CSE 252D: Advanced Computer Vision

Manmohan Chandraker

Lecture 18: Domain Adaptation

Virtual classrooms

- Virtual lectures on Zoom
  - Only host shares the screen
  - Keep video off and microphone muted
  - But please do speak up (remember to unmute!)
  - Slides uploaded on webpage just before class
- Virtual interactions on Zoom
  - Ask and answer plenty of questions
  - “Raise hand” feature on Zoom when you wish to speak
  - Post questions on chat window
  - Happy to try other suggestions!
- Lectures recorded and upload on Canvas
  - Available under “My Media” on Canvas

Overall goals for the course

- Introduce the state-of-the-art in computer vision
- Study principles that make them possible
- Get understanding of tools that drive computer vision
- Enable one or all of several such outcomes
  - Pursue higher studies in computer vision
  - Join industry to do cutting-edge work in computer vision
  - Gain appreciation of modern computer vision technologies
- This is a great time to study computer vision!

Papers for Wed, Jun 2

- Playing for Benchmarks
  - https://arxiv.org/abs/1709.07322
- Domain Adaptive Faster R-CNN for Object Detection in the Wild
  - https://arxiv.org/abs/1803.03243
- Fully Convolutional Adaptation Networks for Semantic Segmentation
  - https://arxiv.org/abs/1804.08286
- Label Efficient Learning of Transferable Representations across Domains and Tasks
  - https://arxiv.org/abs/1712.00123

Papers for Fri, Jun 4

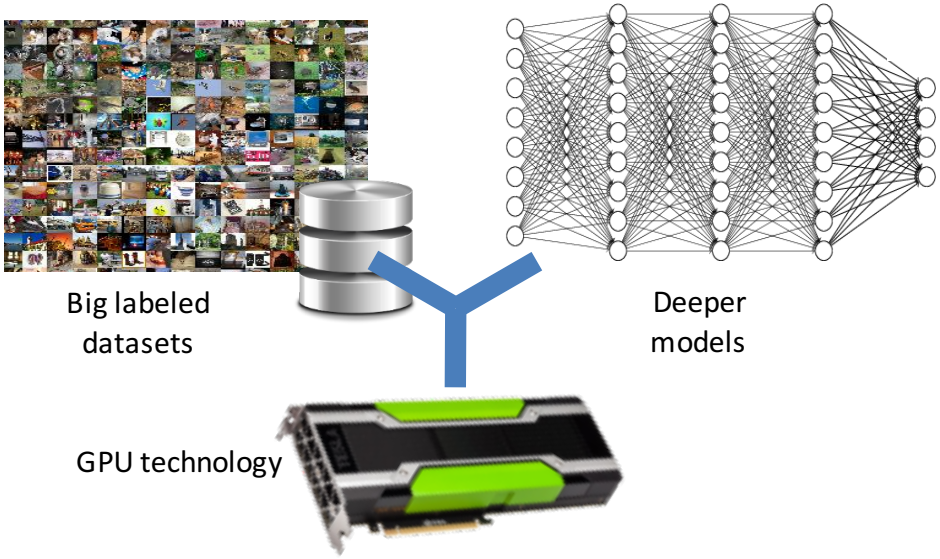
- A Simple Framework for Contrastive Learning of Visual Representations
  - https://arxiv.org/abs/2002.05709
- Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation
  - https://arxiv.org/abs/1805.09806
- Learning to Cluster Faces on an Affinity Graph
  - https://arxiv.org/abs/1904.02749
- UPSNet: A Unified Panoptic Segmentation Network
  - https://arxiv.org/abs/1901.03784
- Towards Universal Object Detection by Domain Attention
  - https://arxiv.org/abs/1904.04402

Announcements

- Final exam: Fri, Jun 11, 7-9:30pm
  - Held “in-class”, on Zoom
  - If time does not work for you, contact instructor by Wed, Jun 2
- Require you to apply or generalize concepts from class
  - Content: topics covered in lectures and lightning presentations
  - Exam will be open notes, can refer any material
  - Keep pen and paper handy
  - Only restriction: no communication with another human ☺
- Sample final questions posted on Piazza
  - Some of those questions *will* be in the actual final!
- **Very flexible, let me know any accommodations**
  - Should enjoy the final, celebrate what we learned

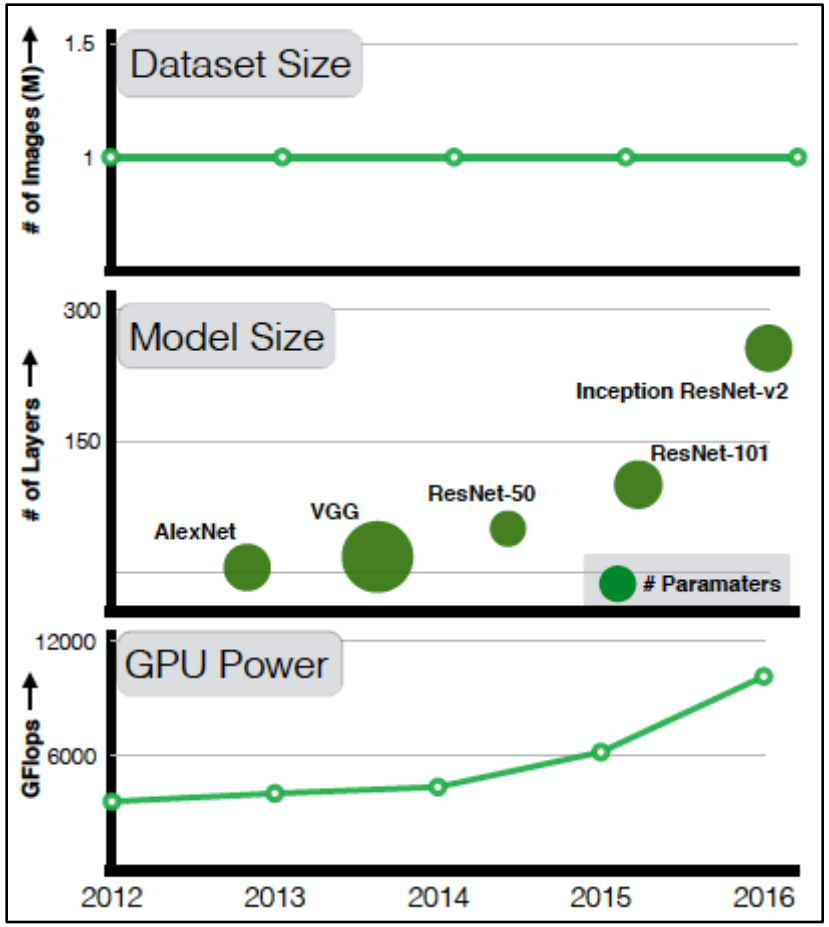
Domain Adaptation

Key Challenge for Deep Learning in Vision



Biggest limiting factor: Lack of **labeled** data.

Domain adaptation: Learn with **unlabeled** data!

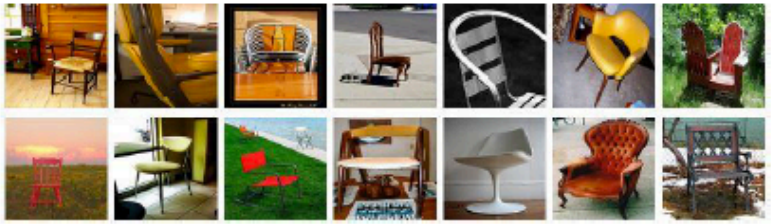


[Sun et al., CVPR 2017]

Train…Learn…Test

PubFig: Public Figures Face Database

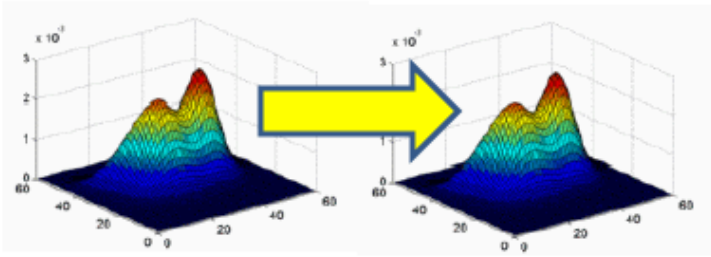
IMAGENET

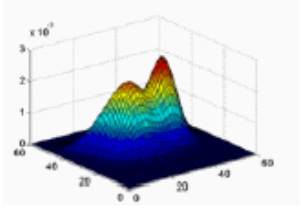


Indoor Scene Recognition

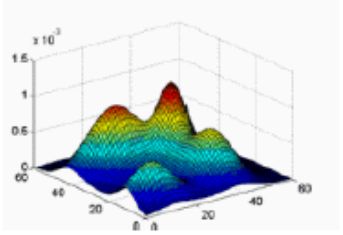
Hollywood Human Actions dataset

- people, faces
- chair
- tables
- monitor
- book
- scene: office, lab
- action: sitting, talking



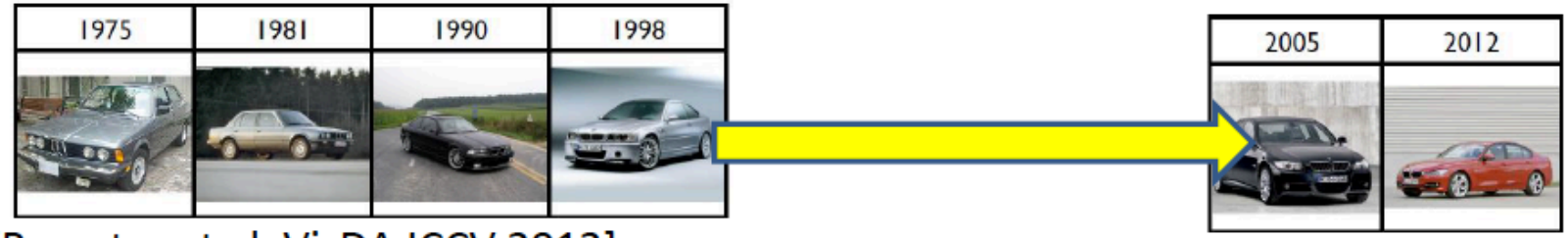


Real Scenarios



[Saenko et al. , ECCV 2010]

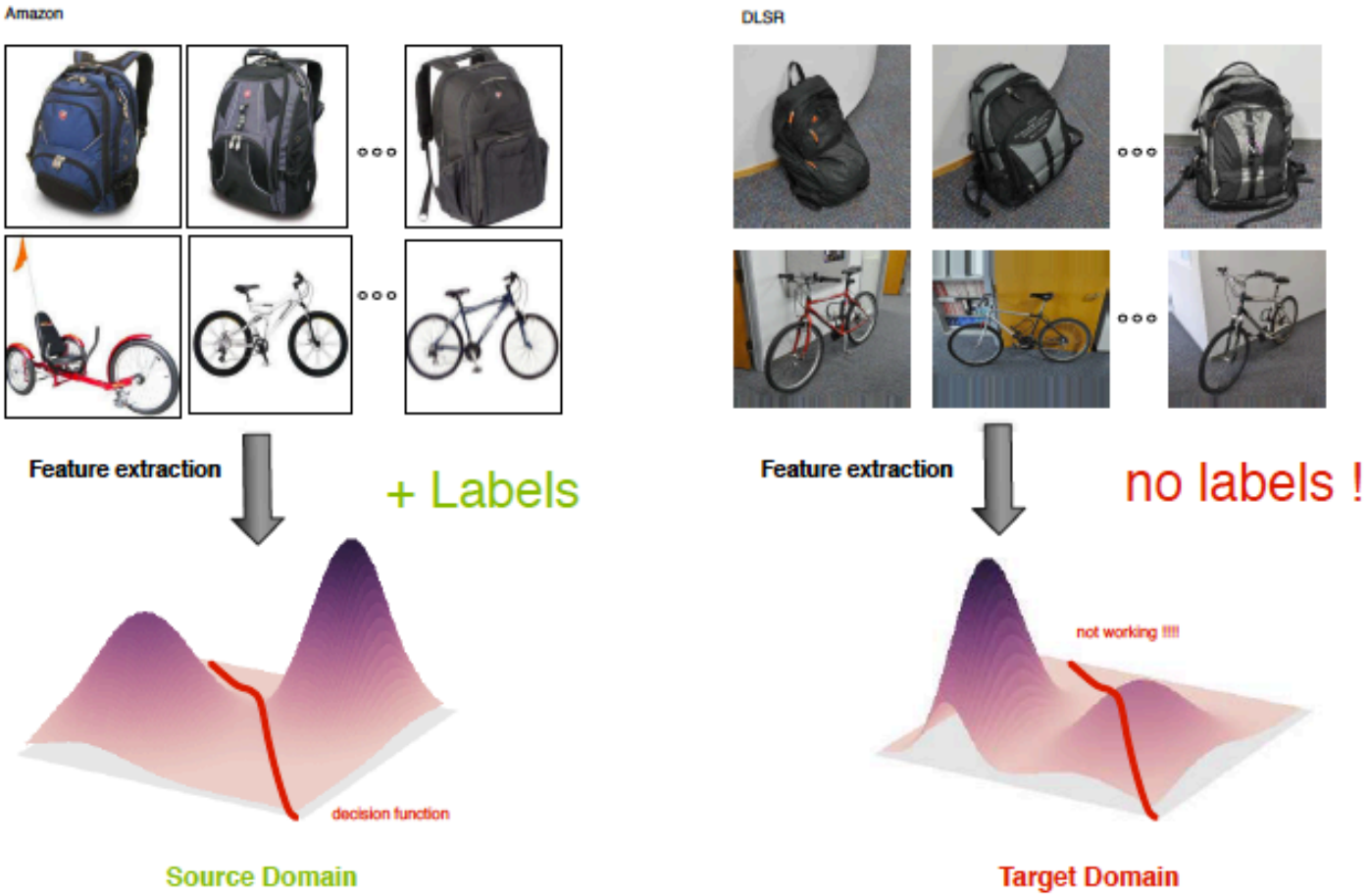
[Torralba, Efros, CVPR 2011]



[Rematas et al, VisDA ICCV 2013]

[Tommasi et al, CVPR 2010]

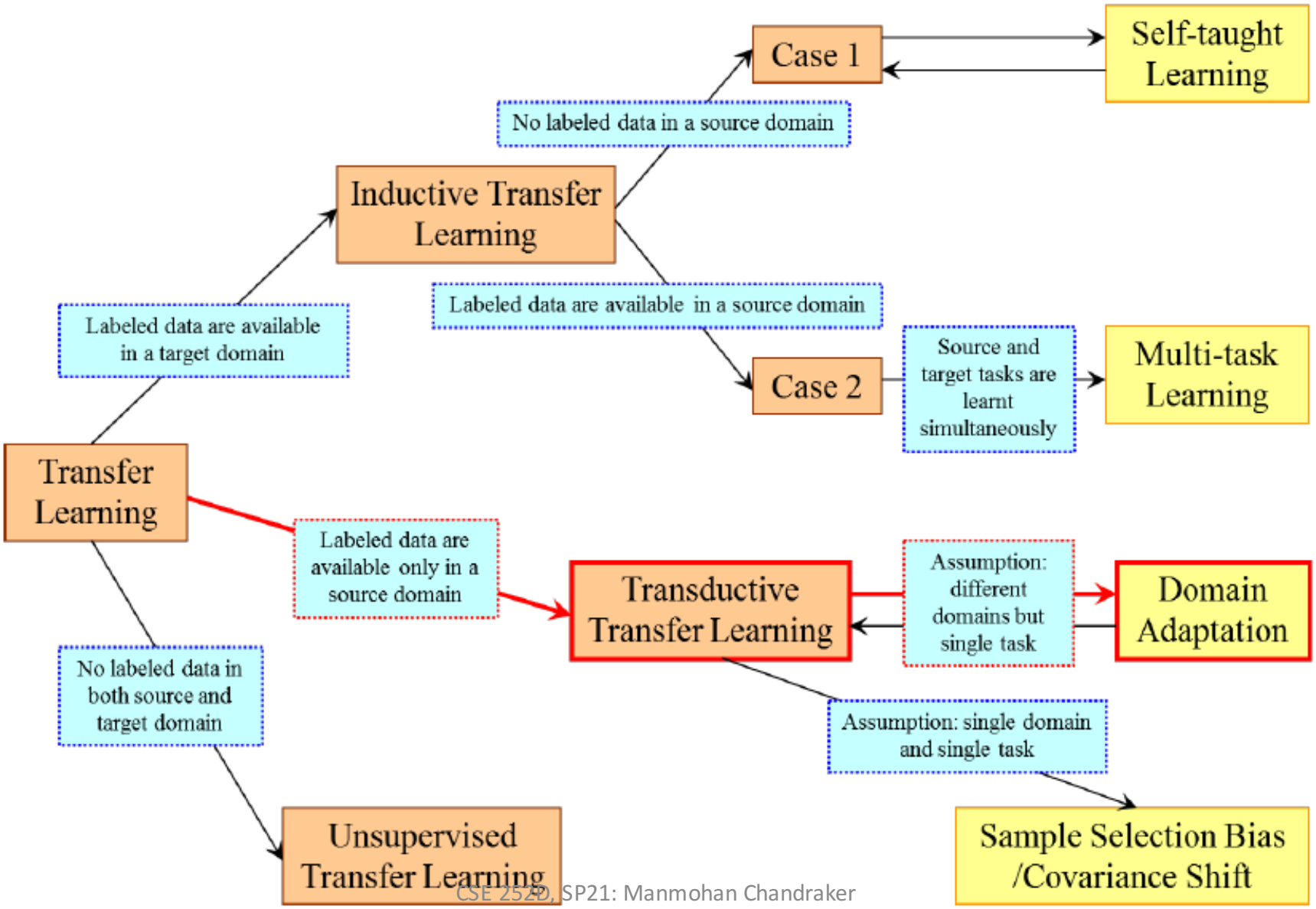
Train on Source, Test on Target



Challenge of domain adaptation:

- Labels only in source domain, classification conducted in target domain
- Classifier trained in source not applicable in target, due to distribution shift

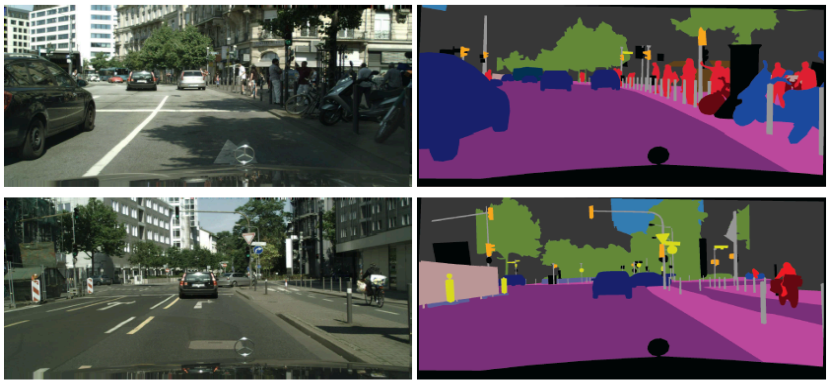
Transfer Learning and Adaptation



Advantages of Domain Adaptation

Advantage 1: Robustness of Solution

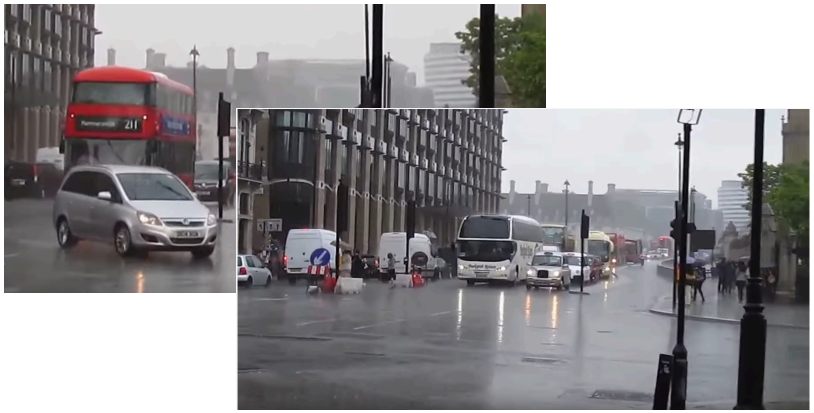
Source domain: good weather, **with** labels



Labels require **1.5 hours** per image!

Train on source, **apply** on target

Target domain: rainy weather, **no** labels

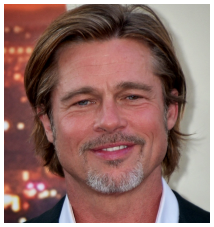


Train on source, **adapt** to target

Advantage 2: Ease of Deployment

Significant expense and time to label data for each new scenario

Existing domain



America

Collect data → Label data → Train model → Deploy

1 week | **1 year** | 1 week

New domain

Asia

Collect data → Label data → Train model → Deploy

1 week | **1 year** | 1 week

New domain

Africa

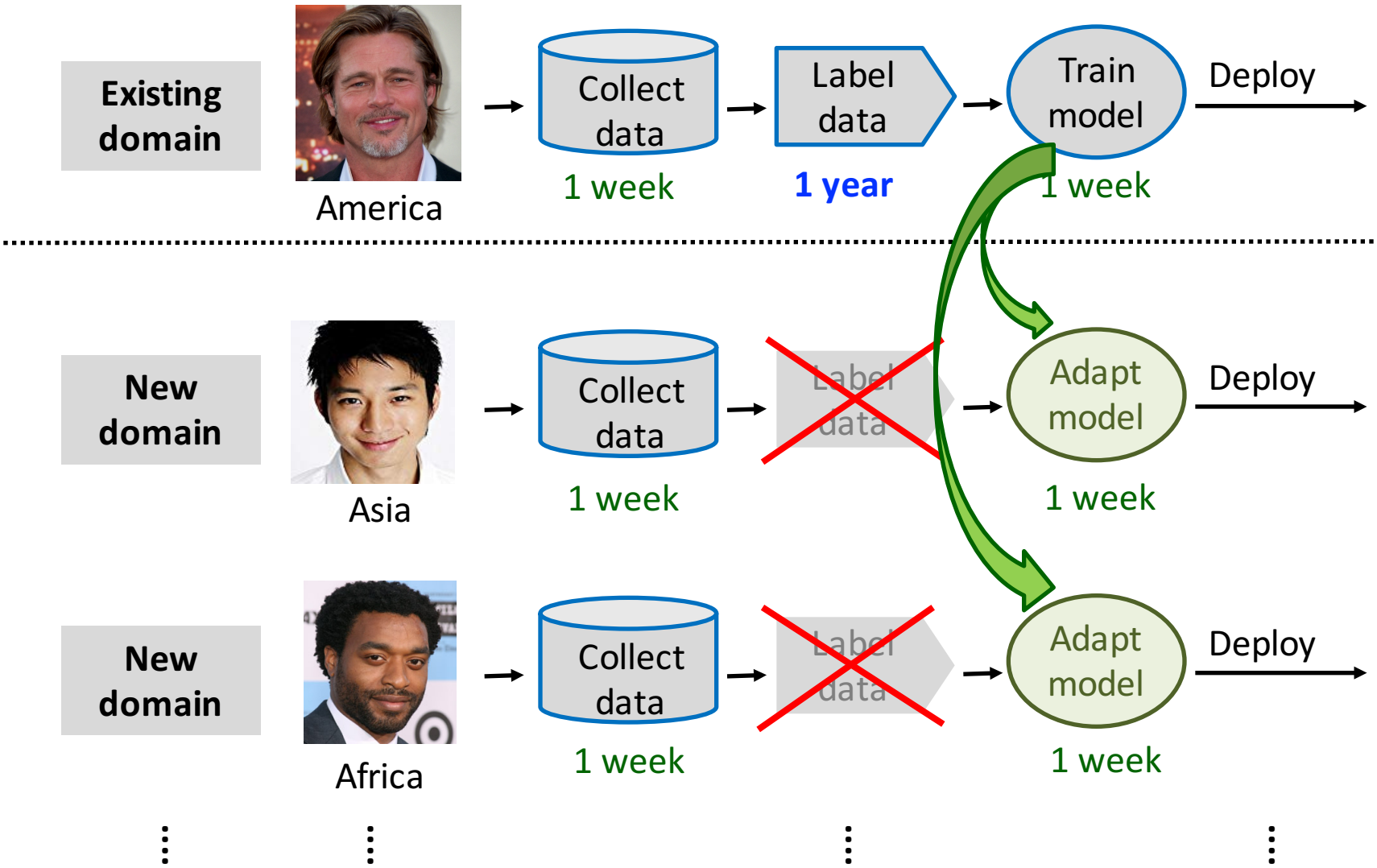
Collect data → Label data → Train model → Deploy

1 week | **1 year** | 1 week

⋮ ⋮ ⋮ ⋮

Advantage 2: Ease of Deployment

Domain adaptation eases deployment by reducing need for data labeling

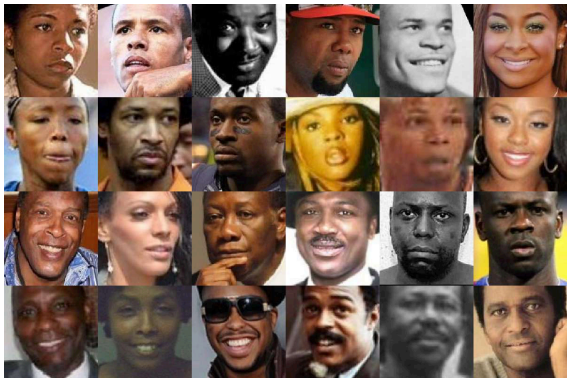


Advantage 3: Fairness to Society

Proportion in datasets: 80%

Caucasian

Proportion in datasets: 10%



African-American

Proportion in datasets: 5%

East-Asian

Training on biased data without domain adaptation		
High accuracy	Low accuracy 😕	Low accuracy 😕

Using domain adaptation to address dataset bias		
High accuracy	High accuracy 🙂	High accuracy 🙂

Advantage 3: Fairness to Society

CNN BUSINESS Markets Tech Media Success Perspectives Videos LIVE TV Edition

California lawmakers ban facial-recognition software from police body cams

By Rachel Metz, CNN Business

Updated 8:04 AM ET, Fri September 13, 2019

BBC Sign in News Sport Reel Worklife Travel Future

NEWS

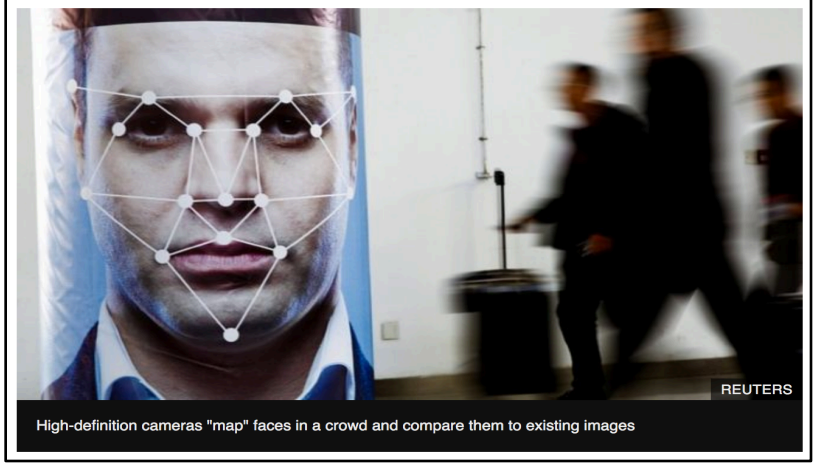
Home Video World US & Canada UK Business Tech Science Stories Enter

Technology

San Francisco is first US city to ban facial recognition

By Dave Lee
North America technology reporter

15 May 2019



High-definition cameras "map" faces in a crowd and compare them to existing images

Distribution Shifts

Source and Target Mismatch

$X \in \mathcal{X}$ Input variable to a learning system (i.e. observation)

$Y \in \mathcal{Y}$ Output variable to a learning system (i.e. label)

D^s, D^t

T^s, T^t

$P^s(X), P^t(X)$

$P^s(Y|X), P^t(Y|X)$

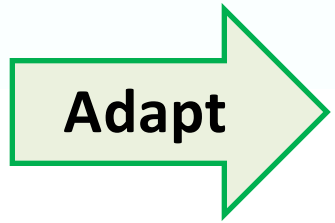
} Different for Source and Target Data

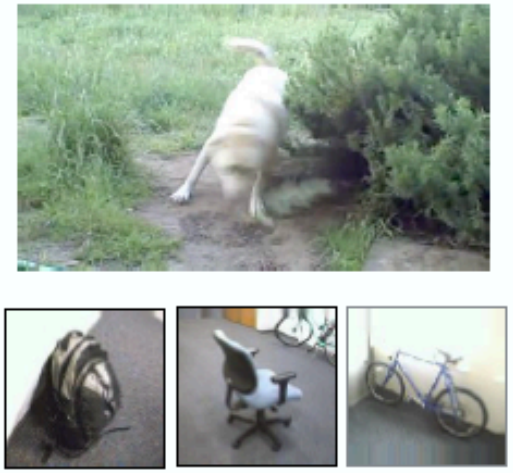
$D = \{\mathcal{X}, P(X)\}$ **Domain**: pair of feature space and marginal distribution.

$T = \{\mathcal{Y}, P(Y|X)\}$ **Task**: pair of label space and conditional probability.

$h(x) : X \to Y$ prediction function, classifier.

Train on Source, Test on Target





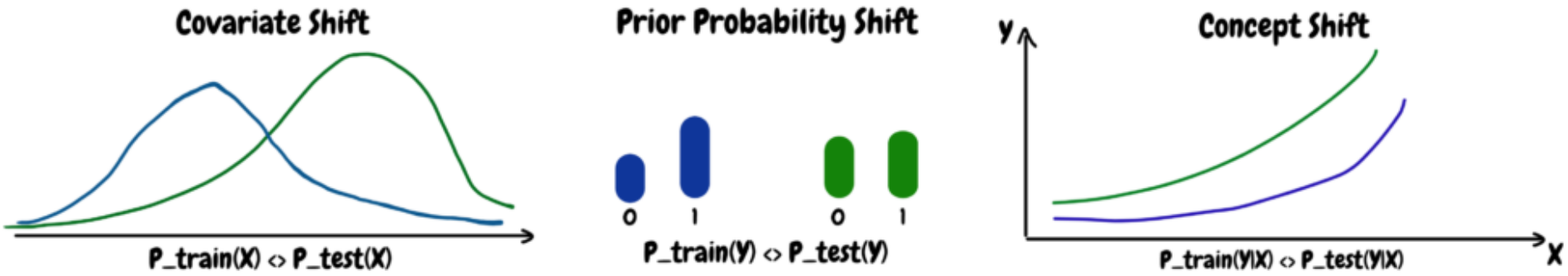
Source Domain $\sim P_S(X_S, Y_S)$

lots of **labeled** data

Target Domain $\sim P_T(X_T, Y_T)$

unlabeled or limited labels

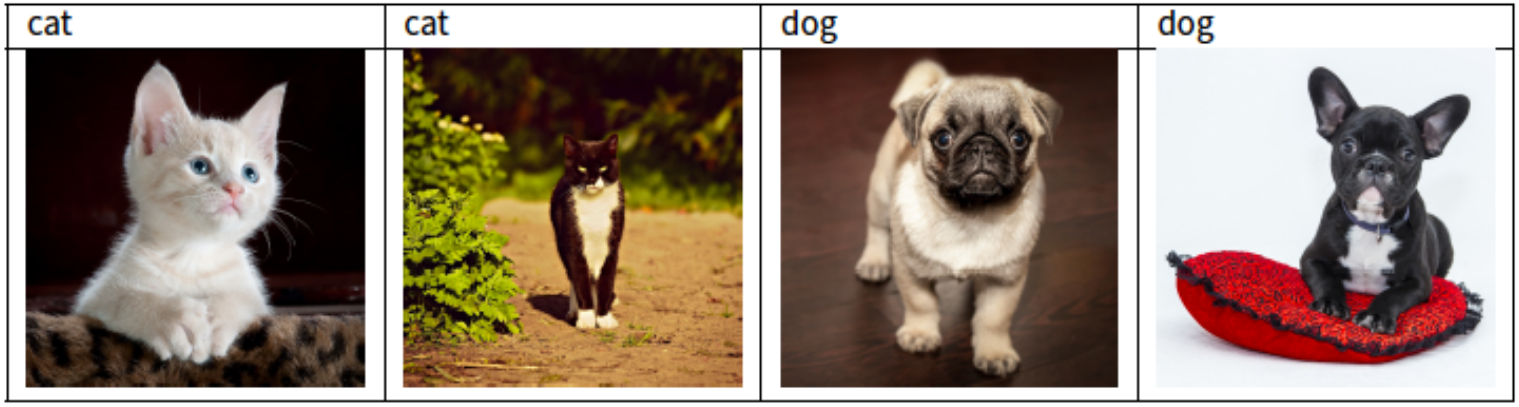
Different types of data shift



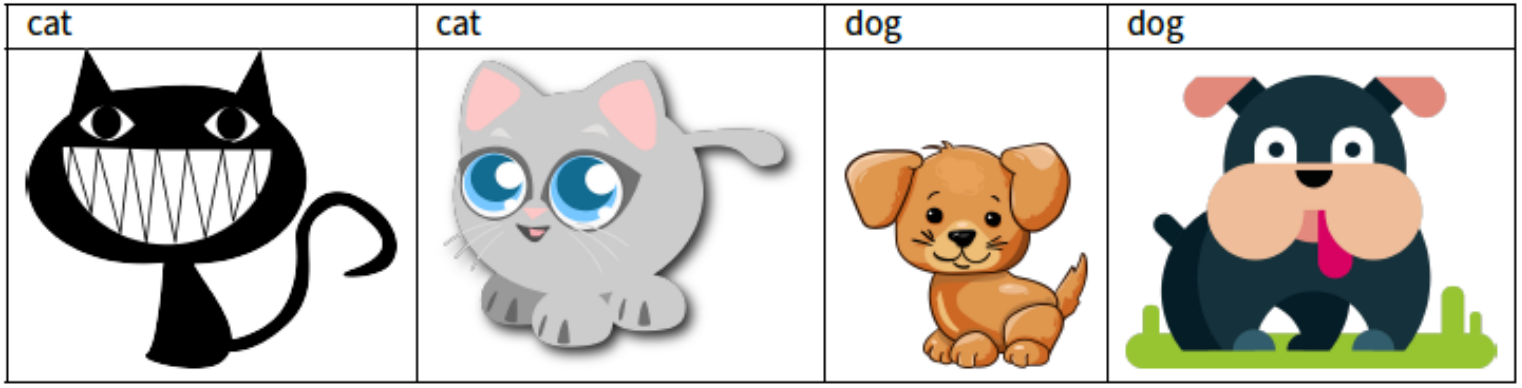
- Data distributions p(x, y) cannot change in arbitrary ways
  - Example: call "cats" as "dogs" and the other way round
- Adaptation possible under reasonable assumptions of data shift

Covariate Shift

Source



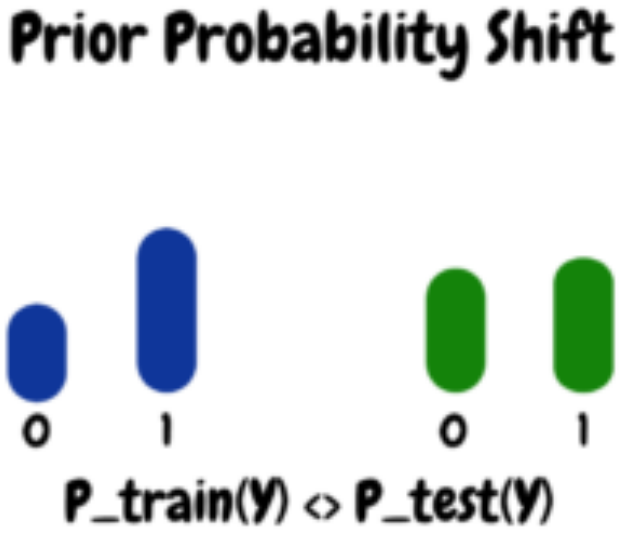
Target



- Input *data distributions* change, but the labeling function is unchanged
- $P^s(x) \neq P^t(x)$, but $P^s(y \mid x) = P^t(y \mid x)$
- Reasonable assumption when we believe x causes y
- S: British, T: American, x: the event that it rains, y: talking about weather

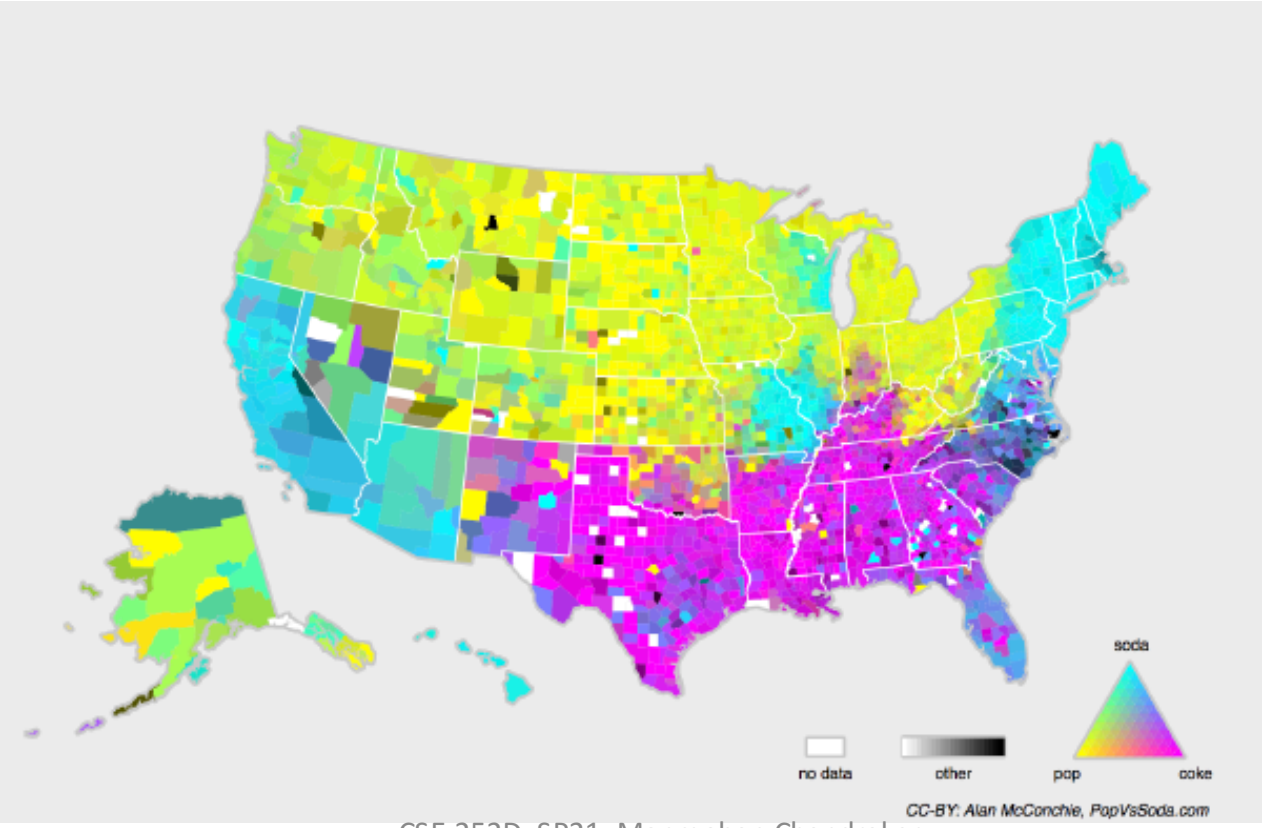
Label Shift

- Marginal distribution over labels changes, $P^s(y) \neq P^t(y)$
- Class conditionals remain unchanged, $P^s(x \mid y) = P^t(x \mid y)$
- Reasonable assumption when we believe y causes x
- Example: y is a diagnosis, x is the symptom
- Typically, when *classes unbalanced* across source and domain

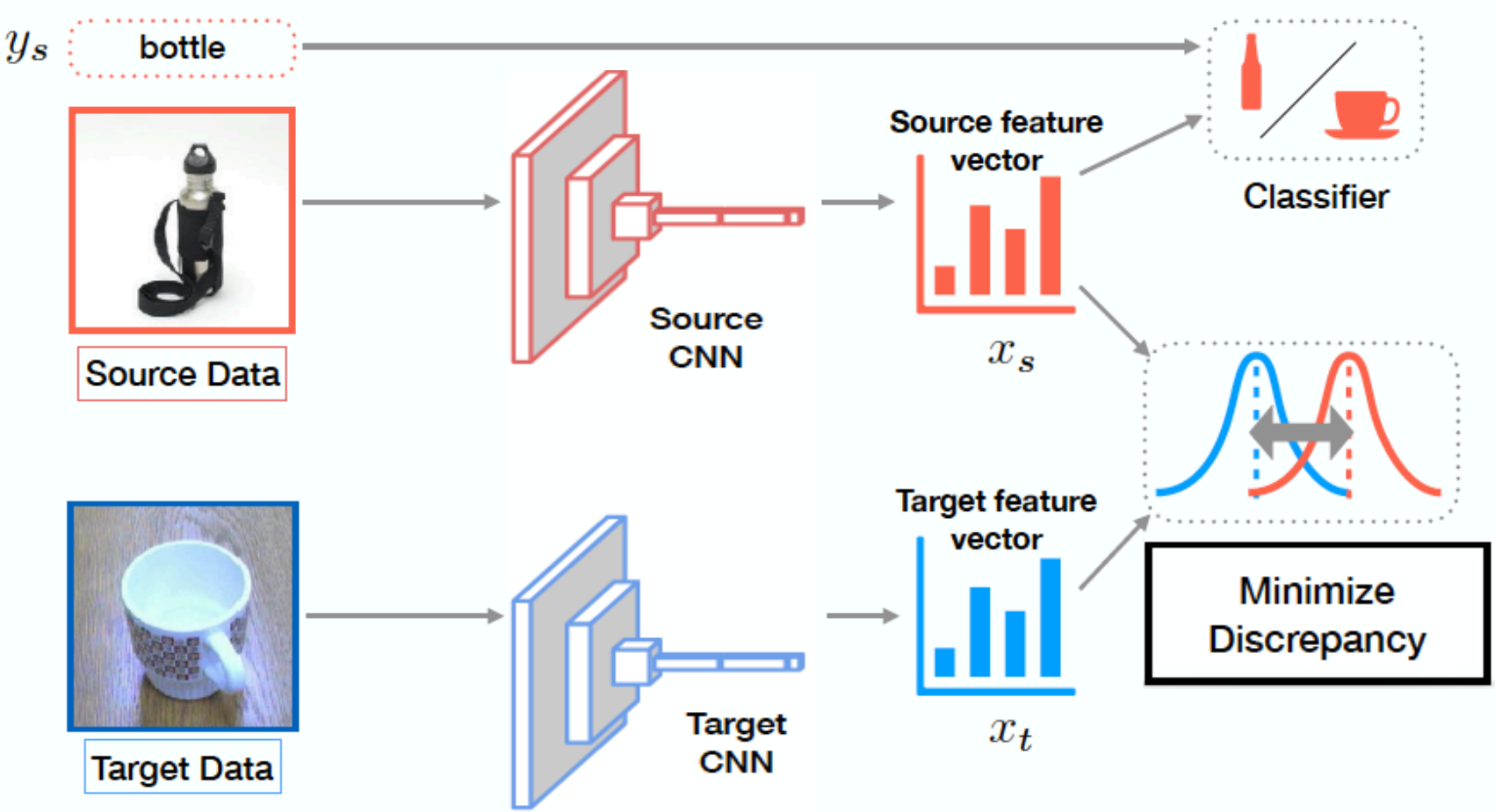


Concept Shift

- *Label definitions* change, $P^s(y \mid x) \neq P^t(y \mid x)$
- Example: x is {soda, pop, coke}, y is whether it is soft drink
- Can assume: P(y | x) changes slowly across domains



Deep Domain Adaptation



Learn a representation to minimize discrepancy

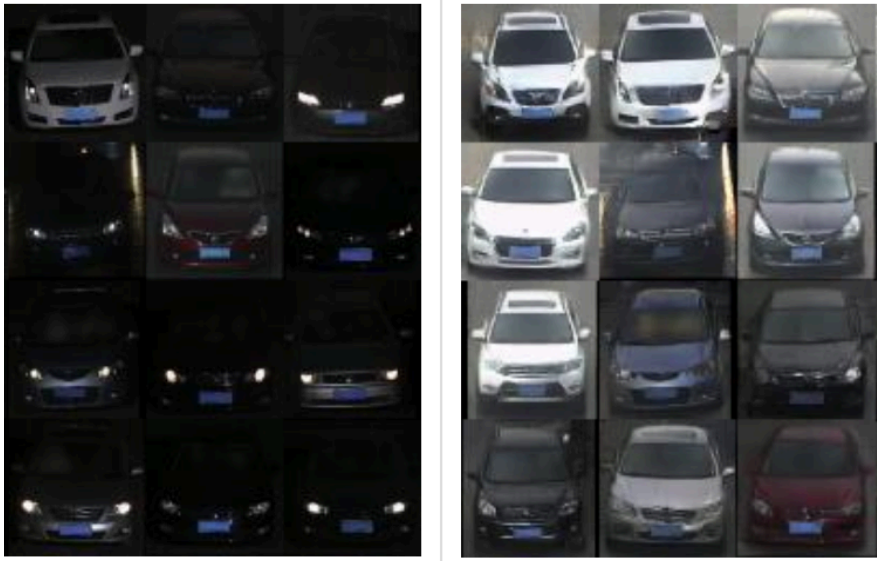
Adversarial Domain Adaptation

Domain adaptation for car recognition

Training domain

- Easy to collect labeled data
- Mostly day time images
- High-quality images
- Low camera elevation angle

Testing domain



- Hard to collect labeled data
- Both day and night time images
- Usually lower resolution images
- Usually high camera elevation angle

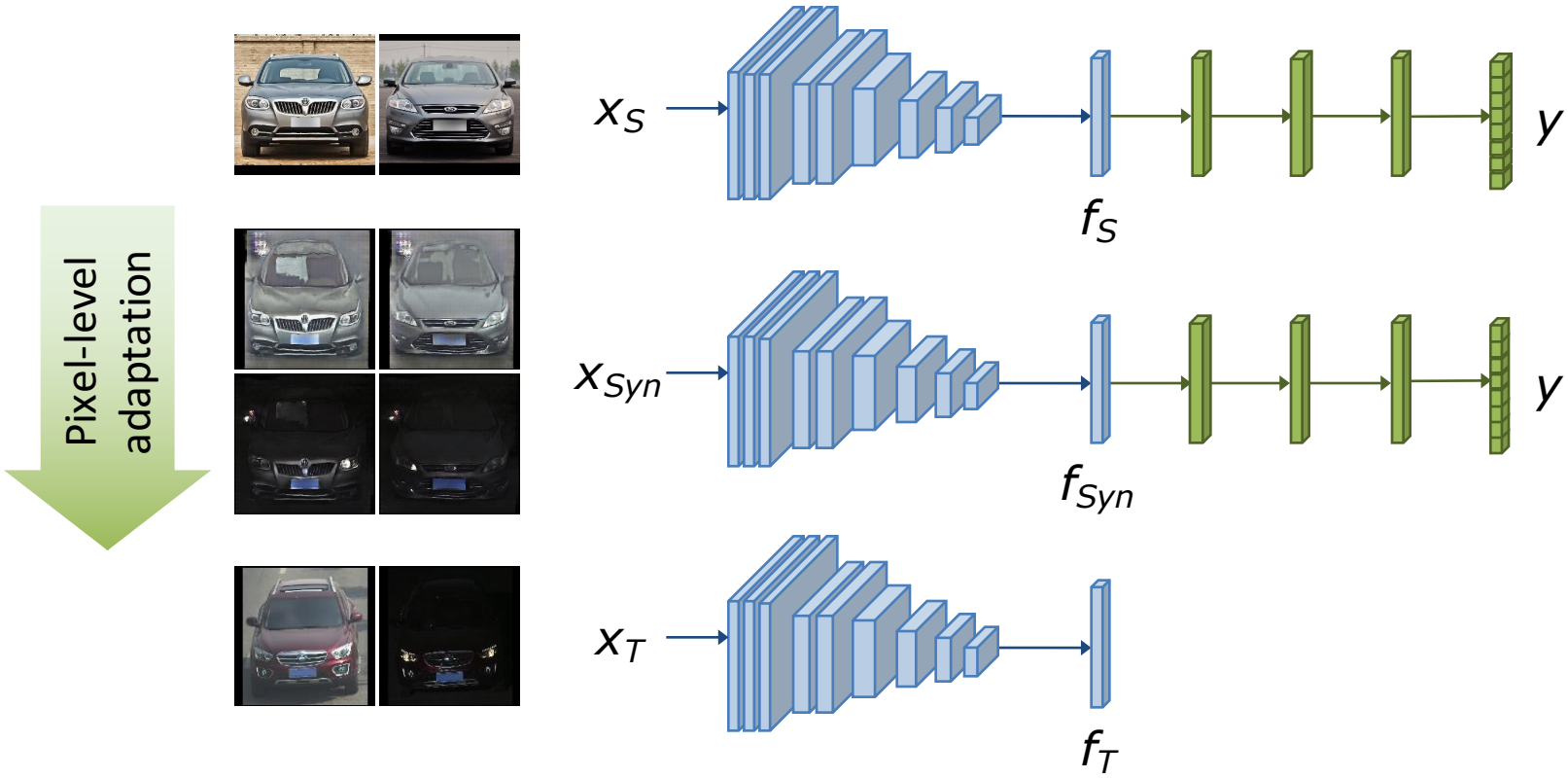
Achieving robustness to domain shift

Web: **96.4%** Day: **72.7%** Night: **19.9%**

Goal: close the gap without labeling effort

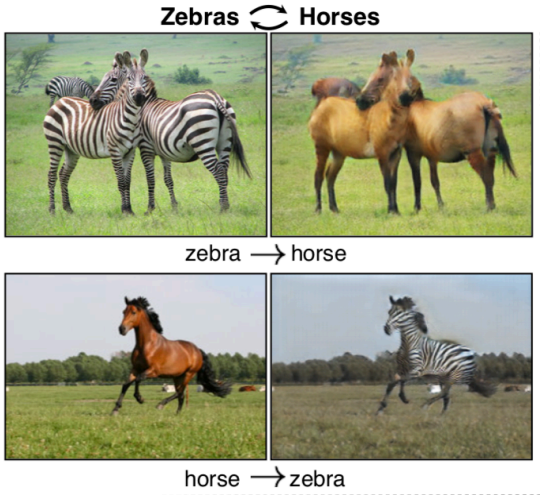
- Data-independent augmentation methods
  - Translation, horizontal flip, chromatic jittering, …
  - May not represent the statistics of target domain examples
- Data-driven methods: **domain adaptation**
  - Learning to *align* distributions between two domains

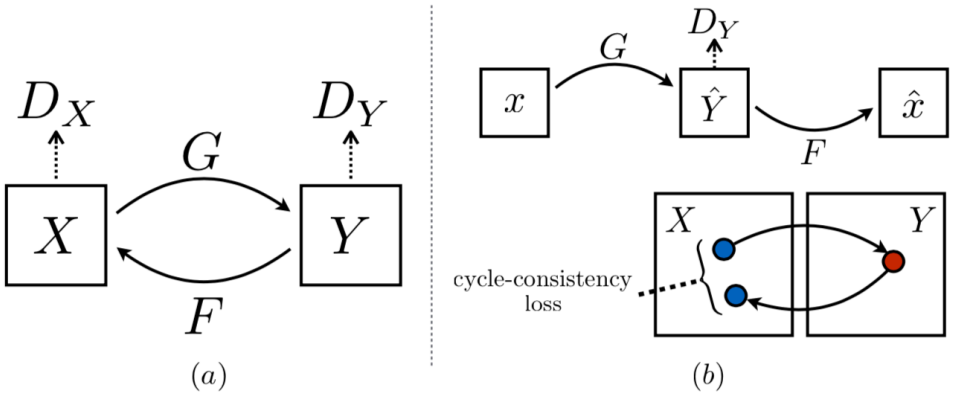
Learning to **align** distributions: pixel-level



Pixel-level adaptation by image synthesis

- General strategy
  - Translate images across domains using a generative neural network
  - Unpaired: no instance-level correspondence





Cycle-consistency and **skip-connection** allows to maintain geometric structure between images

Zhu et al., Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, ICCV 2017

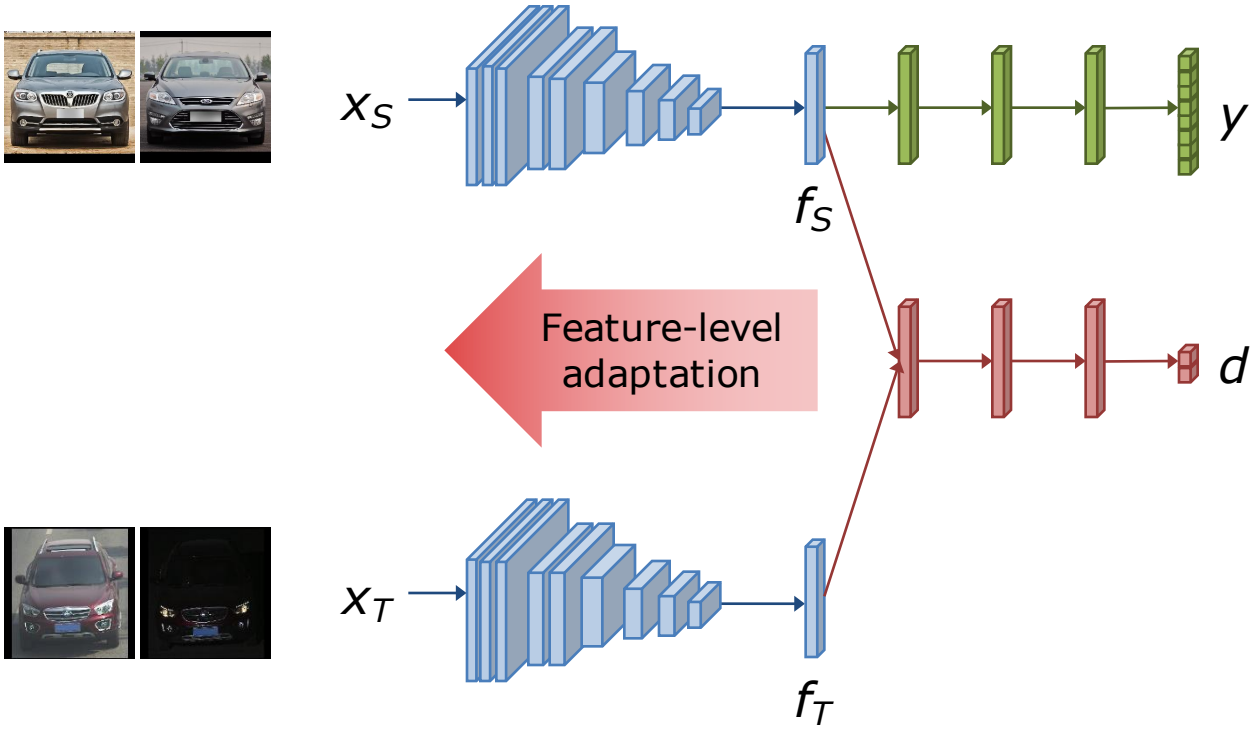
Pixel-level adaptation by image synthesis

- General strategy
  - Translate images across domains using a generative neural network
  - Unpaired: no instance-level correspondence

- Limitations:
  - Underdetermined problem: pixel representation is high-dim
  - Complex image prior: completing an image is difficult
  - Structural consistency: translating beyond color is difficult

Learning to **align** distributions: feature-level

- General Strategy
  - Learn factors that cause divergence between domains
  - Learn to reduce divergence distance between domains
  - While being discriminative with respect to source domain



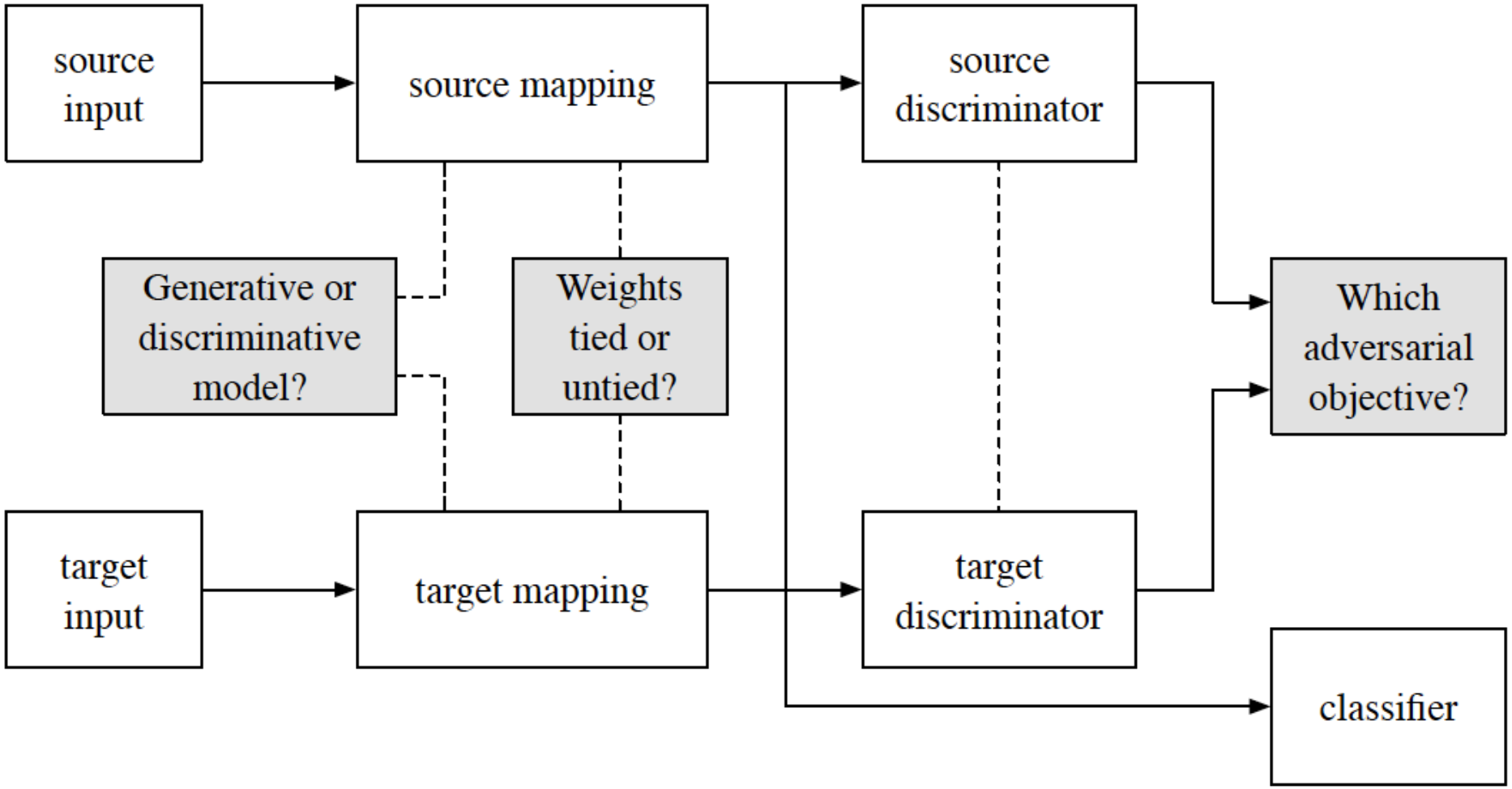
Ben-David et al., Analysis of Representations for DA, NIPS 2006
Ganin et al., UDA by Backpropagation, ICML 2015

Feature-level adaptation by DANN

- General Strategy
  - Learn factors that cause divergence between domains
  - Learn to reduce divergence distance between domains
  - While being discriminative with respect to source domain

- Advantages and limitations:
  - Black-box: works as long as (unlabeled) target data exists
  - Black-box: difficult to inject additional insights or constraints
  - Training instability: adversarial learning without constraints (cycle consistency, UNet) is less stable

Design Choices for Adversarial Adaptation

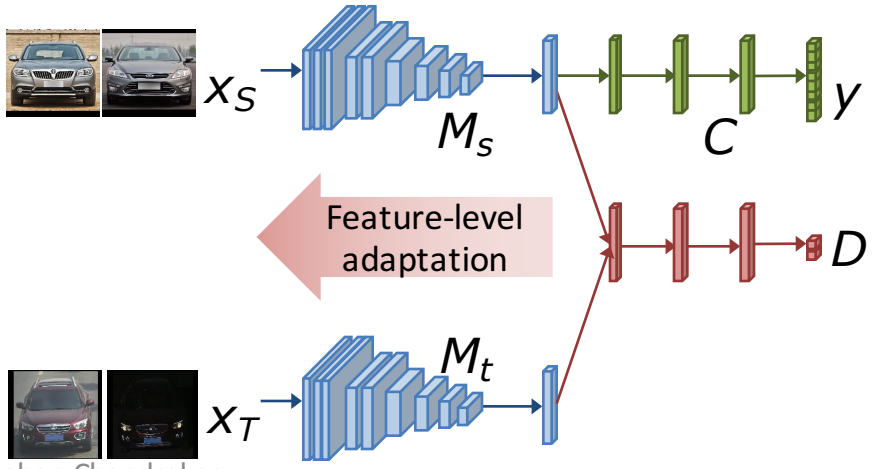


Generalized Adversarial Adaptation

- Source images X_s with labels Y_s, target images X_t with no labels
- Goal: classify target images into one of K categories
- Learn a target representation M_t and classifier C_t
- No labels available in target domain
  - Learn source representation M_s and classifier C_s, then adapt to target

Generalized Adversarial Adaptation

- Source images X_s with labels Y_s, target images X_t with no labels
- Goal: classify target images into one of K categories
- Learn a target representation M_t and classifier C_t
- No labels available in target domain
  - Learn source representation M_s and classifier C_s, then adapt to target
- Adversarial adaptation methods
  - Minimize distance between $M_s(X_s)$ and $M_t(Xt)$
  - Same classifier can be applied to source and target: $C = C_s = C_t$



Source and Target Mappings

- Target mapping
  - Functional form (architecture) usually same as source
  - Initialize with source parameters
  - Different choices for constraints
  - Goal: discriminative, while minimizing distance to source domain
- Impose shared weights for some source and target layers

$$\psi_{\ell_i}(M_s^{\ell_i}, M_t^{\ell_i}) = (M_s^{\ell_i} = M_t^{\ell_i})$$

- Sharing weights for all layers
  - Reduces number of parameters
  - Ensures target mapping is at least discriminative on the source
  - Optimization ill-conditioned: same network must handle source and target
- Share weights only for some layers
  - Allow model to learn parameters for each domain individually
  - Usually more effective for adaptation

Generalized Adversarial Adaptation

- Learn classifier using standard supervised loss on source data

$$\min_{M_s, C} \mathcal{L}_{\text{cls}}(\mathbf{X}_s, Y_t) = \mathbb{E}_{(\mathbf{x}_s, y_s) \sim (\mathbf{X}_s, Y_t)} - \sum_{k=1}^{K} \mathbb{1}_{[k=y_s]} \log C(M_s(\mathbf{x}_s))$$

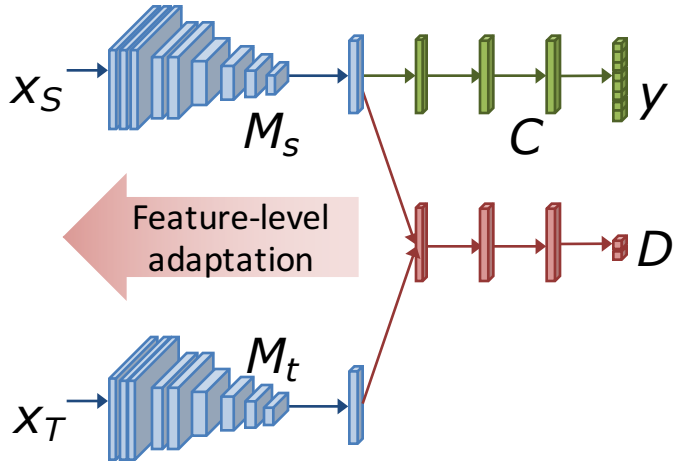
Generalized Adversarial Adaptation

- Learn classifier using standard supervised loss on source data

$$\min_{M_s, C} \mathcal{L}_{\text{cls}}(\mathbf{X}_s, Y_t) = \mathbb{E}_{(\mathbf{x}_s, y_s) \sim (\mathbf{X}_s, Y_t)} - \sum_{k=1}^{K} \mathbb{1}_{[k=y_s]} \log C(M_s(\mathbf{x}_s))$$

- Alternating minimization between two functions
  - Discriminator optimized with supervised loss (labels indicate domain)

$$\mathcal{L}_{\text{adv}_D}(\mathbf{X}_s, \mathbf{X}_t, M_s, M_t) = -\mathbb{E}_{\mathbf{x}_s \sim \mathbf{X}_s}[\log D(M_s(\mathbf{x}_s))] - \mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}_t}[\log(1 - D(M_t(\mathbf{x}_t)))]$$



Generalized Adversarial Adaptation

- Learn classifier using standard supervised loss on source data

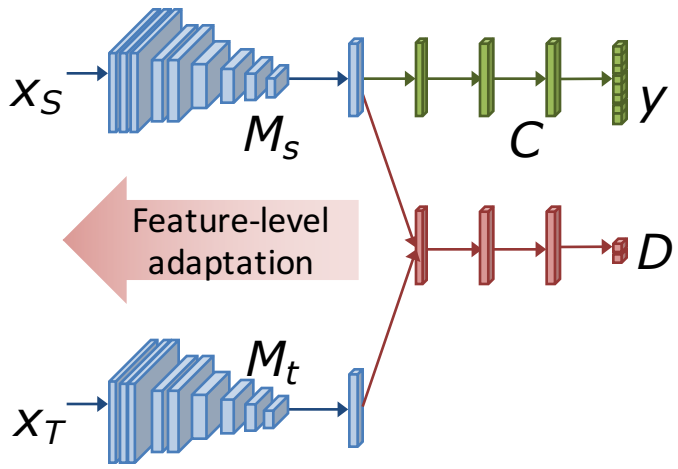
$$\min_{M_s, C} \mathcal{L}_{\text{cls}}(\mathbf{X}_s, Y_t) = \mathbb{E}_{(\mathbf{x}_s, y_s) \sim (\mathbf{X}_s, Y_t)} - \sum_{k=1}^{K} \mathbb{1}_{[k=y_s]} \log C(M_s(\mathbf{x}_s))$$

- Alternating minimization between two functions
  - Discriminator optimized with supervised loss (labels indicate domain)

$$\mathcal{L}_{\text{adv}_D}(\mathbf{X}_s, \mathbf{X}_t, M_s, M_t) = -\mathbb{E}_{\mathbf{x}_s \sim \mathbf{X}_s}[\log D(M_s(\mathbf{x}_s))] - \mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}_t}[\log(1 - D(M_t(\mathbf{x}_t)))]$$

  - Source and target mappings optimized with constrained adversarial objective

$$\begin{aligned} \min_{D} \quad & \mathcal{L}_{\text{adv}_D}(\mathbf{X}_s, \mathbf{X}_t, M_s, M_t) \\ \min_{M_s, M_t} \quad & \mathcal{L}_{\text{adv}_M}(\mathbf{X}_s, \mathbf{X}_t, D) \\ \text{s.t.} \quad & \psi(M_s, M_t) \end{aligned}$$



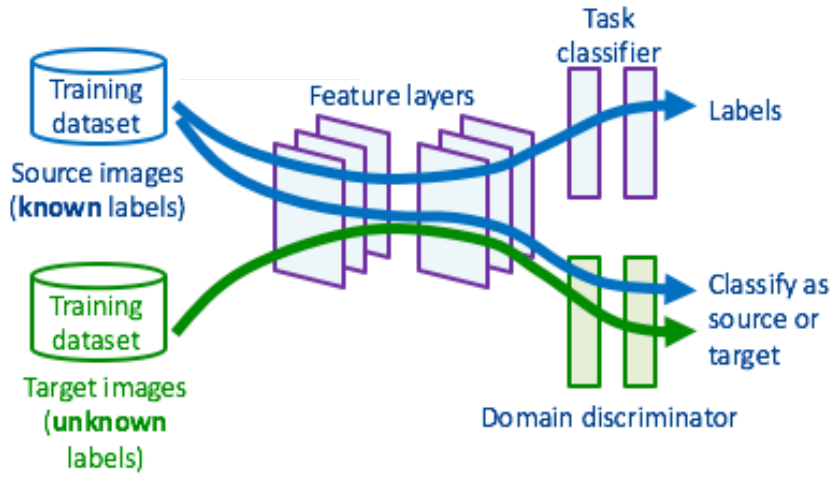
Adversarial Loss Functions

- Loss function for optimizing discriminator same for all methods
- Different choices for loss function for adversarial mapping

- **Gradient reversal layer**
  - Optimizes the true minimax objective

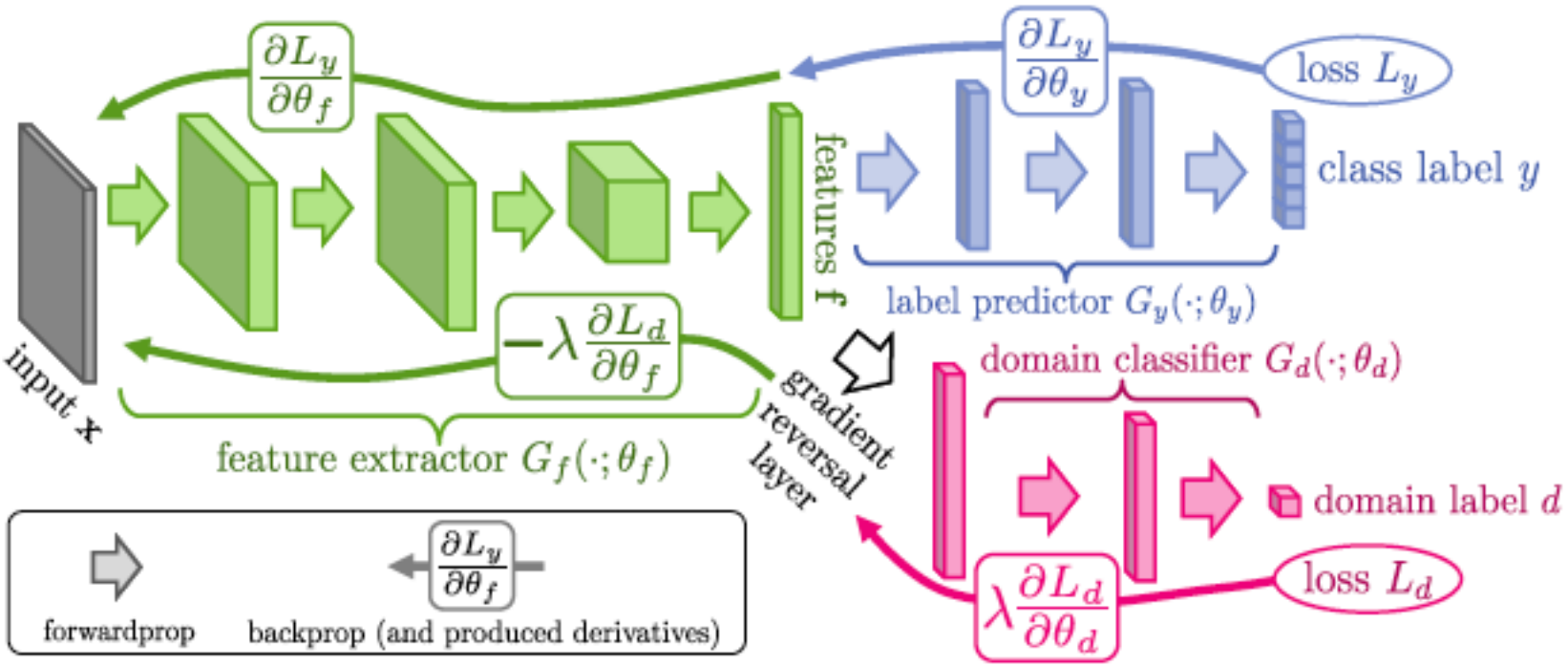
$$\mathcal{L}_{\text{adv}\,M} = -\mathcal{L}_{\text{adv}\,D}$$

  - While training D, keep encoders fixed, encourage $D(M_s)$ = 1 and $D(M_t)$ = 0
  - To train encoders, keep D fixed, but now encourage $D(M_s)$ = 0 and $D(M_t)$ = 1



Training DANN

- Optimize for three sets of parameters
  - Feature extractor G_f, label predictor G_y, domain discriminator G_d
  - Labeled data {1, ..., n} and unlabeled data {n+1, ..., N}



Training DANN

- Optimize for three sets of parameters
  - Feature extractor G_f, label predictor G_y, domain discriminator G_d
  - Labeled data {1, ..., n} and unlabeled data {n+1, ..., N}

- Prediction and domain losses

$$\mathcal{L}_y^i(\theta_f, \theta_y) = \mathcal{L}_y\big(G_y(G_f(\mathbf{x}_i; \theta_f); \theta_y), y_i\big)$$
$$\mathcal{L}_d^i(\theta_f, \theta_d) = \mathcal{L}_d\big(G_d(G_f(\mathbf{x}_i; \theta_f); \theta_d), d_i\big)$$

Training DANN

- Optimize for three sets of parameters
  - Feature extractor G_f, label predictor G_y, domain discriminator G_d
  - Labeled data {1, ..., n} and unlabeled data {n+1, ..., N}
- Prediction and domain losses

$$\mathcal{L}_y^i(\theta_f, \theta_y) = \mathcal{L}_y\big(G_y(G_f(\mathbf{x}_i; \theta_f); \theta_y), y_i\big)$$
$$\mathcal{L}_d^i(\theta_f, \theta_d) = \mathcal{L}_d\big(G_d(G_f(\mathbf{x}_i; \theta_f); \theta_d), d_i\big)$$

- Training: optimize the objective with adversarial learning

$$E(\theta_f, \theta_y, \theta_d) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_y^i(\theta_f, \theta_y) - \lambda\left(\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_d^i(\theta_f, \theta_d) + \frac{1}{n'}\sum_{i=n+1}^{N}\mathcal{L}_d^i(\theta_f, \theta_d)\right)$$

- Find saddle point

$$(\hat{\theta}_f, \hat{\theta}_y) = \underset{\theta_f, \theta_y}{\operatorname{argmin}} E(\theta_f, \theta_y, \hat{\theta}_d)$$
$$\hat{\theta}_d = \underset{\theta_d}{\operatorname{argmax}} E(\hat{\theta}_f, \hat{\theta}_y, \theta_d)$$

Goal for feature: Classify well, maximize domain confusion

Goal for discriminator: maximize domain separability

Gradient Reversal Layer

- Gradient updates

$$\theta_f \longleftarrow \theta_f - \mu\left(\frac{\partial \mathcal{L}_y^i}{\partial \theta_f} - \lambda \frac{\partial \mathcal{L}_d^i}{\partial \theta_f}\right),$$

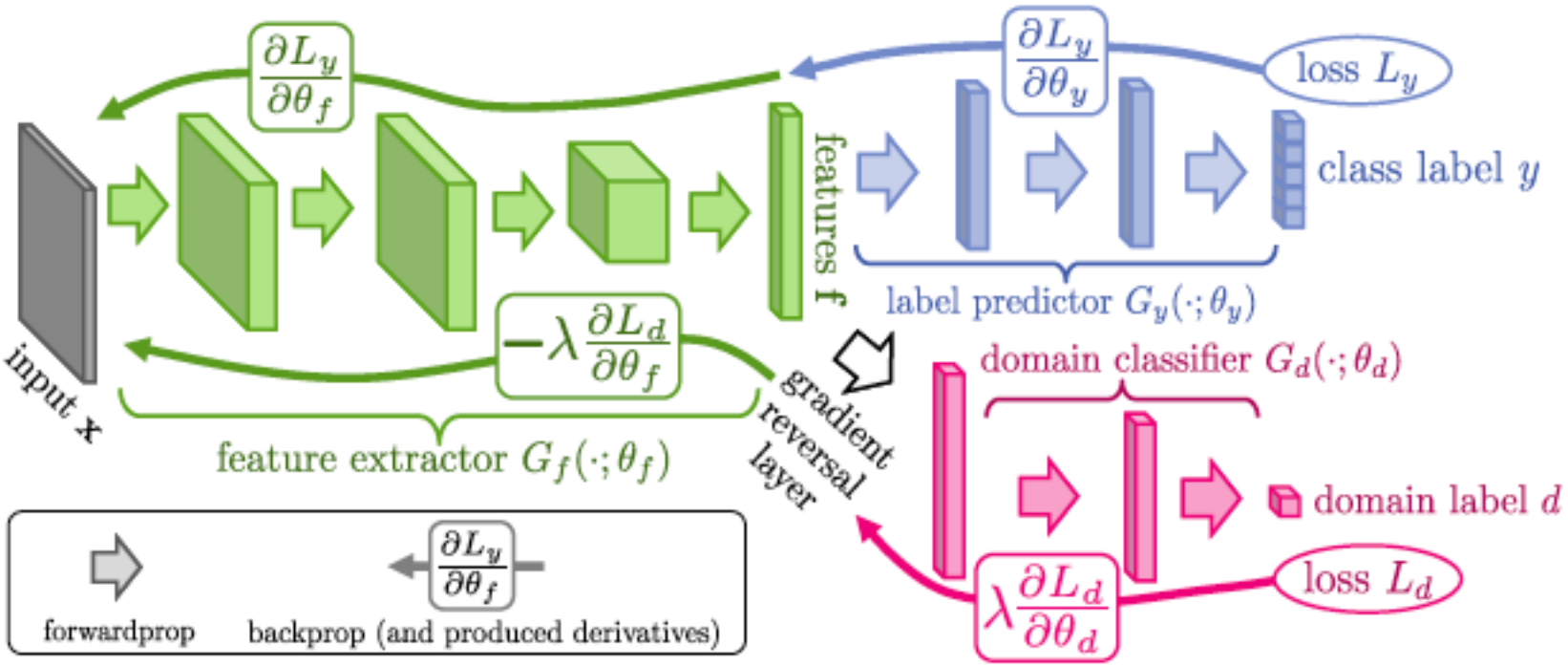
$$\theta_y \longleftarrow \theta_y - \mu \frac{\partial \mathcal{L}_y^i}{\partial \theta_y},$$

$$\theta_d \longleftarrow \theta_d - \mu\lambda \frac{\partial \mathcal{L}_d^i}{\partial \theta_d},$$

- Almost like standard SGD updates for feedforward network
  - Except domain classifier gradients subtracted from label predictor gradients
- Introduce a new gradient reversal layer
  - Identity transformation for forward propagation
  - Multiply gradients by -1 for backward propagation
  - Can be implemented using standard procedures

Gradient Reversal Layer

- Gradient reversal layer
  - Ensures feature distributions between domains are made similar
  - No new parameters introduced



Adversarial Loss Functions

- Loss function for optimizing discriminator same for all methods
- Different choices for loss function for adversarial mapping

- **GAN loss function**
  - Train the encoder with the standard loss function with inverted labels

$$\mathcal{L}_{\text{adv}_M}(\mathbf{X}_s, \mathbf{X}_t, D) = -\mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}_t}[\log D(M_t(\mathbf{x}_t))]$$

  - While training D, keep encoders fixed, encourage $D(M_s)$ = 1 and $D(M_t)$ = 0
  - To train target encoder, keep D and M_s fixed, encourage $D(M_t)$ = 1
  - Same fixed-point properties as minimax loss, but stronger gradients for M_t