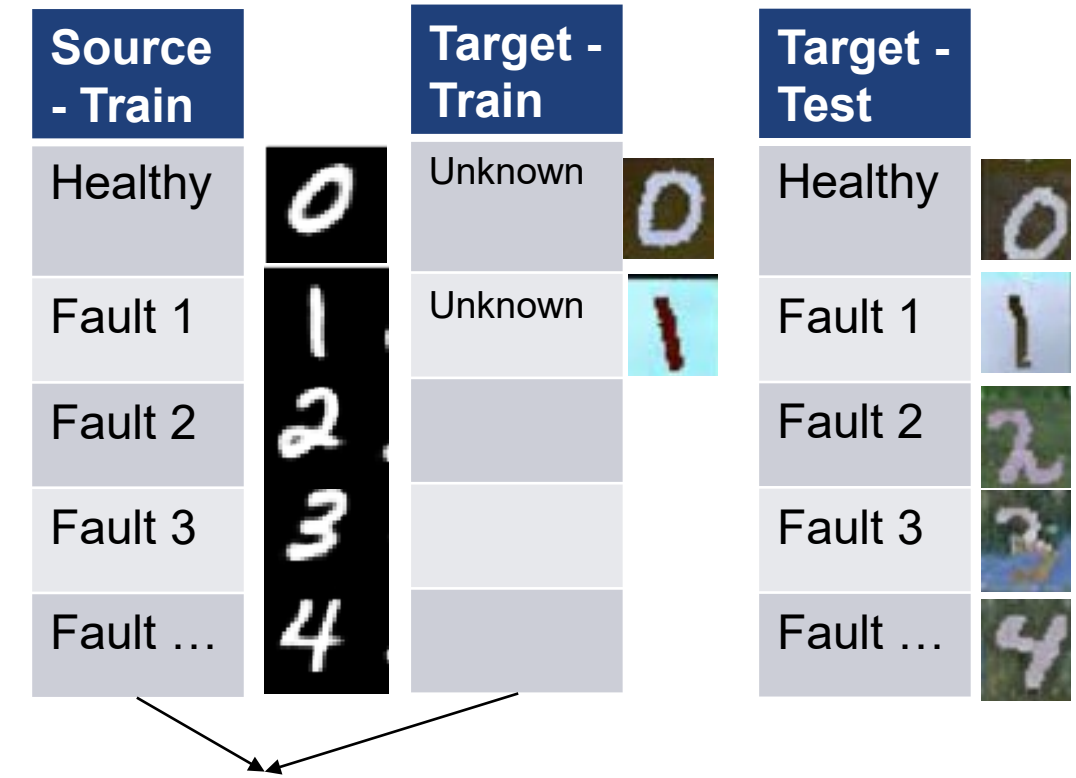
Source - Train	Target - Train	Target - Test
Healthy	Unknown	Healthy
Fault 1	Unknown	Fault 1
Fault 2	Unknown	Fault 2
Fault 3		Fault 3
Fault x		Fault x

Missing-Class Aware Adaptation

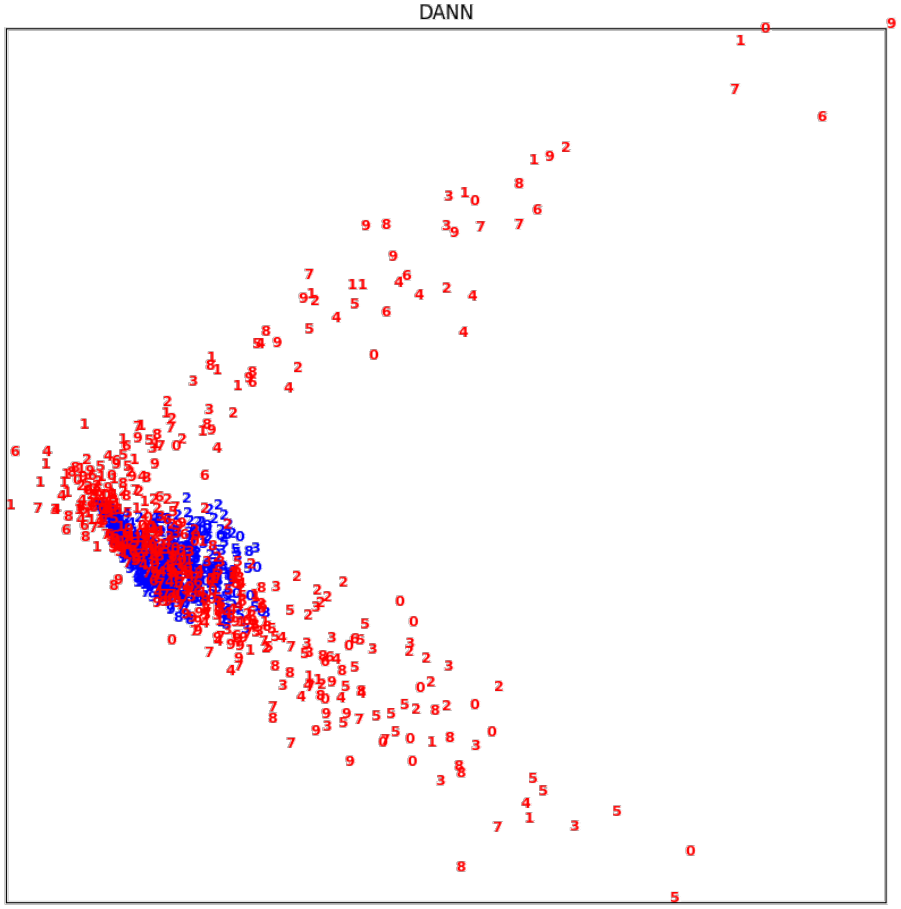
Most faults are rare !

Missing Classes Have an Effect: A MNIST example

Missing-class DA with 10 classes



Missing Classes Have an Effect: A MNIST example



Feature Visualization on DANN Method

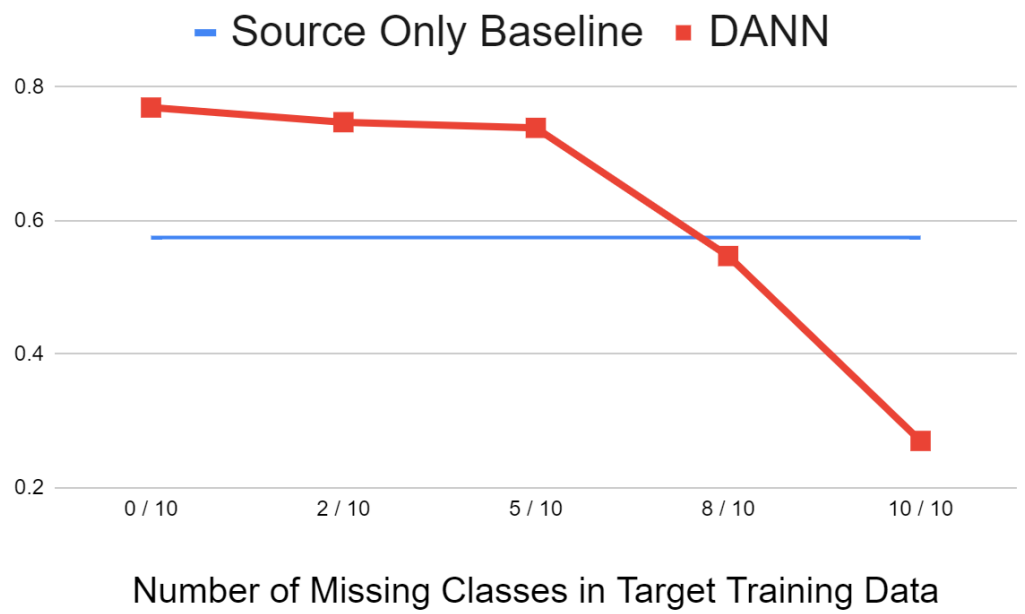
Part of Target Features

are aligned with

Complete source data

A good alignment should see red overlap with blue **on all classes.**

Wrong Alignment Due to Missing Classes

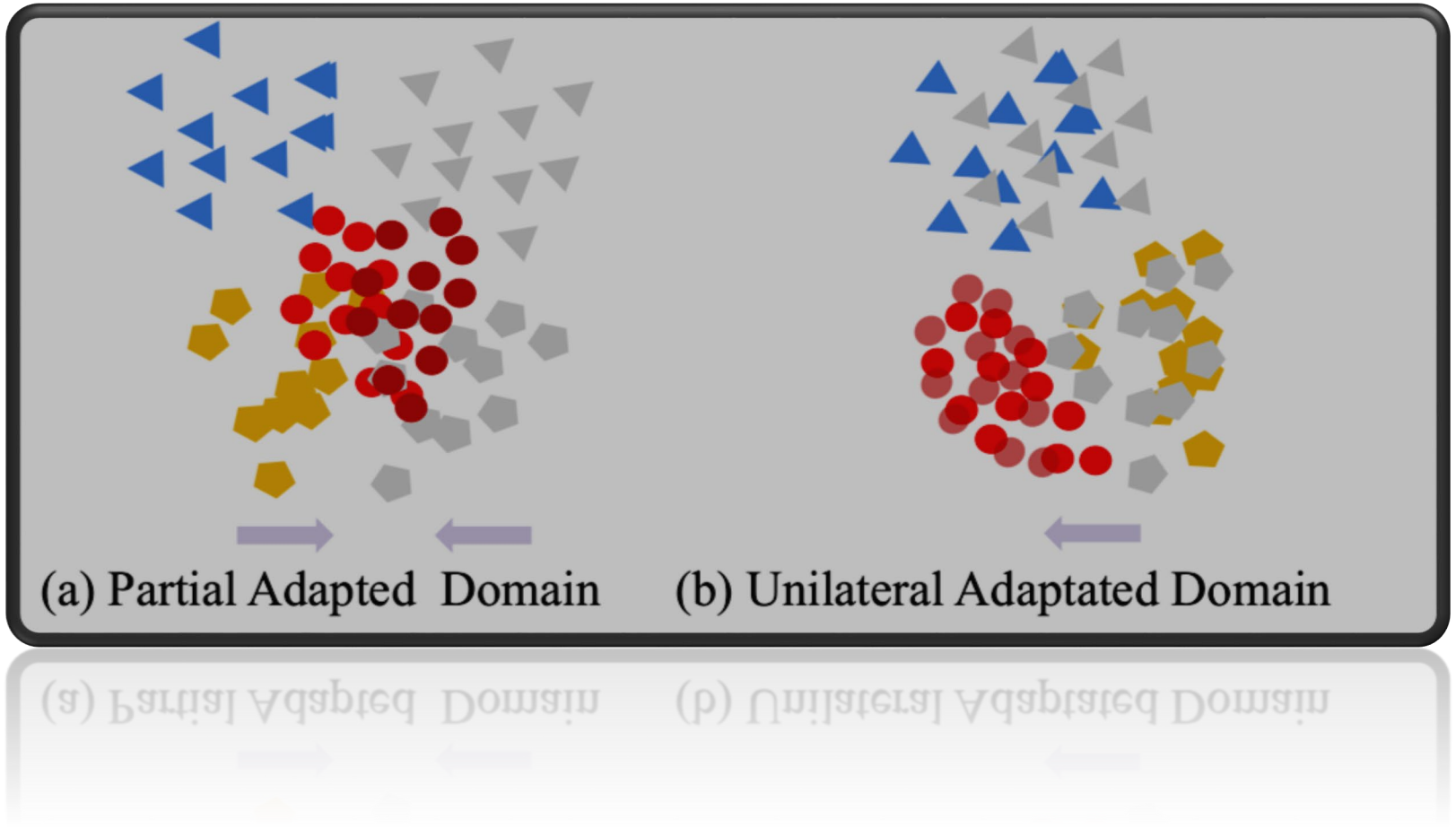


DANN (Ganin et al 2016) performance on MNIST->MNIST-M task.

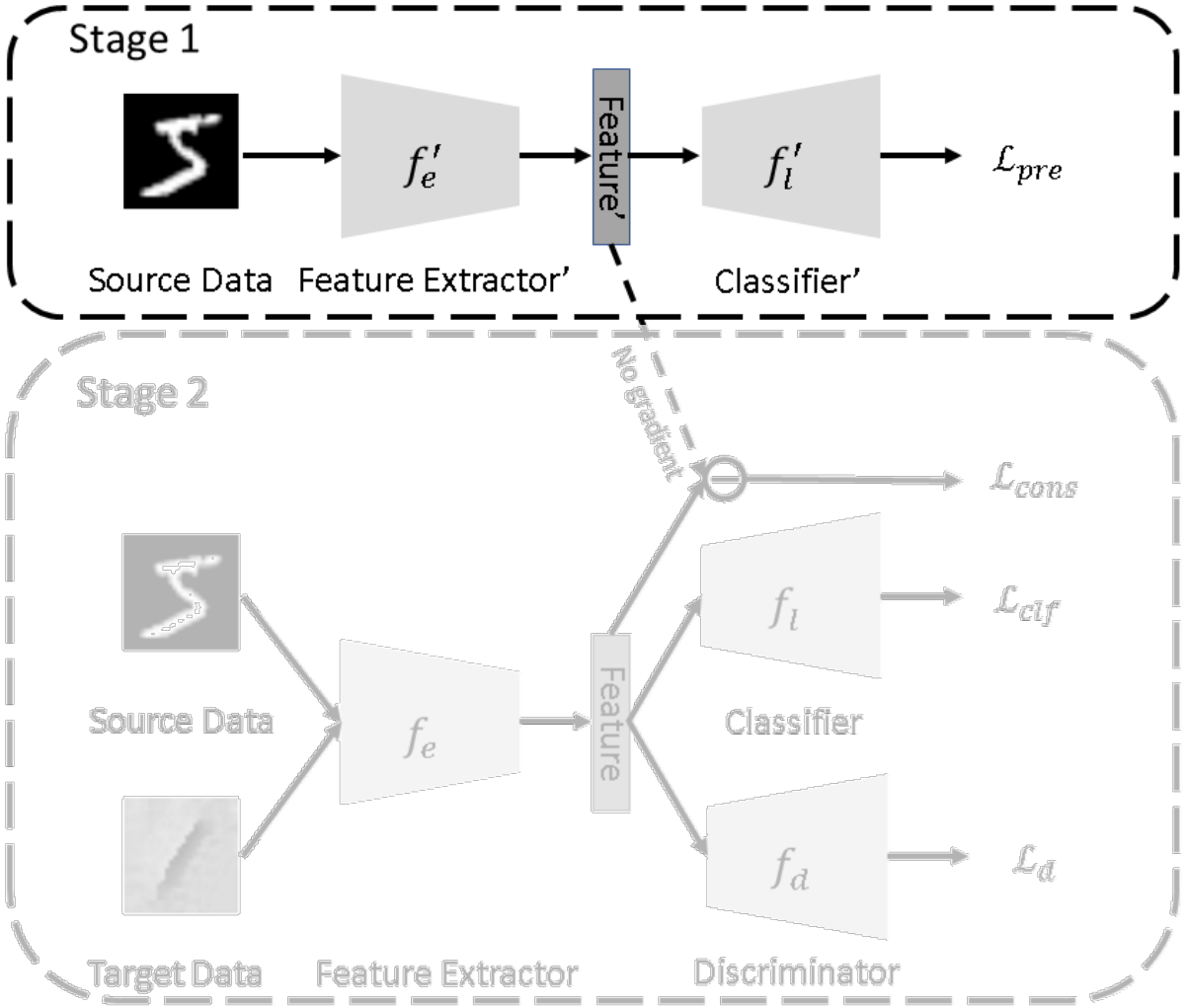
- **Discriminability** is **hurt** because of **misalignment**.

How can we
preserve discriminability
learned **from source** domain?

Our Proposed Method: Unilateral vs Bilateral



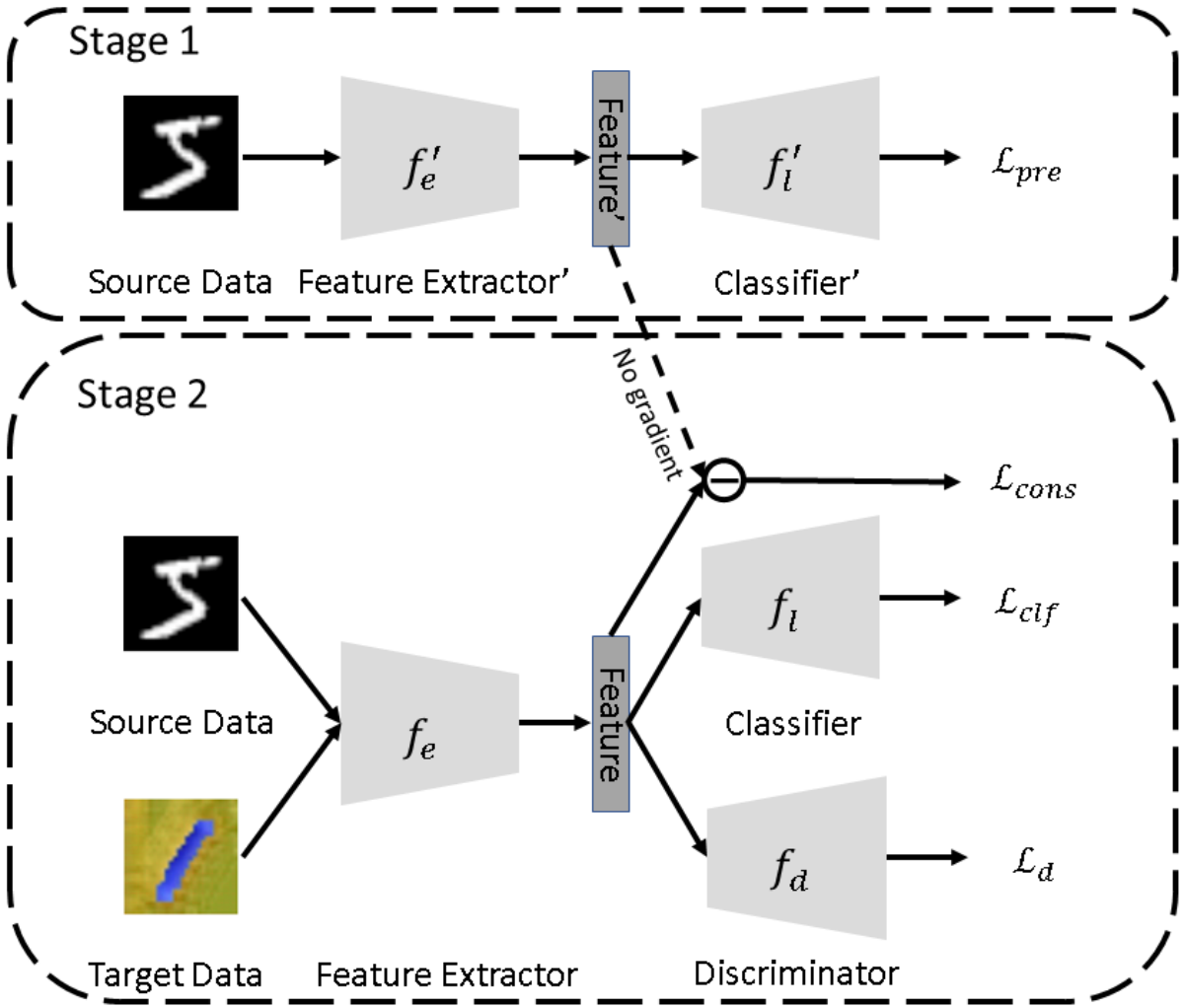
Preserve discriminability learn from source by **unilateral alignment**



Stage 1

- Train a source feature extractor
- Learn discriminability from source

Preserve discriminability learn from source by **unilateral alignment**



Stage 2

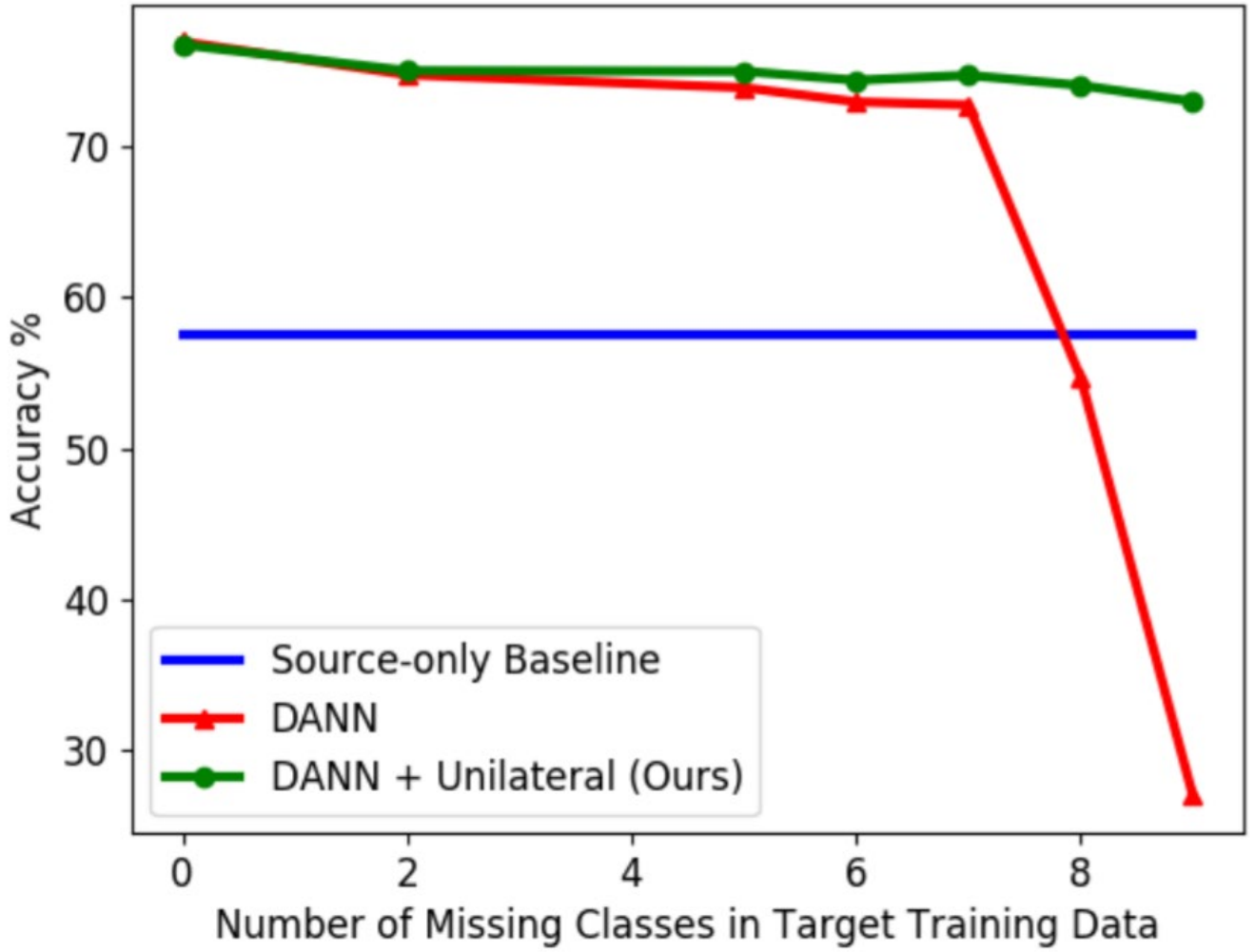
- Make use of this additional info to construct a **consistency loss**

$$\mathcal{L}_{cons} = \frac{1}{K}\sum_{j=1}^{K} ||f(x_s)_j - f'(x_s)_j||_1,$$

- Final loss contains three parts.

$$\mathcal{L} = \mathcal{L}_{clf} + \mathcal{L}_d + \lambda_{cons}\mathcal{L}_{cons}$$

Evaluation


70
60
50
40
30
Accuracy %
0
2
4
6
8
Number of Missing Classes in Target Training Data
Source-only Baseline
DANN
DANN + Unilateral (Ours)

Better alignment in case of missing faults

One step closer to

DA for fault diagnosis in the wild.

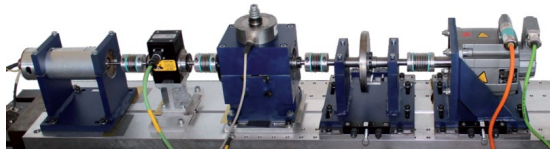
Imbalanced data

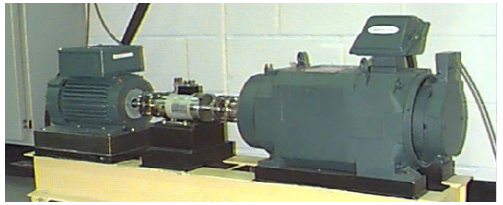
Heterogenous input

Continuous domains

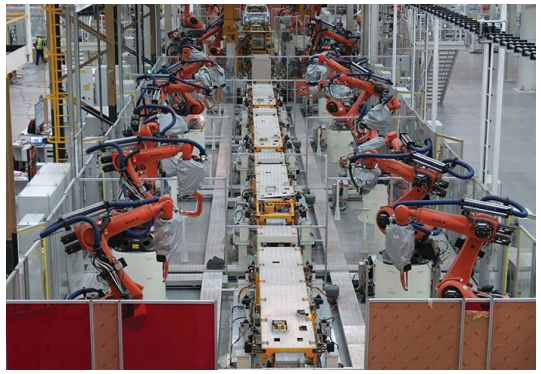
Future Applications

- From one operating condition to another





- From one machine to a fleet



- From one manufacturer to another

Take-home Message

- **Domain Adaptation** is almost ready for real life industry applications.

More about our projects on domain adaptation:

https//ims.ibi.ethz.ch
https://qin.ee