PEARL: Perceptual Adaptive Representation Learning in the Wild

Adversarial Domain Adaptation


Kate Saenko


Trevor Darrell


Eric Tzeng


Judy Hoffman

BOSTON UNIVERSITY

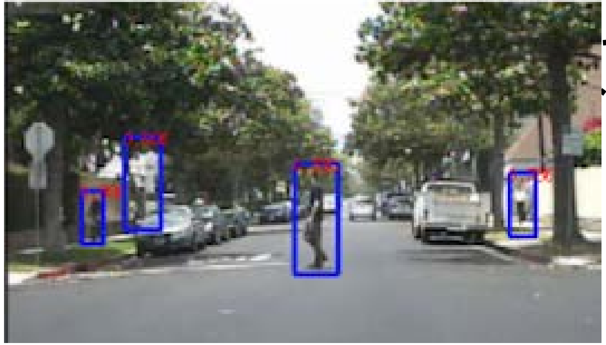
Berkeley
UNIVERSITY OF CALIFORNIA

Has deep learning solved AI?

pedestrian detection FAIL

https://www.youtube.com/watch?v=w2pwxv8rFkU

“What you saw is not what you get”



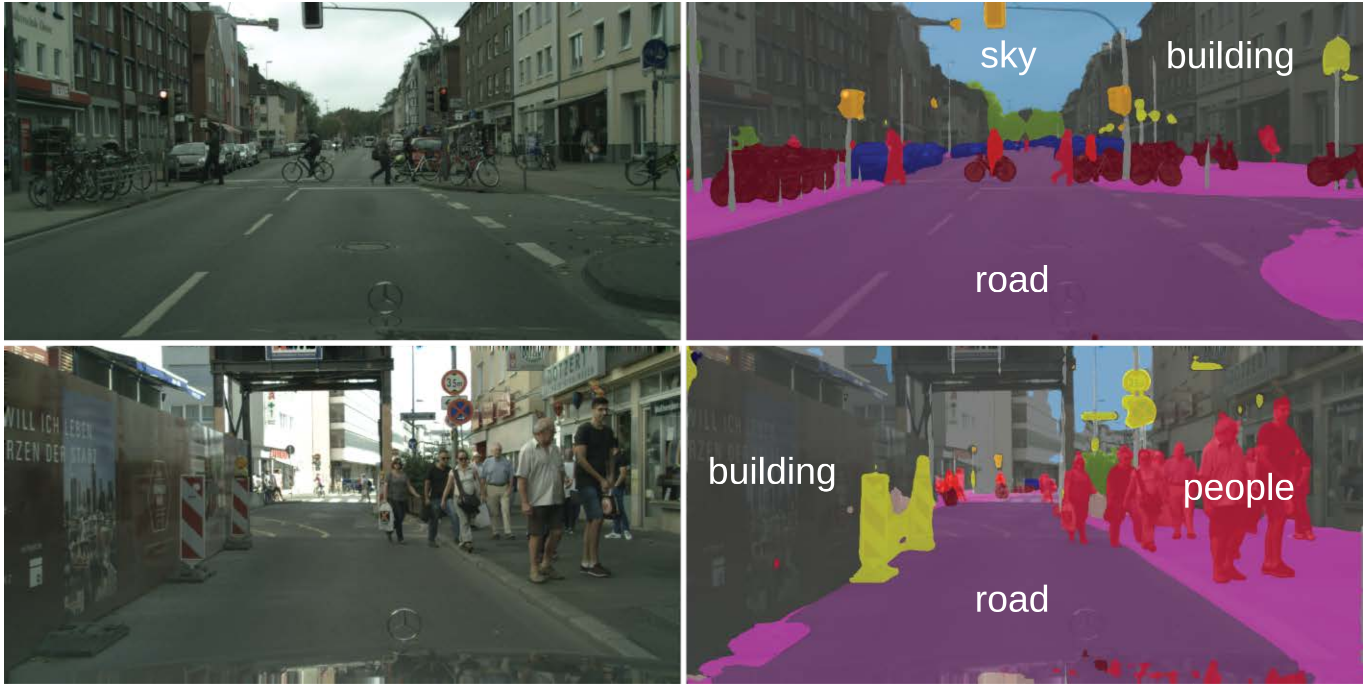
What your net is trained on

What it’s asked to label

“Dataset Bias”
“Domain Shift”
“Domain Adaptation”
“Domain Transfer”

Example shift: scene segmentation

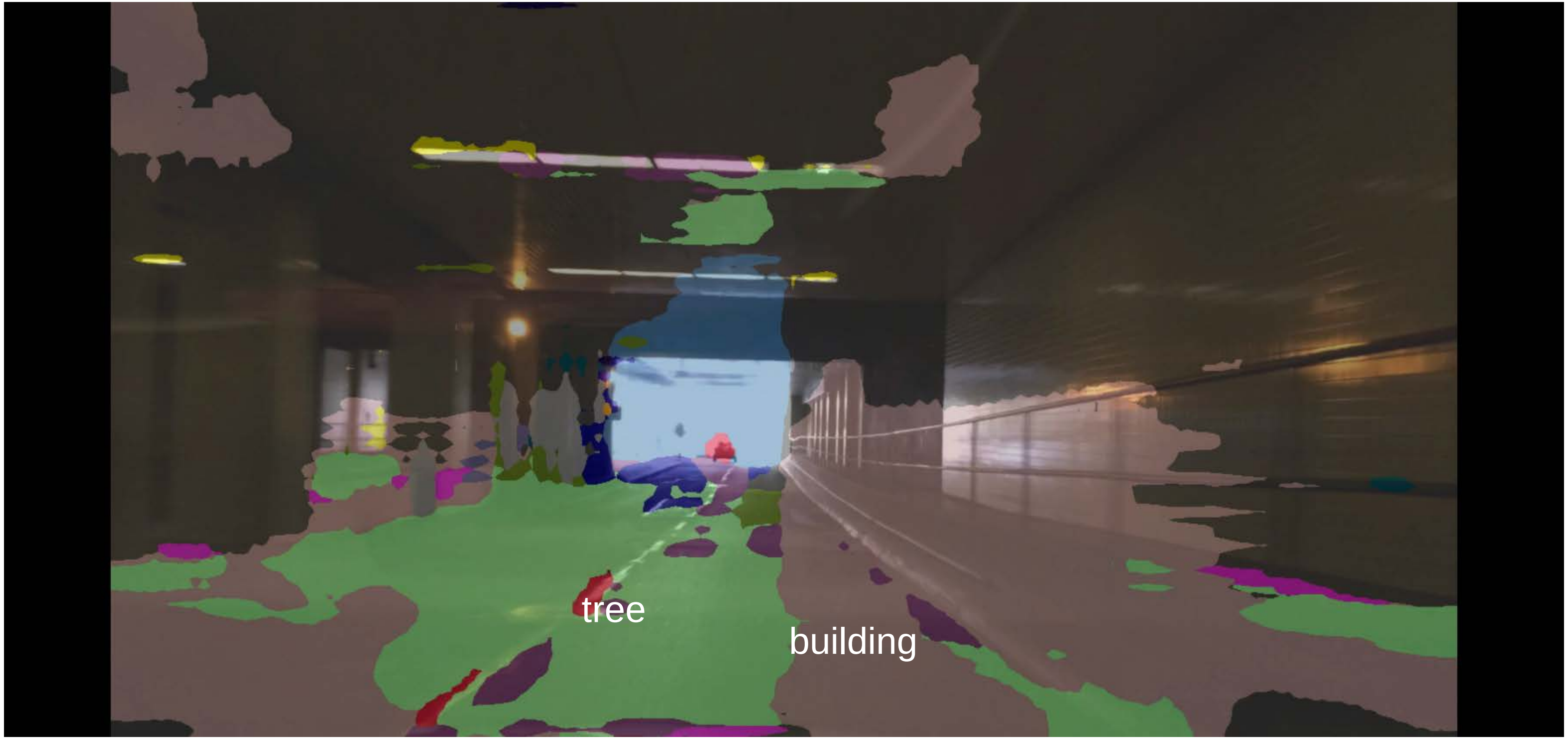
Train on Cityscapes, Test on Cityscapes



FCNs in the Wild: Pixel-level Adversarial and Constraint-based Adaptation, Judy Hoffman, Dequan Wang, Fisher Yu, Trevor Darrell, Arxiv 2016

Example shift: scene segmentation

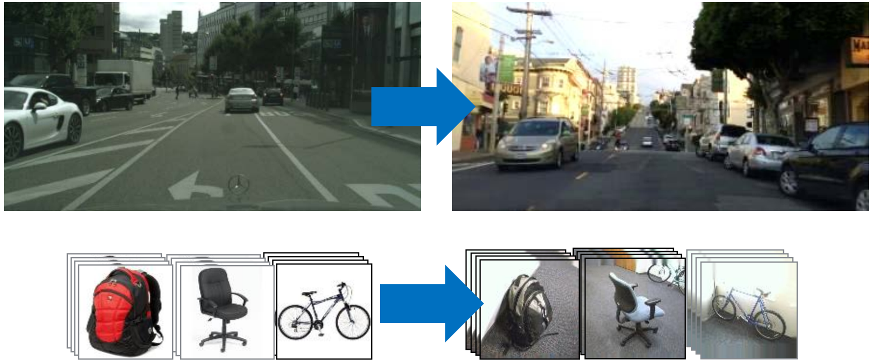
Train on Cityscapes, Test on San Francisco



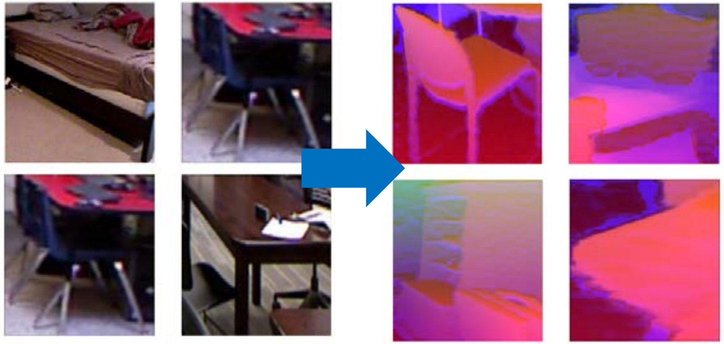
FCNs in the Wild: Pixel-level Adversarial and Constraint-based Adaptation, Judy Hoffman, Dequan Wang, Fisher Yu, Trevor Darrell, Arxiv 2016

Today: solving the domain shift problem

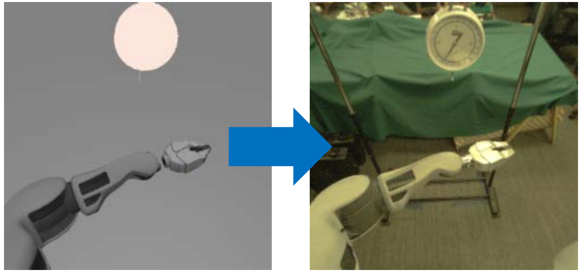
From dataset to dataset



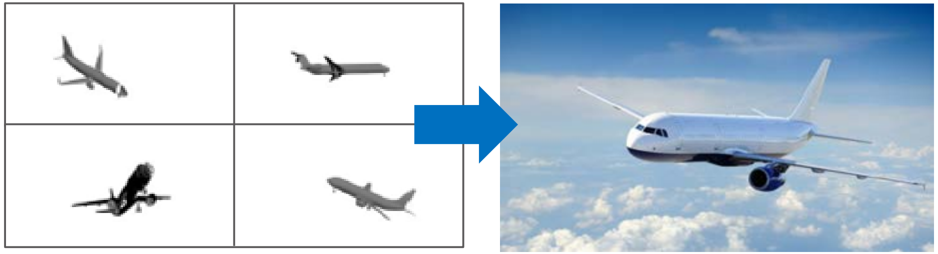
From RGB to depth



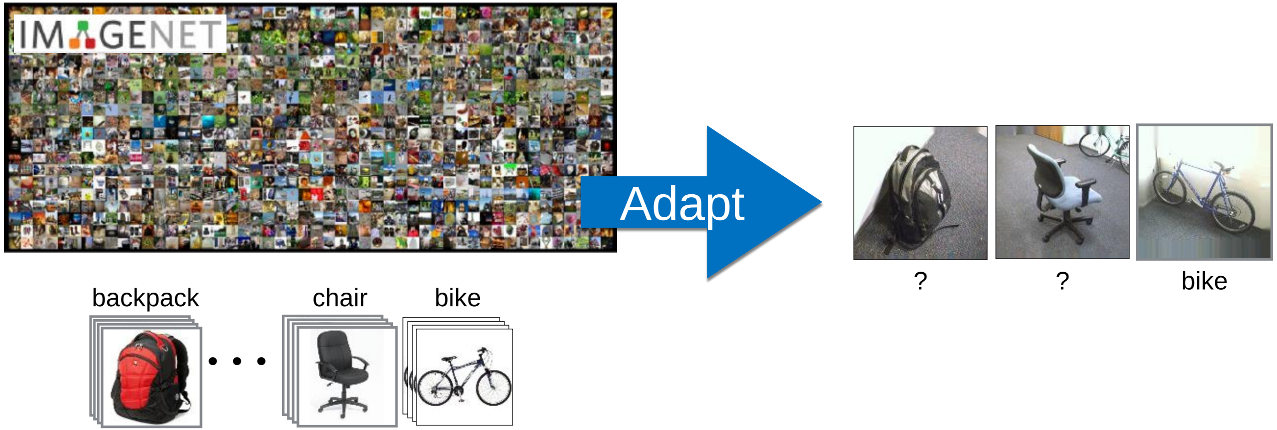
From simulated to real control



From CAD models to real images



Background: Domain Adaptation from source to target distribution



Source Domain $\sim P_S(X, Y)$ $\neq$ Target Domain $\sim P_T(Z, H)$

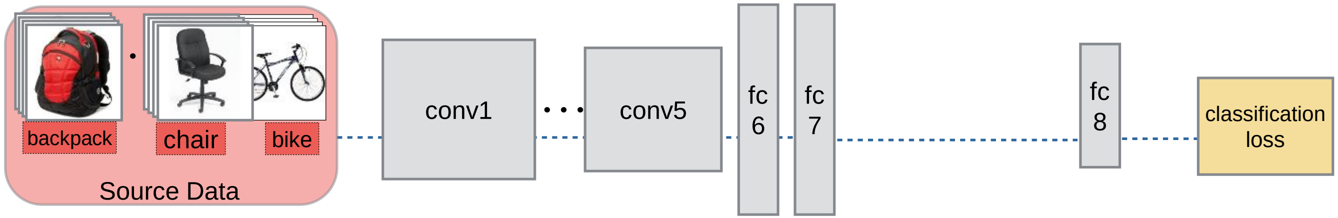
lots of **labeled** data

unlabeled or limited labels

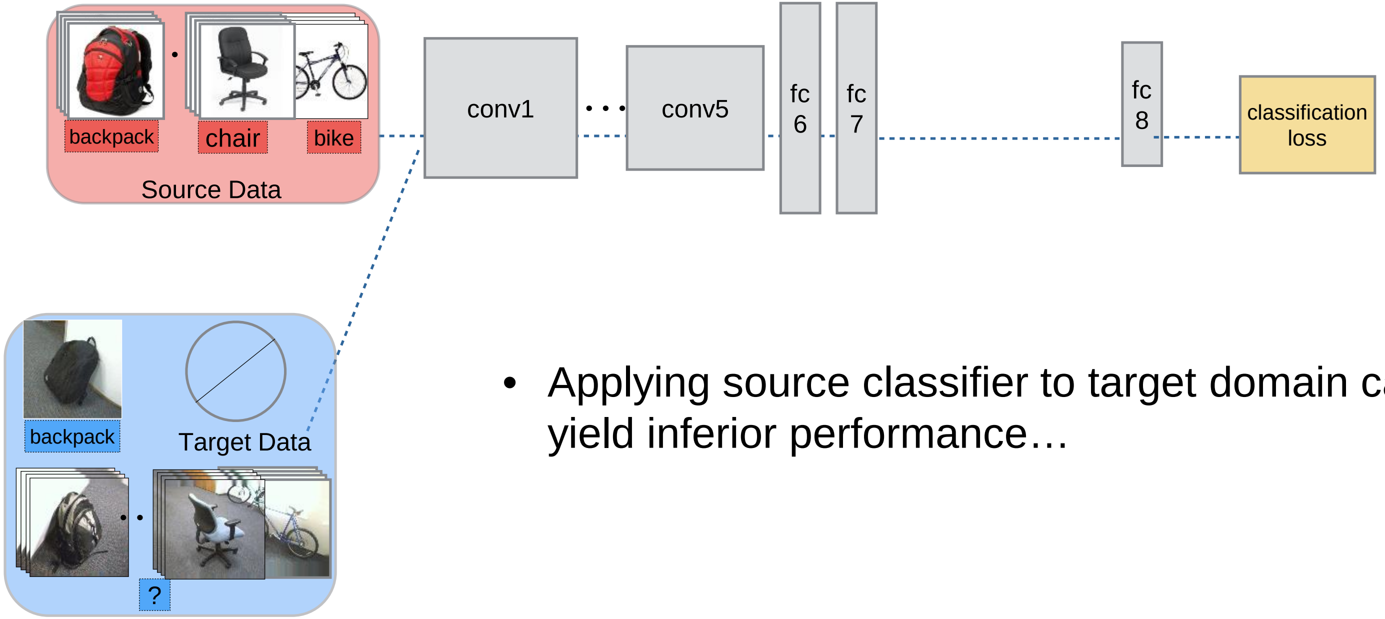
$$D_S = \{(\mathbf{x}_i, y_i), \forall i \in \{1, \ldots, N\}\}$$

$$D_T = \{(\mathbf{z}_j, ?), \forall j \in \{1, \ldots, M\}\}$$

How to adapt a deep network?



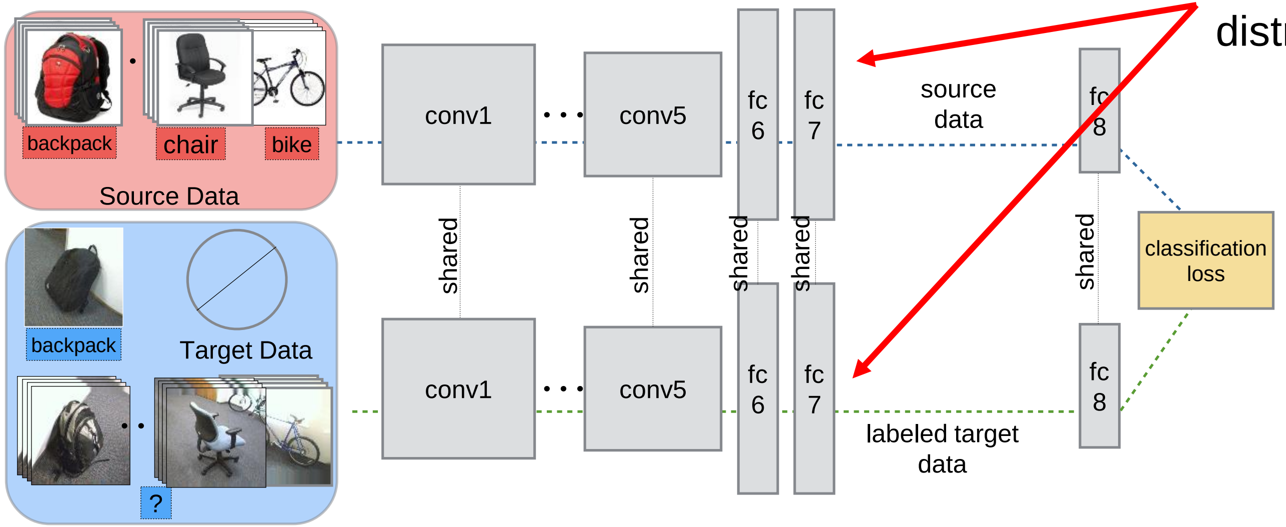
How to adapt a deep network?



- Applying source classifier to target domain can yield inferior performance…

How to adapt a deep network?

IDEA: align feature distributions

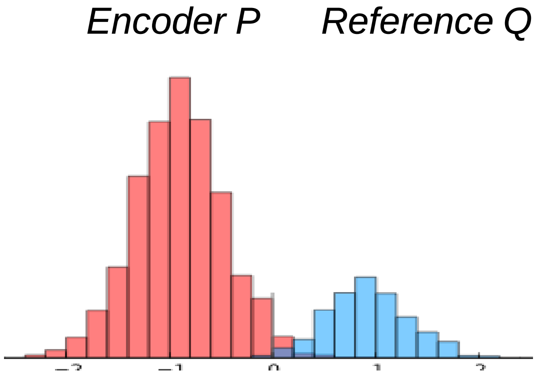


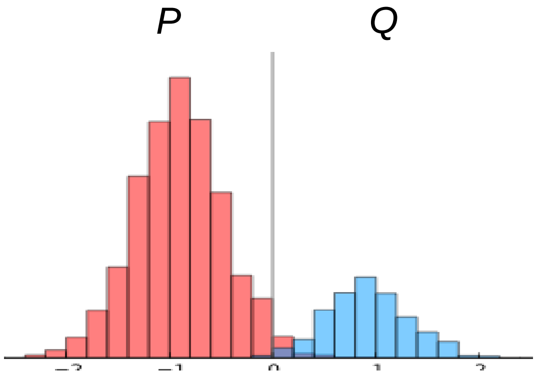
- Fine tune?

  …..Zero or few labels in target domain

Adversarial networks

Adversarial networks

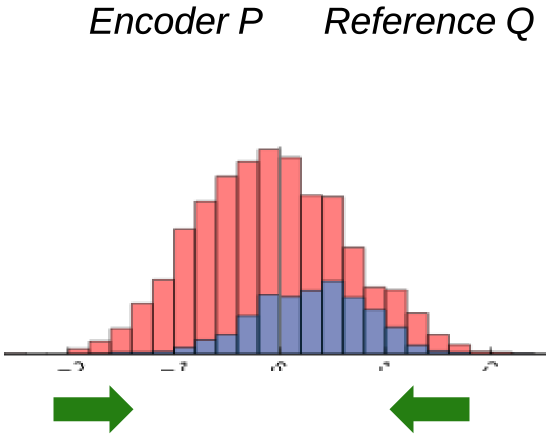


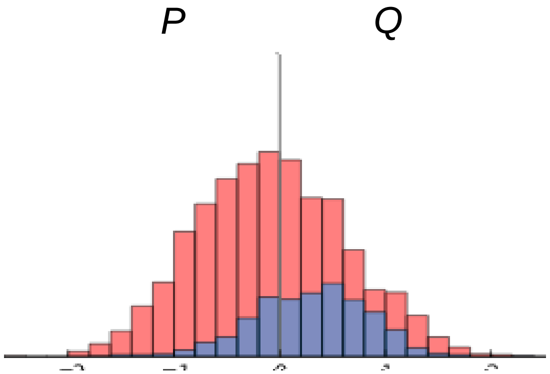


Encoder
Generates features such that their distribution P matches reference distribution Q

Adversary
Tries to discriminate between samples from P and samples from Q

Adversarial networks





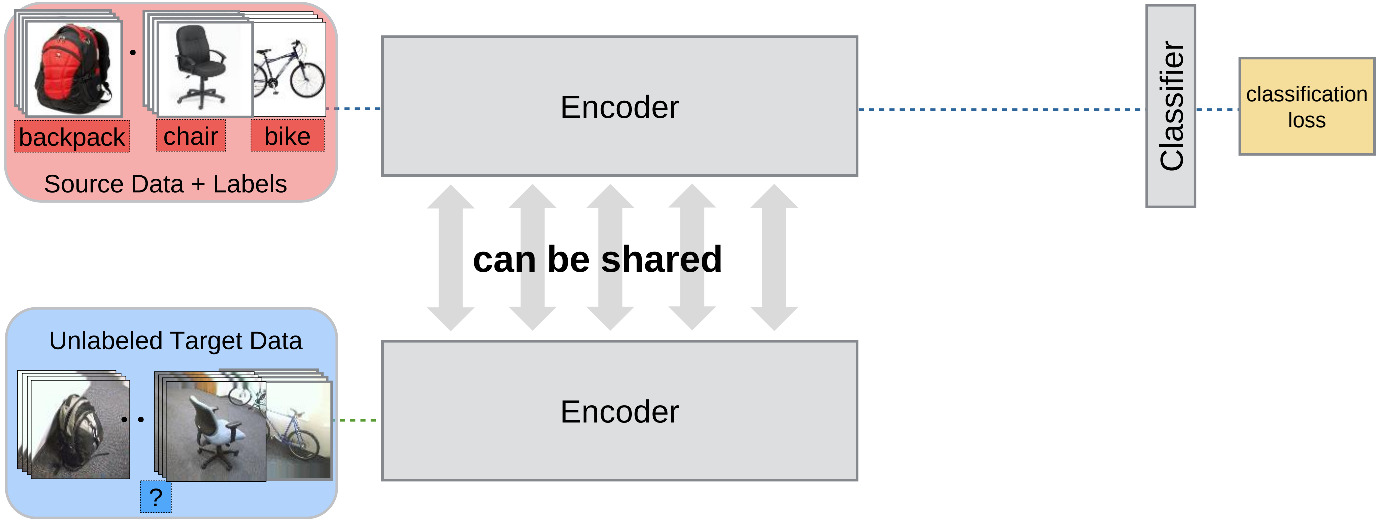
Encoder
Generates features such that their distribution P matches reference distribution Q

fools adversary

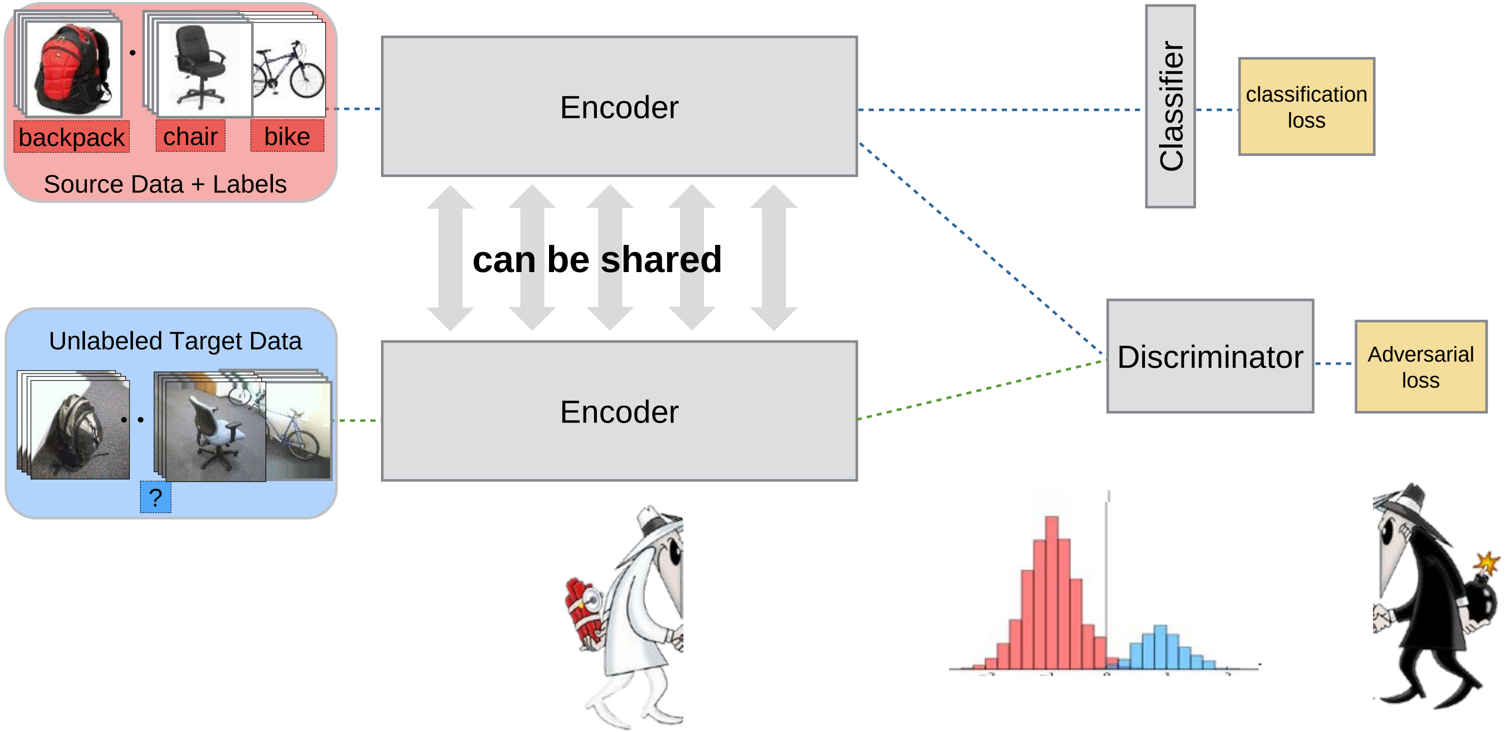
Adversary
Tries to discriminate between samples from P and samples from Q

tries harder

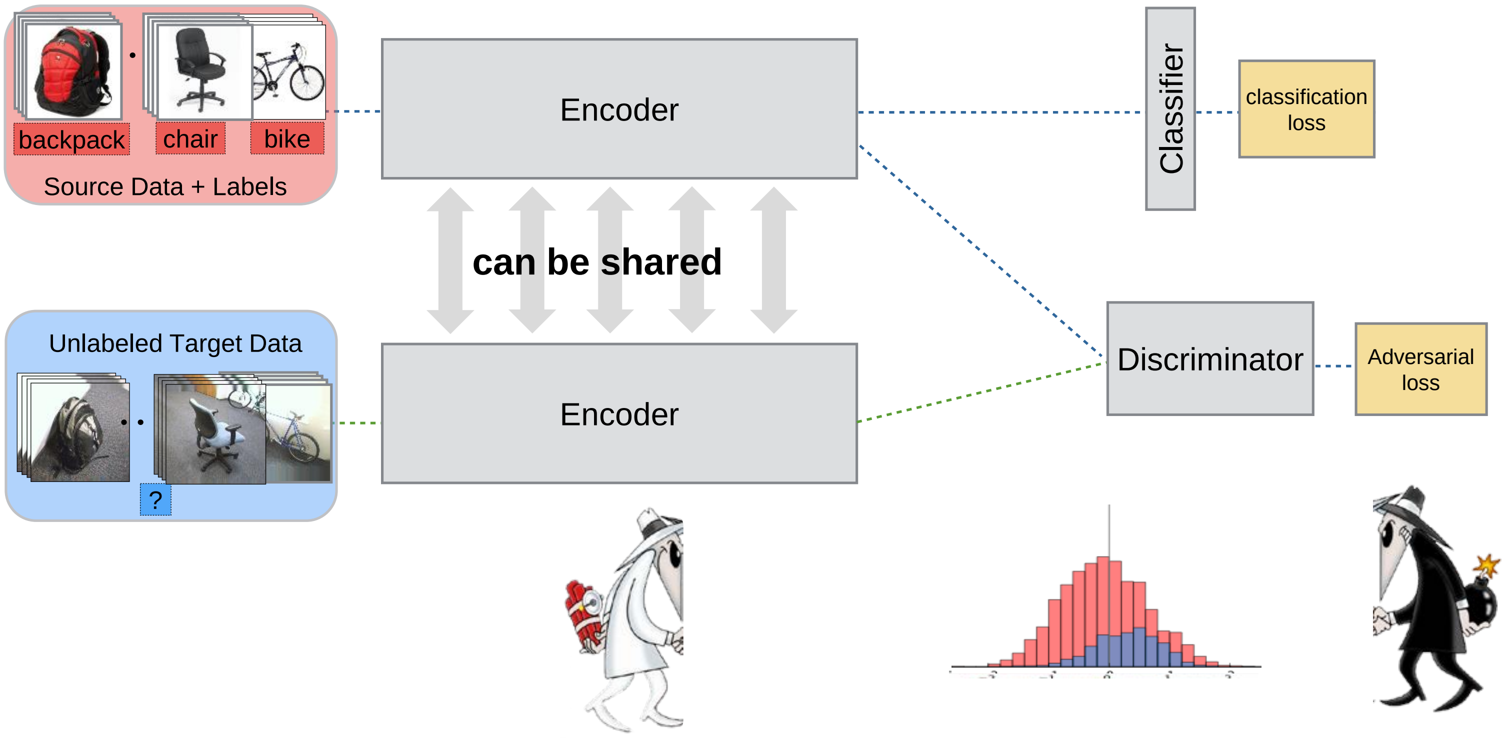
Adversarial domain adaptation



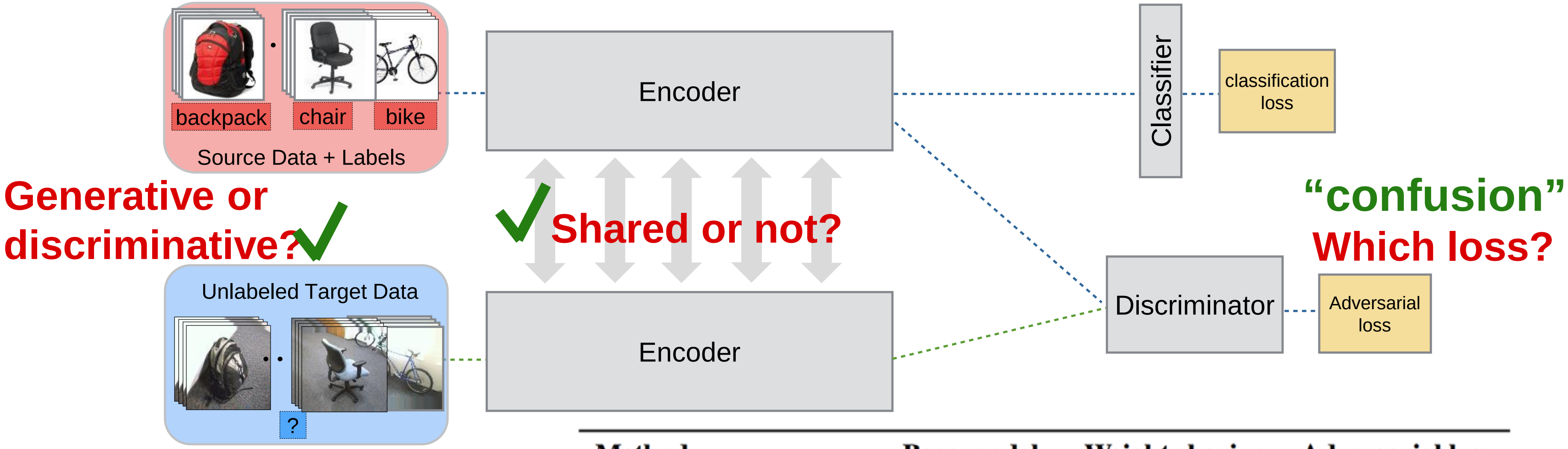
Adversarial domain adaptation



Adversarial domain adaptation



Design choices in adversarial adaptation



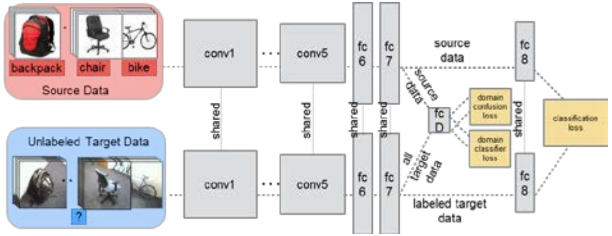
Method	Base model	Weight sharing	Adversarial loss
Gradient reversal [16]	discriminative	shared	minimax
Domain confusion [12]	discriminative	shared	confusion
CoGAN [13]	generative	unshared	GAN

[13] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks, NIPS 2016

Deep domain confusion

[Tzeng ICCV15]

Train a network to minimize classification loss AND confuse two domains



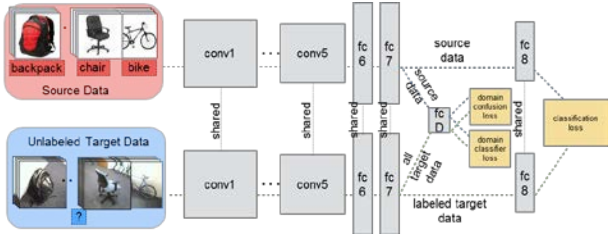
$$\min_{\theta_D} \mathcal{L}_D(x_S, x_T, \theta_{\text{repr}}; \theta_D)$$

$$\min_{\theta_{\text{repr}}} \mathcal{L}_{\text{conf}}(x_S, x_T, \theta_D; \theta_{\text{repr}}).$$

Deep domain confusion

[Tzeng ICCV15]

Train a network to minimize classification loss AND confuse two domains



source inputs · target inputs · network parameters (fixed) · domain classifier (learn) · domain classifier loss

$$\mathcal{L}_D(x_S, x_T, \theta_{\text{repr}}; \theta_D) = -\sum_d \mathbb{1}[y_D = d] \log q_d$$

domain classifier prediction

$$q = \text{softmax}(\theta_D^T f(x; \theta_{\text{repr}})) = p(y_D = 1|x)$$

iterate

$$\min_{\theta_D} \mathcal{L}_D(x_S, x_T, \theta_{\text{repr}}; \theta_D)$$

$$\min_{\theta_{\text{repr}}} \mathcal{L}_{\text{conf}}(x_S, x_T, \theta_D; \theta_{\text{repr}}).$$

domain classifier (fixed) · network parameters (learn) · domain confusion loss

$$\mathcal{L}_{\text{conf}}(x_S, x_T, \theta_D; \theta_{\text{repr}}) = -\sum_d \frac{1}{D} \log q_d$$

(cross-entropy with uniform distribution)

What is a good adversarial loss function?

Confusion loss [Tzeng 2015]

$$\max_{D} \mathbb{E}_{\mathbf{x}\sim p_S(\mathbf{x})}\left[\log D(M_S(\mathbf{x}))\right] + \mathbb{E}_{\mathbf{x}\sim p_T(\mathbf{x})}\left[\log(1 - D(M_T(\mathbf{x})))\right]$$

$$\max_{M_S, M_T} \sum_{d\in\{S,T\}} \mathbb{E}_{\mathbf{x}\sim p_d(\mathbf{x})}\left[\frac{1}{2}\log D(M_d(\mathbf{x})) + \frac{1}{2}\log(1 - D(M_d(\mathbf{x})))\right]$$

Minimax loss [Ganin 2015]

$$\min_{M_S, M_T} \max_{D} V(D, M_S, M_T) = \mathbb{E}_{\mathbf{x}\sim p_S(\mathbf{x})}\left[\log D(M_S(\mathbf{x}))\right] + \mathbb{E}_{\mathbf{x}\sim p_T(\mathbf{x})}\left[\log(1 - D(M_T(\mathbf{x})))\right]$$

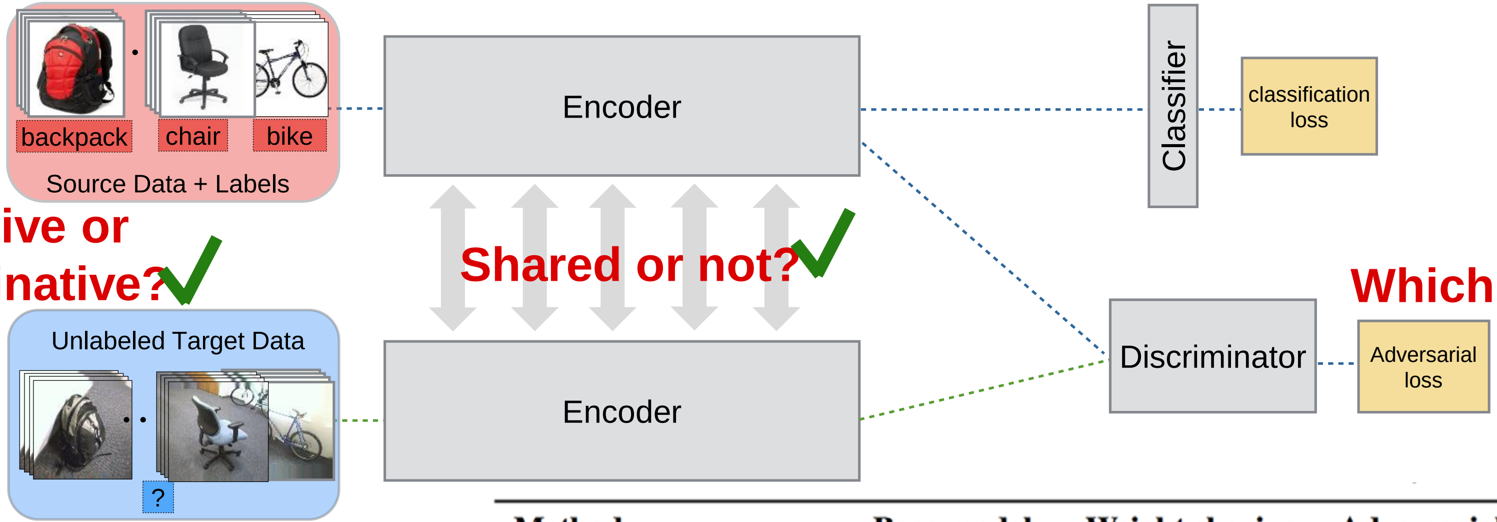
GAN loss [Goodfellow 2014]

$$\max_{D} \mathbb{E}_{\mathbf{x}\sim p_S(\mathbf{x})}\left[\log D(M_S(\mathbf{x}))\right] + \mathbb{E}_{\mathbf{x}\sim p_T(\mathbf{x})}\left[\log(1 - D(M_T(\mathbf{x})))\right]$$

$$\max_{M_T} \mathbb{E}_{\mathbf{x}\sim p_T(\mathbf{x})}\left[\log D(M_T(\mathbf{x}))\right].$$

“stronger gradients”

Adversarial Discriminative Domain Adaptation (ADDA) [Tzeng CVPR17]



Generative or discriminative? ✓

Shared or not? ✓

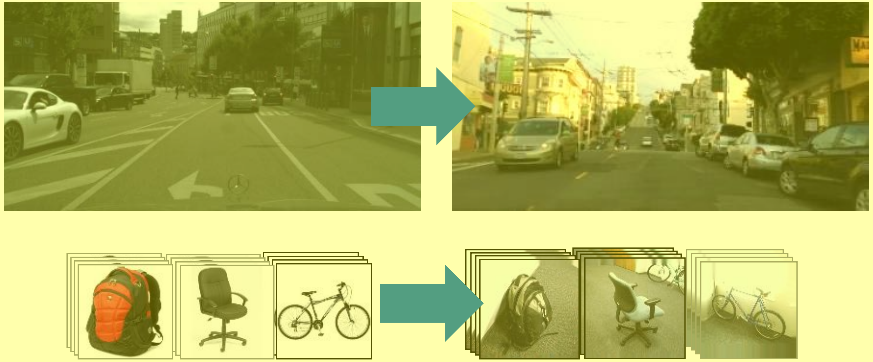
GAN

Which loss?

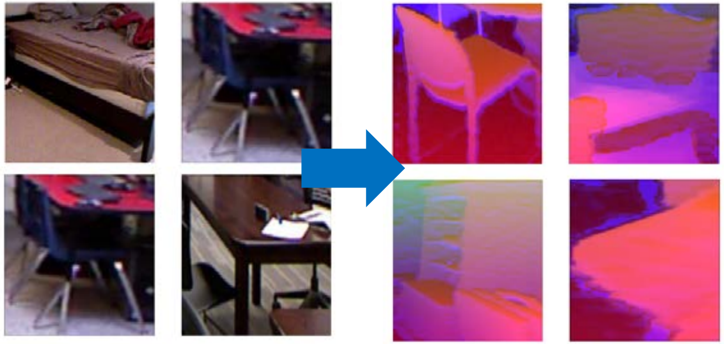
Method	Base model	Weight sharing	Adversarial loss
Gradient reversal [16]	discriminative	shared	minimax
Domain confusion [12]	discriminative	shared	confusion
CoGAN [13]	generative	unshared	GAN
➡ ADDA	discriminative	unshared	GAN

Applications to different types of domain shift

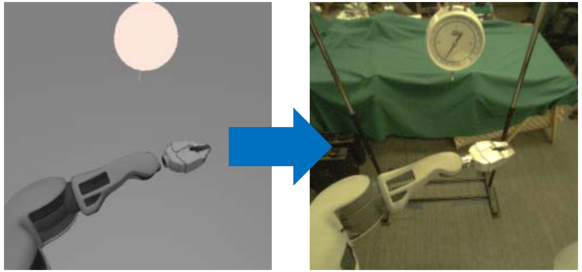
From dataset to dataset



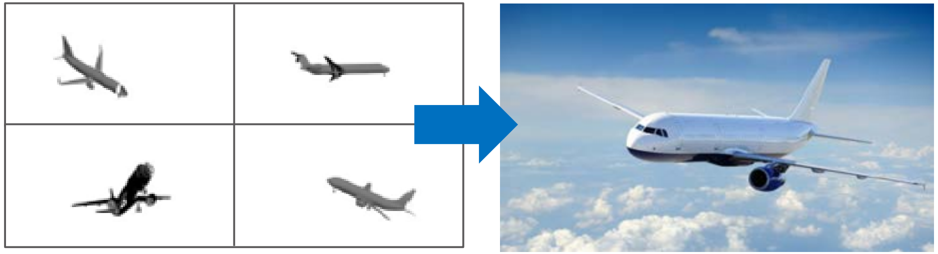
From RGB to depth



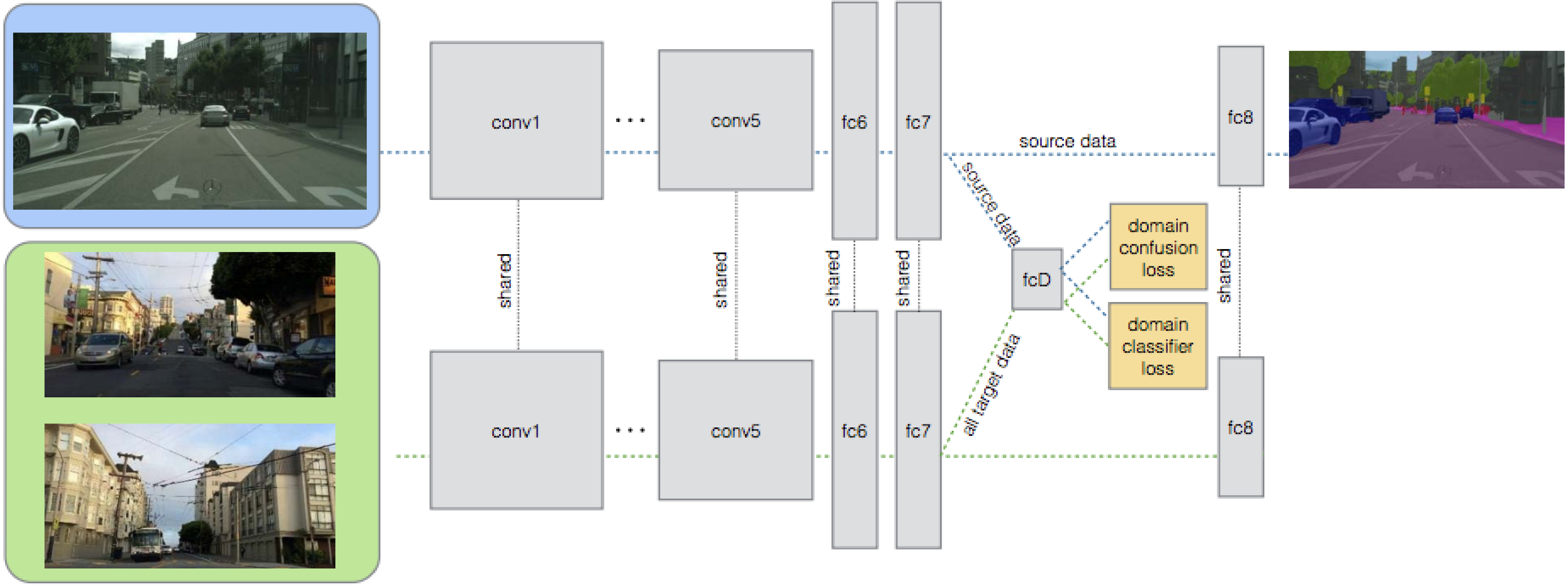
From simulated to real control



From CAD models to real images



Fully Convolutional Network with Domain Confusion Loss [Hoffman 2016]



FCNs in the Wild: Pixel-level Adversarial and Constraint-based Adaptation, Judy Hoffman, Dequan Wang, Fisher Yu, Trevor Darrell, Arxiv 2016

Results on Cityscapes to SF adaptation [Hoffman 2016]

Before domain confusion

After domain confusion

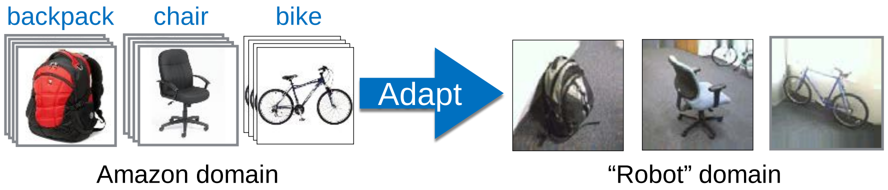
FCNs in the Wild: Pixel-level Adversarial and Constraint-based Adaptation, Judy Hoffman, Dequan Wang, Fisher Yu, Trevor Darrell, Arxiv 2016

ADDA: Adaptation on digits

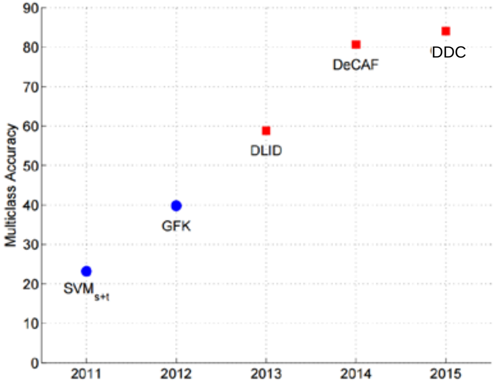
[Tzeng CVPR17]

Method	MNIST → USPS	USPS → MNIST	SVHN → MNIST
Source only	0.752 ± 0.016	0.571 ± 0.017	0.601 ± 0.011
Gradient reversal	0.771 ± 0.018	0.730 ± 0.020	0.739 [16]
Domain confusion	0.791 ± 0.005	0.665 ± 0.033	0.681 ± 0.003
CoGAN	0.912 ± 0.008	0.891 ± 0.008	did not converge
ADDA (Ours)	0.894 ± 0.002	0.901 ± 0.008	0.760 ± 0.018

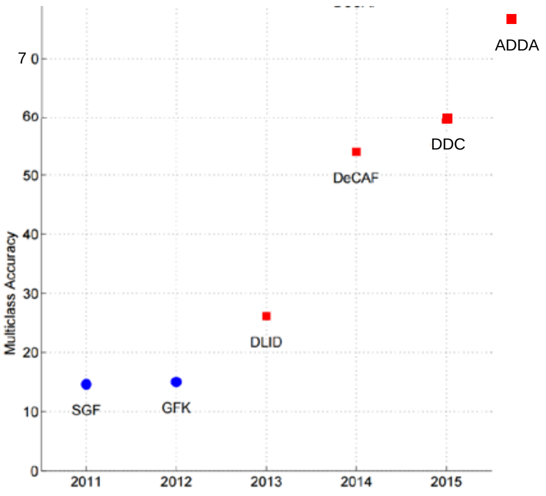
Office dataset: historical progress



ADDA



(a) A→W supervised adaptation



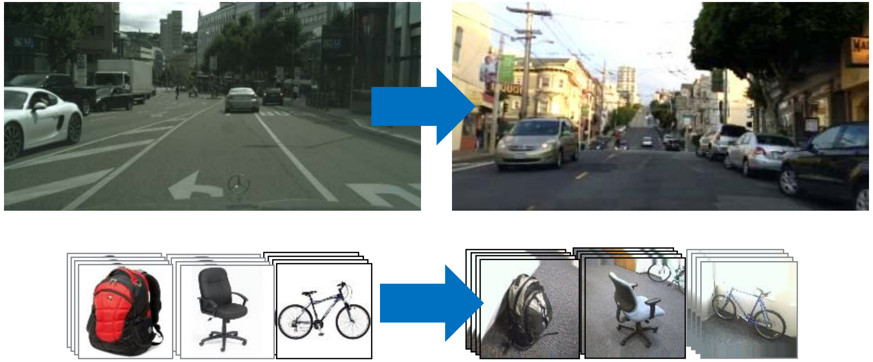
(b) A→W unsupervised adaptation

Unsupervised adaptation in 2016/2017

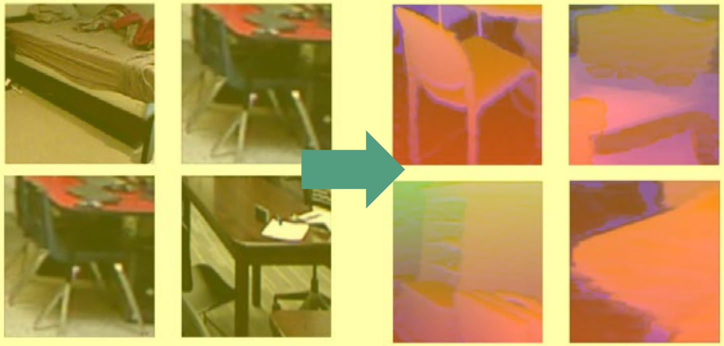
Method	$A \rightarrow W$
DDC (Tzeng et al., 2014)	0.618
DAN (Long & Wang, 2015)	0.685
DRCN (Ghifary et al., 2016)	0.687
DANN (Ganin et al., 2016)	0.730
ADDA (Ours)	**0.751**

Applications to different types of domain shift

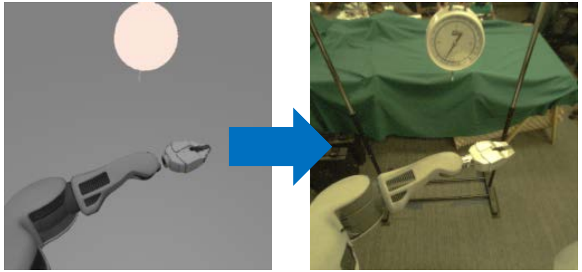
From dataset to dataset



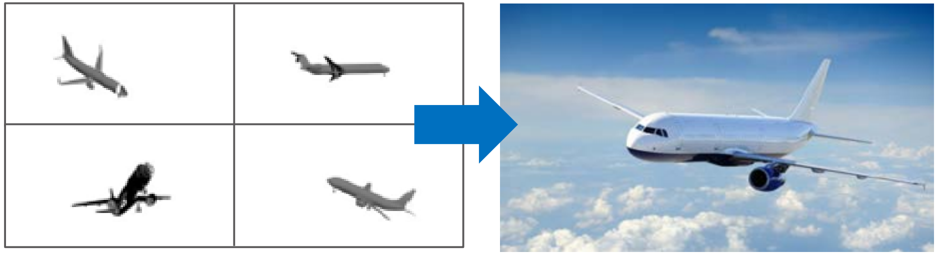
From RGB to depth



From simulated to real control



From CAD models to real images

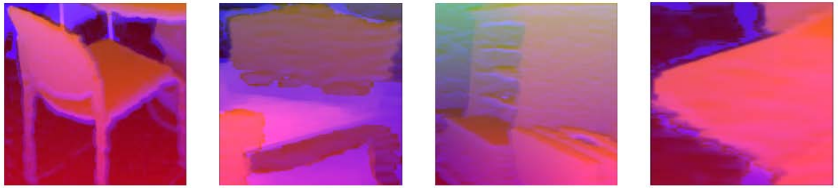


ADDA: Adaptation on RGB-D

[Tzeng CVPR17]

Train on RGB

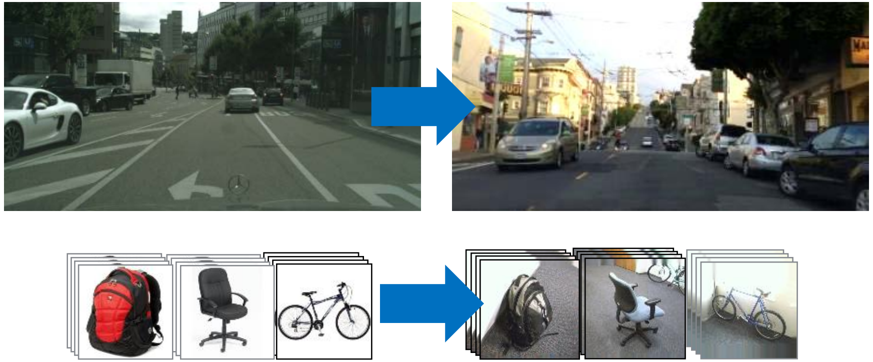
Test on depth



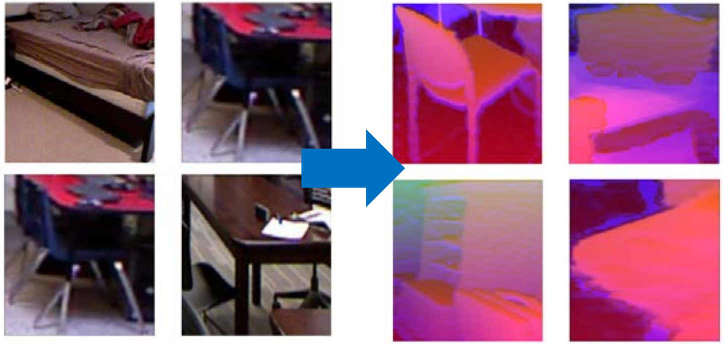
	bathtub	bed	bookshelf	box	chair	counter	desk	door	dresser	garbage bin	lamp	monitor	night stand	pillow	sink	sofa	table	television	toilet	**overall**
# of instances	19	96	87	210	611	103	122	129	25	55	144	37	51	276	47	129	210	33	17	2401
Source only	0.000	0.010	0.011	0.124	0.188	0.029	0.041	0.047	0.000	0.000	0.069	0.000	0.039	0.587	0.000	0.008	0.010	0.000	0.000	0.139
ADDA (Ours)	0.000	0.146	0.046	0.229	0.344	0.447	0.025	0.023	0.000	0.018	0.292	0.081	0.020	0.297	0.021	0.116	0.143	0.091	0.000	0.211
Train on target	0.105	0.531	0.494	0.295	0.619	0.573	0.057	0.636	0.120	0.291	0.576	0.189	0.235	0.630	0.362	0.248	0.357	0.303	0.647	0.468

Not covered today: simulation-to-real shifts

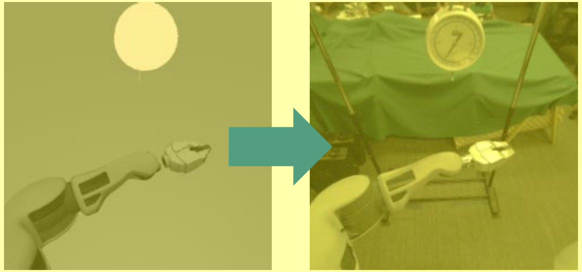
From dataset to dataset



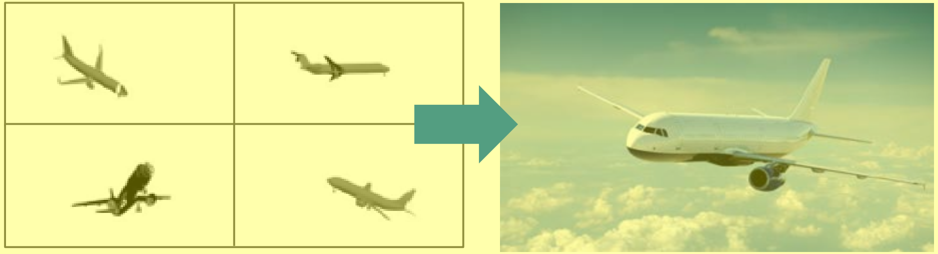
From RGB to depth



From simulated to real control



From CAD models to real images



Thank you

References

- Eric Tzeng, Judy Hoffman, Trevor Darrell, Kate Saenko, Simultaneous Deep Transfer Across Domains and Tasks, ICCV 2015
- Eric Tzeng, Coline Devin, Judy Hoffman, Chelsea Finn, Pieter Abbeel, Sergey Levine, Kate Saenko, Trevor Darrell, Adapting Deep Visuomotor Representations with Weak Pairwise Constraints, WAFR 2016
- Baochen Sun, Jiashi Feng, Kate Saenko, Return of Frustratingly Easy Domain Adaptation, AAAI 2016
- Baochen Sun, Kate Saenko, Deep CORAL: Correlation Alignment for Deep Domain Adaptation, TASK-CV Workshop at ICCV 2016
- Eric Tzeng, Judy Hoffman, Trevor Darrell, Kate Saenko, Adversarial Discriminative Domain Adaptation, accepted to CVPR 2017
- Synthetic to Real Adaptation with Deep Generative Correlation Alignment Networks, arxiv.org