Domain Adaptation and Generalization

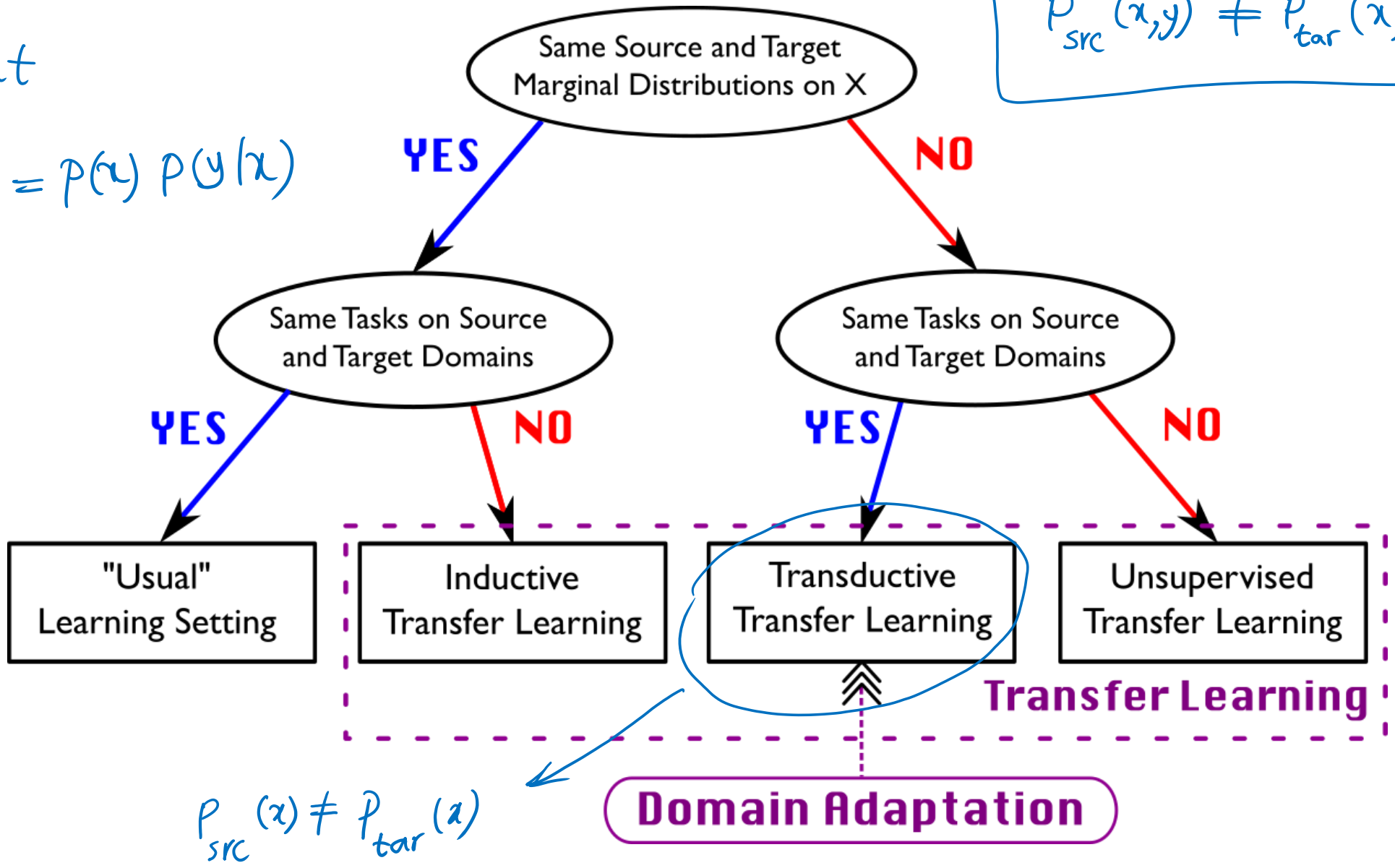
Mostafa Tavassolipour

x: input

y: output

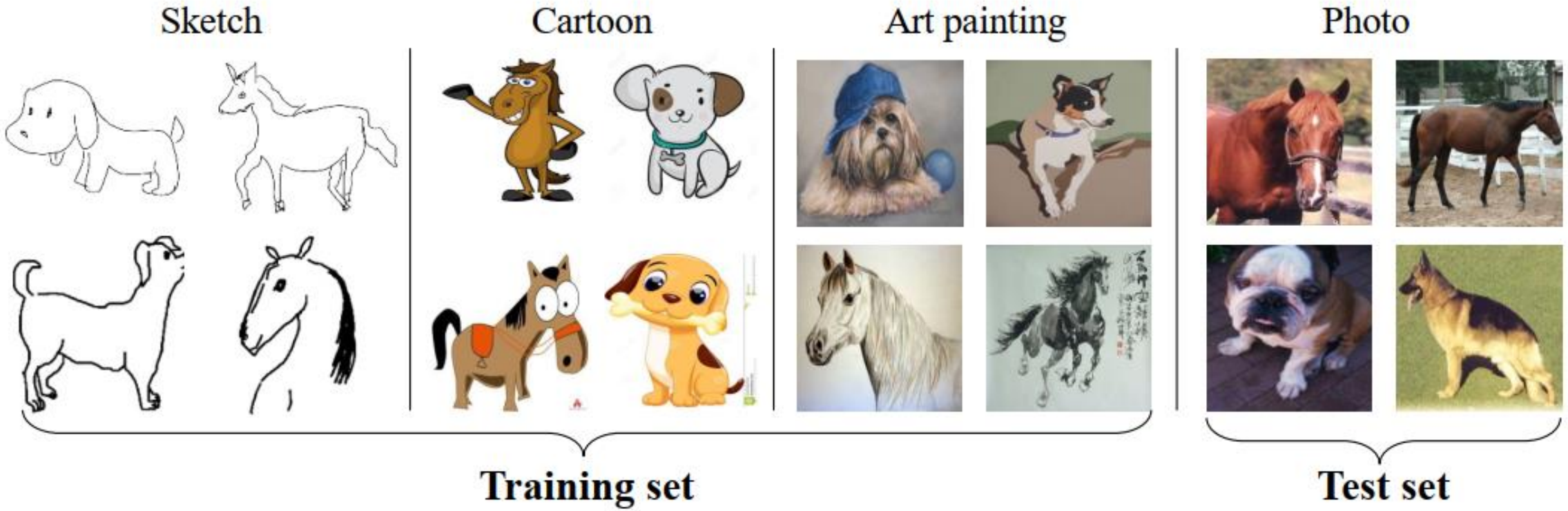
$P(x, y) = P(x)\,P(y|x)$

$P_{src}(x,y) \neq P_{tar}(x,y)$



$P_{src}(x) \neq P_{tar}(x)$

Domain Adaptation



- $P_{Train}(x, y) \neq P_{Test}(x, y)$
- Source: Train, Target: test

Images are copied from PACS dataset [Li et al.'17]

Distribution Shift

Training data

Source domain

Test data

Target domain

A classifier trained on one domain may perform poorly on another domain

Single vs multiple source domains

Source domain 1

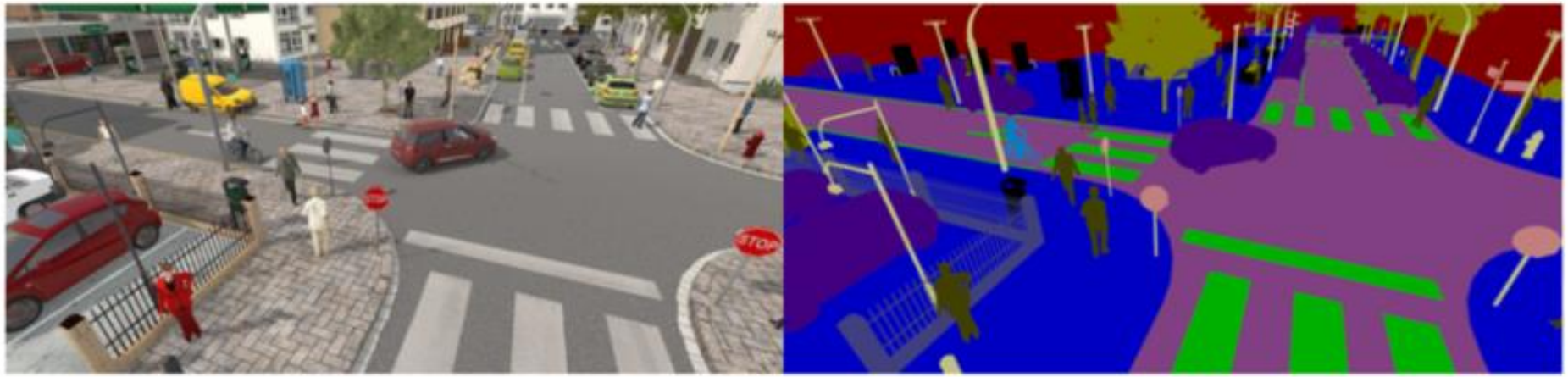
Source domain 2

Target domain

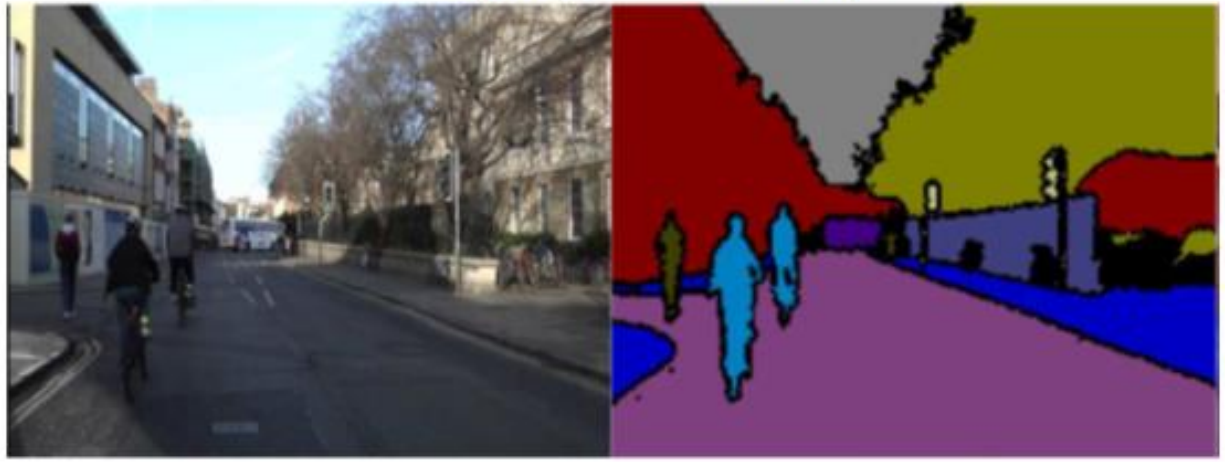
- Moving towards domain generalization

Domain Adaptation: Example 1

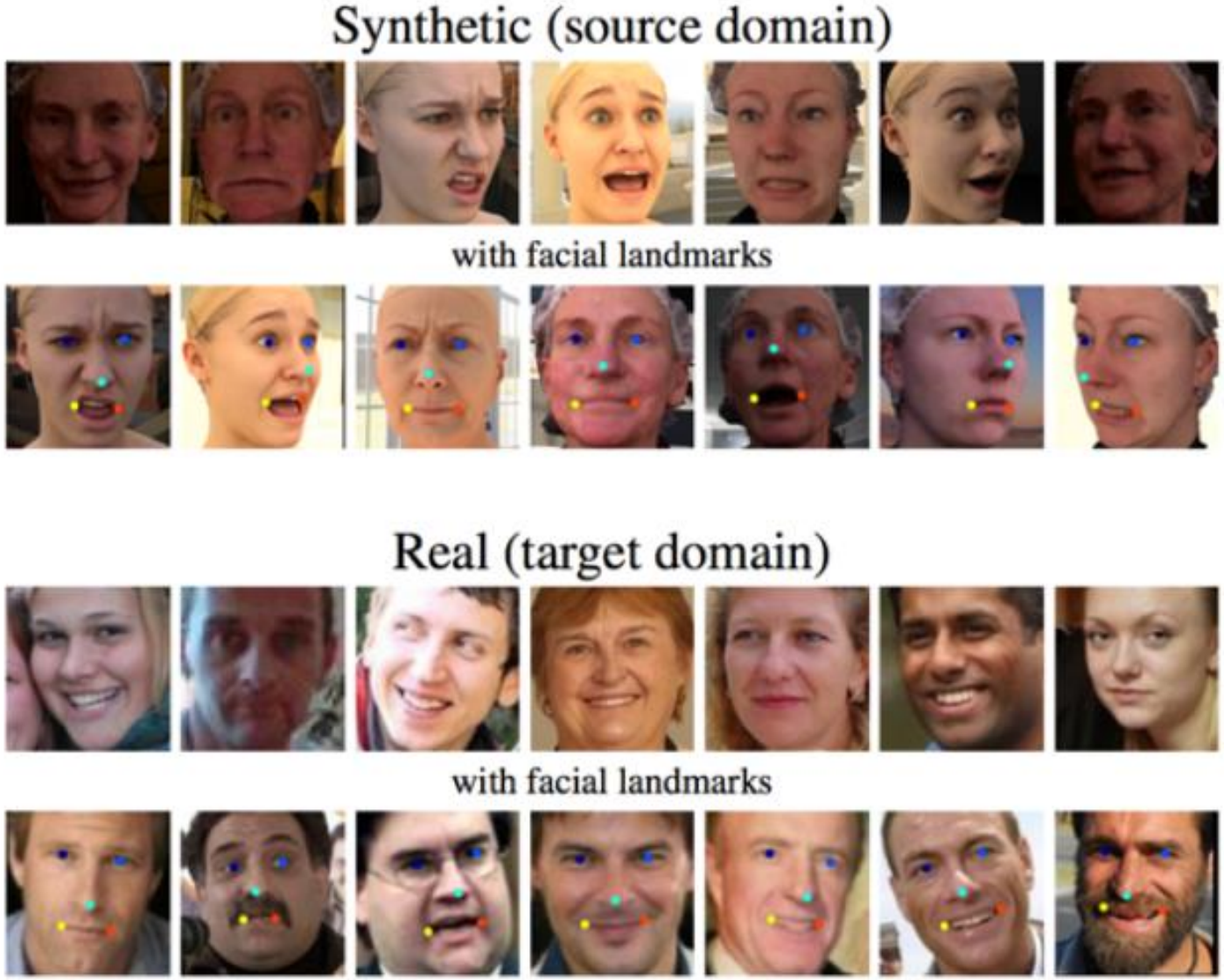
Synthetic (source domain)



Real (target domain)



Domain Adaptation: Example 2



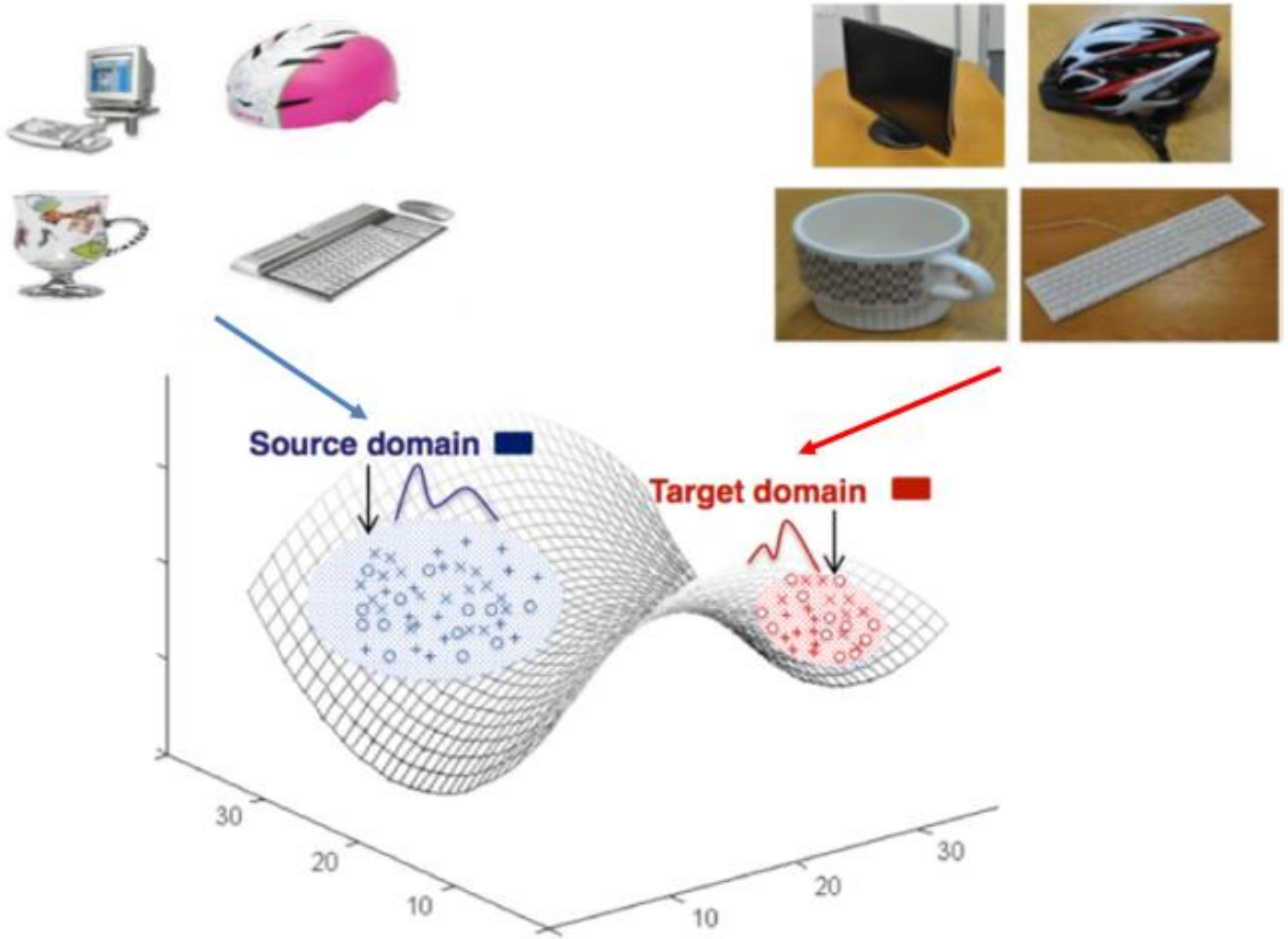
Domain Shift

$P_{src}(x) \neq P_{tar}(x)$

- The domain shift is defined as a difference in the distribution of the source and target samples.

Covariate shift

x: covariate



Domain Shift

- Typically, the literature focuses on the covariate shift case, where

$$p_t(x_t) \neq p_s(x_s)$$

- But

$$p_t(y|x_t) = p_s(y|x_s)$$

Another Distribution Shifts

- Concept Shift:

$$p_t(y|x_t) \neq p_s(y|x_s)$$

- Prior Shift

$$p_t(y) \neq p_s(y)$$

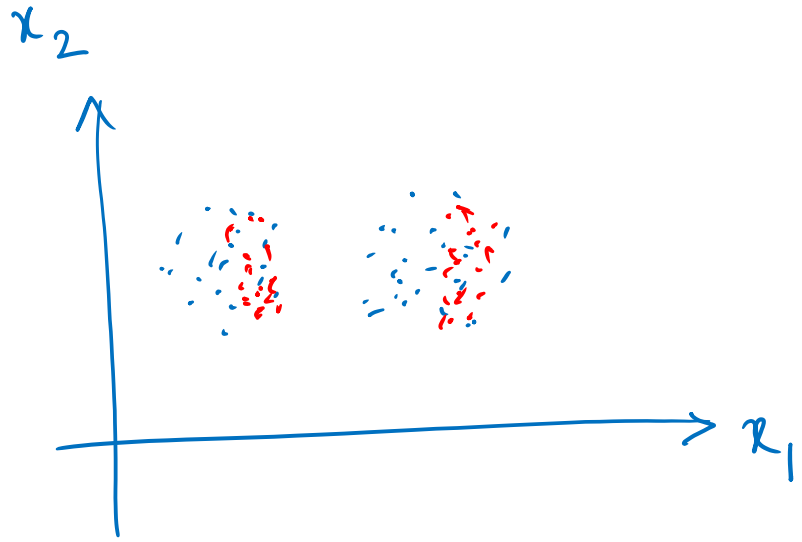
Federated Learning

Metric Learning

- Learning Distance:

$$d_W(x_i, x_j) = (x_i - x_j)^T W (x_i - x_j)$$

$$W = ?$$



Unsupervised Domain Adaptation

- The unsupervised scenario assumes no target labels are available

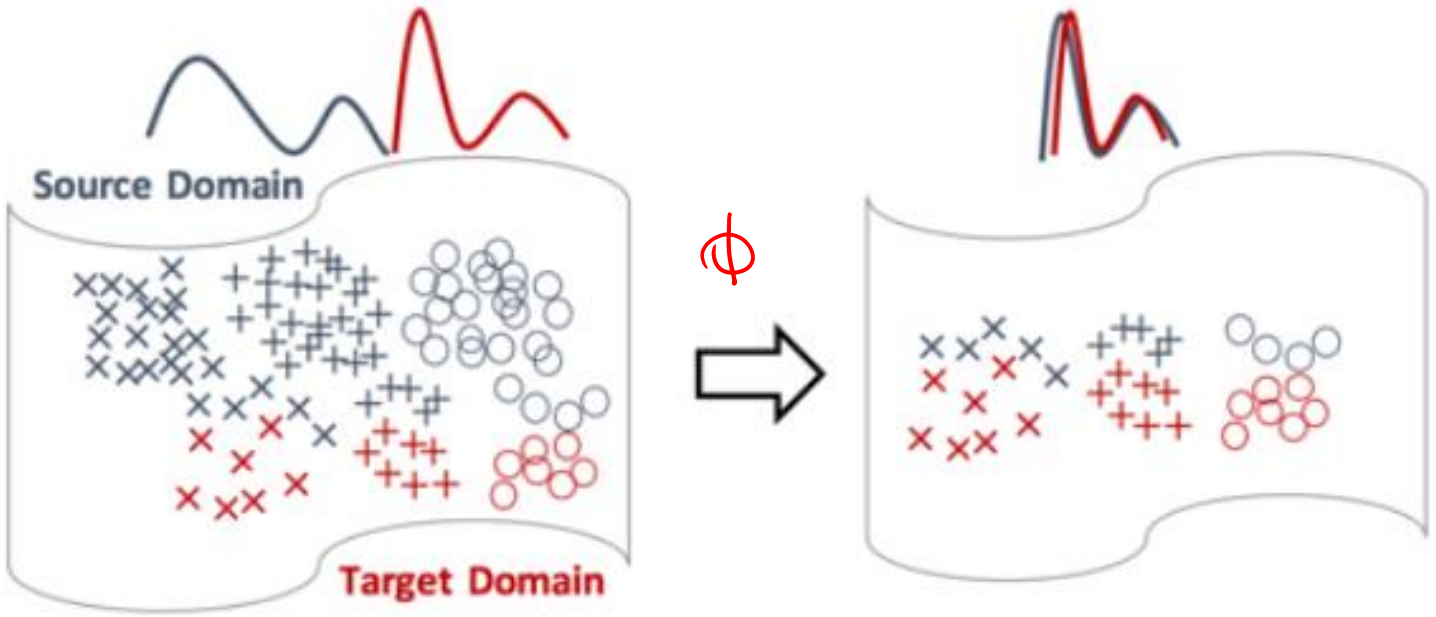
Source data

Fully-labeled

Target data

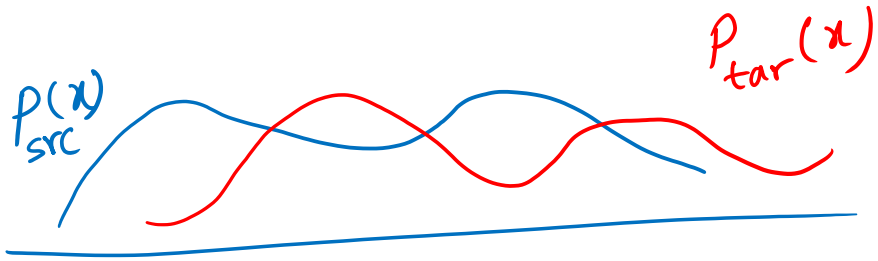
~~A few labels~~

Idea: Learning features which minimize the difference between source and target distributions.



How to measure distance between two distributions?

$$P_{src}(x) \neq P_{tar}(x)$$



$$x' = \phi(x)$$

$$\Rightarrow \boxed{P_{src}(x') \approx P_{tar}(x')}$$

$KL(p(x) \,||\, q(x)) \rightarrow$ KL Divergence

① $d(x,y) \geq 0$ ✓

② $d(x,y) = d(y,x)$ ✗

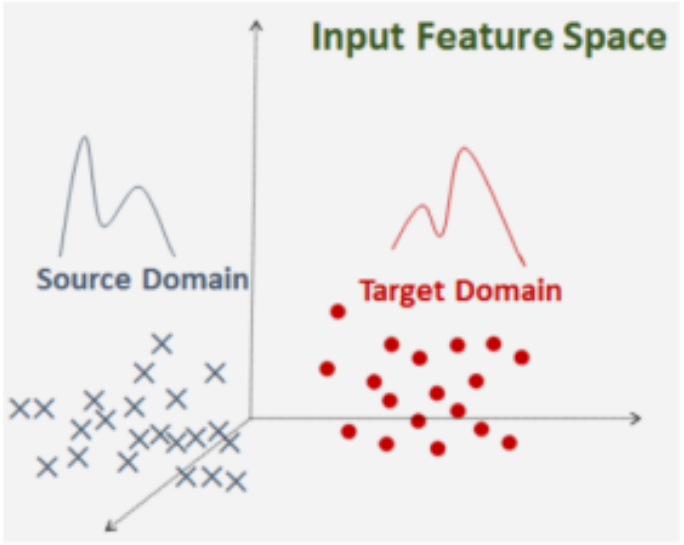
③ $d(x,y) = 0 \Leftrightarrow x = y$ ✓

④ $d(x,y) + d(x,z) > d(y,z)$ ✗

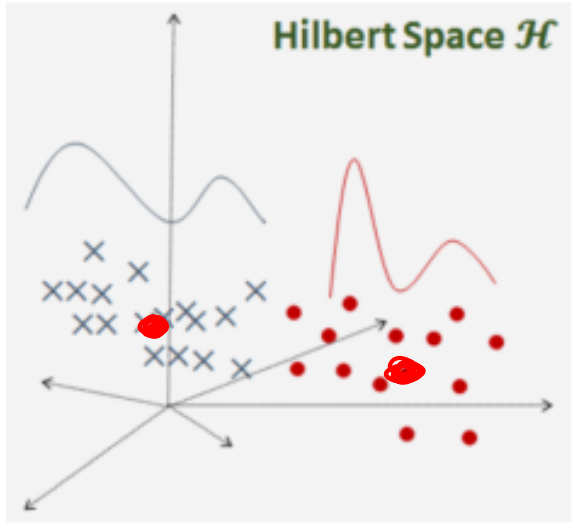
$$\int p(x) \log\left(\frac{p(x)}{q(x)}\right) dx = KL\left(p(x) \,||\, q(x)\right)$$

Maximum Mean Discrepancy (MMD)

- Compare the mean of two samples in Hilbert space
  - Gretton et al., JMLR 2012



$\phi(\mathbf{x})$



$$D_{MMD}(X_s, X_t) = \left\| \frac{1}{n}\sum_{i=1}^{n}\phi(x_s^i) - \frac{1}{m}\sum_{j=1}^{m}\phi(x_t^j) \right\|_{\mathcal{H}}$$

MMD Estimation

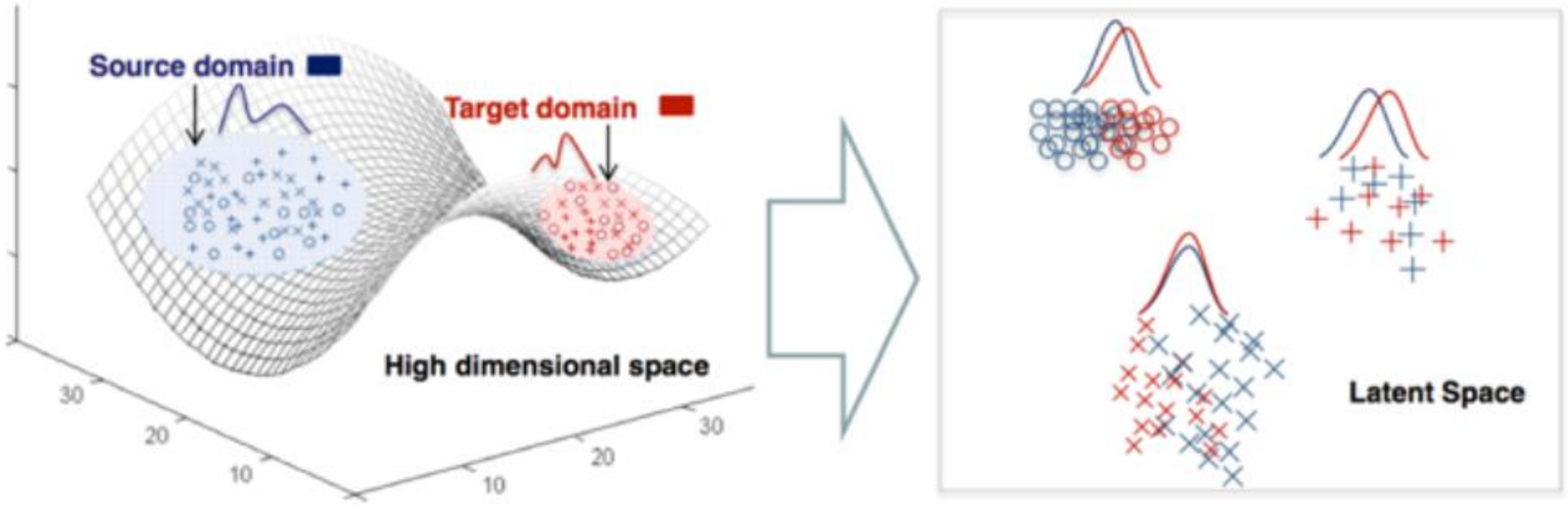
$$\widehat{\mathrm{MMD}}(P,Q) = \left\| \frac{1}{n}\sum_{i=1}^{n}\varphi(x_i) - \frac{1}{m}\sum_{i=1}^{m}\varphi(y_i) \right\|_{\mathcal{H}}^2$$

$$= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}k(x_i,x_j) + \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}k(y_i,y_j) - \frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}k(x_i,y_j)$$

$K(x,y) = \phi(x)^T \phi(y)$

Domain-invariant representation learning

- Learn a mapping to a latent space where the distributions are similar.



Domain-Adversarial Training (JMLR 2016)

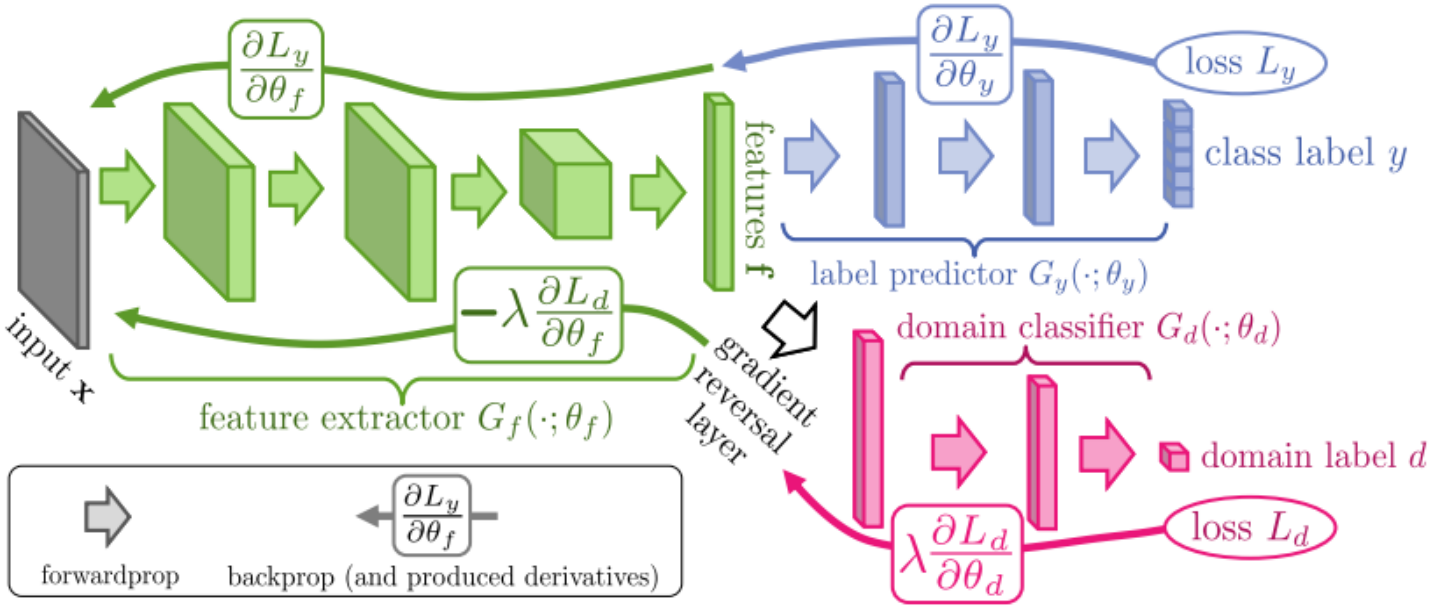
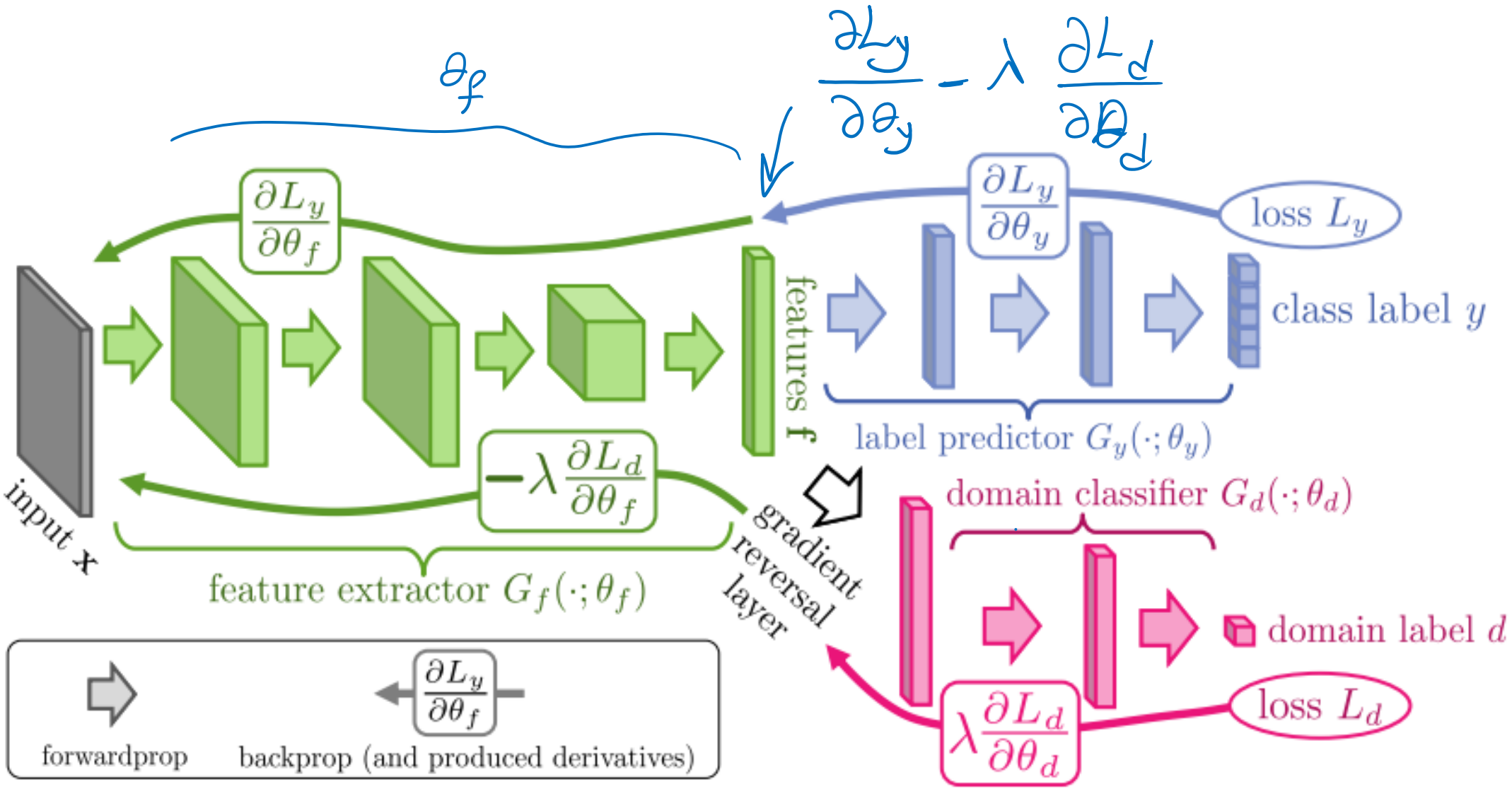
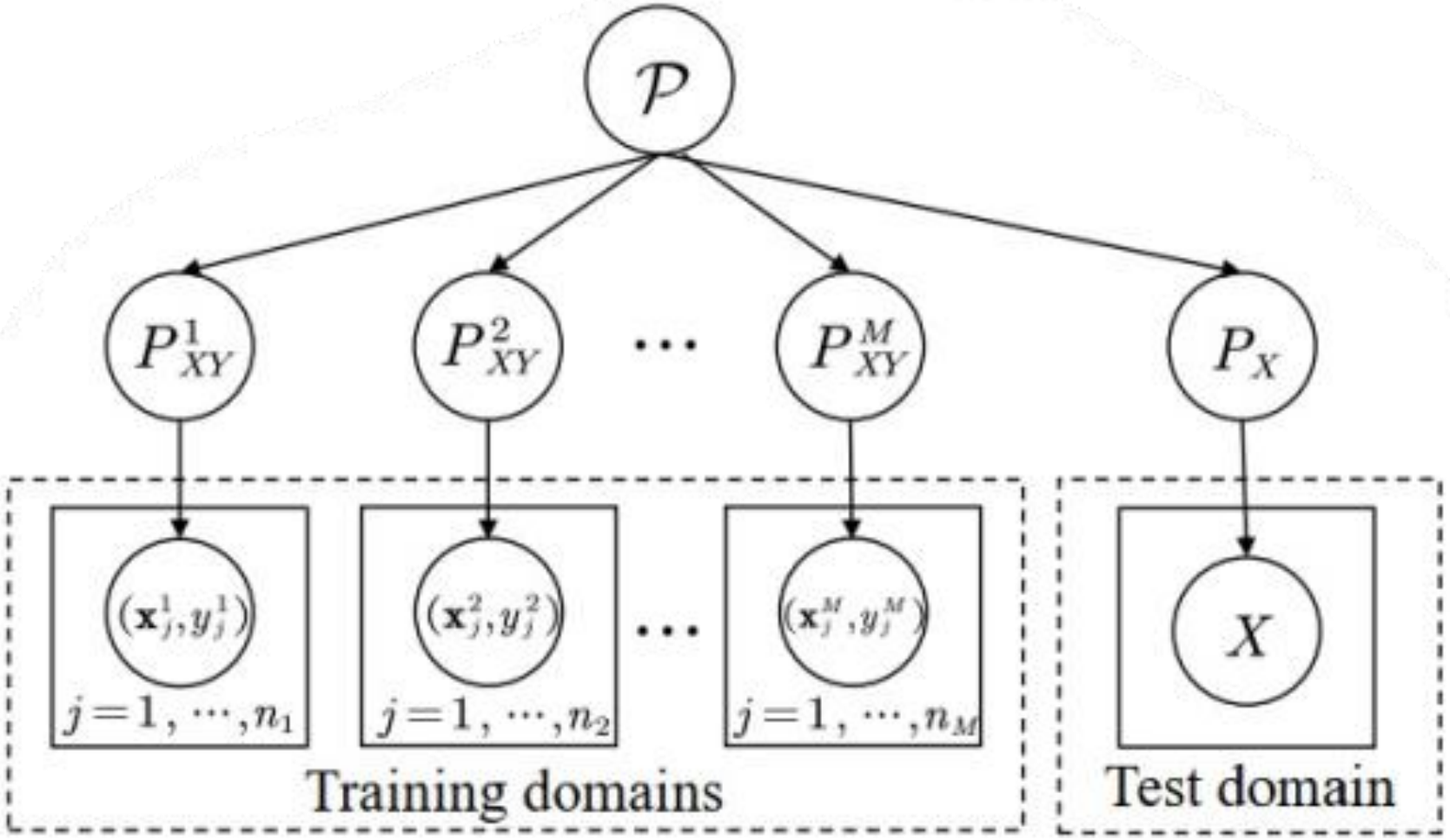


Figure 1: The **proposed architecture** includes a deep *feature extractor* (green) and a deep *label predictor* (blue), which together form a standard feed-forward architecture. Unsupervised domain adaptation is achieved by adding a *domain classifier* (red) connected to the feature extractor via a *gradient reversal layer* that multiplies the gradient by a certain negative constant during the backpropagation-based training. Otherwise, the training proceeds standardly and minimizes the label prediction loss (for source examples) and the domain classification loss (for all samples). Gradient reversal ensures that the feature distributions over the two domains are made similar (as indistinguishable as possible for the domain classifier), thus resulting in the domain-invariant features.

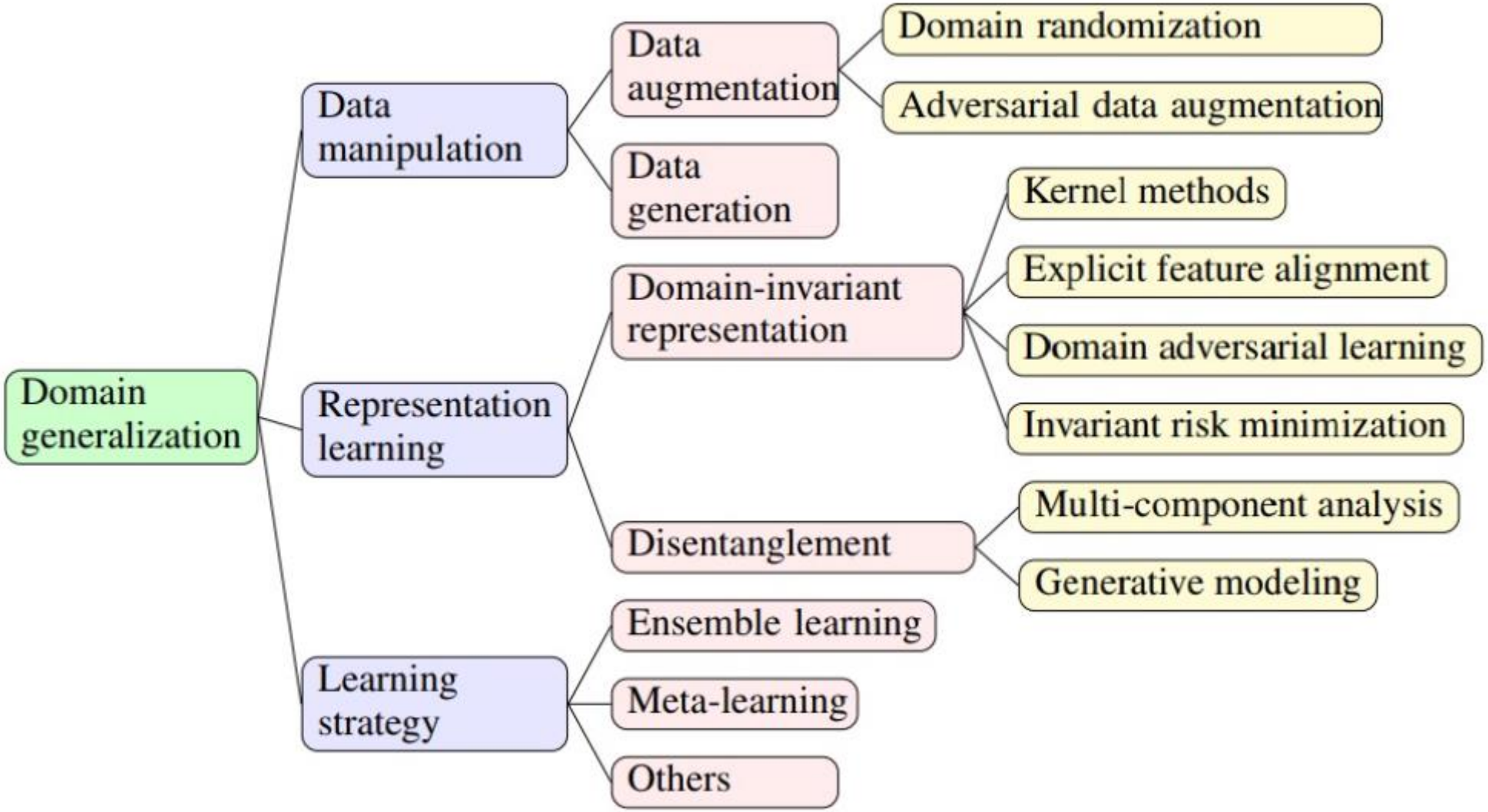


$$\theta = \theta_f \cup \theta_y \cup \theta_d$$

Domain Generalization



Domain Generalization



Data Manipulation

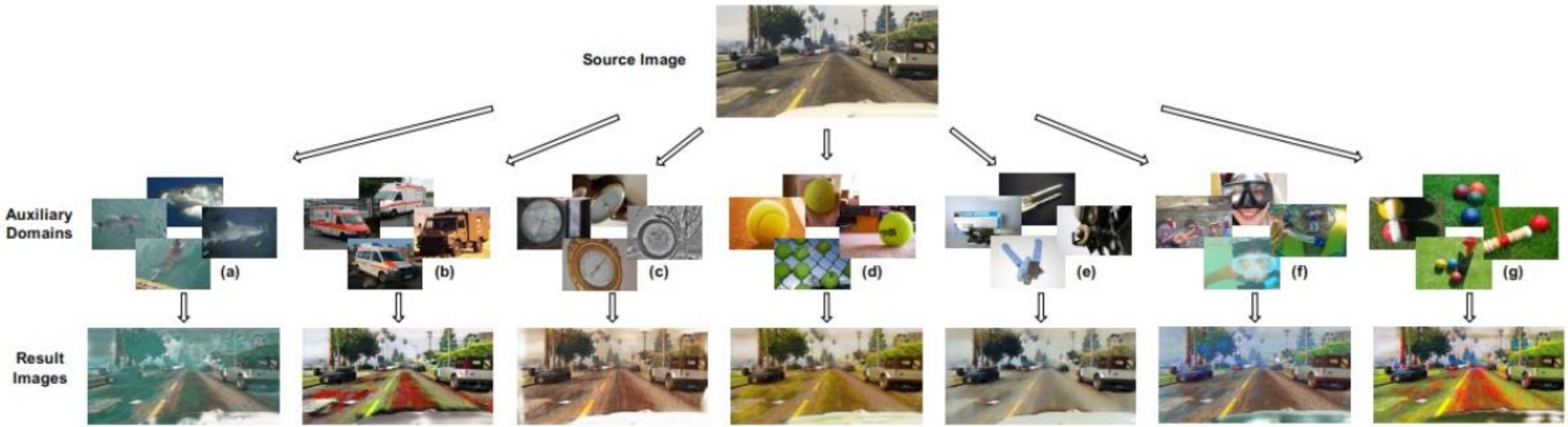
- Data quantity and quality are key factors of generalization

$$\min_h \mathbb{E}_{\mathbf{x},y}[\ell(h(\mathbf{x}), y)] + \mathbb{E}_{\mathbf{x}',y}[\ell(h(\mathbf{x}'), y)]$$

$$\mathbf{x}' = \mathrm{mani}(\mathbf{x}) \begin{cases} \text{Data augmentation} \\ \text{Data generation} \end{cases}$$

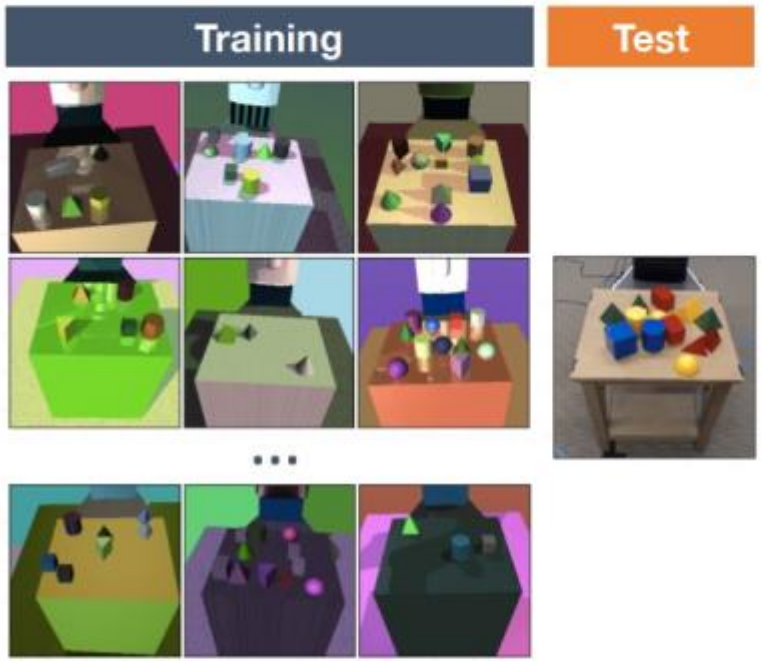
Data augmentation

- Typical augmentation
  - Rotation, noise, color…
- Domain randomization (DR)
  - Shape, position, texture, viewpoint, lighting condition, noise…

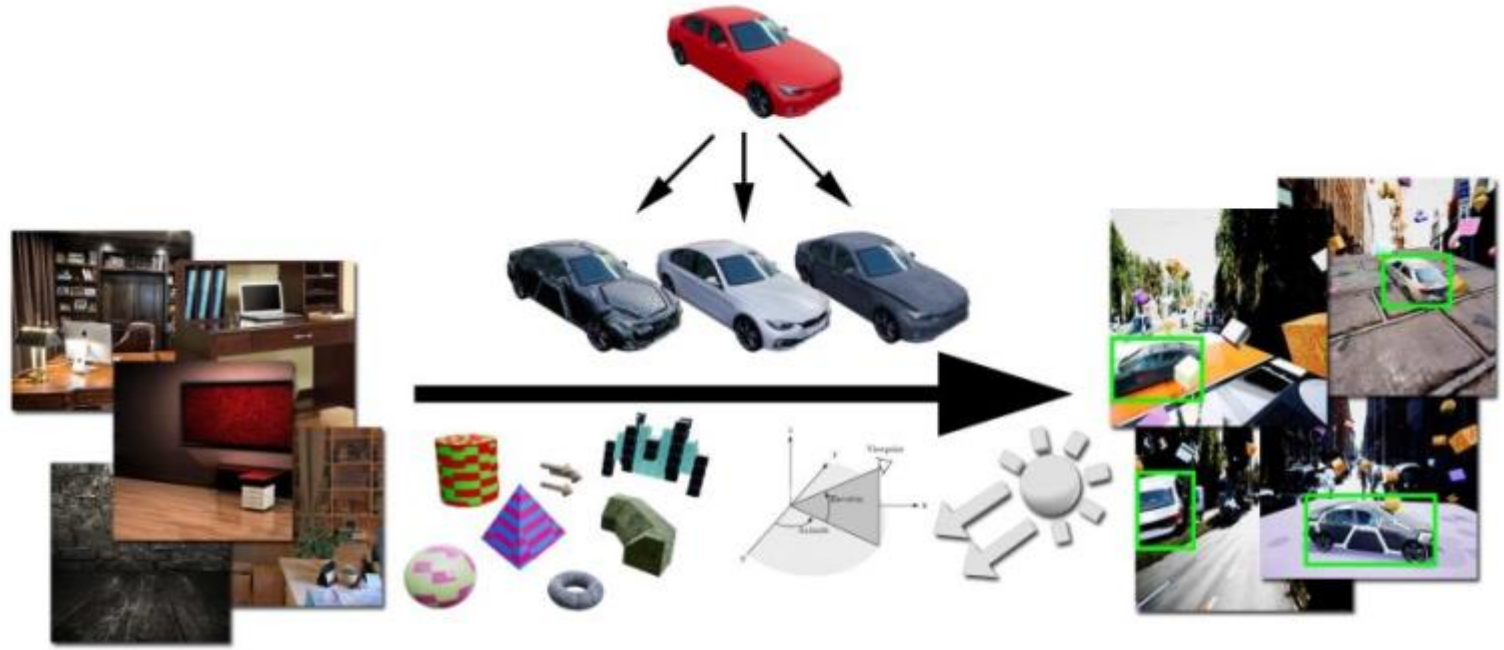


Yue et al. Domain Randomization and Pyramid Consistency: Simulation-to-Real Generalization without Accessing Target Domain Data. 2019.

Domain randomization



Sim->Real robot control



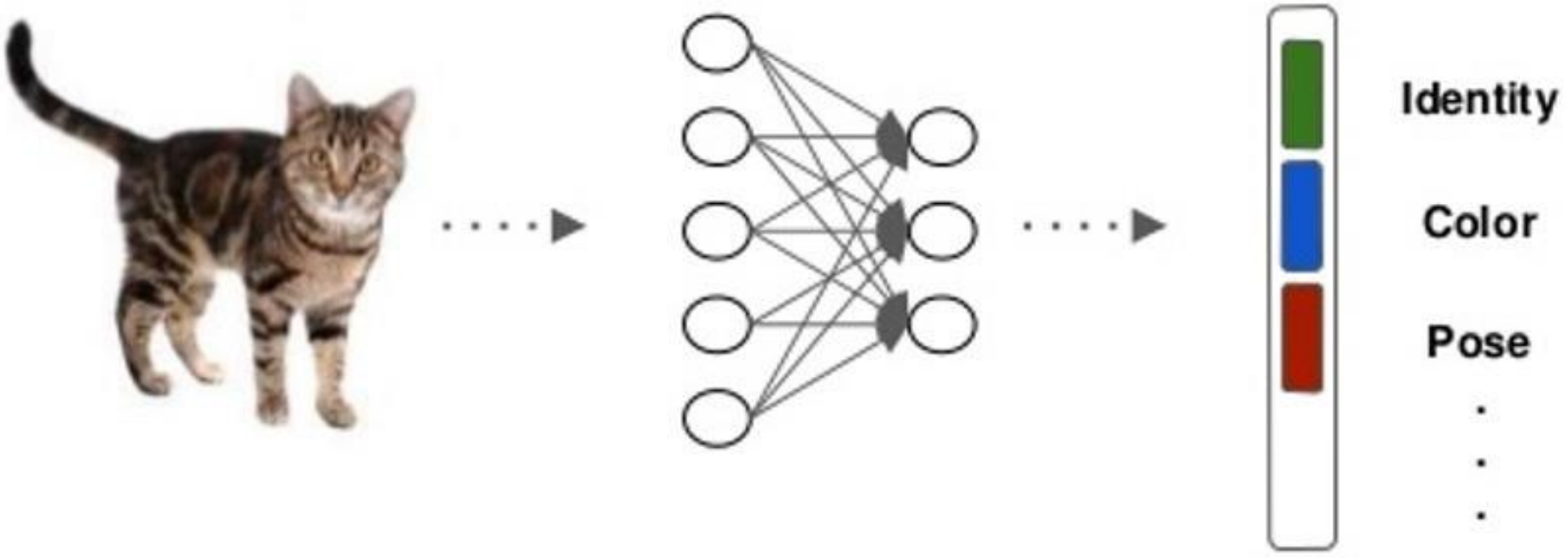
Synthetic images -> Real images

- Tobin, et al. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. IROS 2017.
- Tremblay et al. Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization. CVPR workshop 2018.

Disentangled Representation

Representing data or information in a manner that separates out different factors or elements that make up the data.

improve the interpretability and generalization



Domain Generalization: Learning Strategies

- Ensemble-Learning
  - Multiple models are trained on different subsets of data or with different algorithms, and their predictions are combined to make a final decision.

- Meta-Learning
  - Training a model on a variety of different tasks or domains so that it can quickly adapt and generalize to new, unseen domains with minimal fine-tuning.