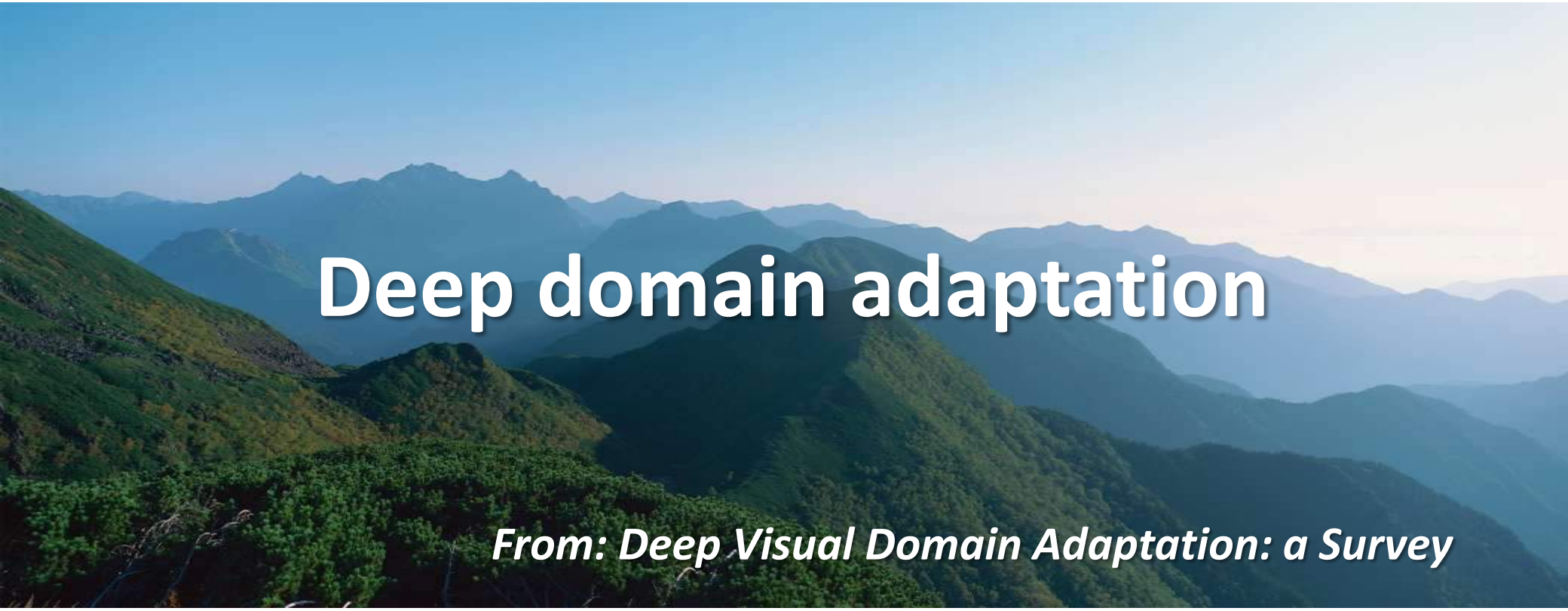
北京郵電大學
BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS



Deep domain adaptation

From: Deep Visual Domain Adaptation: a Survey

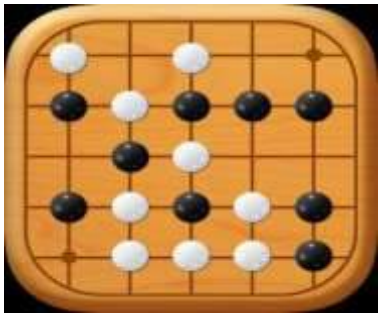
Mei Wang

Beijing University of Posts and Telecommunications

2018.12.28

Why do we need transfer learning

Machine do have its weakness, it has no ability to "transfer learning". The trained models can not be adopted in different related scenarios, for example AlphaGo can't play Chinese chess.



Slide credit: Qiang Yang

Why do we need transfer learning

In the human evolution, the ability of transfer learning is very important. We can extend learned knowledge to other scenarios. For example, after learning riding bicycles, it is very easy to ride motorcycles.

Why do we need transfer learning

Although, deep learning has achieved great success in many tasks, it is non-trivial to address these problems in application:

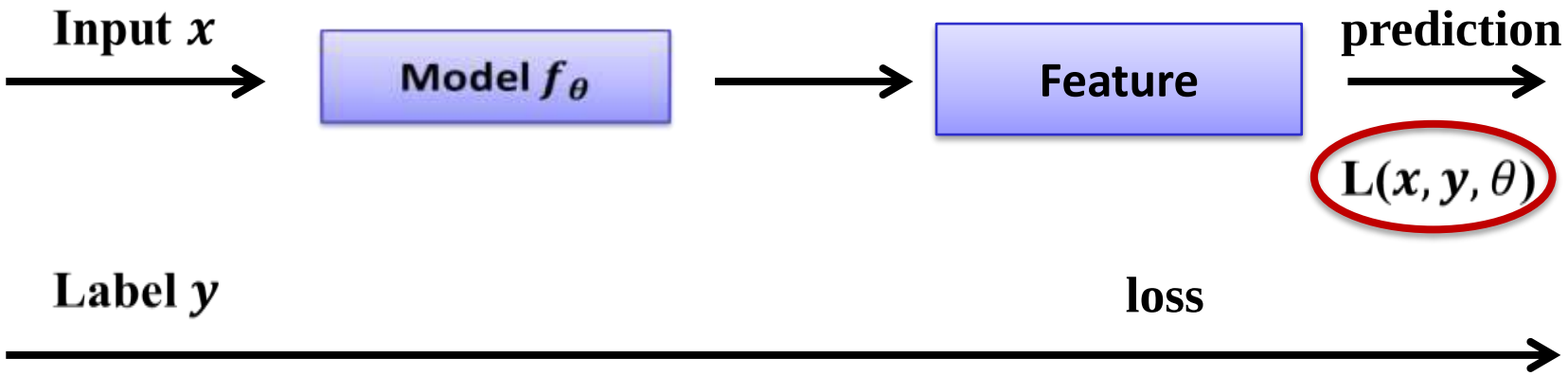
1. **Small data**
2. **Reliability**
3. **Personality**

Slide credit: Qiang Yang

What is transfer learning?

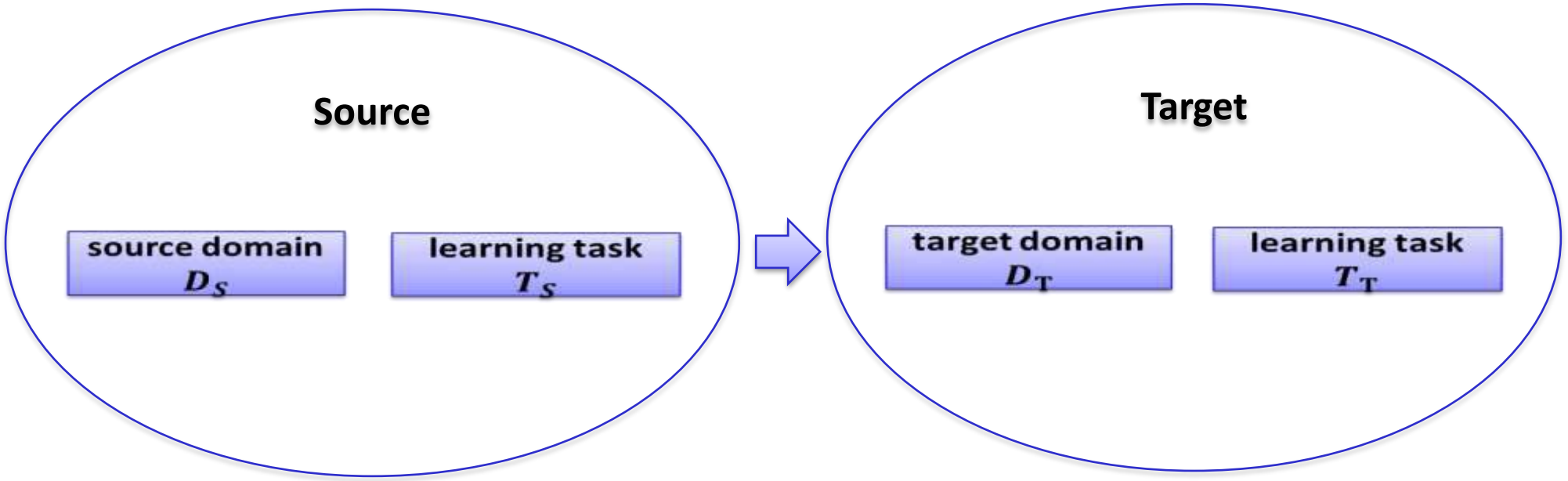
- **Traditional deep learning when training and testing share similar distribution:**

$$\text{Empirical Risk: } \min \frac{1}{n}\sum_{i=1}^{n} L(x_i, y_i, \theta)$$



What is transfer learning?

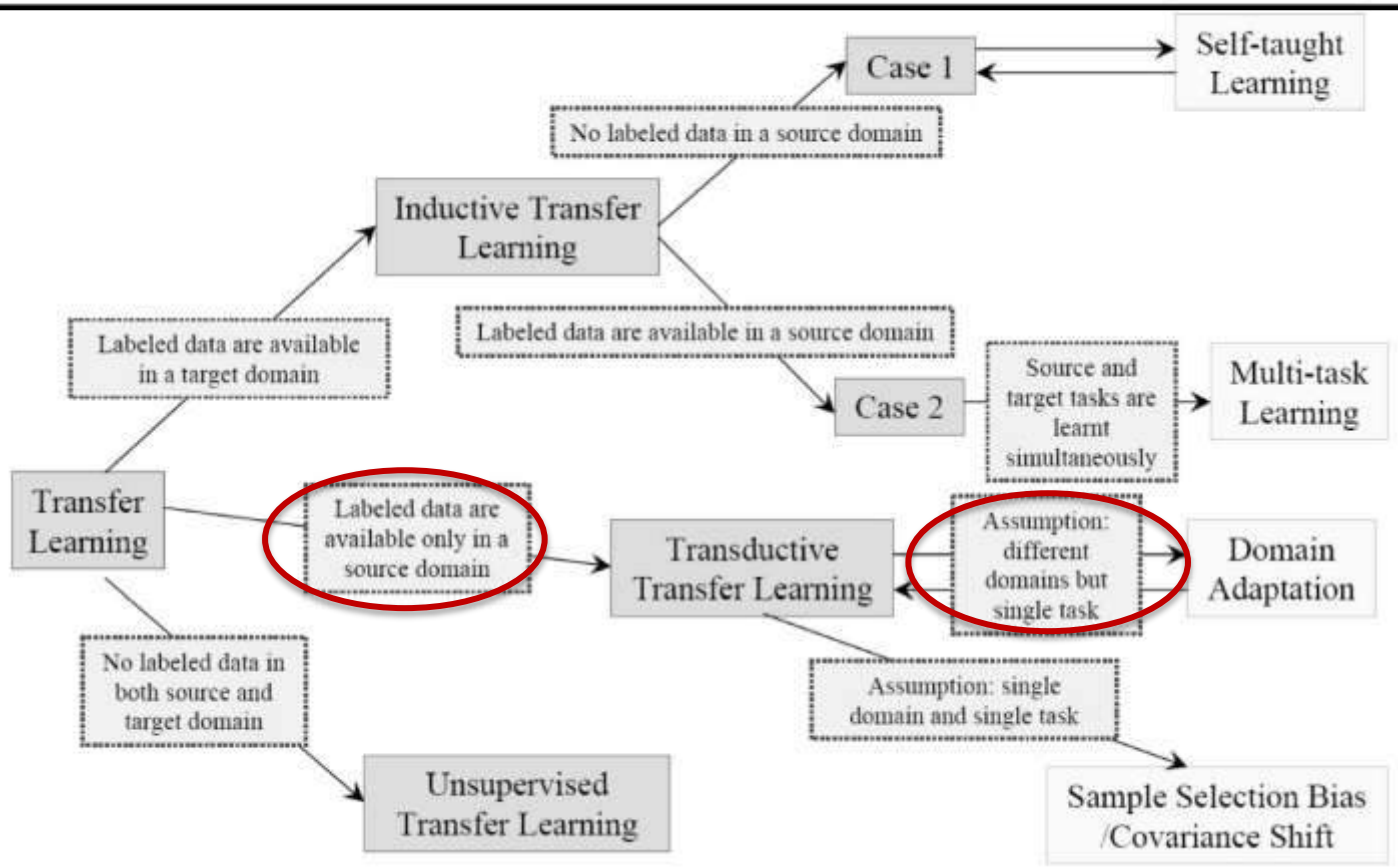
Given a source domain D_S and learning task T_S, a target domain D_T and learning task T_T, transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$, or $T_S \neq T_T$.



What is domain adaptation?

Domain adaptation:

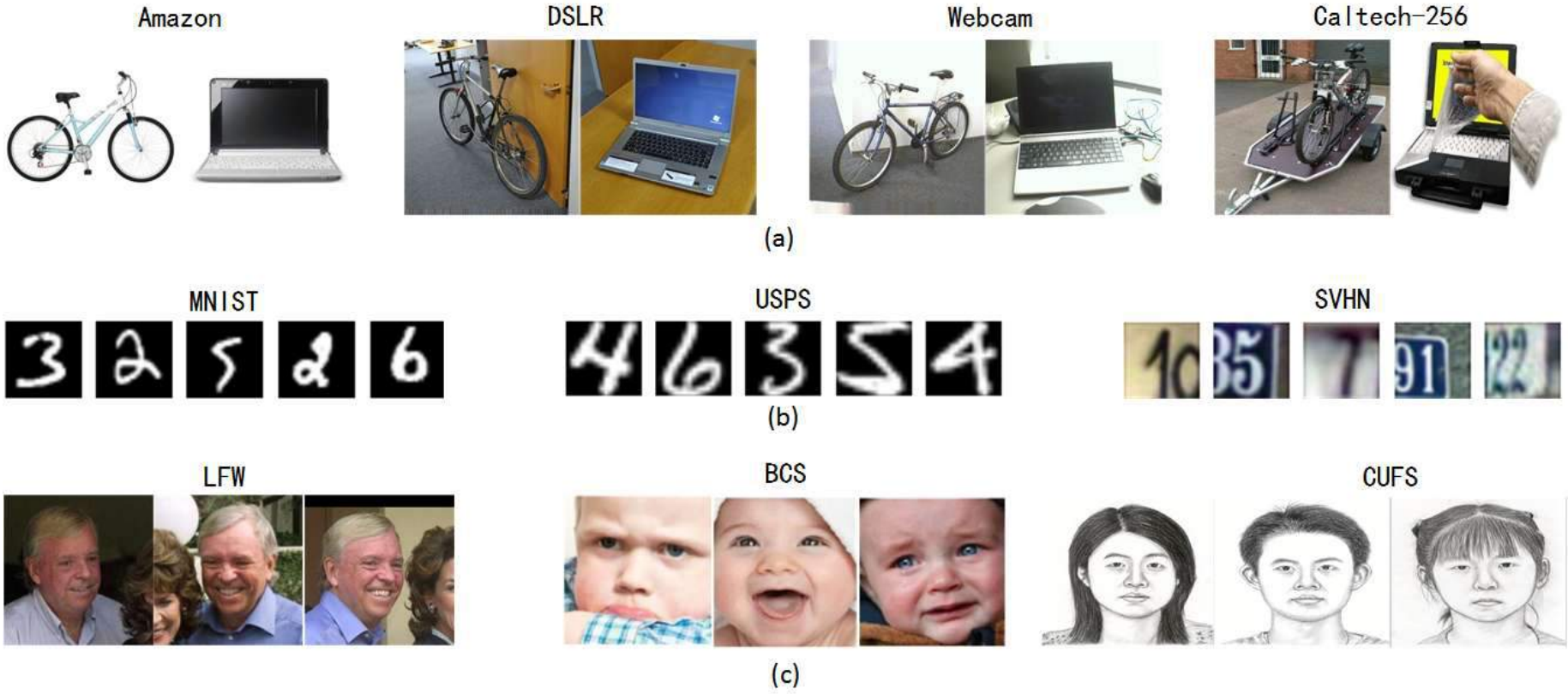
- **Source domain: abandont amout of labeled data.**
- **Target domain: few or no labeled data.**
- **Same tasks, but different data distribution.**



[1] Pan S J, Yang Q. A survey on transfer learning[J]. IEEE Transactions on knowledge and data engineering, 2010, 22(10): 1345-1359.

What is domain adaptation?

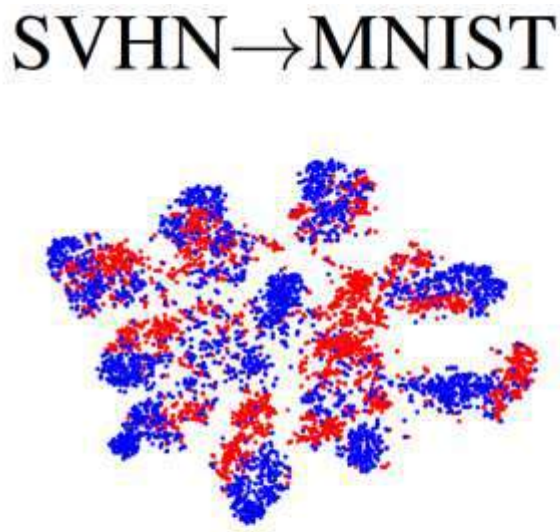
Due to many factors (e.g., illumination, pose, and image quality), there is always a distribution change or domain shift between two domains that can degrade the performance.



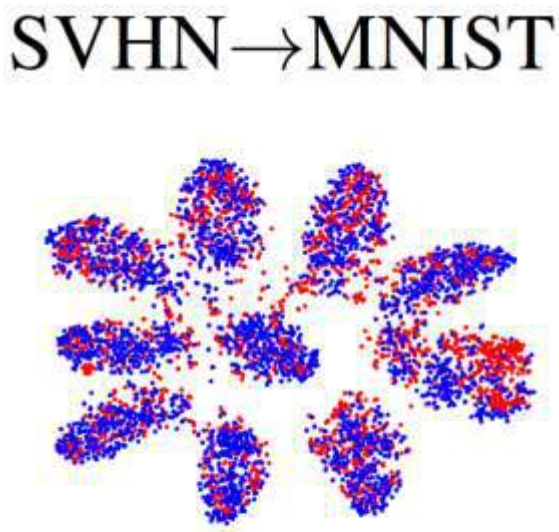
[2] M. Wang and W. Deng. Deep visual domain adaptation: A survey. Neurocomputing, 312:135 – 153, 2018.

What is domain adaptation?

Due to distribution change or domain shift between two domains, the classifier learned for the source domain can not be applied to the target domain.



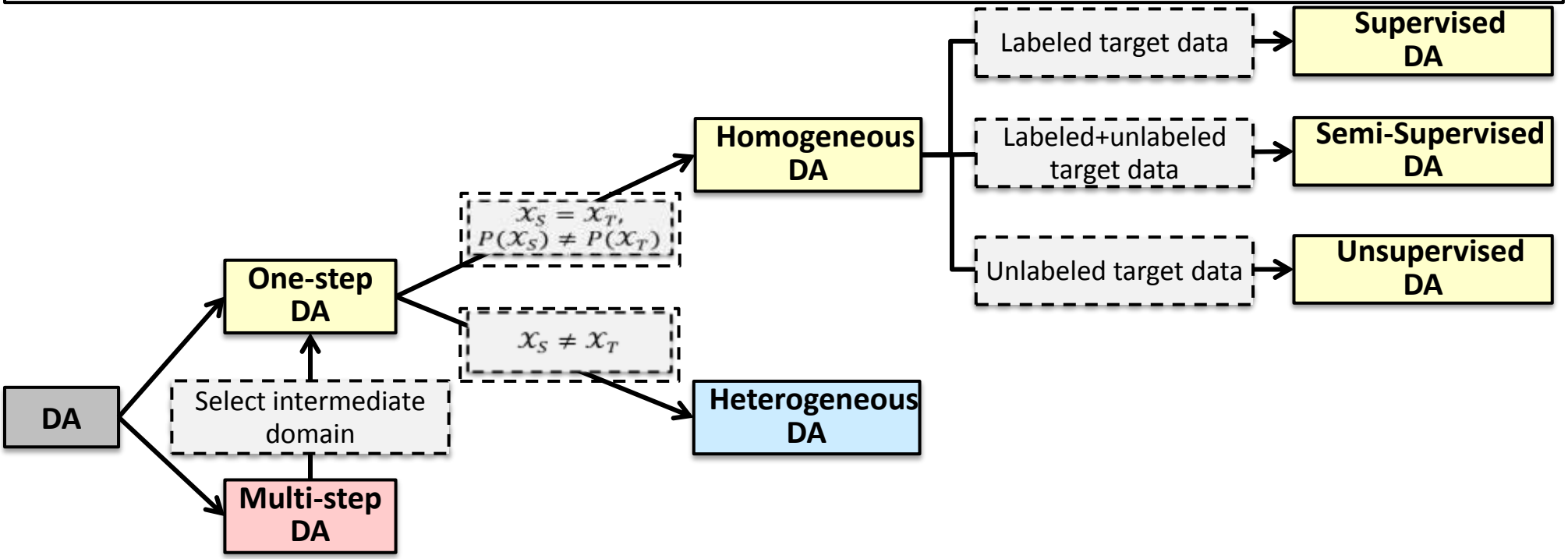
(a) non-adapted



(a) adapted

What is domain adaptation?

- **Supervised DA, a small amount of labeled target data are present. However, the labeled data are commonly not sufficient for tasks.**
- **Semi-supervised DA, both limited labeled data and redundant unlabeled data in the target domain are available in the training stage, which allows the networks to learn the structure information of the target domain.**
- **Unsupervised DA, no labeled but sufficient unlabeled target domain data are observable when training the network.**

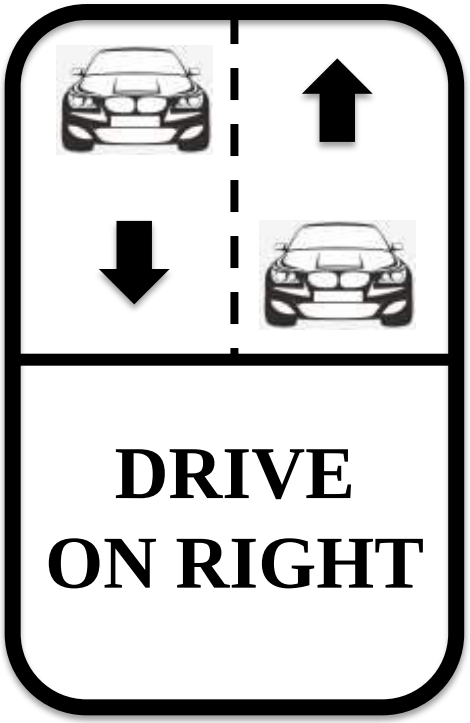


[2] M. Wang and W. Deng. Deep visual domain adaptation: A survey. Neurocomputing, 312:135 – 153, 2018.

The key of transfer learning

The key element of transfer learning is to discover the commonness between the two fields. Once this commonness is discovered, transfer learning becomes easy. We call this commonness as common features in machine learning.

VS



Slide credit: Qiang Yang

Shallow domain adaptation

- **Traditional deep learning**

$$\min \frac{1}{n}\sum_{i=1}^{n} L(x_i, y_i, \theta)$$

- **Feature adaptation**

$$\min \frac{1}{n}\sum_{i=1}^{n} L(\phi(\mathrm{x}_i^s), \mathrm{y}_i^s, \theta)$$

- **Instance adaptation**

$$\min \frac{1}{n}\sum_{i=1}^{n} w_i L(\mathrm{x}_i^s, \mathrm{y}_i^s, \theta)$$

- **Model adaptation**

$$\min \frac{1}{n}\sum_{i=1}^{n} L(\mathrm{x}_i^s, \mathrm{y}_i^s, \theta)$$

Slide credit: Meina Kan

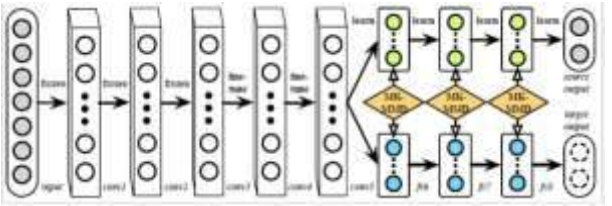
[1] Pan S J, Yang Q. A survey on transfer learning[J]. IEEE Transactions on knowledge and data engineering, 2010, 22(10): 1345-1359.

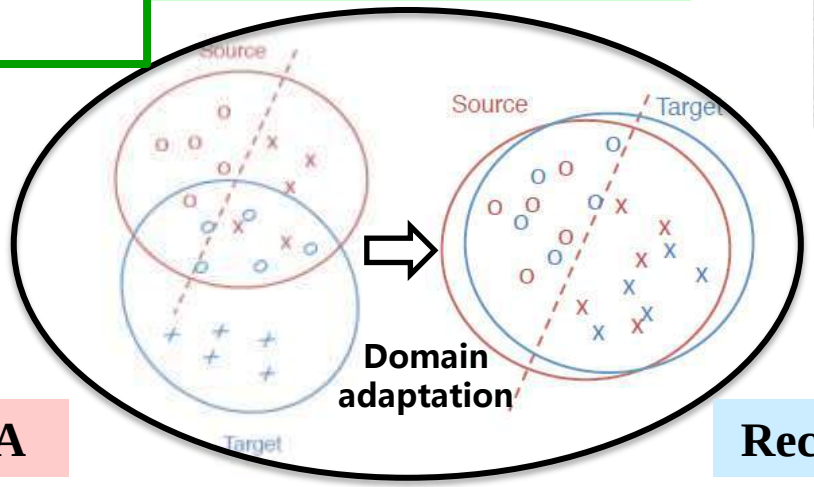
Deep visual domain adaptation: A survey

- class criterion
- statistic criterion
- architecture criterion
- geometric criterion

fine-tuning the deep network with labeled or unlabeled target data to diminish the domain shift

Discrepancy-based DA





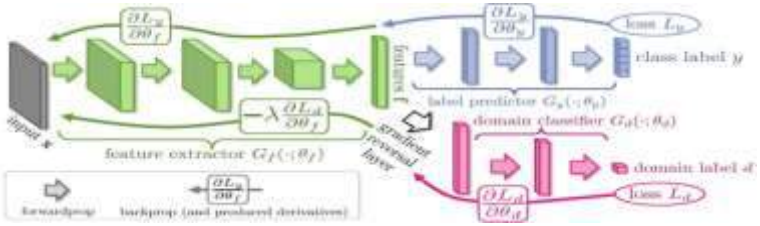
using domain discriminators to encourage domain confusion through an adversarial objective

Adversarial-based DA

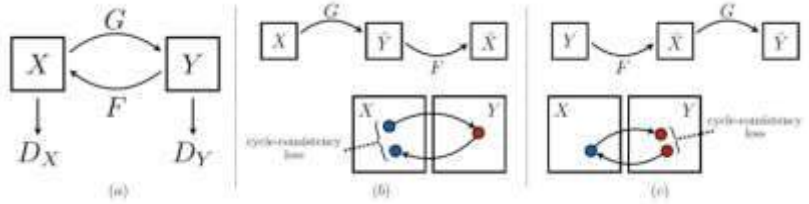
using the data reconstruction as an auxiliary task to ensure feature invariance

Reconstruction-based DA

- generative models
- non-generative models

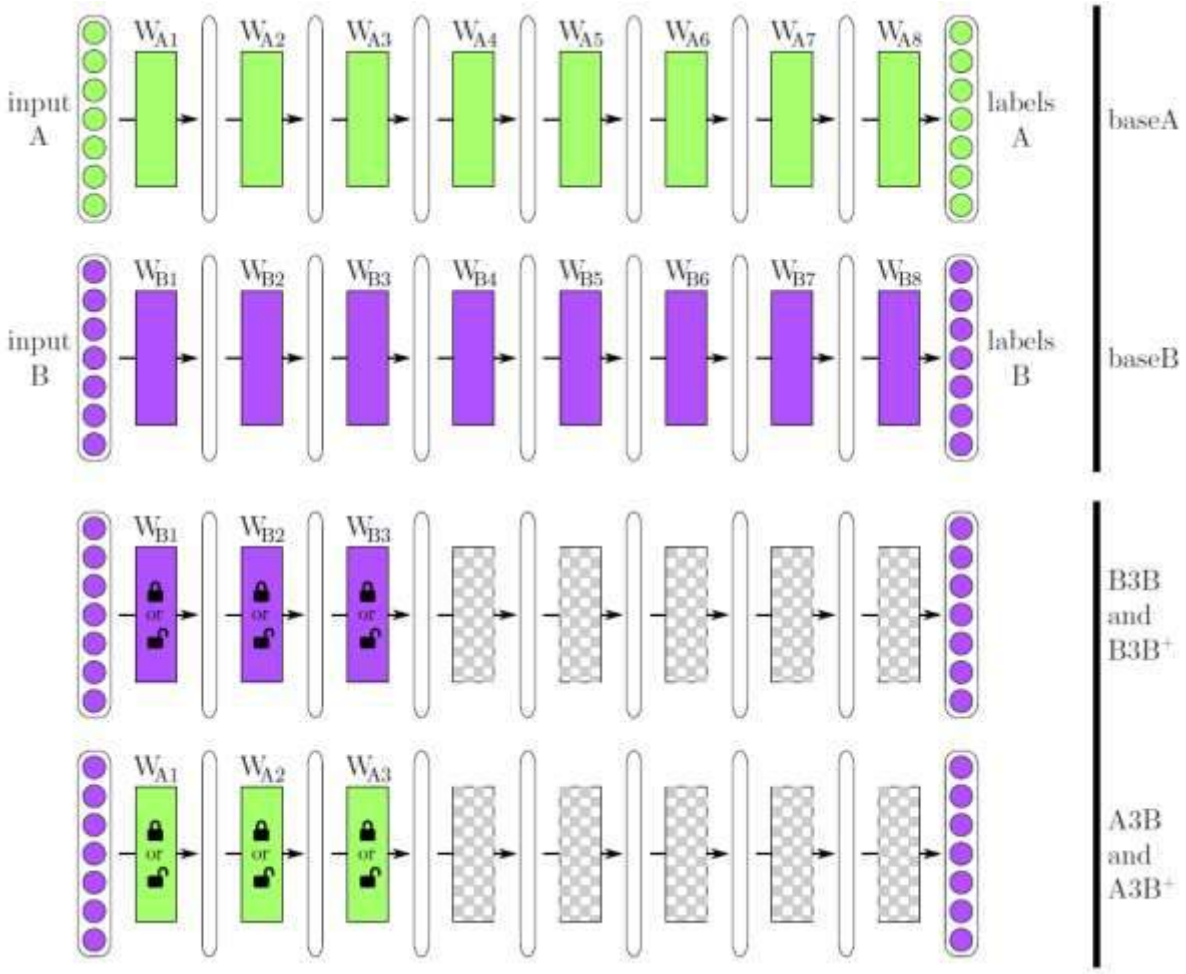


- encoder-decoder reconstruction
- adversarial reconstruction



[2] M. Wang and W. Deng. Deep visual domain adaptation: A survey. Neurocomputing, 312:135 – 153, 2018.

Supervised DA method: fine-tune



First-layer features appear not to be specific to a particular dataset or task, but general in that they are applicable to many datasets and tasks. Features must eventually transition from general to specific by the last layer of the network.

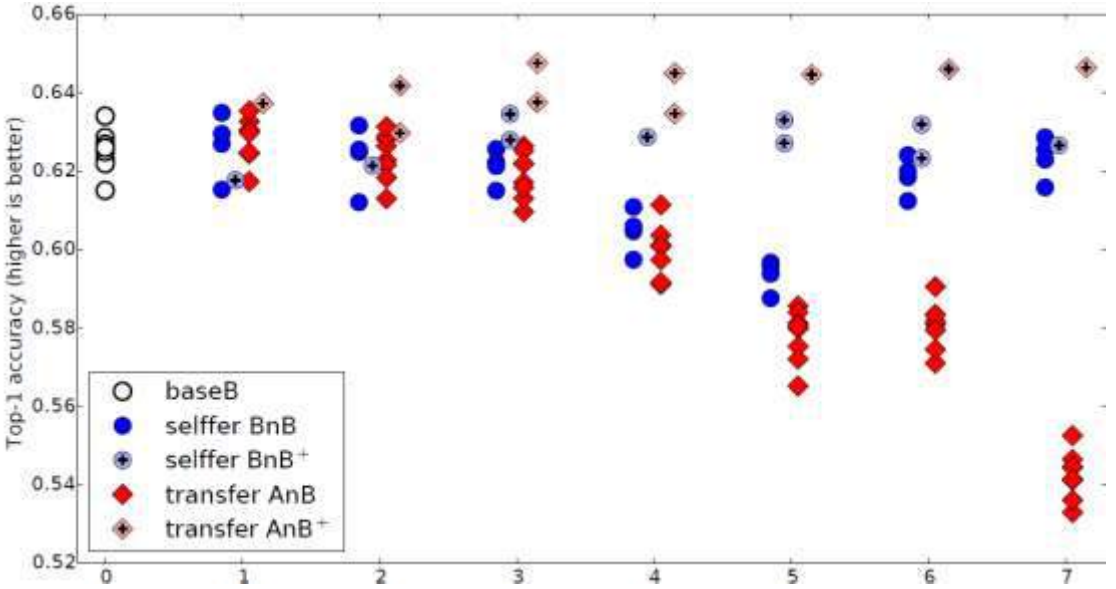
B3B: the first 3 layers are copied from baseB and frozen. The five higher layers are initialized randomly and trained on dataset B.
A3B: the first 3 layers are copied from baseA and frozen. The five higher layers are initialized randomly and trained on dataset B.
B3B+: just like B3B, but all layers learn.
A3B+: just like A3B, but all layers learn.

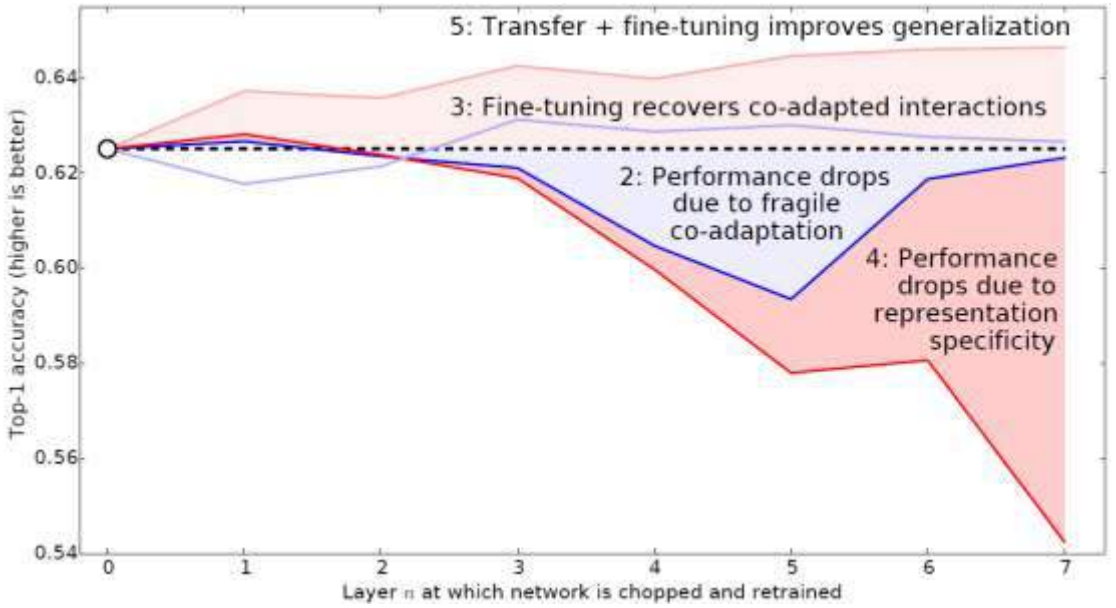
[3] J. Yosinski , J. Clune , Y. Bengio , H. Lipson , How transferable are features in deep neural networks? In NIPS, 2014, pp. 3320–3328 .

Supervised DA method: fine-tune

It showed how transferability is negatively affected by two distinct issues: optimization difficulties related to splitting networks in the middle of fragilely co-adapted layers and the specialization of higher layer features to the original task at the expense of performance on the target task.

		The size of target dataset		
		Low	**Medium**	**High**
The distance between domains	**Low**	Freeze	freeze or tune	Tune
	Medium	freeze or tune	Tune	Tune
	High	freeze or tune	Tune	Tune





Unsupervised DA

- **Traditional deep learning**

$$\min \frac{1}{n}\sum_{i=1}^{n} L(x_i, y_i, \theta)$$

- **Feature adaptation**

$$\min \frac{1}{n}\sum_{i=1}^{n} L(\phi(x_i^s), y_i^s, \theta)$$

- **Instance adaptation**

$$\min \frac{1}{n}\sum_{i=1}^{n} w_i L(x_i^s, y_i^s, \theta)$$

- **Model adaptation**

$$\min \frac{1}{n}\sum_{i=1}^{n} L(x_i^s, y_i^s, \theta)$$

- **KL divergence:**

$$KL(P,Q) = \sum_i P(i) log \frac{P(i)}{Q(i)}$$

- **MMD:**

$$MMD^2(\mathcal{D}_s, \mathcal{D}_t) = \left\| \frac{1}{M}\sum_{i=1}^{M} \phi(x_i^s) - \frac{1}{N}\sum_{i=1}^{N} \phi(x_i^t) \right\|_{\mathcal{H}}^2$$

- **$\mathcal{H}$-divergence:**

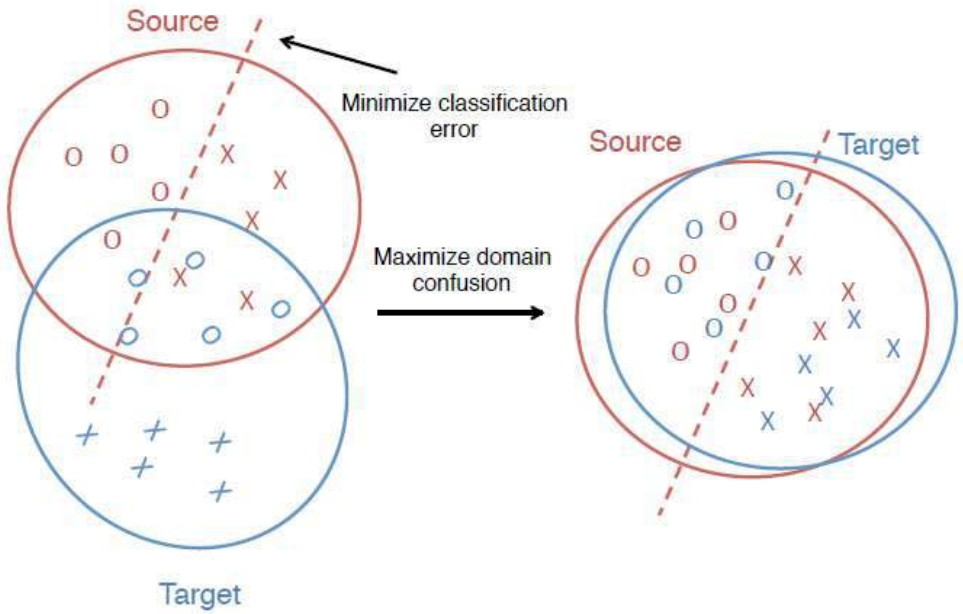
$$\widehat{d_A} = 2(1-2\epsilon)$$

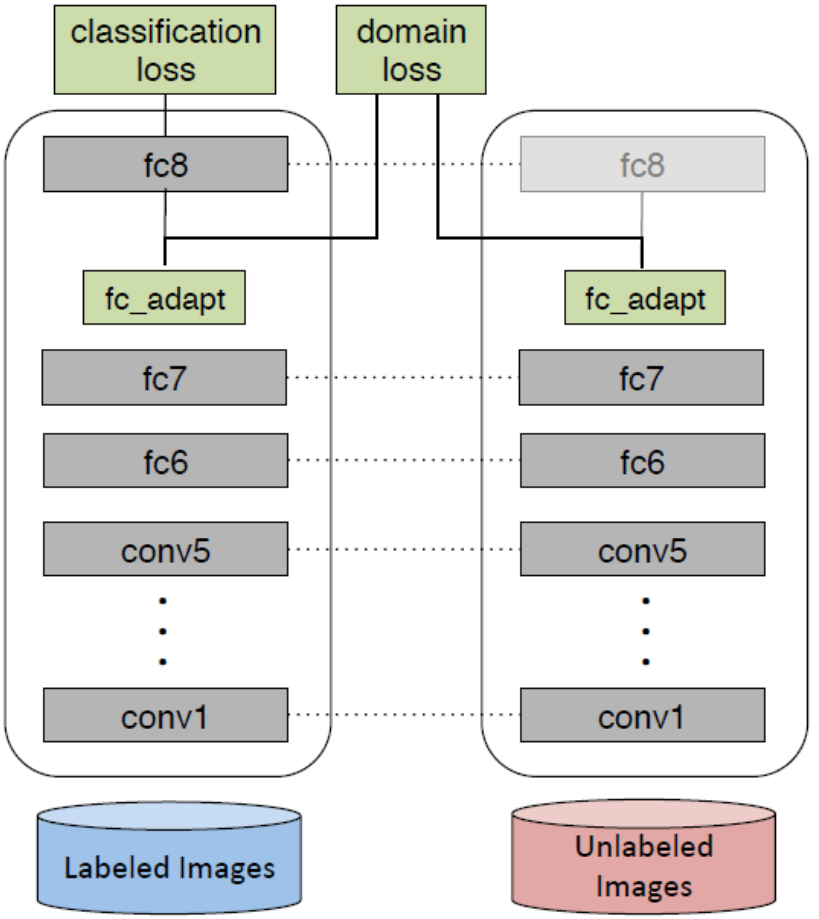
- **Wasserstein distance (Earth mover distance)**

$$W(P_1, P_2) = \inf_{\gamma \sim \prod(P_1, P_2)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|]$$

Unsupervised DA method

Most deep DA methods try to learn more transferable representations through mapping both domains into a domain-invariant feature space, and then directly apply the classifier learned from only source labels to target domain.





Some papers explore domain-invariant feature spaces by minimizing some measures of domain discrepancy such as statistic loss, adversarial loss and reconstruction loss

Unsupervised DA method: DDC (1.1)

Maximum mean discrepancy (MMD) is a commonly-used statistic loss for unsupervised DA. The hidden representations of images of different domain are embedded in a reproducing kernel Hilbert space, and the mean embeddings of distributions cross domains can be explicitly matched.

1. Given two distributions s and t, the MMD between them is defined as:

$$MMD^2(s,t) = \sup_{\|\phi\|_{\mathcal{H}} \leq 1} \left\| E_{x^s \sim s}[\phi(x^s)] - E_{x^t \sim t}[\phi(x^t)] \right\|_{\mathcal{H}}^2$$

2. Denote by $\mathcal{D}_s = \{x_i^s\}_{i=1}^M$ and $\mathcal{D}_t = \{x_i^t\}_{i=1}^N$ drawn from the distributions s and t, respectively, an empirical estimate of MMD is given as:

$$MMD^2(\mathcal{D}_s, \mathcal{D}_t) = \left\| \frac{1}{M} \sum_{i=1}^{M} \phi(x_i^s) - \frac{1}{N} \sum_{i=1}^{N} \phi(x_i^t) \right\|_{\mathcal{H}}^2$$

3. The main idea of DDC and DAN is to integrate this MMD estimator:

$$\mathcal{L} = \mathcal{L}_C(X_s, y) + \lambda \sum_{l \in \mathcal{L}} L_M(D_s^l, D_t^l)$$

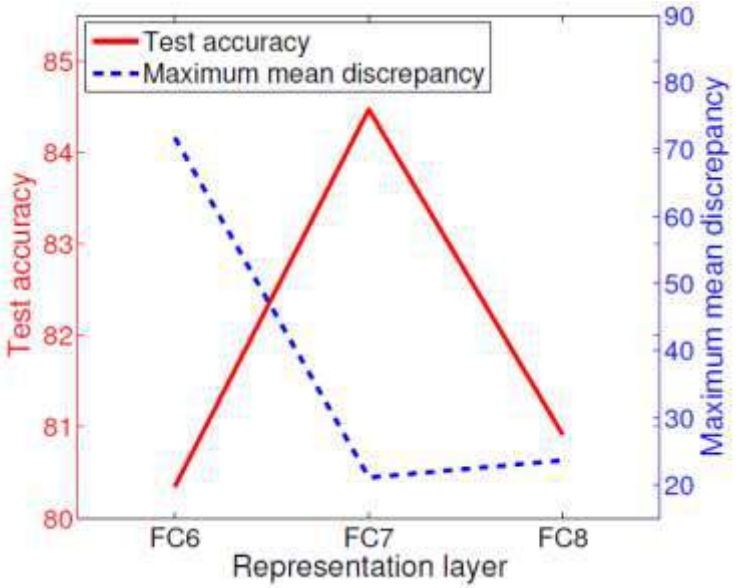
[4] Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. Deep domain confusion: Maximizing for domain invariance. CoRR, abs/1412.3474, 2014.

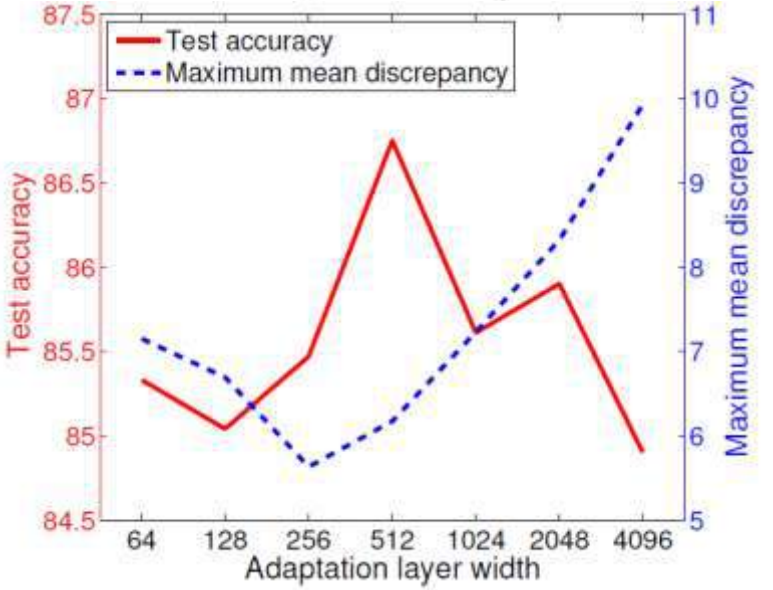
Unsupervised DA method: DDC (1.1)

A

D

W





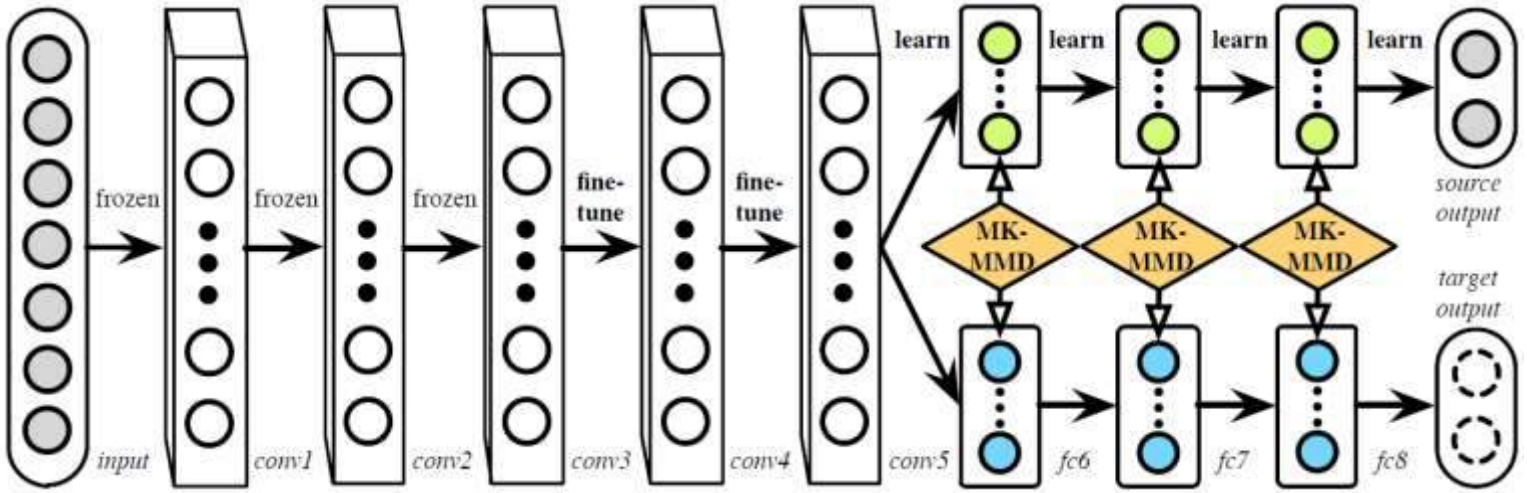
	$A \to W$	$D \to W$	$W \to D$	Average
GFK(PLS,PCA) [16]	15.0 ± 0.4	44.6 ± 0.3	49.7 ± 0.5	36.4
SA [13]	15.3	50.1	56.9	40.8
DA-NBNN [31]	23.3 ± 2.7	67.2 ± 1.9	67.4 ± 3.0	52.6
DLID [8]	26.1	68.9	84.9	60.0
$DeCAF_6$ S [11]	52.2 ± 1.7	91.5 ± 1.5	–	–
DaNN [14]	35.0 ± 0.2	70.5 ± 0.0	74.3 ± 0.0	59.9
Ours	$\mathbf{59.4 \pm 0.8}$	$\mathbf{92.5 \pm 0.3}$	$\mathbf{91.7 \pm 0.8}$	**81.2**

Unsupervised DA method: DAN (1.2)

DAN that matches the shift in marginal distributions across domains by adding multiple adaptation layers and exploring multiple kernels, assuming that the conditional distributions remain unchanged.

Adapting a single layer cannot undo the dataset bias between the source and the target, since there are other layers that are not transferable. The multiple kernels with different bandwidths can match both the low-order moments and high-order moments to minimize the domain discrepancy.

$$\mathcal{K} = \left\{ k = \sum_{u=1}^{m} \beta_u k_u : \sum_{u=1}^{m} \beta_u = 1, \beta_u \geq 0, \forall u \right\}$$

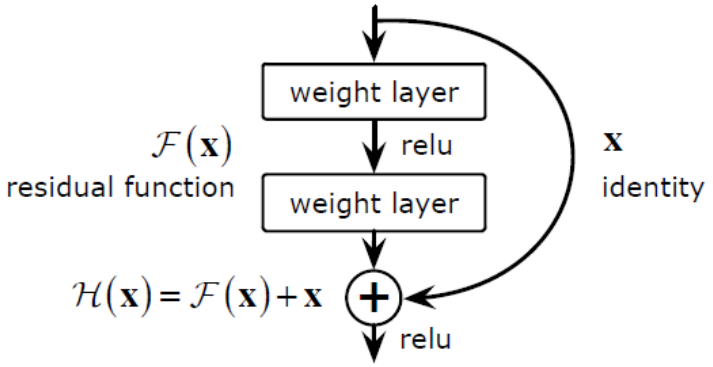


[5] M. Long , Y. Cao , J. Wang , M. Jordan , Learning transferable features with deep adaptation networks, in ICML, 2015, pp. 97–105 .

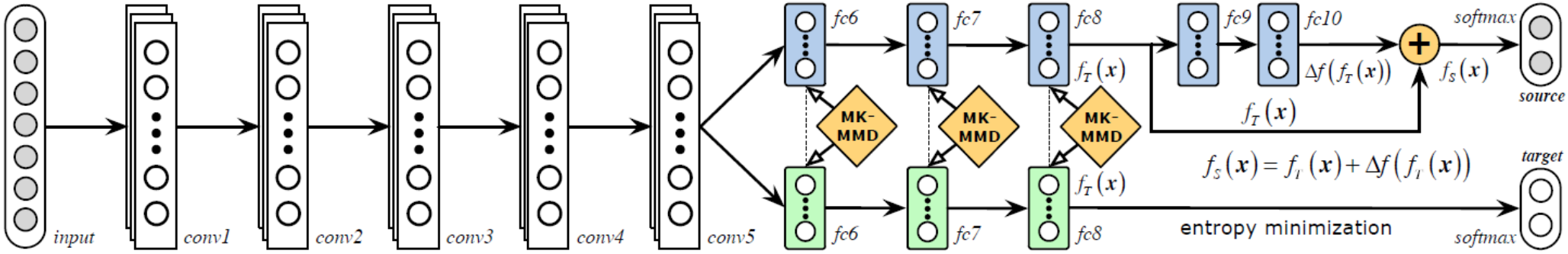
Unsupervised DA method: RTN (1.3)

$$\min \frac{1}{n}\sum_{i=1}^{n} L(x_i^s, y_i^s, \theta)$$

Since deep features eventually transition from general to specific along the network: (1) fully connected layers fc6–fc8 are tailored to model task-specific structures, hence they are not safely transferable and should be adapted with MK-MMD minimization; (2) supervised classifiers are not safely transferable, hence they are bridged by the residual layers fc9–fc10 such that $f_S(x) = f_T(x) + \Delta f(f_T(x))$.



Instead of using the residual block to model the feature mapping, we use it to bridge the source classifier $f_S(x)$ and target classifier $f_T(x)$ by $x \triangleq f_T(x)$, $\mathcal{H}(x) \triangleq f_S(x)$, and $\mathcal{F}(x) \triangleq \Delta f(f_T(x))$



[6] M. Long , H. Zhu , J. Wang , M.I. Jordan , Unsupervised domain adaptation with residual transfer networks, in NIPS, 2016, pp. 136–144 .

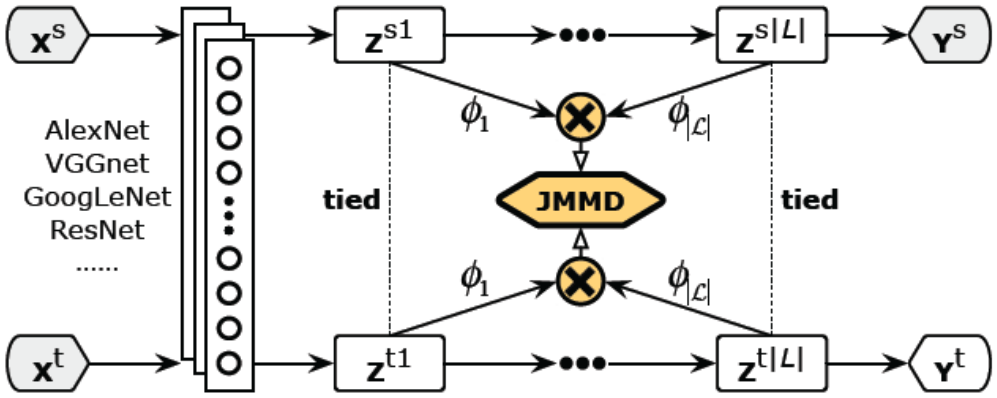
Unsupervised DA method: JAN (1.4)

Transfer learning will become more challenging as domains may change by the joint distributions P(X,Y) of input features X and output labels Y. The distribution shifts may stem from the marginal distributions P(X), the conditional distributions P(Y|X), or both.

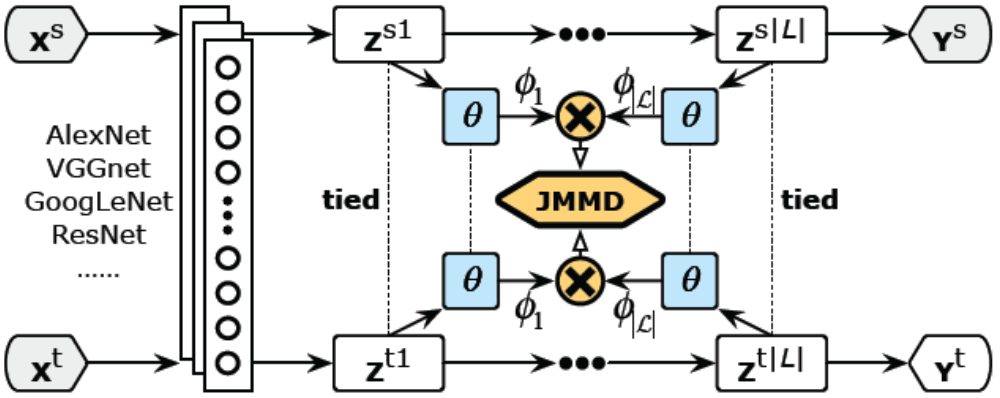
Kernel embeddings can be readily generalized to joint distributions of two or more variables using tensor product feature spaces.

$$\mathcal{C}_{\mathbf{X}^{1:m}}(P) \triangleq \mathbb{E}_{\mathbf{X}^{1:m}}\left[\otimes_{\ell=1}^{m}\phi^{\ell}\left(\mathbf{X}^{\ell}\right)\right] = \int_{\times_{\ell=1}^{m}\Omega^{\ell}}\left(\otimes_{\ell=1}^{m}\phi^{\ell}\left(\mathbf{x}^{\ell}\right)\right)\mathrm{d}P\left(\mathbf{x}^{1},\ldots,\mathbf{x}^{m}\right),$$

$$D_{\mathcal{L}}(P,Q) \triangleq \left\|\mathcal{C}_{\mathbf{Z}^{s,1:|\mathcal{L}|}}(P) - \mathcal{C}_{\mathbf{Z}^{t,1:|\mathcal{L}|}}(Q)\right\|^{2}_{\otimes_{\ell=1}^{|\mathcal{L}|}\mathcal{H}^{\ell}}$$



(a) Joint Adaptation Network (JAN)



(b) Adversarial Joint Adaptation Network (JAN-A)

[7] M. Long, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. arXiv preprint arXiv:1605.06636, 2016.

Unsupervised DA method: JAN (1.4)

Table 1. Classification accuracy (%) on *Office-31* dataset for unsupervised domain adaptation (AlexNet and ResNet)

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
AlexNet (Krizhevsky et al., 2012)	61.6±0.5	95.4±0.3	99.0±0.2	63.8±0.5	51.1±0.6	49.8±0.4	70.1
TCA (Pan et al., 2011)	61.0±0.0	93.2±0.0	95.2±0.0	60.8±0.0	51.6±0.0	50.9±0.0	68.8
GFK (Gong et al., 2012)	60.4±0.0	95.6±0.0	95.0±0.0	60.6±0.0	52.4±0.0	48.1±0.0	68.7
DDC (Tzeng et al., 2014)	61.8±0.4	95.0±0.5	98.5±0.4	64.4±0.3	52.1±0.6	52.2±0.4	70.6
DAN (Long et al., 2015)	68.5±0.5	96.0±0.3	99.0±0.3	67.0±0.4	54.0±0.5	53.1±0.5	72.9
RTN (Long et al., 2016)	73.3±0.3	**96.8**±0.2	**99.6**±0.1	71.0±0.2	50.5±0.3	51.0±0.1	73.7
RevGrad (Ganin & Lempitsky, 2015)	73.0±0.5	96.4±0.3	99.2±0.3	72.3±0.3	53.4±0.4	51.2±0.5	74.3
JAN (ours)	74.9±0.3	96.6±0.2	99.5±0.2	71.8±0.2	**58.3**±0.3	55.0±0.4	76.0
JAN-A (ours)	**75.2**±0.4	96.6±0.2	**99.6**±0.1	**72.8**±0.3	57.5±0.2	**56.3**±0.2	**76.3**
ResNet (He et al., 2016)	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
TCA (Pan et al., 2011)	72.7±0.0	96.7±0.0	99.6±0.0	74.1±0.0	61.7±0.0	60.9±0.0	77.6
GFK (Gong et al., 2012)	72.8±0.0	95.0±0.0	98.2±0.0	74.5±0.0	63.4±0.0	61.0±0.0	77.5
DDC (Tzeng et al., 2014)	75.6±0.2	96.0±0.2	98.2±0.1	76.5±0.3	62.2±0.4	61.5±0.5	78.3
DAN (Long et al., 2015)	80.5±0.4	97.1±0.2	99.6±0.1	78.6±0.2	63.6±0.3	62.8±0.2	80.4
RTN (Long et al., 2016)	84.5±0.2	96.8±0.1	99.4±0.1	77.5±0.3	66.2±0.2	64.8±0.3	81.6
RevGrad (Ganin & Lempitsky, 2015)	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
JAN (ours)	85.4±0.3	**97.4**±0.2	**99.8**±0.2	84.7±0.3	68.6±0.3	70.0±0.4	84.3
JAN-A (ours)	**86.0**±0.4	96.7±0.3	99.7±0.1	**85.1**±0.4	**69.2**±0.4	**70.7**±0.5	**84.6**

Unsupervised domain adaptation

- **Traditional deep learning**

$$\min \frac{1}{n}\sum_{i=1}^{n} L(x_i, y_i, \theta)$$

- **Feature adaptation**

$$\min \frac{1}{n}\sum_{i=1}^{n} L(\phi(x_i^s), y_i^s, \theta)$$

- **Instance adaptation**

$$\min \frac{1}{n}\sum_{i=1}^{n} w_i L(x_i^s, y_i^s, \theta)$$

- **Model adaptation**

$$\min \frac{1}{n}\sum_{i=1}^{n} L(x_i^s, y_i^s, \theta)$$

- **KL divergence:**

$$KL(P,Q) = \sum_i P(i) log \frac{P(i)}{Q(i)}$$

- **MMD:**

$$MMD^2(\mathcal{D}_s, \mathcal{D}_t) = \left\| \frac{1}{M}\sum_{i=1}^{M} \phi(x_i^s) - \frac{1}{N}\sum_{i=1}^{N} \phi(x_i^t) \right\|_{\mathcal{H}}^2$$

- **$\mathcal{H}$-divergence:**

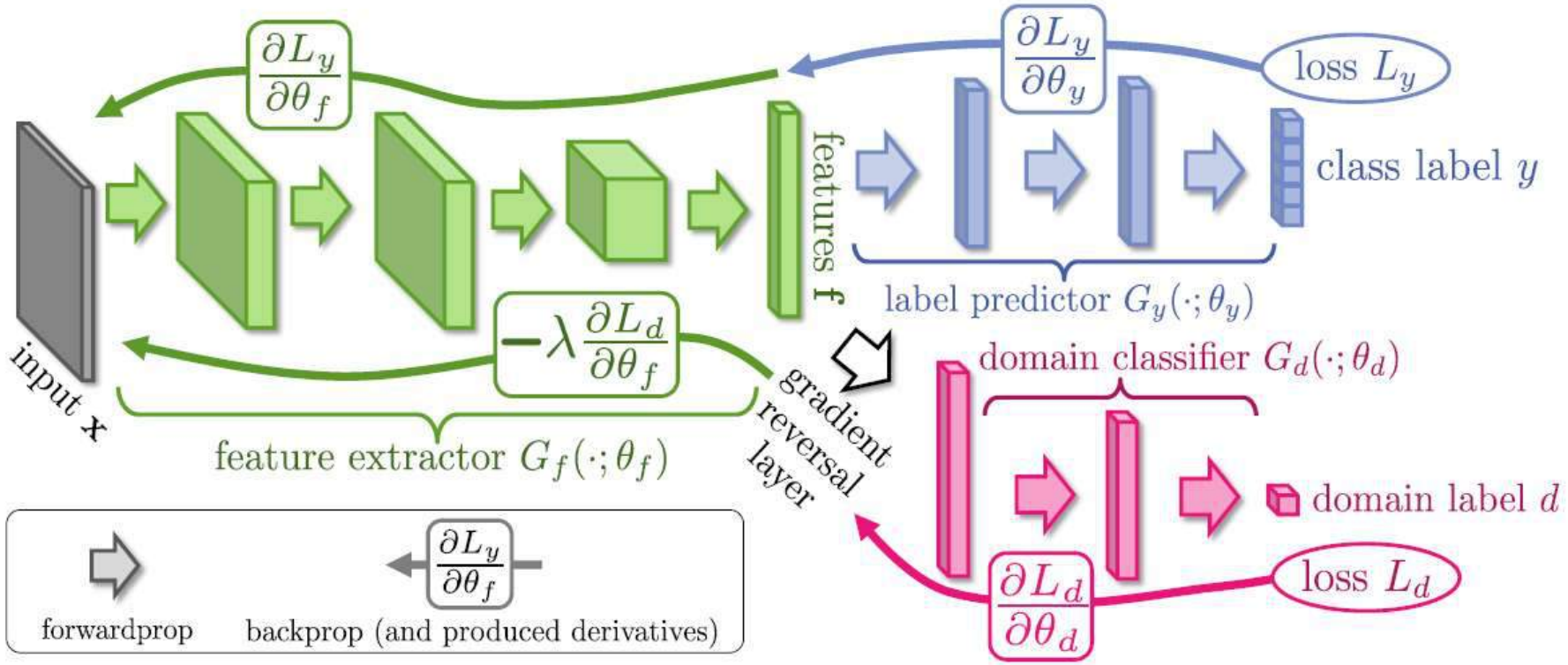
$$\widehat{d_A} = 2(1-2\epsilon)$$

- **Wasserstein distance (Earth mover distance)**

$$W(P_1, P_2) = \inf_{\gamma \sim \prod(P_1, P_2)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|]$$

Unsupervised DA method: RevGrad (2.1)

The domain-adversarial neural network (DANN) integrated a gradient reversal layer (GRL) to train a feature extractor by maximizing the domain classifier loss and simultaneously minimizing the label predictor loss.



[8] Y. Ganin , V. Lempitsky , Unsupervised domain adaptation by backpropagation, in ICML, 2015, pp. 1180–1189 .

Unsupervised DA method: RevGrad (2.1)

1 Max-min optimization

$$E(\theta_f, \theta_y, \theta_d) = \sum_{\substack{i=1..N \\ d_i=0}} L_y\left(G_y(G_f(\mathbf{x}_i;\theta_f);\theta_y), y_i\right) - \lambda \sum_{i=1..N} L_d\left(G_d(G_f(\mathbf{x}_i;\theta_f);\theta_d), y_i\right) = \sum_{\substack{i=1..N \\ d_i=0}} L_y^i(\theta_f, \theta_y) - \lambda \sum_{i=1..N} L_d^i(\theta_f, \theta_d) \quad (1)$$

$$(\hat{\theta}_f, \hat{\theta}_y) = \arg\min_{\theta_f, \theta_y} E(\theta_f, \theta_y, \hat{\theta}_d) \quad (2)$$

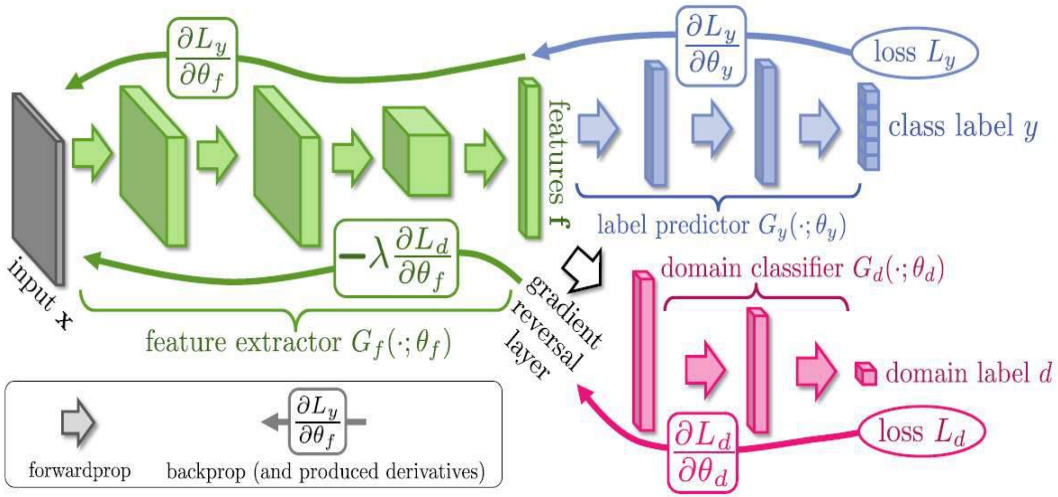
$$\hat{\theta}_d = \arg\max_{\theta_d} E(\hat{\theta}_f, \hat{\theta}_y, \theta_d)\,. \quad (3)$$

2 Gradient reversal layer

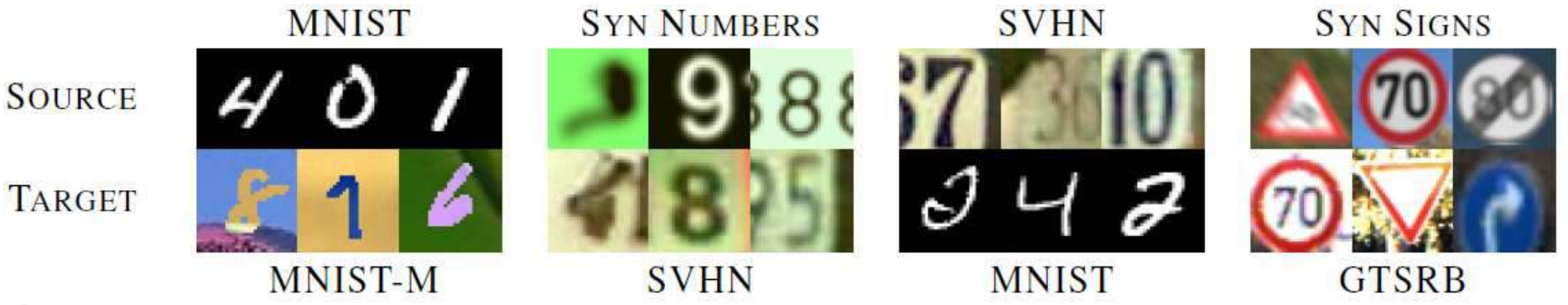
$$\tilde{E}(\theta_f, \theta_y, \theta_d) = \sum_{\substack{i=1..N \\ d_i=0}} L_y\left(G_y(G_f(\mathbf{x}_i;\theta_f);\theta_y), y_i\right) + \sum_{i=1..N} L_d\left(G_d(R_\lambda(G_f(\mathbf{x}_i;\theta_f));\theta_d), y_i\right) \quad (9)$$

$$R_\lambda(\mathbf{x}) = \mathbf{x}$$

$$\frac{dR_\lambda}{d\mathbf{x}} = -\lambda\mathbf{I}$$

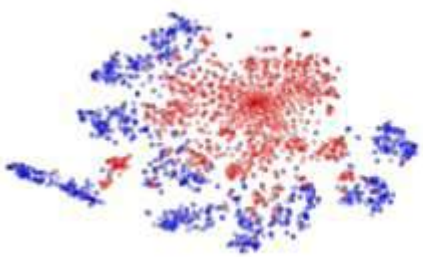


Unsupervised DA method: RevGrad (2.1)

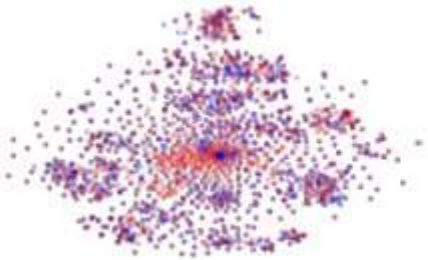


METHOD	SOURCE / TARGET	MNIST / MNIST-M	SYN NUMBERS / SVHN	SVHN / MNIST	SYN SIGNS / GTSRB
SOURCE ONLY		.5749	.8665	.5919	.7400
SA (FERNANDO ET AL., 2013)		.6078 (7.9%)	.8672 (1.3%)	.6157 (5.9%)	.7635 (9.1%)
PROPOSED APPROACH		**.8149** (57.9%)	**.9048** (66.1%)	**.7107** (29.3%)	**.8866** (56.7%)
TRAIN ON TARGET		.9891	.9244	.9951	.9987

MNIST → MNIST-M: top feature extractor layer

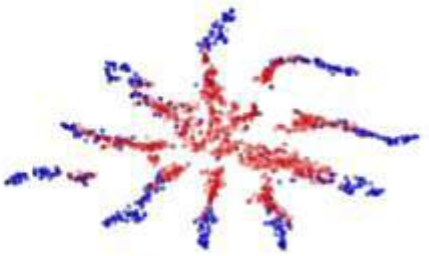


(a) Non-adapted

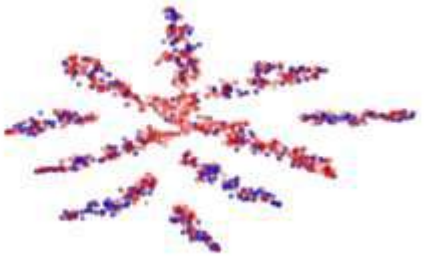


(b) Adapted

SYN NUMBERS → SVHN: last hidden layer of the label predictor



(a) Non-adapted

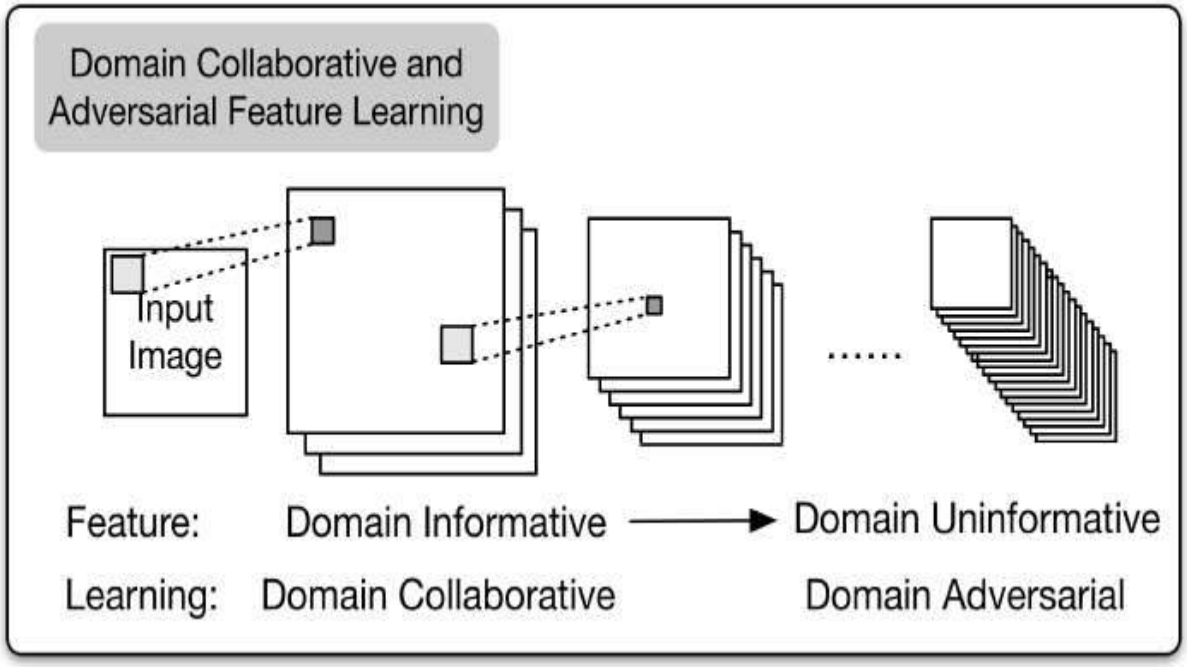


(b) Adapted

Unsupervised DA method: iCAN (2.2)

Collaborative and Adversarial Network (CAN) through domain-collaborative and domain adversarial training of neural networks.

- **Domain informative representations: from lower blocks through collaborative learning**
- **Domain uninformative representations: from higher blocks through adversarial learning.**



1

$$\min_{\boldsymbol{\theta},\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_D\left(D(F(\mathbf{x}_i;\boldsymbol{\theta});\mathbf{w}),d_i\right)$$

2

$$\max_{\boldsymbol{\theta}}\min_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_D\left(D(F(\mathbf{x}_i;\boldsymbol{\theta});\mathbf{w}),d_i\right)$$

3

$$\min_{\Theta_F,\boldsymbol{\lambda}} \mathcal{L}_{CAN} = \sum_{k=1}^{m-1}\lambda_k \min_{\mathbf{w}_k}\mathcal{L}_D(\boldsymbol{\theta}_k,\mathbf{w}_k) + \lambda_m \min_{\mathbf{w}_m}\mathcal{L}_D(\boldsymbol{\theta}_m,\mathbf{w}_m),$$

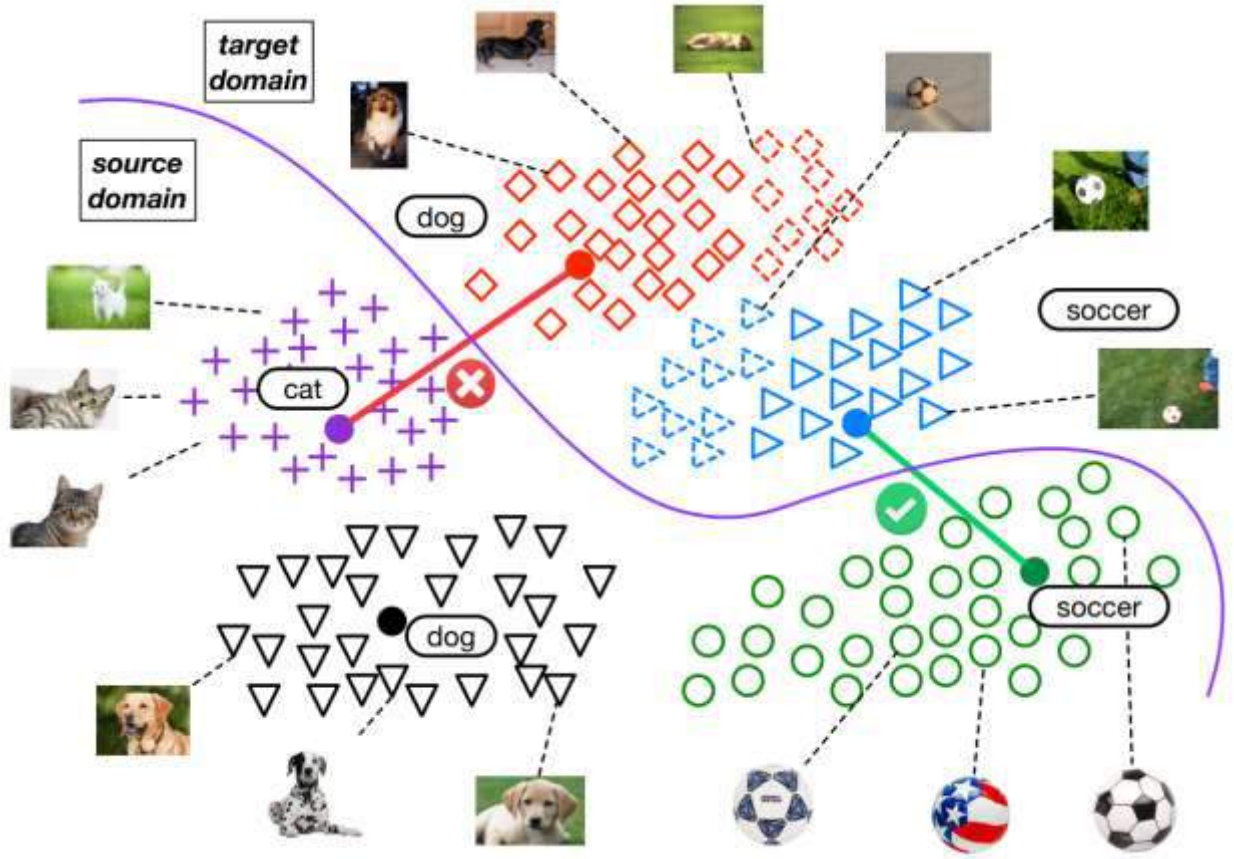
$$\text{s.t.} \quad \sum_{k=1}^{m-1}\lambda_k = \lambda_0, \quad |\lambda_k| \le \lambda_0,$$

[9] W. Zhang, W. Ouyang, W. Li, and D. Xu. Collaborative and adversarial network for unsupervised domain adaptation. In CVPR, pages 3801–3809, 2018.

Unsupervised DA method: MADA (2.3)

The difficulty of domain adaptation:

- **target label space is only a subspace of the source label space.**
- **discriminative structures may be mixed up or falsely aligned across domains.**

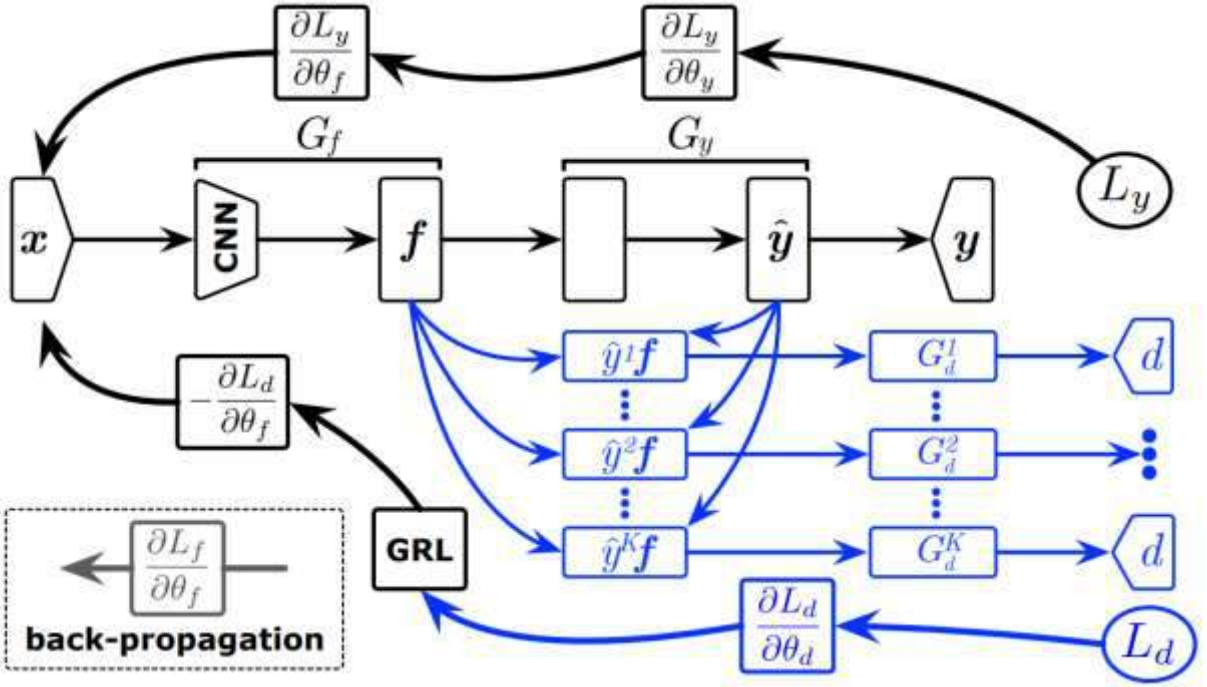


[10] Z. Pei, Z. Cao, M. Long, and J. Wang. Multi-adversarial domain adaptation. In AAAI Conference on Artificial Intelligence, 2018

Unsupervised DA method: MADA (2.3)

$$\min \frac{1}{n}\sum_{i=1}^{n} w_i L(x_i^s, y_i^s, \theta)$$

A multi-adversarial domain adaptation (MADA) approach captures multimode structures enable fine-grained alignment of different data distributions based on multiple domain discriminators.



it is a natural idea to use $\widehat{y_i}$ as the probability to indicate how much each data point x_i should be attended to the K domain discriminators ${G_d}^k$. The attention of each point x_i to a domain discriminator ${G_d}^k$ can be modeled by weighting its features $G_f(x_i)$ with probability $\widehat{y_{ik}}$.

$$C\left(\theta_f, \theta_y, \theta_d^k|_{k=1}^K\right) = \frac{1}{n_s}\sum_{\mathbf{x}_i \in \mathcal{D}_s} L_y\left(G_y\left(G_f\left(\mathbf{x}_i\right)\right), y_i\right) - \frac{\lambda}{n}\sum_{k=1}^{K}\sum_{\mathbf{x}_i \in \mathcal{D}} L_d^k\left(G_d^k\left(\hat{y}_i^k G_f\left(\mathbf{x}_i\right)\right), d_i\right),$$

$$\left(\hat{\theta}_f, \hat{\theta}_y\right) = \arg\min_{\theta_f, \theta_y} C\left(\theta_f, \theta_y, \theta_d^k|_{k=1}^K\right),$$

$$\left(\hat{\theta}_d^1, \ldots, \hat{\theta}_d^K\right) = \arg\max_{\theta_d^1, \ldots, \theta_d^K} C\left(\theta_f, \theta_y, \theta_d^k|_{k=1}^K\right).$$

Unsupervised DA method: MADA (2.3)

Table 1: Accuracy (%) on *Office-31* for unsupervised domain adaptation (AlexNet and ResNet)

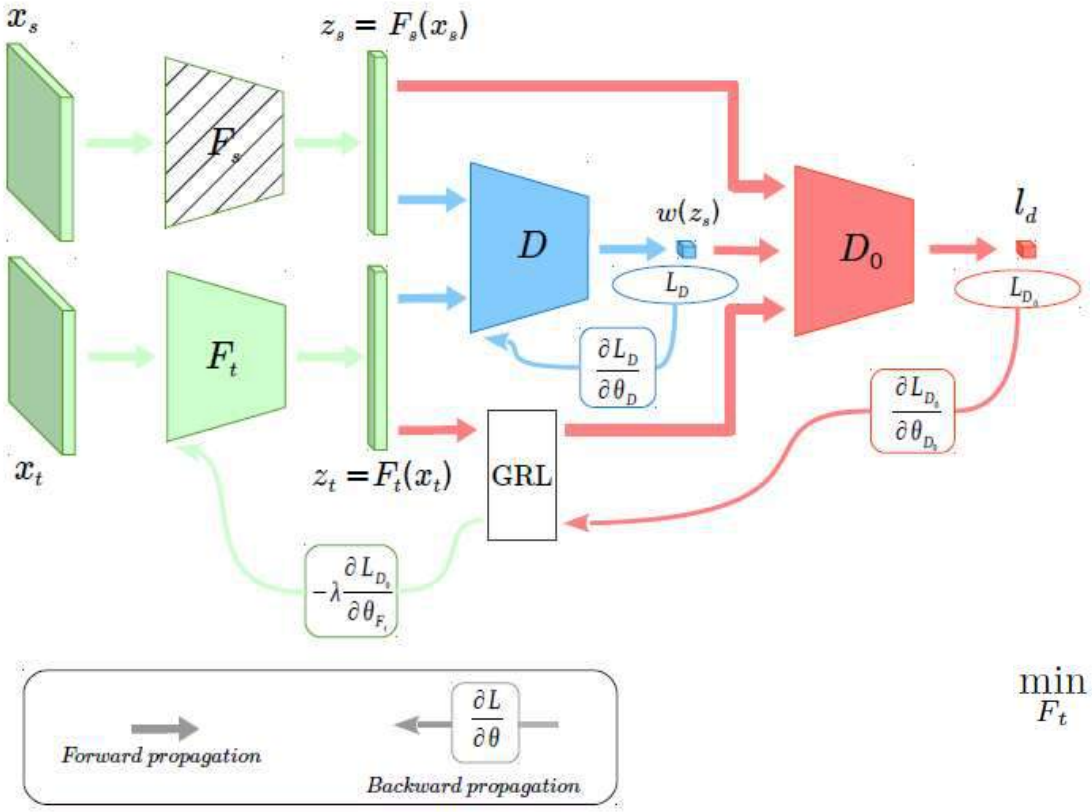
Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
AlexNet (Krizhevsky, Sutskever, and Hinton 2012)	60.6±0.4	95.4±0.2	99.0±0.1	64.2±0.3	45.5±0.5	48.3±0.5	68.8
TCA (Pan et al. 2011)	59.0±0.0	90.2±0.0	88.2±0.0	57.8±0.0	51.6±0.0	47.9±0.0	65.8
GFK (Gong et al. 2012)	58.4±0.0	93.6±0.0	91.0±0.0	58.6±0.0	52.4±0.0	46.1±0.0	66.7
DDC (Tzeng et al. 2014)	61.0±0.5	95.0±0.3	98.5±0.3	64.9±0.4	47.2±0.5	49.4±0.4	69.3
DAN (Long et al. 2015)	68.5±0.3	96.0±0.1	99.0±0.1	66.8±0.2	50.0±0.4	49.8±0.3	71.7
RTN (Long et al. 2016)	73.3±0.2	96.8±0.2	99.6±0.1	71.0±0.2	50.5±0.3	51.0±0.1	73.7
RevGrad (Ganin and Lempitsky 2015)	73.0±0.5	96.4±0.3	99.2±0.3	72.3±0.3	52.4±0.4	50.4±0.5	74.1
MADA	**78.5**±0.2	**99.8**±0.1	**100.0**±.0	**74.1**±0.1	**56.0**±0.2	**54.5**±0.3	**77.1**
ResNet (He et al. 2016)	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
TCA (Pan et al. 2011)	74.7±0.0	96.7±0.0	99.6±0.0	76.1±0.0	63.7±0.0	62.9±0.0	79.3
GFK (Gong et al. 2012)	74.8±0.0	95.0±0.0	98.2±0.0	76.5±0.0	65.4±0.0	63.0±0.0	78.8
DDC (Tzeng et al. 2014)	75.8±0.2	95.0±0.2	98.2±0.1	77.5±0.3	67.4±0.4	64.0±0.5	79.7
DAN (Long et al. 2015)	83.8±0.4	96.8±0.2	99.5±0.1	78.4±0.2	66.7±0.3	62.7±0.2	81.3
RTN (Long et al. 2016)	84.5±0.2	96.8±0.1	99.4±0.1	77.5±0.3	66.2±0.2	64.8±0.3	81.6
RevGrad (Ganin and Lempitsky 2015)	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	**67.4**±0.5	82.2
MADA	**90.0**±0.1	**97.4**±0.1	**99.6**±0.1	**87.8**±0.2	**70.3**±0.3	66.4±0.3	**85.2**

Table 3: Accuracy (%) on *Office-31* for domain adaptation from 31 classes to 25 classes (AlexNet)

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
AlexNet (Krizhevsky, Sutskever, and Hinton 2012)	58.2±0.4	95.9±0.2	99.0±0.1	60.4±0.3	49.8±0.5	47.3±0.5	68.4
RevGrad (Ganin and Lempitsky 2015)	65.1±0.5	91.7±0.3	97.1±0.3	60.6±0.3	42.1±0.4	42.9±0.5	66.6
MADA	**70.8**±0.2	**96.6**±0.1	**99.5**±.0	**69.6**±0.1	**51.4**±0.2	**54.2**±0.3	**73.7**

Unsupervised DA method: Weighted Adversarial Nets (2.4)

The importance weighted adversarial nets-based partial domain adaptation method can identify the source samples that are potentially from the outlier classes and, at the same time, reduce the shift of shared classes between domains.



The intuition of the weighting scheme is that if the activation of the first domain classifier is large, the sample can be almost perfectly discriminated from the target domain by the discriminator.

1

$$D(\mathbf{z}) = p(y = 1|\mathbf{z}) = \sigma(a(\mathbf{z}))$$

$$\tilde{w}(\mathbf{z}) = 1 - D^*(\mathbf{z}) = \frac{1}{\frac{p_s(\mathbf{z})}{p_t(\mathbf{z})} + 1}$$

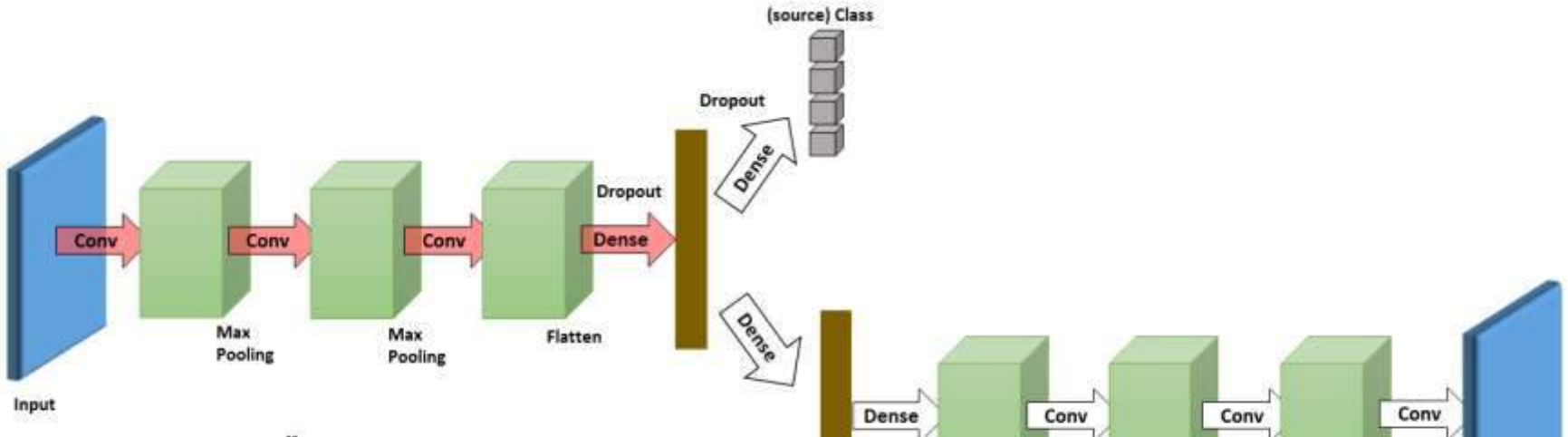
2

$$\min_{F_t} \max_{D_0} \mathcal{L}_w(D_0, F_s, F_t) = \mathbb{E}_{\mathbf{x} \sim p_s(\mathbf{x})}[w(\mathbf{z}) \log D_0(F_s(\mathbf{x}))] + \mathbb{E}_{\mathbf{x} \sim p_t(\mathbf{x})}[\log(1 - D_0(F_t(\mathbf{x})))]$$

[11] Zhang J, Ding Z, Li W, et al. Importance Weighted Adversarial Nets for Partial Domain Adaptation. In CVPR. 2018: 8156-8164.

Unsupervised DA method: DRCN (3.1)

The model is optimized through multitask learning, that is, jointly learns the (supervised) source label prediction and the (unsupervised) target data reconstruction tasks. The aim is that the encoding shared representation should learn the commonality between those tasks that provides useful information for cross-domain object recognition



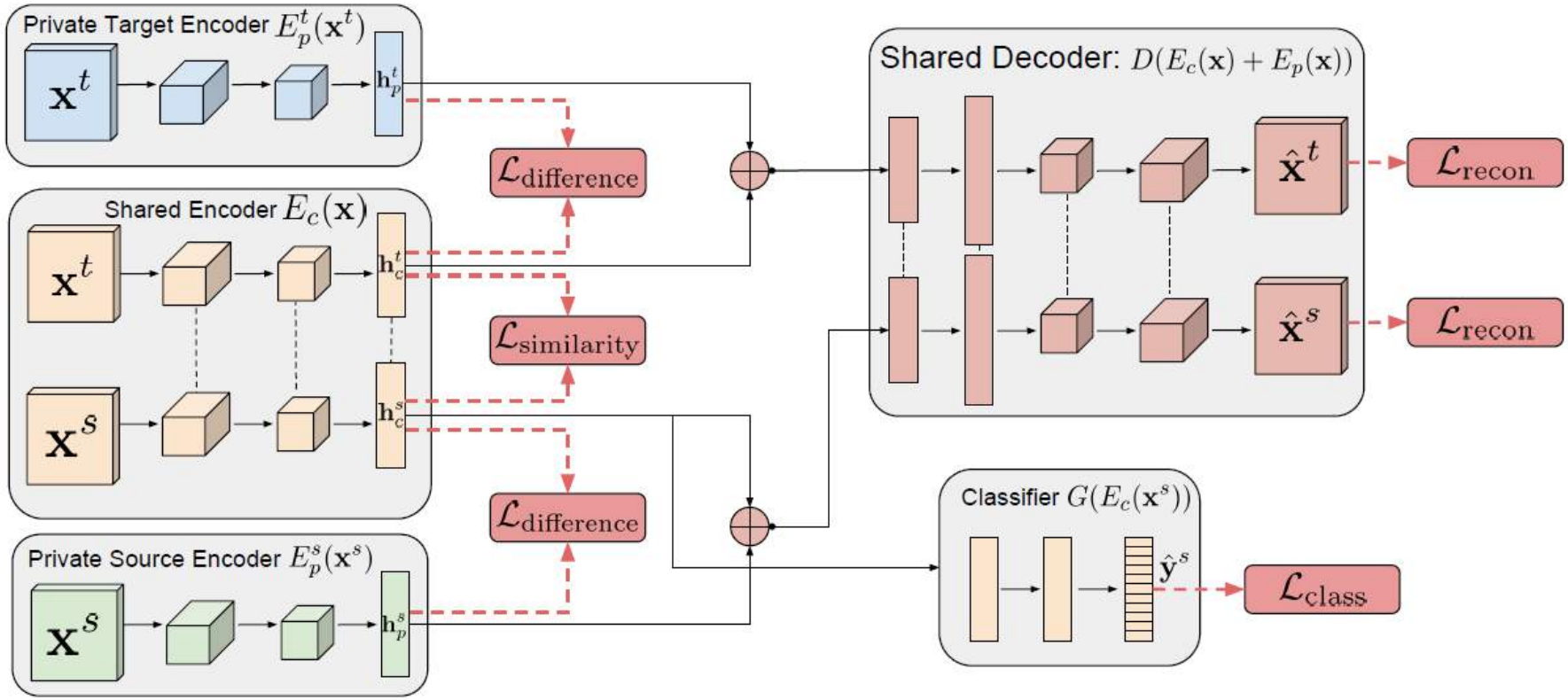
$$\mathcal{L}_c^{n_s}(\{\Theta_{\text{enc}}, \Theta_{\text{lab}}\}) := \sum_{i=1}^{n_s} \ell_c \left(f_c(\mathbf{x}_i^s; \{\Theta_{\text{enc}}, \Theta_{\text{lab}}\}), \mathbf{y}_i^s \right),$$

$$\mathcal{L}_r^{n_t}(\{\Theta_{\text{enc}}, \Theta_{\text{dec}}\}) := \sum_{j=1}^{n_t} \ell_r \left(f_r(\mathbf{x}_j^t; \{\Theta_{\text{enc}}, \Theta_{\text{dec}}\}), \mathbf{x}_j^t \right)$$

[12] M. Ghifary , W.B. Kleijn , M. Zhang , D. Balduzzi , W. Li , Deep reconstruction–classification networks for unsupervised domain adaptation, in ECCV, Springer, 2016, pp. 597–613

Unsupervised DA method: DSN (3.2)

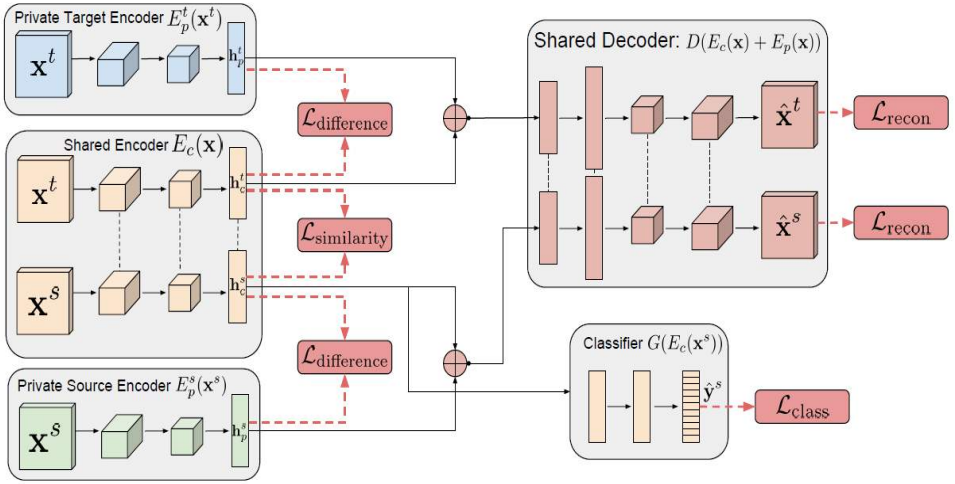
DSNs explicitly and jointly model both private and shared components of the domain representations. The private component of the representation is specific to a single domain and the shared component of the representation is shared by both domains.



[13] K. Bousmalis , G. Trigeorgis , N. Silberman , D. Krishnan , D. Erhan , Domain separation networks, in NIPS, 2016, pp. 343–351 .

Unsupervised DA method: DSN (3.2)

DSNs explicitly and jointly model both private and shared components of the domain representations. The private component of the representation is specific to a single domain and the shared component of the representation is shared by both domains.



1

$$\mathcal{L}_{\text{task}} = -\sum_{i=0}^{N_s} \mathbf{y}_i^s \cdot \log \hat{\mathbf{y}}_i^s,$$

2

$$\mathcal{L}_{\text{recon}} = \sum_{i=1}^{N_s} \mathcal{L}_{\text{si\_mse}}(\mathbf{x}_i^s, \hat{\mathbf{x}}_i^s) + \sum_{i=1}^{N_t} \mathcal{L}_{\text{si\_mse}}(\mathbf{x}_i^t, \hat{\mathbf{x}}_i^t)$$

$$\mathcal{L}_{\text{si\_mse}}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{k}\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 - \frac{1}{k^2}([\mathbf{x} - \hat{\mathbf{x}}] \cdot \mathbf{1}_k)^2,$$

3

$$L_{\text{difference}} = \left\|\mathbf{H}_c^{s\top}\mathbf{H}_p^s\right\|_F^2 + \left\|\mathbf{H}_c^{t\top}\mathbf{H}_p^t\right\|_F^2$$

4

$$\mathcal{L}_{\text{similarity}}^{\text{DANN}} = \sum_{i=0}^{N_s+N_t} \left\{ d_i \log \hat{d}_i + (1 - d_i) \log(1 - \hat{d}_i) \right\}$$

$$\mathcal{L}_{\text{similarity}}^{\text{MMD}} = \frac{1}{(N^s)^2} \sum_{i,j=0}^{N^s} \kappa(\mathbf{h}_{ci}^s, \mathbf{h}_{cj}^s) - \frac{2}{N^s N^t} \sum_{i,j=0}^{N^s, N^t} \kappa(\mathbf{h}_{ci}^s, \mathbf{h}_{cj}^t) + \frac{1}{(N^t)^2} \sum_{i,j=0}^{N^t} \kappa(\mathbf{h}_{ci}^t, \mathbf{h}_{cj}^t)$$

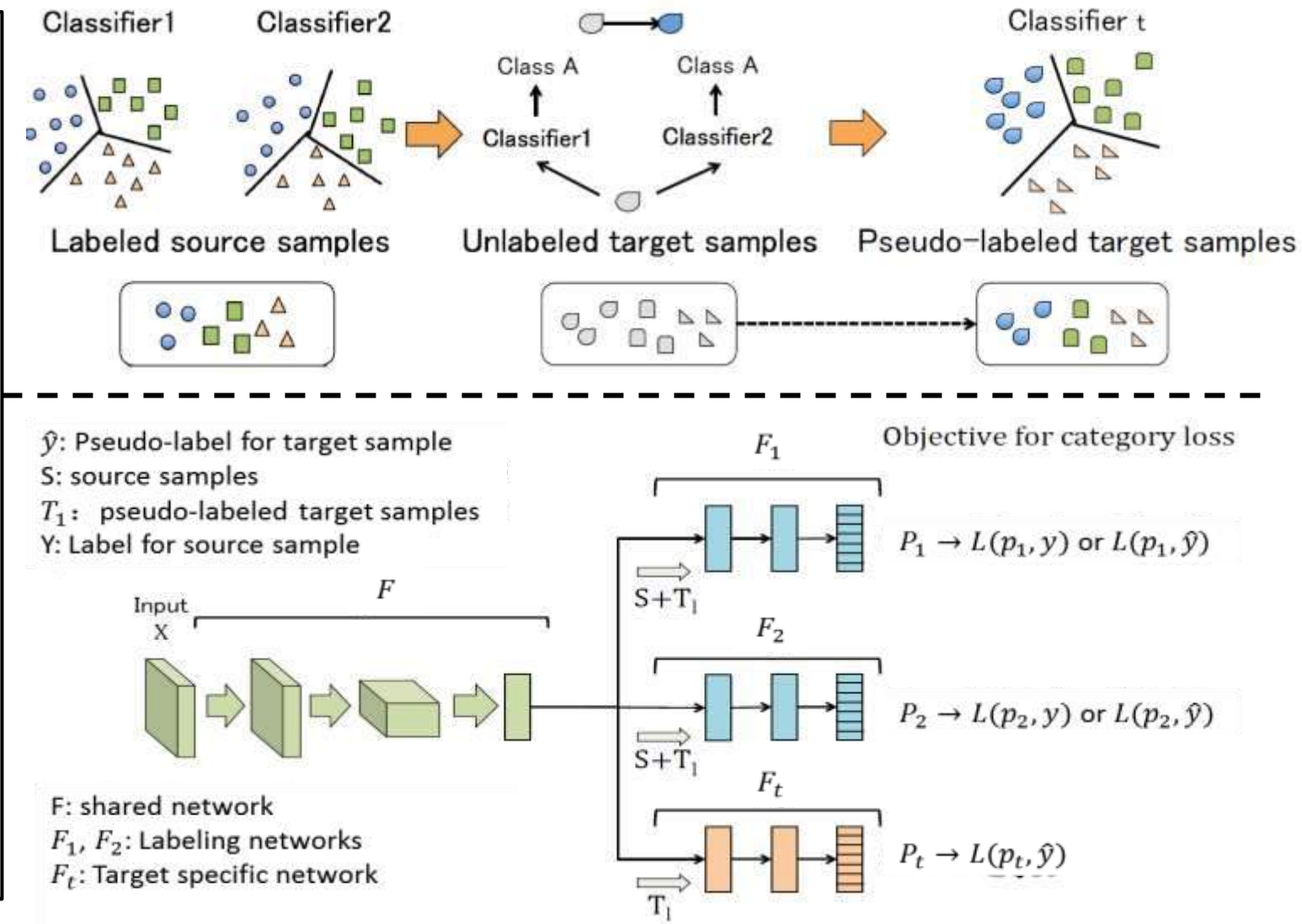
Unsupervised DA method: DSN (3.2)

Table 1: Mean classification accuracy (%) for the unsupervised domain adaptation scenarios we evaluated all the methods on. We have replicated the experiments from Ganin et al. [8] and in parentheses we show the results reported in their paper. The "Source–only" and "Target–only" rows are the results on the target domain when using no domain adaptation and training only on the source or the target domain respectively. The results that perform best in each domain adaptation task are in bold font.

Model	**MNIST to MNIST-M**	**Synth Digits to SVHN**	**SVHN to MNIST**	**Synth Signs to GTSRB**
Source-only	56.6 (52.2)	86.7 (86.7)	59.2 (54.9)	85.1 (79.0)
CORAL [27]	57.7	85.2	63.1	86.9
MMD [30, 18]	76.9	88.0	71.1	91.1
DANN [8]	77.4 (76.6)	90.3 (91.0)	70.7 (73.8)	92.9 (88.6)
DSN w/ MMD (ours)	80.5	88.5	72.2	92.6
DSN w/ DANN (ours)	**83.2**	**91.2**	**82.7**	**93.1**
Target-only	98.7	92.4	99.5	99.8

Unsupervised DA method: Tri-training (4.1)

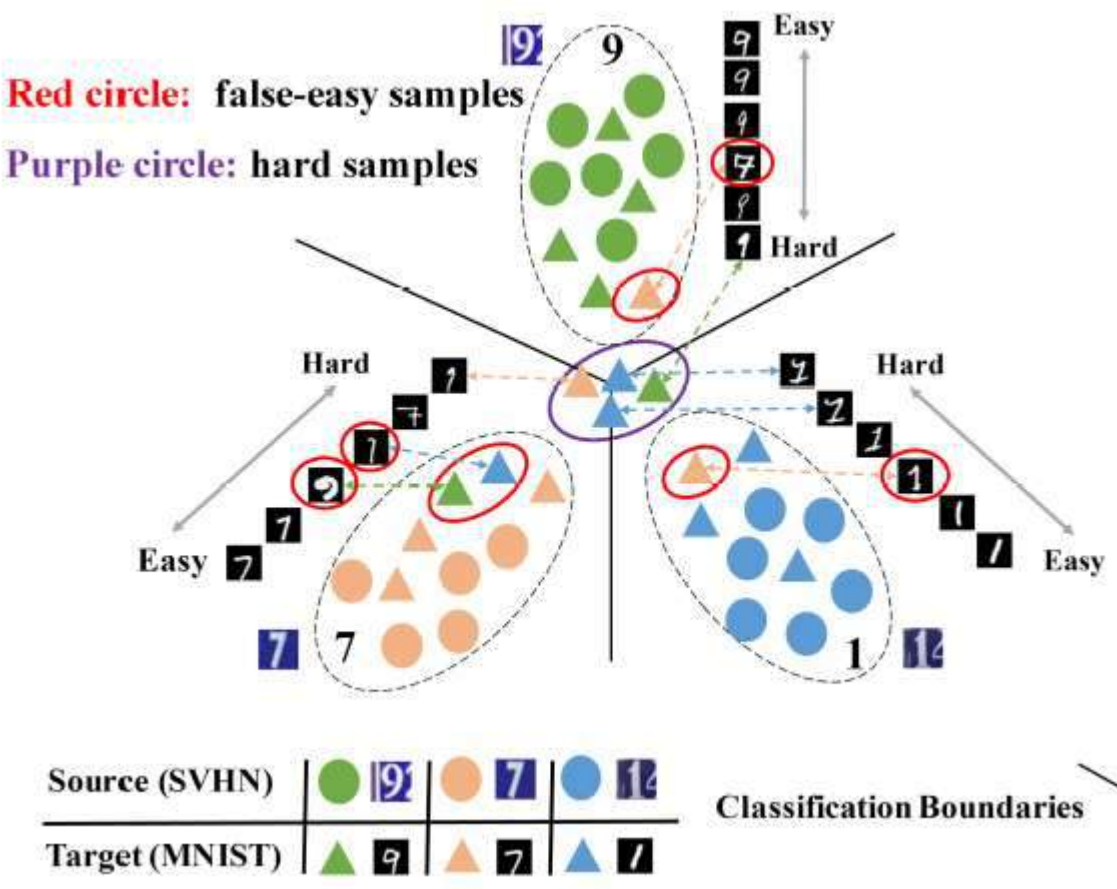
Tri-training method use three networks asymmetrically to generate pseudo labels. By asymmetric, two networks are used to label unlabeled target samples and one network is trained by the samples to obtain target discriminative representations.



[14] Saito K, Ushiku Y, Harada T. Asymmetric tri-training for unsupervised domain adaptation. In ICML 2017.

Unsupervised DA method: EHTS+APA (4.2)

Firstly, an Easy-to-Hard Transfer Strategy (EHTS) progressively selects reliable pseudo-labeled target samples with cross-domain similarity measurements.



The source prototype is a mean vector of the embedded source samples in each class

$$c_k^{\mathcal{S}} = \frac{1}{N_s^k} \sum_{(x_i^s, y_i^s) \in D_s^k} G(x_i^s),$$

It uses a similarity measurement to cluster unlabeled target sample to the corresponding source prototypes

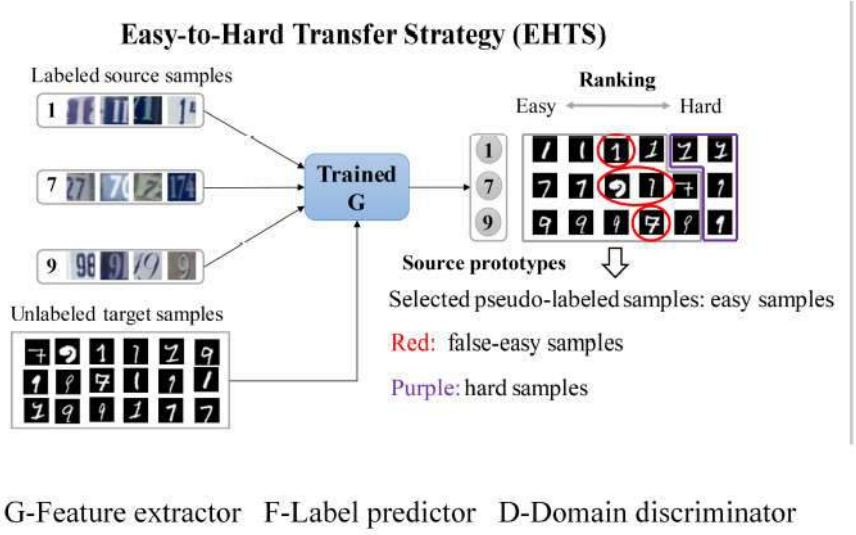
$$\psi(x_j^t) = CS(G(x_j^t), c_k^{\mathcal{S}}), k = \{1, 2, ..., C\}$$

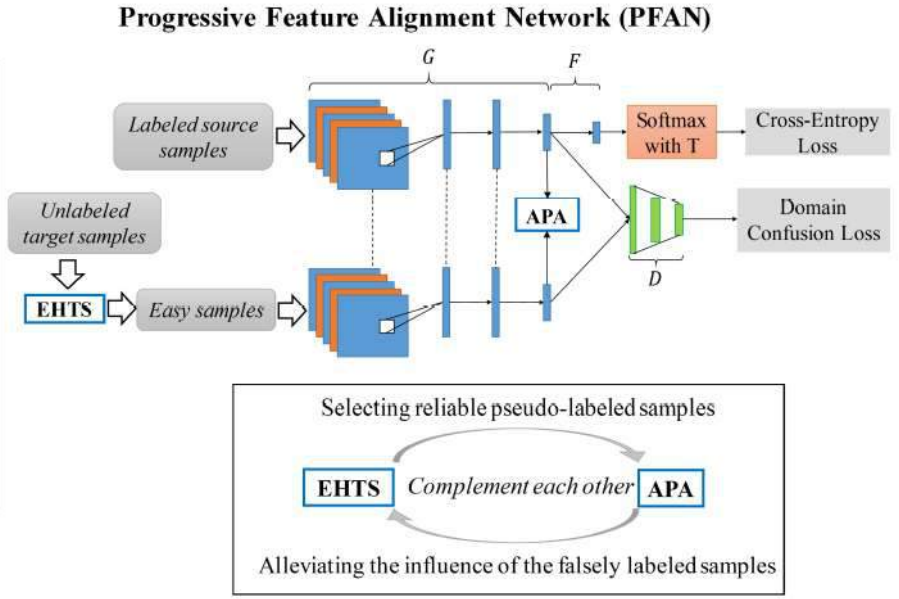
$$\tau = \frac{1}{1 + e^{-\mu \cdot (m+1)}} - 0.01,$$

[15] Chen C, Xie W, Xu T, et al. Progressive Feature Alignment for Unsupervised Domain Adaptation. arXiv preprint arXiv:1811.08585, 2018.

Unsupervised DA method: EHTS+APA (4.2)

Adaptive Prototype Alignment (APA) is proposed to align the source and target prototypes for each category. Rather than back-propagating the category loss for target samples based on pseudo-labeled samples, our work statistically align the cross-domain class distributions based on the source samples and the selected pseudo-labeled target samples.





1.Initial global prototypes

$$c^{\mathcal{T}}_{k(0)} = \frac{1}{\hat{D}^k_t} \sum_{(x^t_j, y^t_j) \in \hat{D}^k_t} G(x^t_j).$$

2. Updated prototypes

$$\rho_t = CS(\bar{c}^t_{k(I)}, c^{\mathcal{T}}_{k(I-1)}),$$
$$c^{\mathcal{T}}_{k(I)} \leftarrow {\rho_t}^2 \bar{c}^t_{k(I)} + (1 - {\rho_t}^2) c^{\mathcal{T}}_{k(I-1)}$$

3. Prototypes align

$$\mathcal{L}_{apa}(\theta_g) = \sum_{k=1}^{C} d(c^{\mathcal{S}}_{k(I)}, c^{\mathcal{T}}_{k(I)}).$$

Deep visual domain adaptation: A survey

1 One-step DA

		Supervised DA	Unsupervised DA
Discrepancy-based DA	Class criterion	√	
	Statistic criterion		√
	Architecture criterion	√	√
	Geometric criterion	√	
Adversarial-based DA	Generative model		√
	Non-generative model		√
Reconstruction-based DA	Encoder-decoder model		√
	Adversarial model		√

2 Multi-step DA

5. Multi-step domain adaptation

For multi-step DA, the selection of the intermediate domain is problem specific, and different problems may have different strategies.

3 Heterogeneous DA

4.2. Heterogeneous domain adaptation

In heterogeneous DA, the feature spaces
get domains are not the same, $Xs \neq Xt$, ar
the feature spaces may also differ. Accordi
of feature spaces, heterogeneous DA can be
two scenarios. In one scenario, the source ar
contain images, and the divergence of feat
caused by different sensory devices (e.g., visu
infrared (NIR) or RGB vs. depth) and different
sketches vs. photos). In the other scenario, th
of media in source and target domain (e.g., t

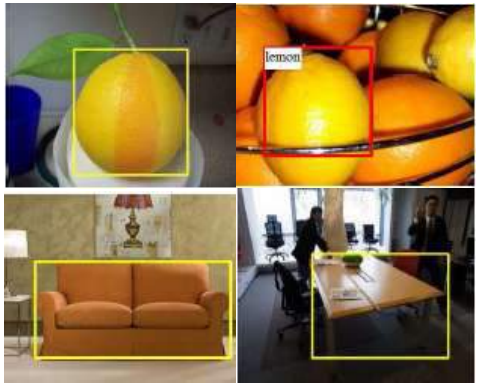
[2] M. Wang and W. Deng. Deep visual domain adaptation: A survey. Neurocomputing, 312:135 – 153, 2018.

Application of domain adaptation

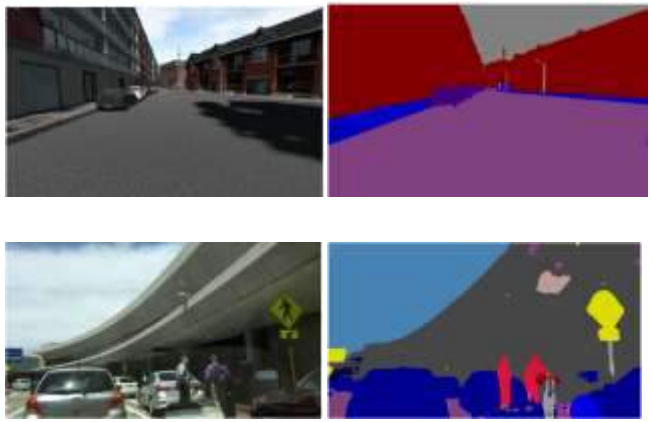
1 Image classification

2 Face recognition

3 Object detection



4 Semantic segmentation



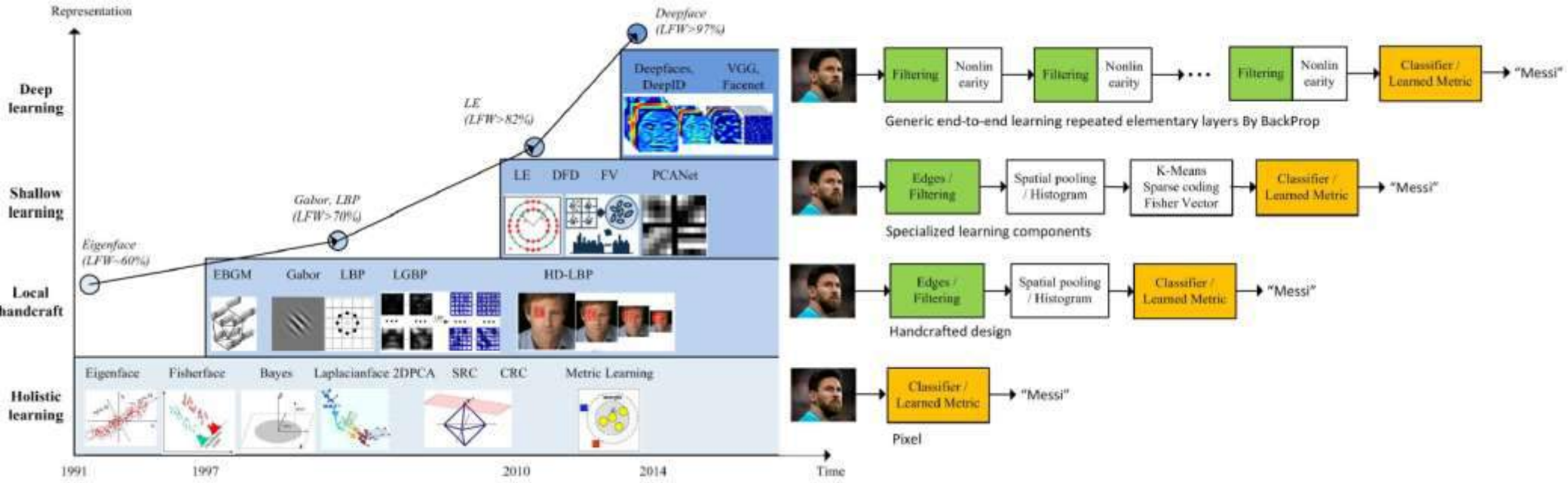
5 Person re-identification

6 Image-to-image translation

[2] M. Wang and W. Deng. Deep visual domain adaptation: A survey. Neurocomputing, 312:135 – 153, 2018.

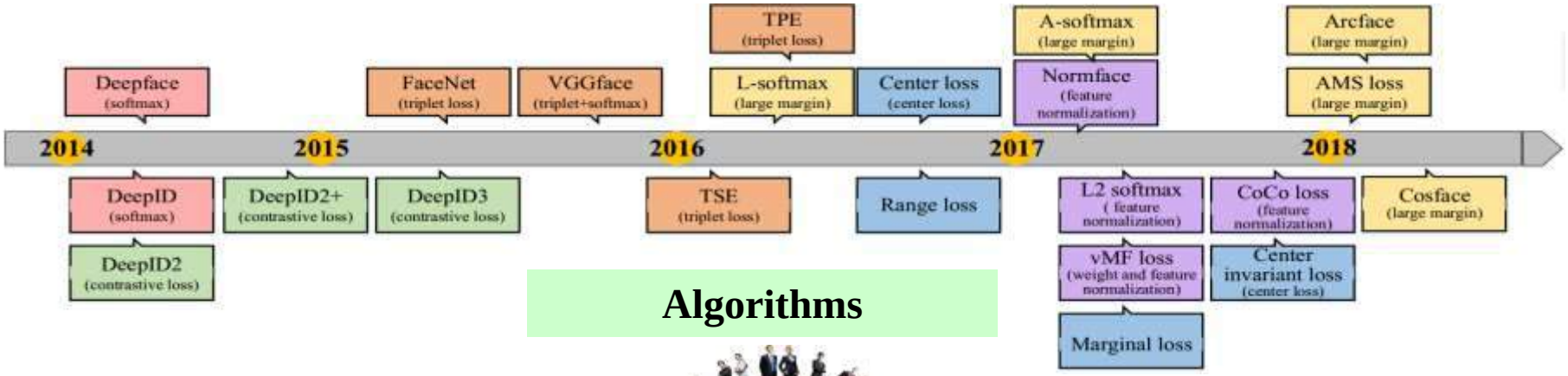
Face recognition

In 2014, DeepFace and DeepID achieved state-of-the-art accuracy, and research focus has shifted to deep-learning-based approaches. As the representation pipeline becomes deeper and deeper, the LFW (Labeled Face in-the-Wild) performance steadily improves from around 60% to above 90%, while deep learning boosts the performance to 99.80% in only three years.



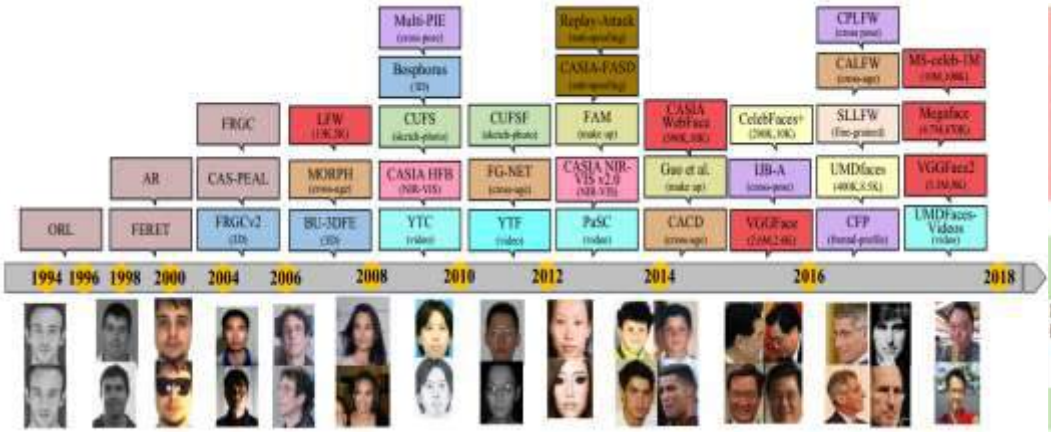
[16] M. Wang and W. Deng. Deep face recognition: A survey. arXiv preprint arXiv:1804.06655, 2018. 1, 2, 3

Face recognition

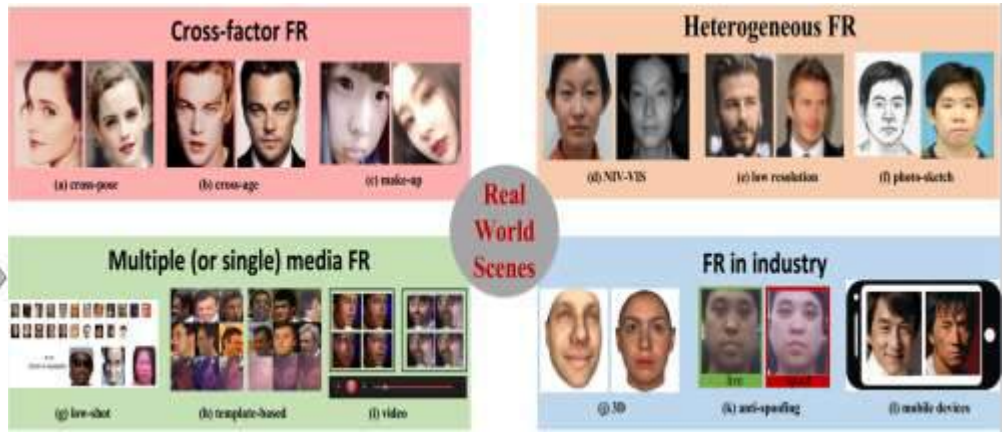


Algorithms

Data



Scenes



[16] M. Wang and W. Deng. Deep face recognition: A survey. arXiv preprint arXiv:1804.06655, 2018. 1, 2, 3

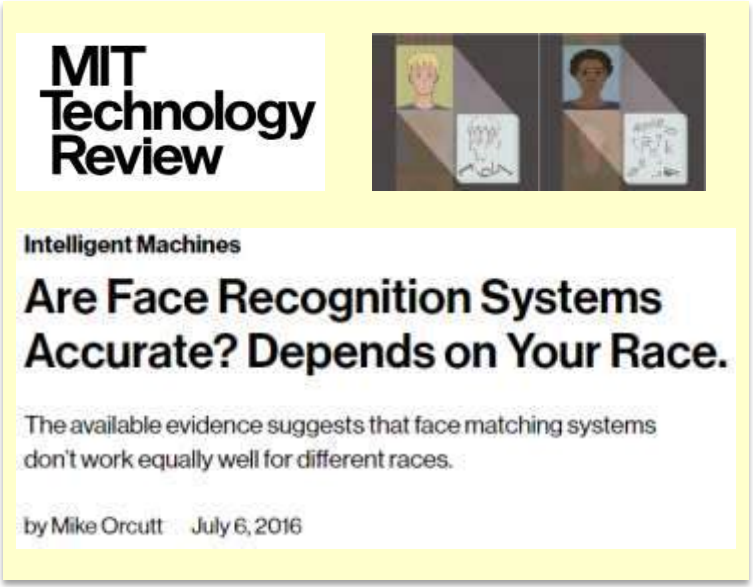
Background

More and more people find that a problematic issue, namely racial bias, has always been concealed in the previous studies due to biased benchmarks but explicitly degrades the performance in realistic FR systems.

1 Amazon's Rekognition Tool incorrectly matched 28 U.S. congressmen with criminals, especially non-Caucasian people.

2 The accuracies of 3 commercial gender classification algorithms drop largely on darker female faces.

3 MIT Technology Reviewer suggested that racial bias in databases will reflect in algorithms, hence the performances of FR systems depend on the race.


MIT Technology Review

Intelligent Machines

Are Face Recognition Systems Accurate? Depends on Your Race.

The available evidence suggests that face matching systems don't work equally well for different races.

by Mike Orcutt July 6, 2016

[17]Wang M, Deng W, Hu J, et al. Racial Faces in-the-Wild: Reducing Racial Bias by Deep Unsupervised Domain Adaptation. arXiv preprint arXiv:1812.00194, 2018.

Blocks

So little testing information available makes it hard to measure the racial bias in existing FR algorithms and there has yet to be a comprehensive study that investigates how deep FR algorithms are affected by it.

Train/ Test	Database	Racial distribution (%)			
		Caucasian	Asian	Indian	African
train	CASIA-WebFace [17]	84.5	2.6	1.6	11.3
	VGGFace2 [18]	74.2	6.0	4.0	15.8
	MS-Celeb-1M [19]	76.3	6.6	2.6	14.5
test	LFW [20]	69.9	13.2	2.9	14.0
	IJB-A [21]	66.0	9.8	7.2	17.0
	RFW	25.0	25.0	25.0	25.0

[18] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. arXiv preprint arXiv:1411.7923, 2014.
[19] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. arXiv preprint arXiv:1710.08092, 2017.
[20] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In ECCV, pages 87–102. Springer, 2016.
[21] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007
[22] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In CVPR, pages 1931–1939, 2015.

[17]Wang M, Deng W, Hu J, et al. Racial Faces in-the-Wild: Reducing Racial Bias by Deep Unsupervised Domain Adaptation. arXiv preprint arXiv:1812.00194, 2018.

Racial Faces in-the-Wild (RFW)

Racial Faces in-the-Wild (RFW) is a large-scale face database for studying racial bias in face recognition which has two important uses:

- **Measure racial bias of FR algorithms**. Four testing subsets, namely Caucasian, Asian, Indian and African, are constructed, and each contains about 3000 individuals with 6000 image pairs for face verification.
- **Reduce racial bias by transfer learning**. Four training subsets are offered as well. The Caucasian subset consists of about 500K labeled images of 10k identities and other-race subsets contain 50K unlabeled images, respectively.

[17]Wang M, Deng W, Hu J, et al. Racial Faces in-the-Wild: Reducing Racial Bias by Deep Unsupervised Domain Adaptation. arXiv preprint arXiv:1812.00194, 2018.

Collection process

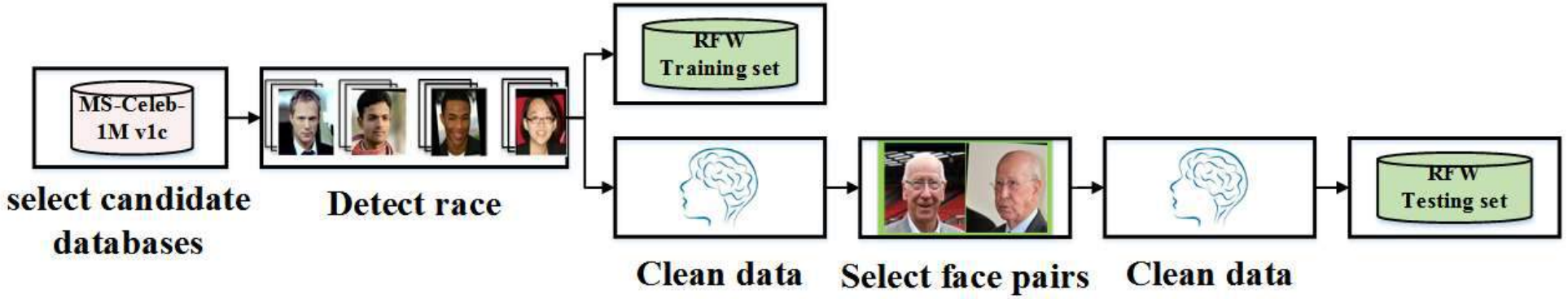
We estimate the race of images in MS-Celeb-1M using Face++ API and select about 625K images of 25K celebrities of different races to construct a new RFW database. The collection process is as following:

1 Race detection

For each identity in MS-Celeb-1M, it will be selected only if almost images are estimated as the same race by Face++ API.

2 Data re-cleaning

We manually remove outlier faces for each identity as well as outlier identities for each race manually.



[17]Wang M, Deng W, Hu J, et al. Racial Faces in-the-Wild: Reducing Racial Bias by Deep Unsupervised Domain Adaptation. arXiv preprint arXiv:1812.00194, 2018.

Collection process

We estimate the race of images in MS-Celeb-1M using Face++ API and select about 625K images of 25K celebrities of different races to construct a new RFW database. The collection process is as following:

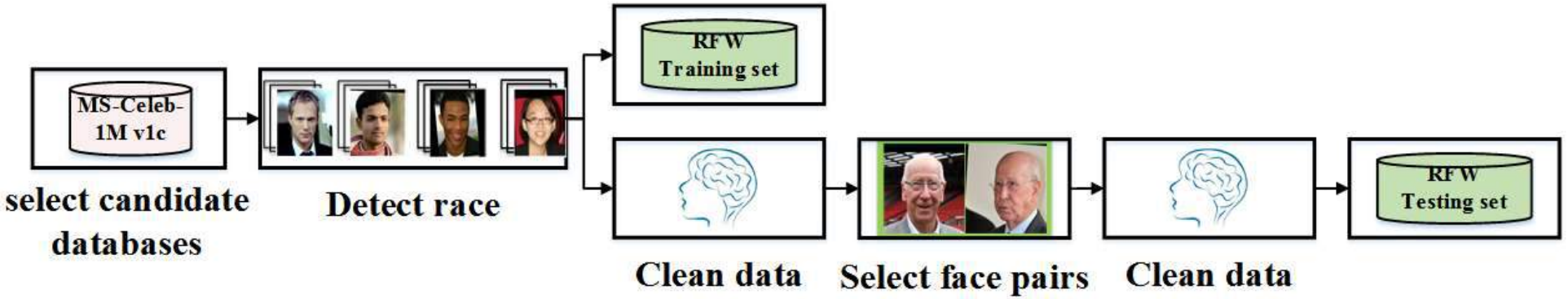
3 Testing set construction

For each identity A, we randomly select one positive pair $\{A_i, A_j\}$ from 50% of pairs with smaller cosine; and we randomly select one negative pair $\{A_i, B_j\}$ from 1% of pairs with larger cosine similarity

4 Training set construction

Subsets	Train		Test	
	#subjects	#Images	#subjects	#Images
Caucasian	10000	468139	2959	10196
Indian	-	52285	2984	10308
Asian	-	54188	2492	9688
African	-	50588	2995	10415

Table 1. The number of identities and images in RFW



[17]Wang M, Deng W, Hu J, et al. Racial Faces in-the-Wild: Reducing Racial Bias by Deep Unsupervised Domain Adaptation. arXiv preprint arXiv:1812.00194, 2018.

Statistics and analyses

We construct our testing set similar to LFW. Besides,

- RFW considers both the large intra-class variance and the small inter-class variance simultaneously to get close to more realistic scenarios.
- Four testing subsets ensure to exclude other factors (e.g. pose, age and gender) except for race which can cause difference.

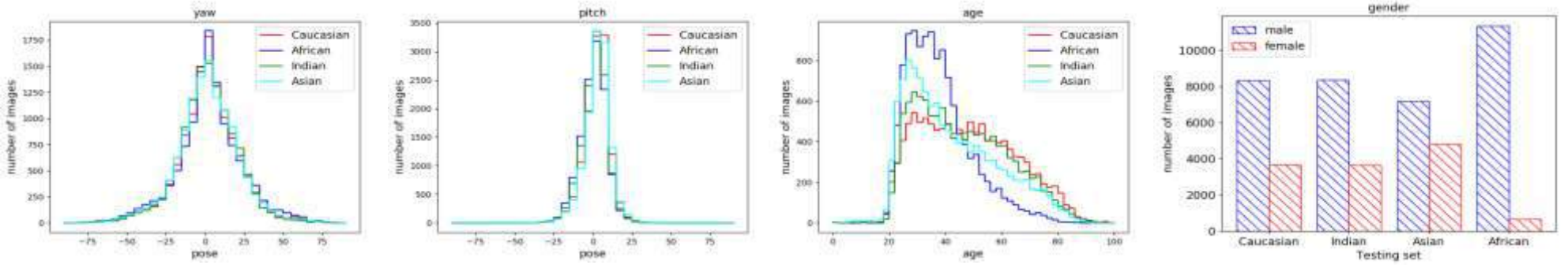


Figure 1. The pose (yaw and pitch), age and gender distribution of four testing subsets

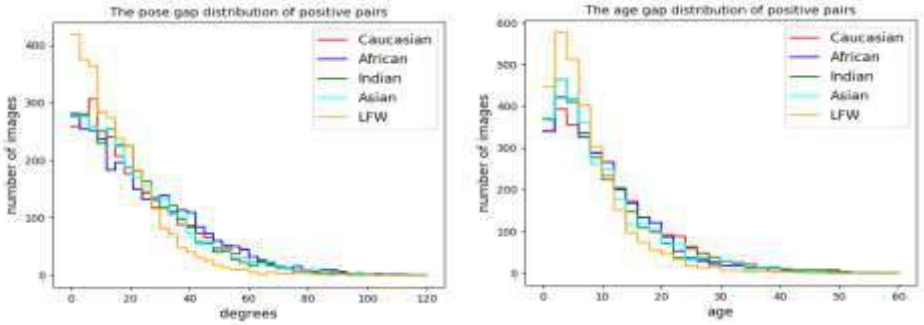


Figure 2. The pose and age gap distribution of RFW and LFW

From Fig. 1, there are no large differences in pose, age and gender distribution between Caucasian, Indian and Asian testing sets. African set has a smaller age gap and the least females which makes Africans are easier to be recognized.

From Fig. 2, Compared to LFW, the pose and age gap of positive pairs in RFW are larger which shows we successfully add variations to intra-class.

[17]Wang M, Deng W, Hu J, et al. Racial Faces in-the-Wild: Reducing Racial Bias by Deep Unsupervised Domain Adaptation. arXiv preprint arXiv:1812.00194, 2018.

Statistics and analyses

We construct our testing set similar to LFW. Besides,

- RFW considers both the large intra-class variance and the small inter-class variance simultaneously to get close to more realistic scenarios.
- Four testing subsets ensure to exclude other factors (e.g. pose, age and gender) except for race which can cause difference.

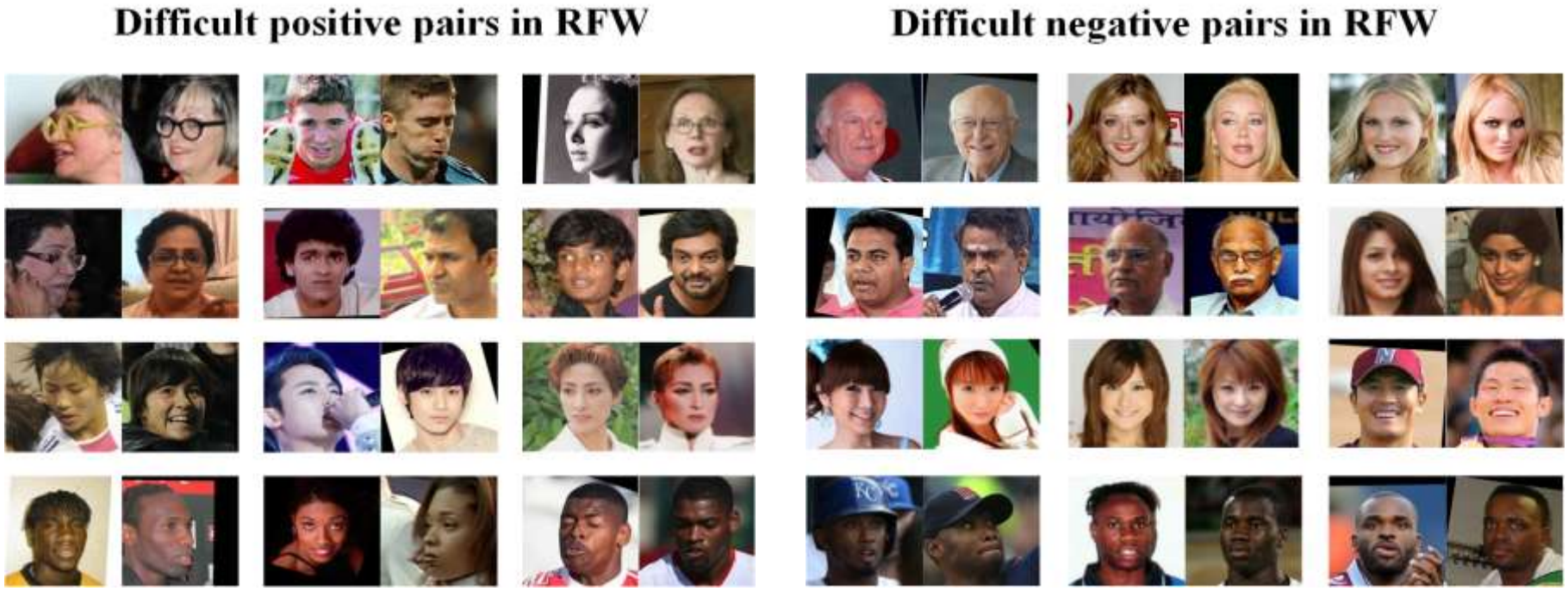


Figure 3. Examples of difficult pairs in RFW dataset, which challenge the recognizer by the pose, age, expression and make-up variations of same people and the similar appearance of different people.

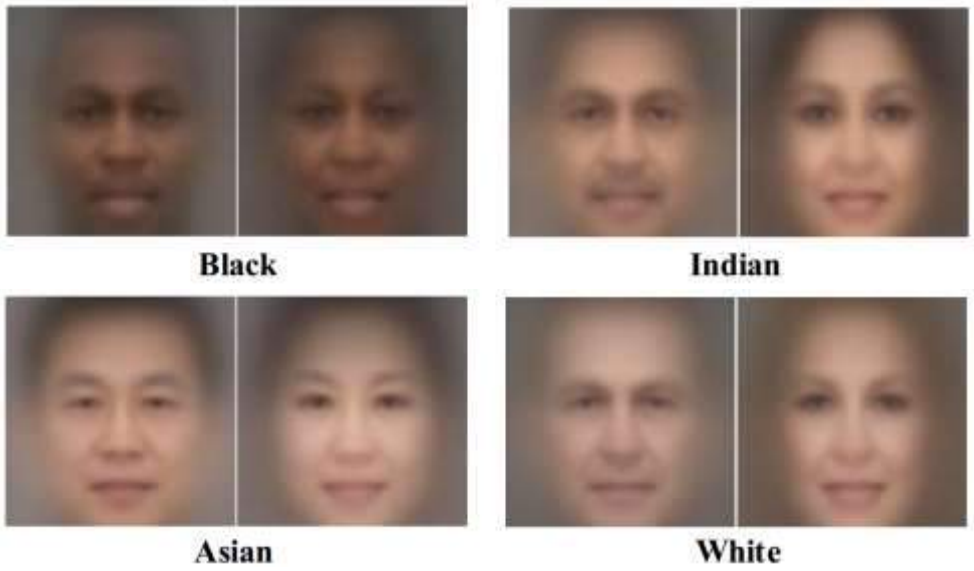
[17]Wang M, Deng W, Hu J, et al. Racial Faces in-the-Wild: Reducing Racial Bias by Deep Unsupervised Domain Adaptation. arXiv preprint arXiv:1812.00194, 2018.

Domain gap

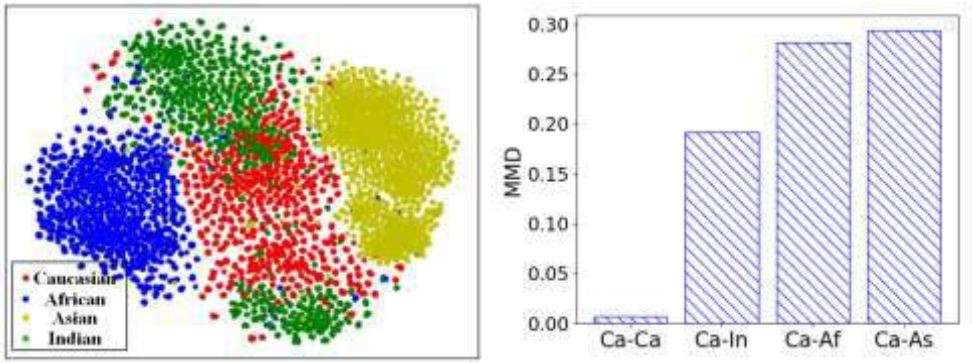
Through experiments on our RFW, we first prove that:

- From results of average faces, T-SNE and distribution discrepancy measured by MMD, there is domain gap between Caucasians and other races.
- FR systems indeed work unequally well for different races (racial bias); the deep models trained on the current benchmarks do not perform well on non-Caucasian faces (other-race effect).

1 Image level



2 Feature level



(a) The feature space of four testing subsets visualized by t-SNE. Each dot color represents a image belong to Caucasian, Indian, Asian or African. (b) The distribution discrepancy measured by MMD. 'Ca', 'As', 'In' and 'Af' represent Caucasian, Asian, Indian and African, respectively. 'Ca-As' represents the distribution discrepancy between Caucasian and Asian, and so on

[17]Wang M, Deng W, Hu J, et al. Racial Faces in-the-Wild: Reducing Racial Bias by Deep Unsupervised Domain Adaptation. arXiv preprint arXiv:1812.00194, 2018.

Racial bias

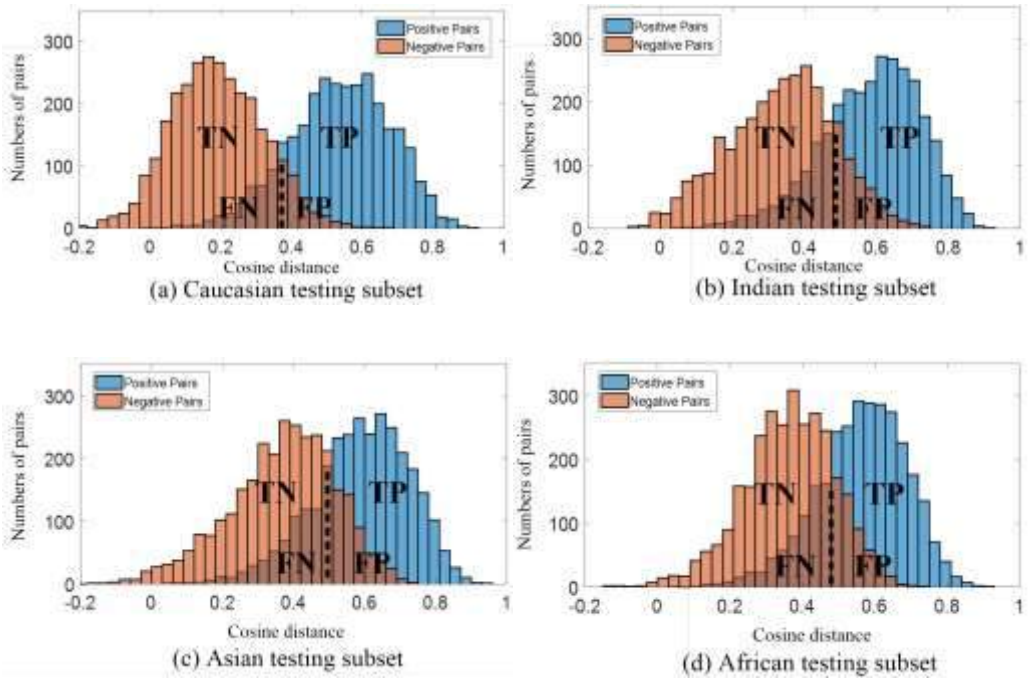
Through experiments on our RFW, we first prove that:

- From results of average faces, T-SNE and distribution discrepancy measured by MMD, there is domain gap between Caucasians and other races.
- FR systems indeed work unequally well for different races (racial bias); the deep models trained on the current benchmarks do not perform well on non-Caucasian faces (other-race effect).

Table 2. Face Verification Accuracy (%) on RFW dataset

	Model	LFW	RFW			
			Caucasian	Indian	Asian	African
Algorithms	Center-loss	98.75	87.18	81.92	79.32	78.00
	SphereFace	99.27	90.80	87.02	82.95	82.28
	ArcFace	99.40	92.15	88.00	83.98	84.93
	VGGFace2	99.30	89.90	86.13	84.93	83.38
	Mean	**99.18**	**90.01**	**85.77**	**82.80**	**82.15**
Commercial APIs	Face++	97.03	93.90	88.55	92.47	87.50
	Baidu	98.67	89.13	86.53	90.27	77.97
	Amazon	98.50	90.45	87.20	84.87	86.27
	Microsoft	98.22	87.60	82.83	79.67	75.83
	Mean	**98.11**	**90.27**	**86.28**	**86.82**	**81.89**

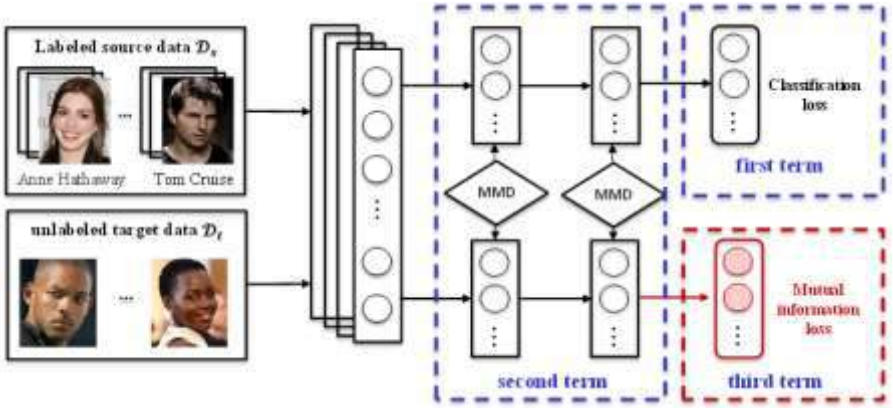
Figure 4. Distribution of cosine-distances of 6000 pairs



[17]Wang M, Deng W, Hu J, et al. Racial Faces in-the-Wild: Reducing Racial Bias by Deep Unsupervised Domain Adaptation. arXiv preprint arXiv:1812.00194, 2018.

Deep information maximization adaptation network (IMAN)

We propose a new domain adaptation method, i.e. IMAN. It identifies a feature space where data in the source and the target domains are similarly distributed, it also learns the target feature space discriminatively, optimizing an mutual-information loss as an proxy to maximize the decision margin on the unlabeled target domain.



Softmax loss: $\mathcal{L}_{Softmax} = \frac{1}{N}\sum_i y_i log(p_i)$

Mutual information loss:

$$\mathcal{L}_M = \frac{1}{M}\sum_i H[p_i] - \lambda H[\overline{p_0}] = -I_t(X;\hat{y}),$$

where $\overline{p_0}=[\frac{1}{N}\sum_i p_{i1}, \frac{1}{N}\sum_i p_{i2}, \ldots, \frac{1}{N}\sum_i p_{ic}]$

MMD loss:

$$\mathcal{L}_{MMD} = \left\| \frac{1}{M}\sum_{i=1}^{M}\phi(x_i^s) - \frac{1}{N}\sum_{j=1}^{N}\phi(x_j^t) \right\|_H^2$$

Methods	Caucasian	Indian	Asian	African
Baseline	94.78	90.48	86.27	85.13
DDC [4]	-	91.63	87.55	86.28
DAN [5]	-	91.78	87.78	86.30
PL5 [22]	-	92.00	88.33	87.67
PL5+PL1	-	92.08	88.80	88.12
PL5+MMD	-	92.00	88.65	87.92
IMAN (ours)	-	93.55	89.87	88.88
IMAN* (ours)	-	**94.15**	**91.15**	**91.42**

[4] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. Computer Science, 2014.
[5] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In ICML, pages 97–105, 2015.
[23] A. Nech and I. Kemelmacher-Shlizerman. Level playing field for million scale face recognition. In CVPR, pages 3406–3415. IEEE, 2017.
[17]Wang M, Deng W, Hu J, et al. Racial Faces in-the-Wild: Reducing Racial Bias by Deep Unsupervised Domain Adaptation. arXiv preprint arXiv:1812.00194, 2018.

Related works

[1] Pan S J, Yang Q. A survey on transfer learning[J]. IEEE Transactions on knowledge and data engineering, 2010, 22(10): 1345-1359.

[2] M. Wang and W. Deng. Deep visual domain adaptation: A survey. Neurocomputing, 312:135 – 153, 2018.

[3] J. Yosinski , J. Clune , Y. Bengio , H. Lipson , How transferable are features in deep neural networks? In NIPS, 2014, pp. 3320–3328 .

[4] Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. Deep domain confusion: Maximizing for domain invariance. CoRR, abs/1412.3474, 2014.

[5] M. Long , Y. Cao , J. Wang , M. Jordan , Learning transferable features with deep adaptation networks, in ICML, 2015, pp. 97–105 .

[6] M. Long , H. Zhu , J. Wang , M.I. Jordan , Unsupervised domain adaptation with residual transfer networks, in NIPS, 2016, pp. 136–144 .

[7] M. Long, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. arXiv preprint arXiv:1605.06636, 2016.

[8] Y. Ganin , V. Lempitsky , Unsupervised domain adaptation by backpropagation, in ICML, 2015, pp. 1180–1189 .

[9] W. Zhang, W. Ouyang, W. Li, and D. Xu. Collaborative and adversarial network for unsupervised domain adaptation. In CVPR, pages 3801–3809, 2018.

[10] Z. Pei, Z. Cao, M. Long, and J. Wang. Multi-adversarial domain adaptation. In AAAI Conference on Artificial Intelligence, 2018

[11] Zhang J, Ding Z, Li W, et al. Importance Weighted Adversarial Nets for Partial Domain Adaptation. In CVPR. 2018: 8156-8164.

[12] M. Ghifary , W.B. Kleijn , M. Zhang , D. Balduzzi , W. Li , Deep reconstruction–classification networks for unsupervised domain adaptation, in ECCV, Springer, 2016, pp. 597–613

[13] K. Bousmalis , G. Trigeorgis , N. Silberman , D. Krishnan , D. Erhan , Domain sep- aration networks, in NIPS, 2016, pp. 343–351 .

Related works

[14] Saito K, Ushiku Y, Harada T. Asymmetric tri-training for unsupervised domain adaptation. In ICML 2017.
[15] Chen C, Xie W, Xu T, et al. Progressive Feature Alignment for Unsupervised Domain Adaptation. arXiv preprint arXiv:1811.08585, 2018.
[16] M. Wang and W. Deng. Deep face recognition: A survey. arXiv preprint arXiv:1804.06655, 2018. 1, 2, 3
[17] Wang M, Deng W, Hu J, et al. Racial Faces in-the-Wild: Reducing Racial Bias by Deep Unsupervised Domain Adaptation. arXiv preprint arXiv:1812.00194, 2018.
[18] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. arXiv preprint arXiv:1411.7923, 2014.
[19] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. arXiv preprint arXiv:1710.08092, 2017.
[20] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In ECCV, pages 87–102. Springer, 2016.
[21] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007
[22] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In CVPR, pages 1931–1939, 2015.
[23] A. Nech and I. Kemelmacher-Shlizerman. Level playing field for million scale face recognition. In CVPR, pages 3406–3415. IEEE, 2017.