Master IMA – RDFIA 2026-2027
Reconnaissance des formes pour l'analyse et l'interprétation d'images
Transfer Learning and Domain Adaptation

Clément Rambour

clement.rambour@isir.upmc.fr

Sorbonne Université
ISIR - équipe MLIA

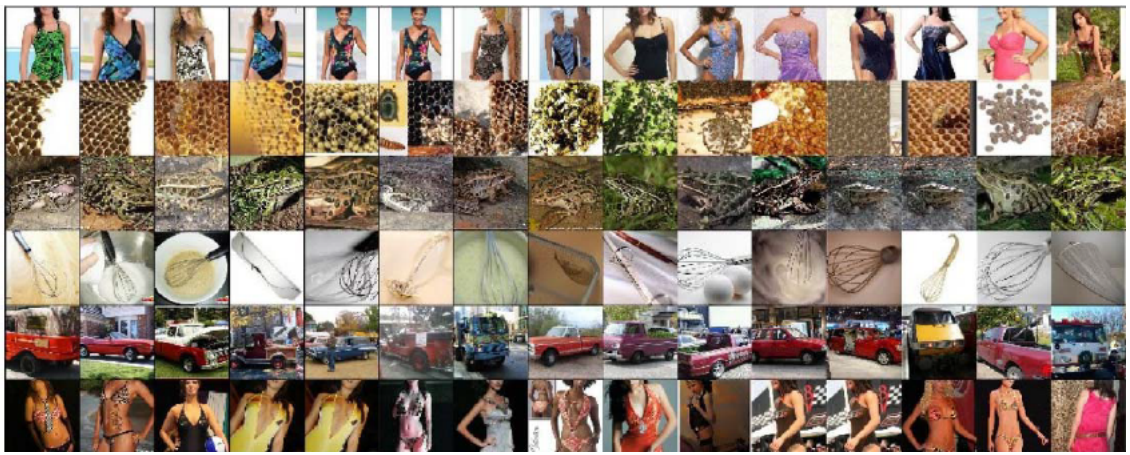
Année universitaire 2026-2027

Outline

1. Transfer learning: recap
2. A general framework for transfer learning
3. Unsupervised Domain Adaptation
4. Supervised Domain Adaptation
5. Use case: domain adaptation for autonomous driving
6. Going further: domain shift after 2019

Transfer from ImageNet (source)

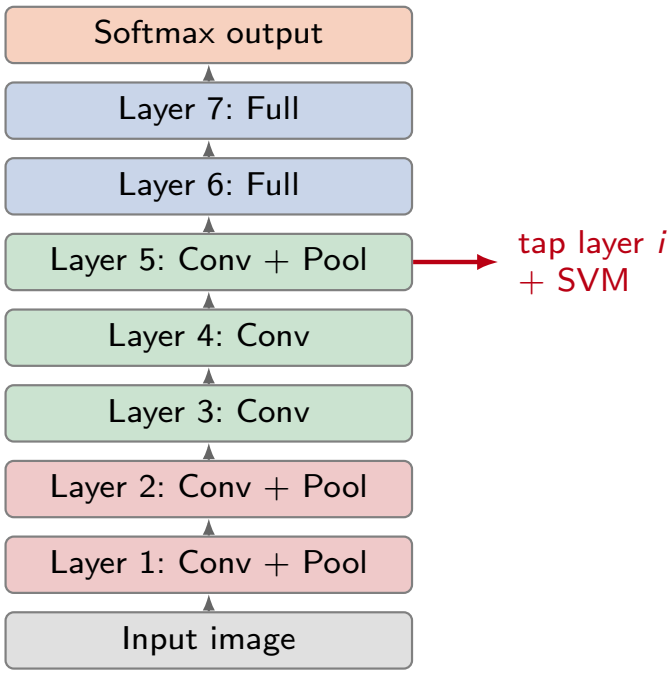
- **Transfer as generic features**
  - Raw deep features (learned from ImageNet)
    = a learned embedding from image to vector representation
  - Retrieval



- **Transfer learning (from source to target)**
  - Frozen features + SVM $\Rightarrow$ solution to small datasets
  - Frozen features + deep classifier
  - Fine-tuning not easy in that case (small datasets)

Transfer from source (= ImageNet task) to target task

- **Source:** ImageNet (dataset + 1000 classes) $\Rightarrow$ AlexNet trained
- **Target:** new dataset Cal-101 and new classification task with 101 classes $\Rightarrow$ chopped AlexNet (layer i) + SVM trained on top



AlexNet: 8 layers, trained on ImageNet

Tapping off features at each layer

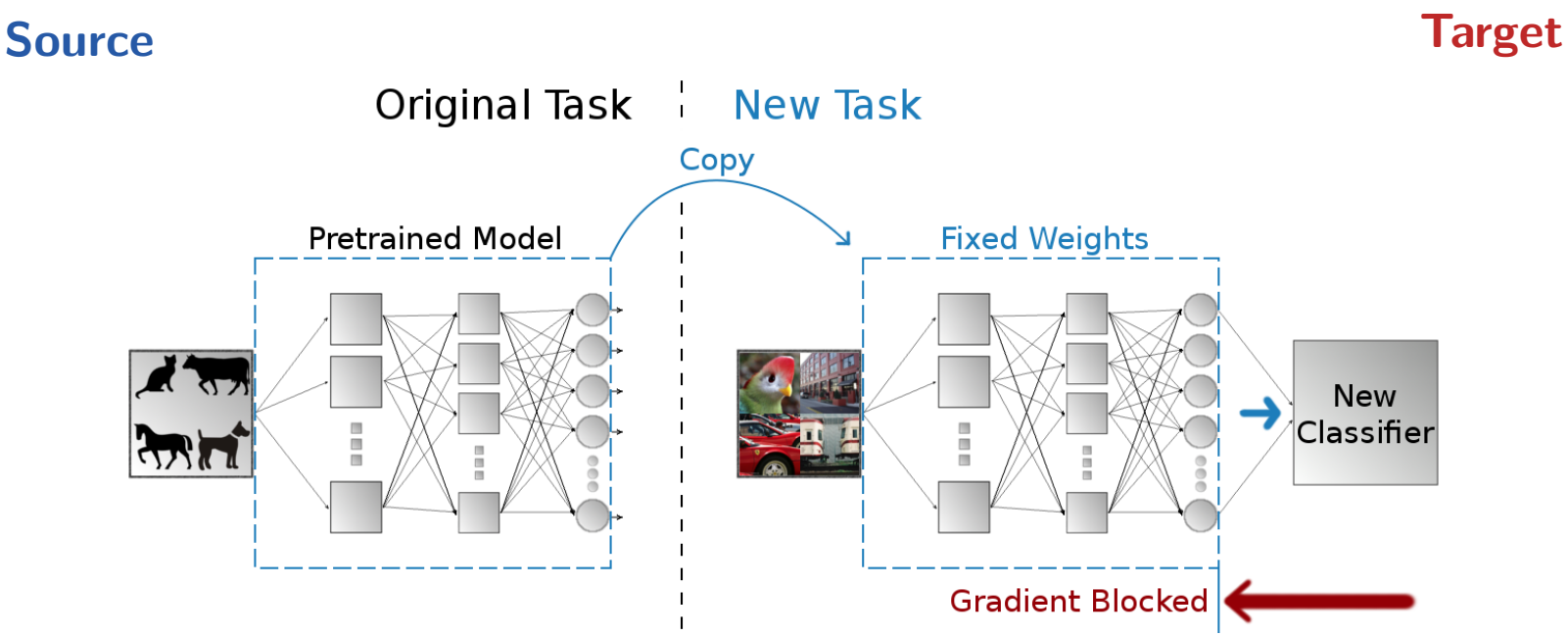
	Cal-101 (30/class)	Cal-256 (60/class)
SVM (1)	44.8 ± 0.7	24.6 ± 0.4
SVM (2)	66.2 ± 0.5	39.6 ± 0.3
SVM (3)	72.3 ± 0.4	46.0 ± 0.3
SVM (4)	76.6 ± 0.4	51.3 ± 0.1
SVM (5)	$\mathbf{86.2 \pm 0.8}$	65.6 ± 0.3
SVM (7)	$\mathbf{85.5 \pm 0.4}$	$\mathbf{71.7 \pm 0.2}$
Softmax (5)	82.9 ± 0.4	65.7 ± 0.5
Softmax (7)	$\mathbf{85.4 \pm 0.4}$	$\mathbf{72.6 \pm 0.1}$

Features from layer i plugged into a linear SVM or softmax [Zeiler and Fergus, 2014]

$\Rightarrow$ **Results better than SoA computer vision methods on Cal-101!**

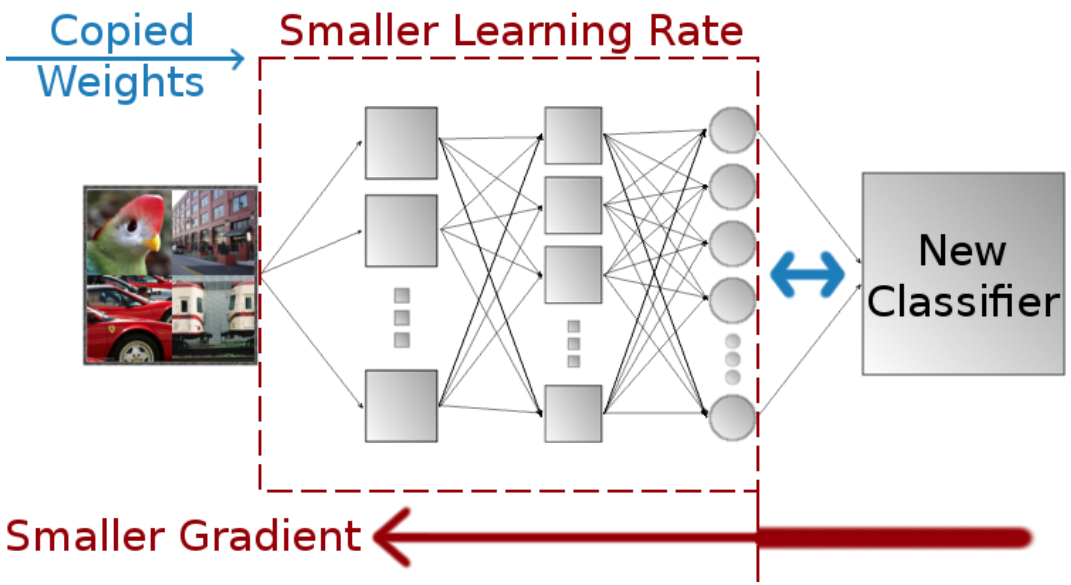
Transfer: fine-tuning of a deep model on target task

- Train a deep network (AlexNet) on source (ImageNet)
- Keep the deep parameters for target and complete with a small network on top (fully trained on target task)
- Fine-tune the whole model on target data
  - **Challenge:** only limited target data, careful about overfitting
  - **Solution:** freeze the gradient's update for the AlexNet part



Transfer: fine-tuning of a deep model on target task

- Train a deep network (AlexNet) on source (ImageNet)
- Keep the deep parameters for target and complete with a small network on top (fully trained on target task)
- Fine-tune the whole model on target data
  - **Challenge:** only limited target data, careful about overfitting
  - **Solution:** freeze the gradient's update for the AlexNet part
  - **Other solution:** use smaller gradient's update for the AlexNet part



Transfer: which supervision?

- **Task description**
  - Source data: (x^s, y^s) $\longleftarrow$ **a large amount**
  - Target data: (x^t, y^t) $\longleftarrow$ **(very) little**

  Rq: Few/one-shot learning: only a few/one examples in target domain

- **Many different contexts:**
  - In vision: from large dataset (ImageNet) to small datasets (VOC2007)
  - In speech: (supervised) speaker adaptation
    - Source data: audio data and transcriptions from many speakers
    - Target data: audio data and its transcriptions of a specific user

Outline

1. Transfer learning: recap
2. A general framework for transfer learning
3. Unsupervised Domain Adaptation
4. Supervised Domain Adaptation
5. Use case: domain adaptation for autonomous driving
6. Going further: domain shift after 2019

Transfer learning: definitions

Domain and task [Pan and Yang, 2010]

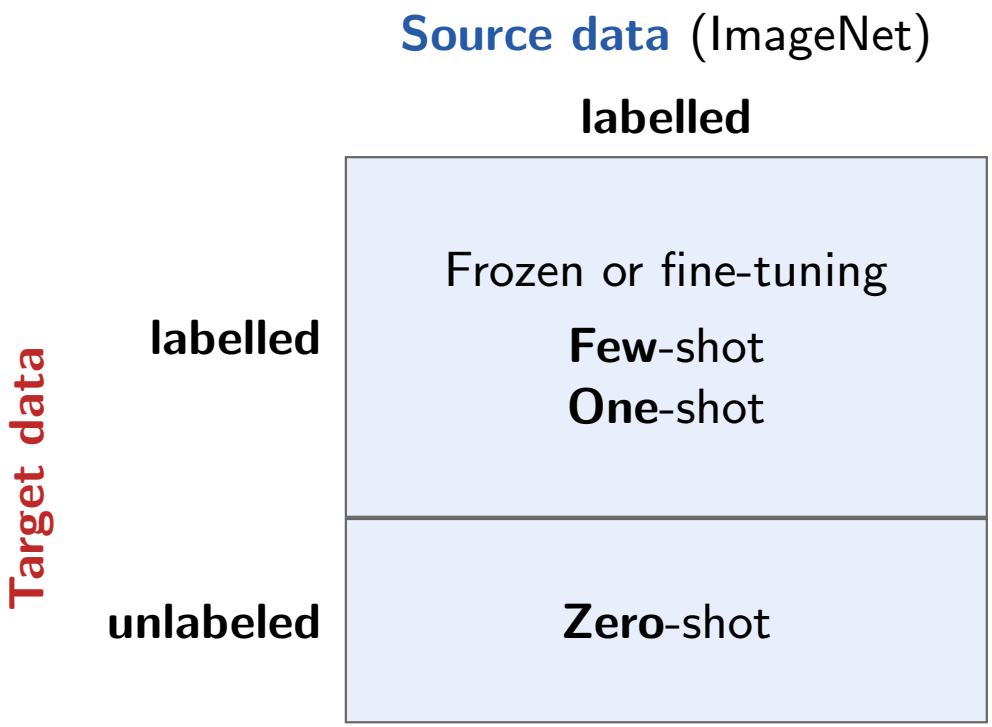
- A **domain** $\mathcal{D} = \big(\mathcal{X},\, P(X)\big)$: an input space and a marginal distribution over it
- A **task** $\mathcal{T} = \big(\mathcal{Y},\, P(Y \mid X)\big)$: a label space and the predictive function to learn

- **Transfer learning:** given a source $(\mathcal{D}_S, \mathcal{T}_S)$ and a target $(\mathcal{D}_T, \mathcal{T}_T)$ with $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$, use the knowledge acquired on the source to learn a better predictor on the target

What differs	Formally	In this lecture
Task	$\mathcal{Y}_S \neq \mathcal{Y}_T$ or $P_S(Y \mid X) \neq P_T(Y \mid X)$	ImageNet $\rightarrow$ Cal-101, VOC: deep features, fine-tuning
Domain	$P_S(X) \neq P_T(X)$, same $\mathcal{X}$ and $\mathcal{Y}$	domain adaptation: MNIST $\rightarrow$ MNIST-M, GTA5 $\rightarrow$ Cityscapes
Both		zero-shot + domain adaptation

- Fine-tuning = minimising the target risk starting from θ_S; freezing layers or a small learning rate keeps θ close to θ_S (explicit version: add $\frac{\alpha}{2}\|\theta - \theta_S\|^2$ [Li et al., 2018])

More on transfer framework



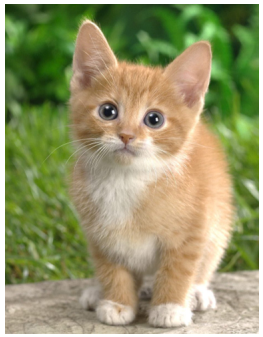
Main questions:

Similar visual domain?

Same task (*i.e.* classes)?

Similar domain: ImageNet task ⇒ Dog/Cat task

Target: Dog/Cat classifier



cat



dog

Data not directly related to the task considered





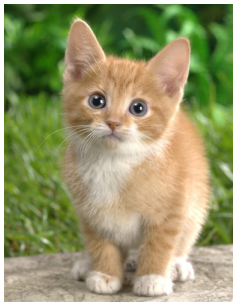




ImageNet: similar domain,
different task (1000 classes but NOT the Dog and Cat classes)

General Framework for Transfer Learning

Target: Dog/Cat classifier



cat



dog

Data not directly related to the task considered



elephant



tiger

Similar domain, completely different tasks



dog



cat

Different domains, same task

Same task, different domain: which shift?

- Same $\mathcal{X}$ and $\mathcal{Y}$ but $P_S(X, Y) \neq P_T(X, Y)$. Two factorisations:

$$P(X, Y) \;=\; P(X)\,P(Y \mid X) \;=\; P(Y)\,P(X \mid Y)$$

	changes	assumed fixed	Example
Covariate shift	$P(X)$	$P(Y \mid X)$	day $\rightarrow$ night, synthetic $\rightarrow$ real, new sensor
Label shift	$P(Y)$	$P(X \mid Y)$	class frequencies: pedestrians in a city vs on a highway
Concept shift	$P(Y \mid X)$	–	annotation conventions differ between datasets

Covariate shift: importance weighting [Shimodaira, 2000]

$$\varepsilon_T(h) \;=\; \mathbb{E}_{(x,y)\sim P_T}\big[\ell(h(x), y)\big] \;=\; \mathbb{E}_{(x,y)\sim P_S}\left[\frac{p_T(x)}{p_S(x)}\;\ell(h(x), y)\right]$$

- Requires $\mathrm{supp}(P_T) \subseteq \mathrm{supp}(P_S)$. For images the supports hardly overlap: the weights cannot be estimated
  $\Rightarrow$ learn a **representation** in which the two domains overlap

General Framework for Transfer Learning

Source data (not directly related to the task)

Target data	**labelled**	**unlabeled**
labelled	Fine-tuning Multitask learning	Self-supervised Self-taught learning
unlabeled	Domain-adversarial training Zero-shot learning	Self-taught clustering

Right column (unlabeled source data): *Not considered here*

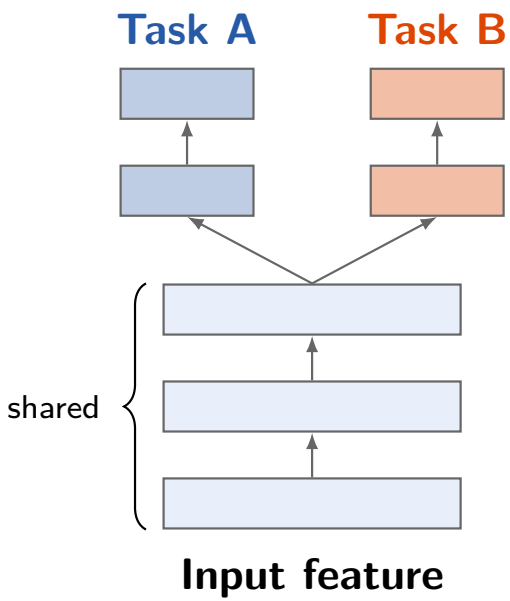
General Framework for Transfer Learning

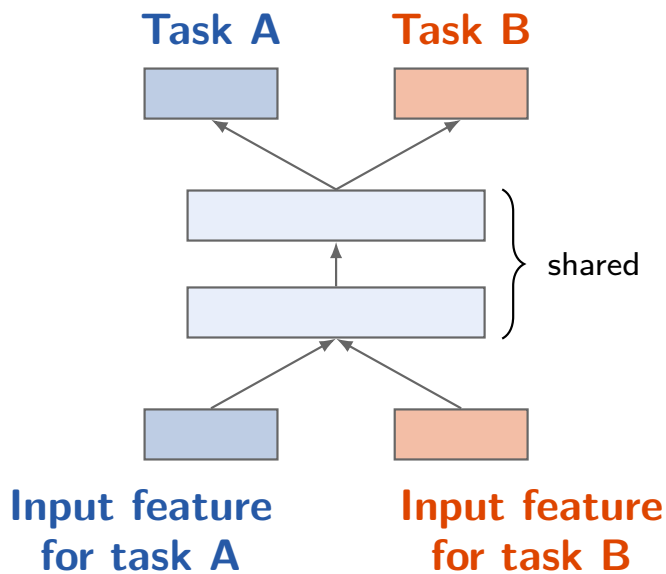
Source data (not directly related to the task)

Target data	**labelled**	**unlabeled**
labelled	**Fine-tuning** **Multitask learning**	*Not considered here*
unlabeled		*Not considered here*

Multitask Learning

- The multi-layer structure makes neural networks suitable for multitask learning





Transfer Learning – Overview

Source data (not directly related to the task)

Target data	**labelled**	**unlabeled**
labelled	Fine-tuning Multitask learning	*Not considered here*
unlabeled	**Domain adaptation** **adversarial training**	*Not considered here*

Outline

1. Transfer learning: recap
2. A general framework for transfer learning
3. Unsupervised Domain Adaptation
4. Supervised Domain Adaptation
5. Use case: domain adaptation for autonomous driving
6. Going further: domain shift after 2019

Unsupervised Domain Adaptation (UDA)

- Source data: (x^s, y^s)
- Target data: x^t

} **training data**

Same task, domain mismatch

MNIST

SOURCE



with labels

TARGET



without labels

MNIST-M

Final test on target domain!

UDA: formal setting

- **Data**

$$S = \{(x_i^s, y_i^s)\}_{i=1}^{n_s} \sim P_S(X, Y) \qquad T = \{x_j^t\}_{j=1}^{n_t} \sim P_T(X)$$

- **Hypothesis** $h = c \circ g$, with a feature extractor $g : \mathcal{X} \to \mathcal{Z}$ and a classifier $c : \mathcal{Z} \to \mathcal{Y}$

- **What we want:** a small target risk

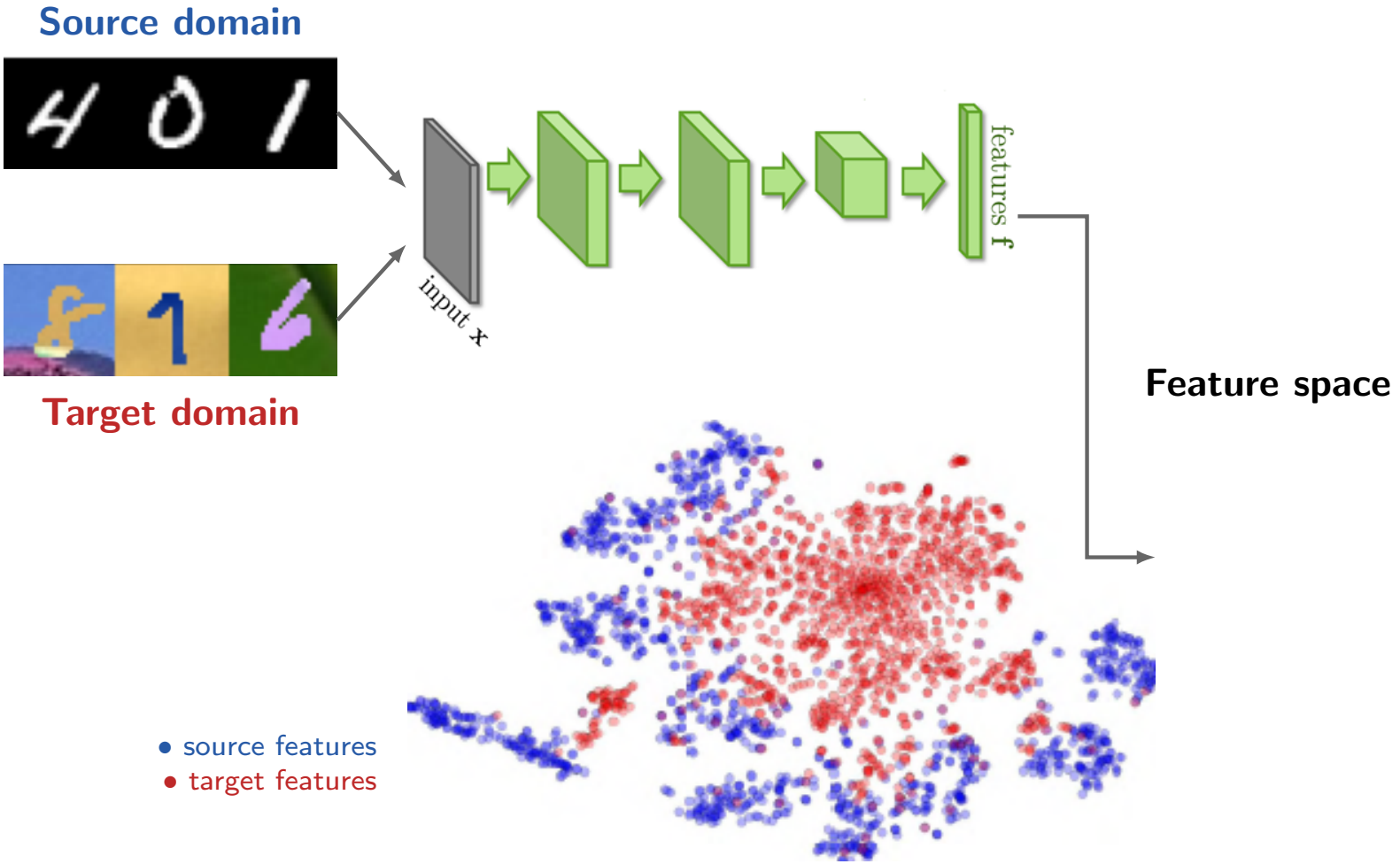
$$\varepsilon_T(h) = \mathbb{E}_{(x,y)\sim P_T}\left[\ell(h(x), y)\right] \qquad \text{✗ cannot be estimated: no } y^t$$

- **What we can minimise:** the empirical source risk

$$\hat{\varepsilon}_S(h) = \frac{1}{n_s}\sum_{i=1}^{n_s} \ell\left(h(x_i^s), y_i^s\right) \approx \varepsilon_S(h) \qquad \checkmark$$

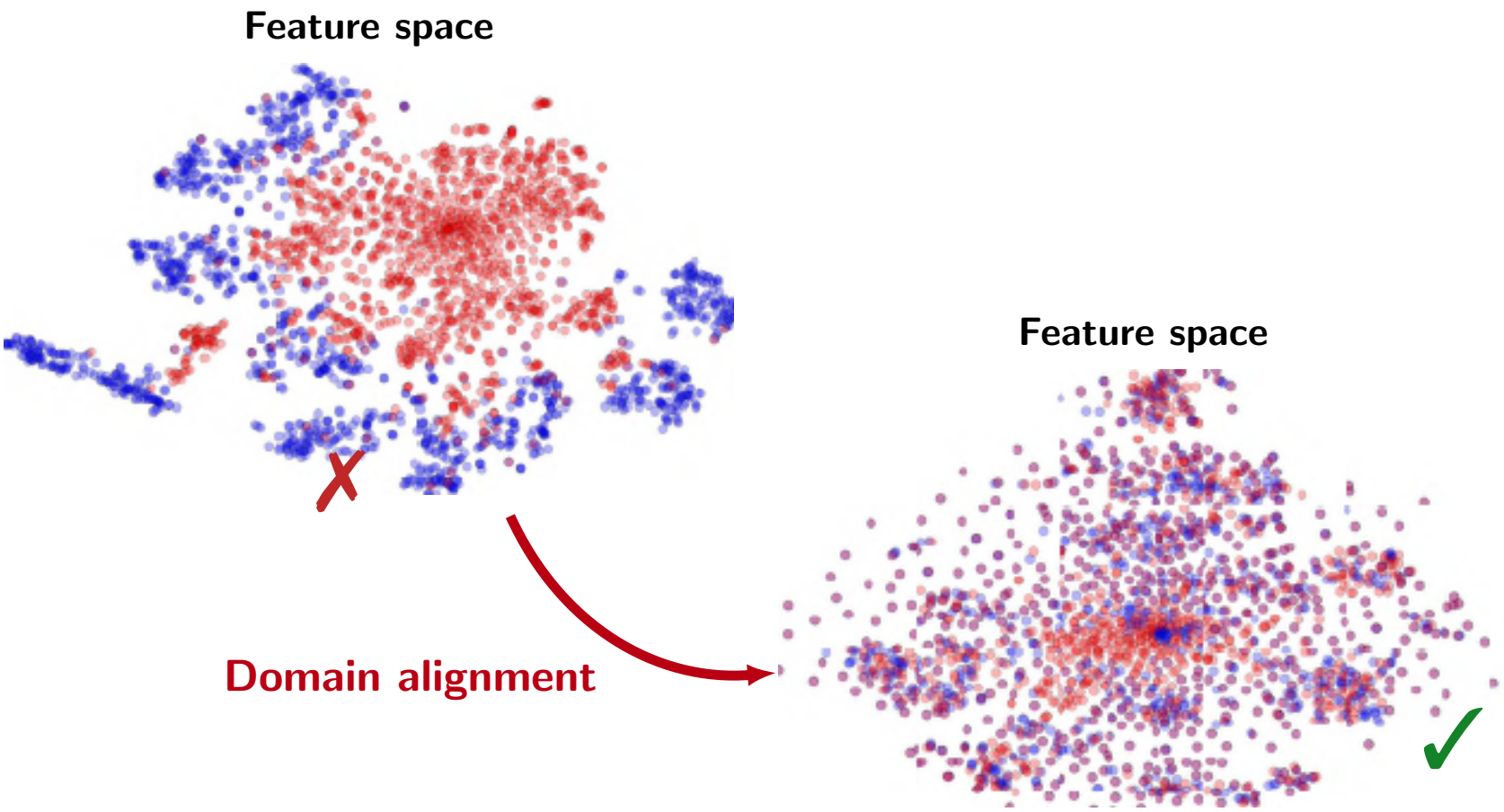
Question: when does a small $\varepsilon_S(h)$ guarantee a small $\varepsilon_T(h)$?
It depends on how different P_S and P_T are *as seen by the hypothesis class*

Unsupervised Domain Adaptation (UDA): objectives



- **Main principle:** diminish the domain shift in the learned features, encourage **domain confusion**

UDA strategy: align both domains



A bound on the target risk

Theorem [Ben-David et al., 2010] (binary classification, hypothesis class $\mathcal{H}$)

For every $h \in \mathcal{H}$:

$$\varepsilon_T(h) \leq \varepsilon_S(h) + \tfrac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(P_S, P_T) + \lambda$$

- **Source risk**: estimated with the labelled source samples
- **Divergence between domains**, measured through the hypothesis class:

$$d_{\mathcal{H}\Delta\mathcal{H}}(P_S, P_T) = 2 \sup_{h,h' \in \mathcal{H}} \left| \Pr_{x \sim P_S} \left[h(x) \neq h'(x)\right] - \Pr_{x \sim P_T} \left[h(x) \neq h'(x)\right] \right|$$

  - only needs **unlabelled** samples from both domains
- **Joint error** $\lambda = \min_{h \in \mathcal{H}} \left[\varepsilon_S(h) + \varepsilon_T(h)\right]$
  - is there one classifier that is good on *both* domains?
  - cannot be estimated without target labels: it is an **assumption**

From the bound to an algorithm

- **Estimating the divergence:** train a domain classifier D to tell source ($d = 0$) from target ($d = 1$); with ε_D its error,

$$\hat{d} = 2(1 - 2\varepsilon_D) \qquad \begin{array}{l} \varepsilon_D = 0 \Rightarrow \hat{d} = 2 \quad \text{(domains perfectly separable)} \\ \varepsilon_D = \frac{1}{2} \Rightarrow \hat{d} = 0 \quad \text{(domains indistinguishable)} \end{array}$$

- **Apply the bound in feature space** $z = g(x)$, with P_S^Z, P_T^Z the feature distributions:

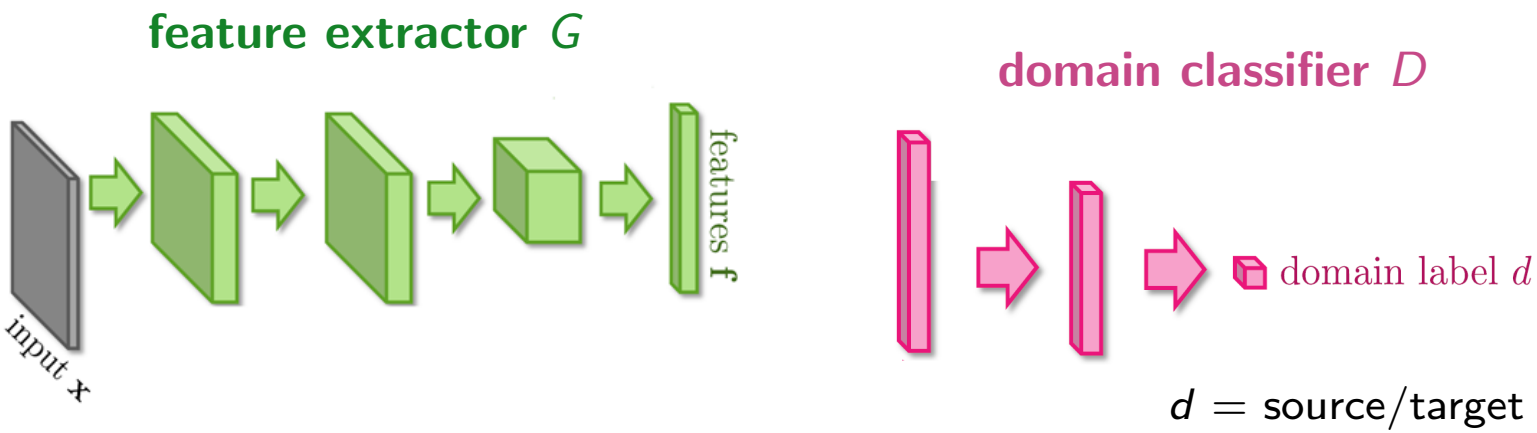
$$\varepsilon_T(c \circ g) \leq \underbrace{\varepsilon_S(c \circ g)}_{\substack{\text{label predictor} \\ \text{(source labels)}}} + \underbrace{\tfrac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(P_S^Z, P_T^Z)}_{\substack{\text{domain classifier} \\ \text{on the features}}} + \underbrace{\lambda_g}_{\substack{\text{not observable,} \\ \text{depends on } g}}$$

- $\Rightarrow$ learn g such that c is accurate on the source **and** D cannot separate the domains:
  this is **domain-adversarial training**
  [Ganin and Lempitsky, 2015, Ganin et al., 2016]

- Other ways to measure and reduce the discrepancy between P_S^Z and P_T^Z: MMD [Long et al., 2015], second-order statistics [Sun and Saenko, 2016], optimal transport [Courty et al., 2017]

UDA strategy: 1/ domain-adversarial training

- Add to the feature generator (G) a domain classifier (discriminator D) **for which labels are available!**
- Learn G and D:
  - G tries to align domains
  - D tries to identify domains

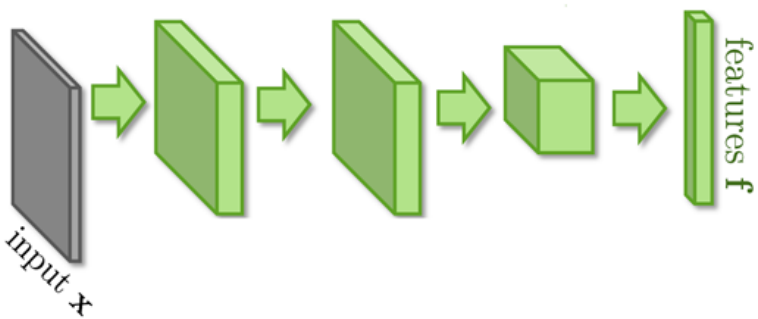


Rq: similar to GANs (coming soon)

UDA strategy: 1/ domain-adversarial training
2/ classification task (same for source and target here)

Maximize label classification accuracy
+ **minimize** domain classification accuracy

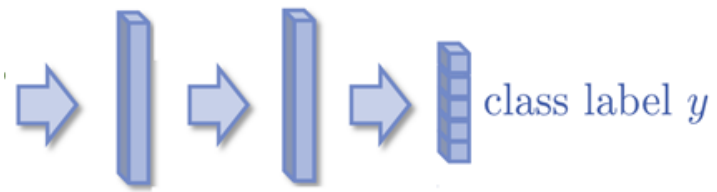
feature extractor



Not only cheat the domain classifier, but satisfy the label classifier at the same time

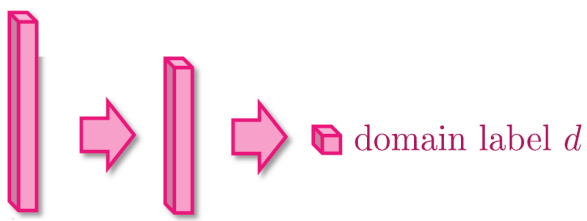
Maximize label classification accuracy

label predictor



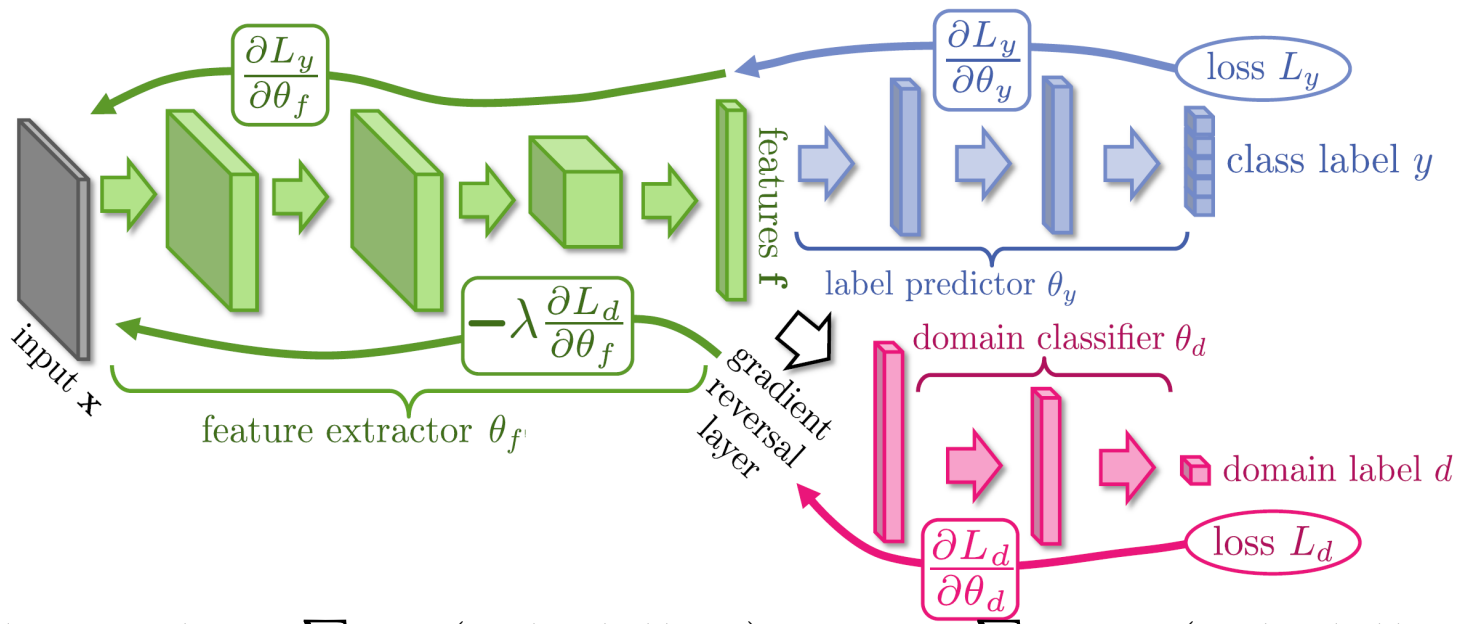
Source only!

domain classifier



Maximize domain classification accuracy

UDA strategy: joint learning



$$E(\theta_f, \theta_y, \theta_d) = \sum_{i \in \text{source}} L_y\big(G_y(G_f(x_i)),\, y_i\big) \;-\; \lambda \sum_{i \in \text{source} \cup \text{target}} L_d\big(G_d(G_f(x_i)),\, d_i\big)$$

$$(\hat{\theta}_f, \hat{\theta}_y) = \arg\min_{\theta_f, \theta_y} E(\theta_f, \theta_y, \hat{\theta}_d) \qquad \hat{\theta}_d = \arg\max_{\theta_d} E(\hat{\theta}_f, \hat{\theta}_y, \theta_d)$$

- Domain classifier fails in the end . . . but it should struggle!
- Optimisation from [Ganin and Lempitsky, 2015] (gradient reversal layer), reconsidered and better formulated in the GAN framework (later in the course)

DANN in detail (1/2): the gradient reversal layer

- Saddle point = two opposite gradient directions for θ_f. With learning rate μ:

$$\theta_f \leftarrow \theta_f - \mu\left(\frac{\partial L_y}{\partial \theta_f} - \lambda \frac{\partial L_d}{\partial \theta_f}\right) \qquad \theta_y \leftarrow \theta_y - \mu \frac{\partial L_y}{\partial \theta_y} \qquad \theta_d \leftarrow \theta_d - \mu \frac{\partial L_d}{\partial \theta_d}$$

- **Gradient reversal layer (GRL)** between G_f and G_d: identity in the forward pass, multiplies the gradient by $-\lambda$ in the backward pass

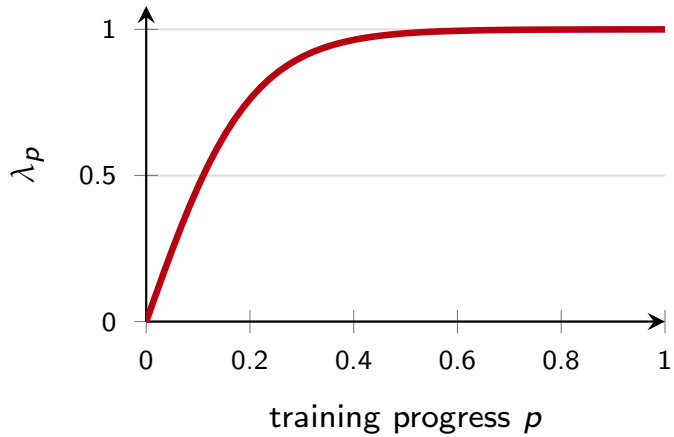
$$R_\lambda(z) = z \qquad \frac{\partial R_\lambda}{\partial z} = -\lambda\, I$$

$\Rightarrow$ plain SGD on $L_y\big(G_y(z)\big) + L_d\big(G_d(R_\lambda(z))\big)$ performs the three updates at once

DANN in detail (2/2): training recipe

- **Mini-batch:** half source (labels y, $d = 0$), half target (no label, $d = 1$)
- **Warm-up of the adversary:** the domain signal is noisy at the start, so λ grows from 0 to 1 with the training progress $p \in [0, 1]$

$$\lambda_p = \frac{2}{1 + \exp(-\gamma\, p)} - 1 \qquad (\gamma = 10)$$



- **Learning rate decay:** $\mu_p = \dfrac{\mu_0}{(1 + \alpha\, p)^\beta}$

  - **At test time:** only $G_y \circ G_f$ is kept; the domain classifier is thrown away
  - **Practical difficulties**
    - Min-max optimisation: oscillations, sensitivity to λ and to the capacity of G_d
    - **Model selection:** there is no labelled target validation set! Hyper-parameters are chosen on source accuracy + domain classifier error, or on a handful of target labels
    - A fooled domain classifier does not prove that classes are aligned (see next slides)

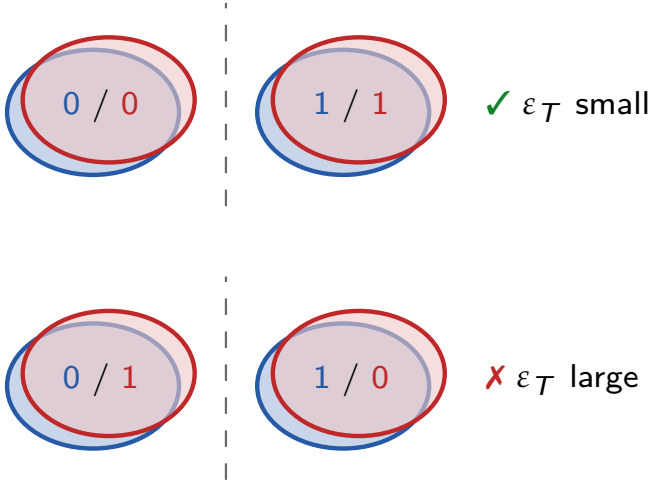
Domain-adversarial training



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

[Ganin and Lempitsky, 2015]

When is invariance not enough?



Feature space $\mathcal{Z}$: source and target classes.
Same ε_S, same (zero) divergence in both cases

- Aligning the **marginals** P_S^Z and P_T^Z says nothing about the classes: $P_S(Z \mid Y)$ and $P_T(Z \mid Y)$ may be mismatched
- In the bound: λ_g depends on g, and a flexible g can make it large while driving the two other terms to zero

Label shift makes it unavoidable [Zhao et al., 2019]

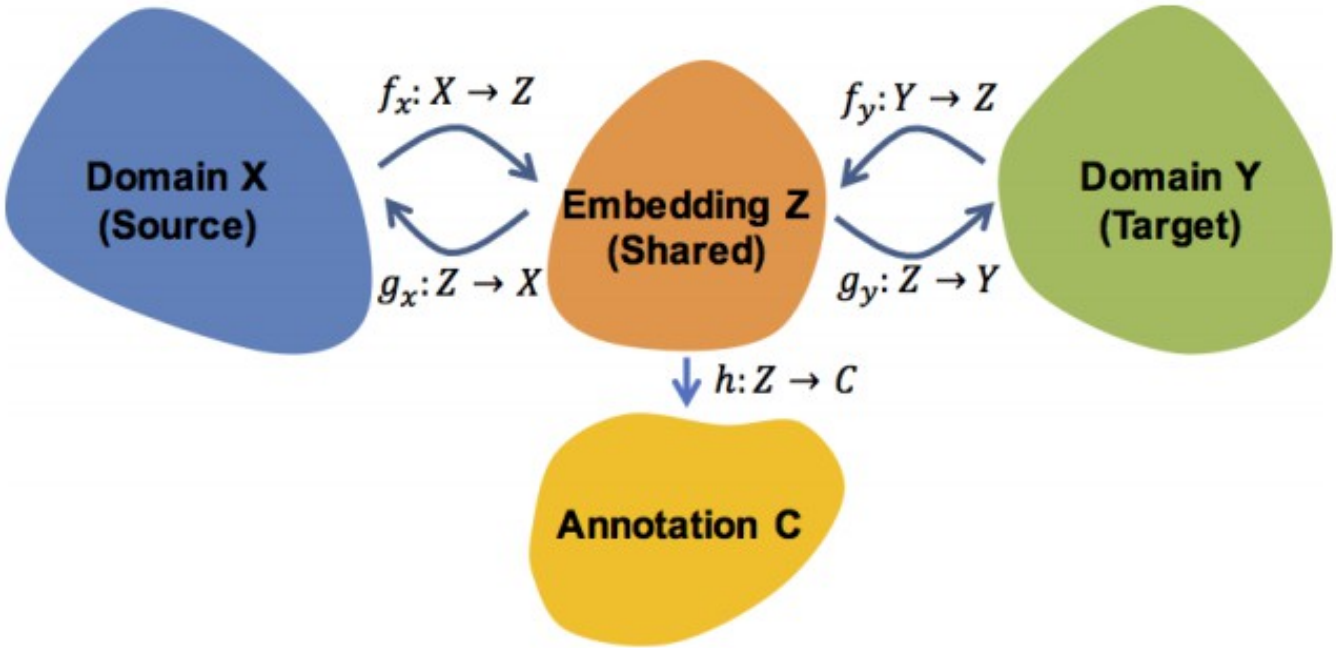
If $P_S(Y) \neq P_T(Y)$, any representation with perfectly aligned marginals satisfies

$$\varepsilon_S(c \circ g) + \varepsilon_T(c \circ g) \;\geq\; \tfrac{1}{2}\, d_{\mathrm{JS}}\big(P_S(Y), P_T(Y)\big)^2$$

- $\Rightarrow$ use the structure of the target predictions: class-conditional alignment, **entropy minimisation**, pseudo-labels (use case and last section)

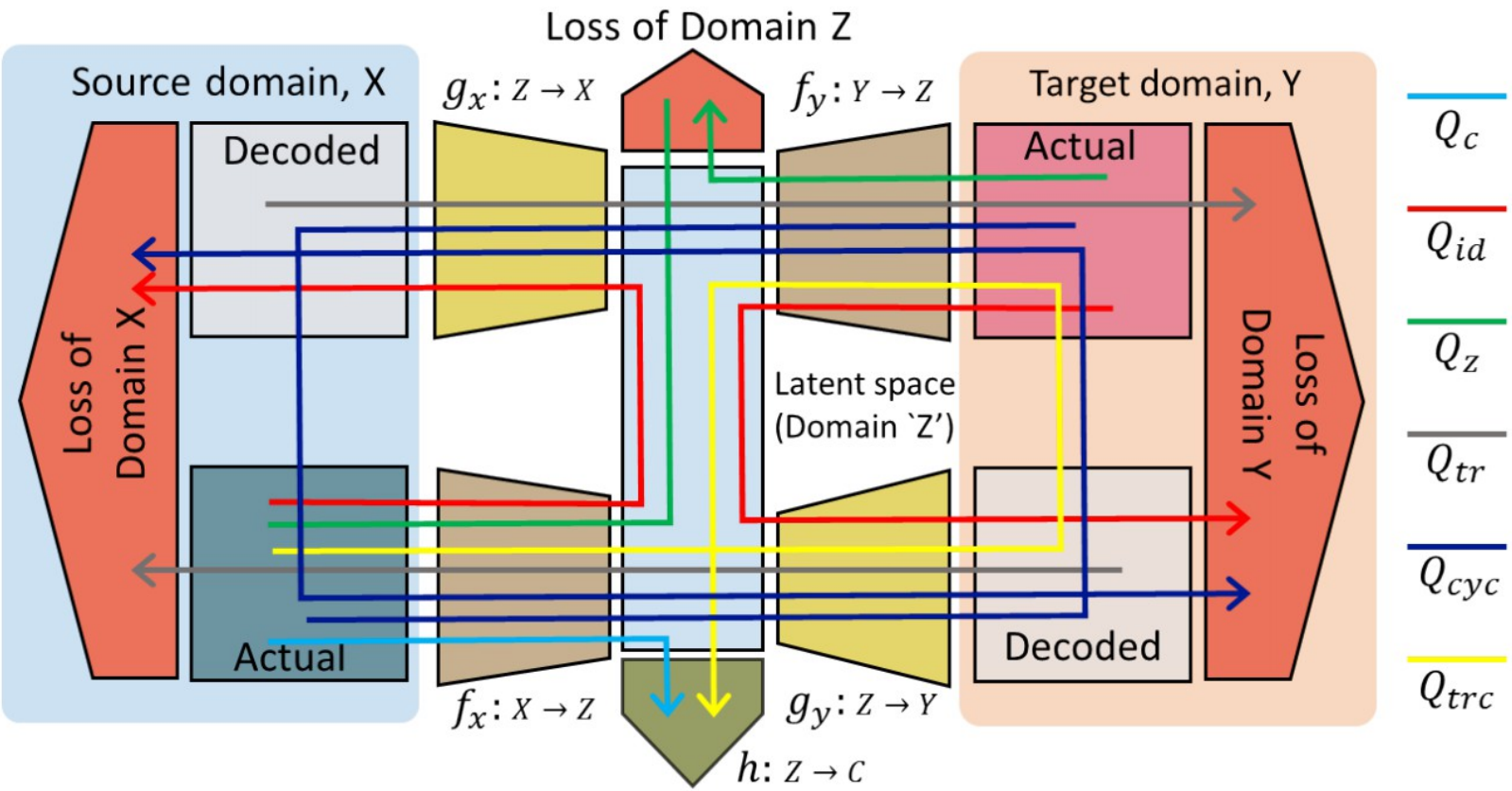
Domain adaptation

- General formulation



Domain adaptation

- General formulation



Outline

1. Transfer learning: recap
2. A general framework for transfer learning
3. Unsupervised Domain Adaptation
4. Supervised Domain Adaptation
5. Use case: domain adaptation for autonomous driving
6. Going further: domain shift after 2019

Supervised Domain Adaptation (SDA): setting

- **Data:** same task, two domains, and a **few labels in the target**

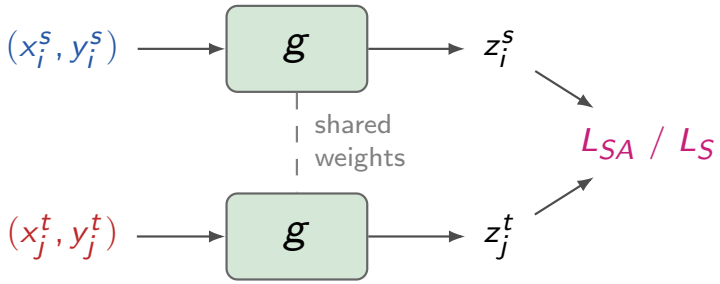
$$S = \{(x_i^s, y_i^s)\}_{i=1}^{n_s} \sim P_S(X, Y) \qquad T = \{(x_j^t, y_j^t)\}_{j=1}^{n_t} \sim P_T(X, Y)$$

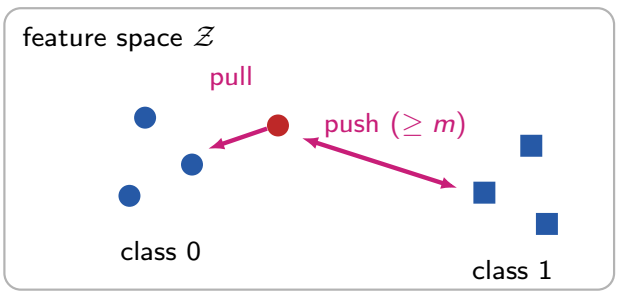
  - $n_t \ll n_s$: typically 1 to 10 target samples per class
  - *semi-supervised* DA: the same, plus unlabelled target samples
- **Baselines**
  - Train on T only: overfits
  - Train on $S \cup T$: the target samples are drowned ($n_t \ll n_s$)
  - Train on S, then fine-tune on T: strong when n_t is not too small
- **What the target labels buy**
  - UDA can only align the marginals P_S^Z and P_T^Z
  - With labels we can align **class by class**: $P_S(Z \mid Y = k) \approx P_T(Z \mid Y = k)$
    $\Rightarrow$ removes the failure case of marginal alignment (swapped classes)

Generic objective

$$\min_{g,\,c} \; L_C(c \circ g;\, S) + L_C(c \circ g;\, T) + \gamma\, L_{align}(g;\, S, T, \text{labels})$$

SDA method 1: contrastive semantic alignment (CCSA)





- Siamese network: every target sample is paired with every source sample [Motiian et al., 2017b]
- **Semantic alignment**: same class, different domains $\Rightarrow$ close

$$L_{SA} = \sum_{y_i^s = y_j^t} \frac{1}{2} \|z_i^s - z_j^t\|^2$$

- **Separation**: different classes $\Rightarrow$ at least a margin m apart

$$L_S = \sum_{y_i^s \neq y_j^t} \frac{1}{2} \max\left(0,\ m - \|z_i^s - z_j^t\|\right)^2$$

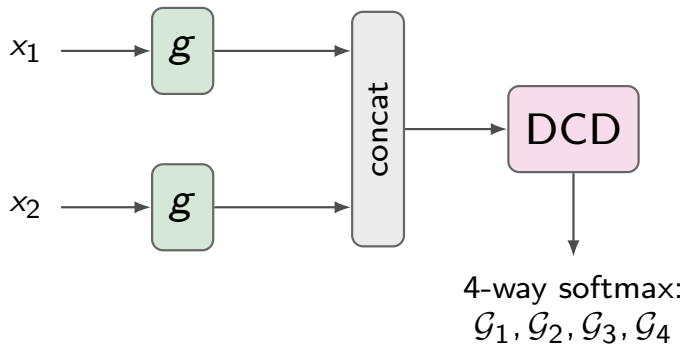
Total loss: $L_{CCSA} = (1-\gamma)\, L_C(c \circ g) + \gamma \left(L_{SA}(g) + L_S(g)\right) \qquad z = g(x)$

- Too few target samples to estimate $P_T(Z \mid Y)$: **point-wise** distances replace distribution distances
- $n_s \times n_t$ pairs from n_t target samples: limits overfitting, works with 1 sample per class

SDA method 2: few-shot adversarial domain adaptation (FADA)

- Too few target samples to train a source-vs-target discriminator as in DANN
- **Idea:** discriminate **pairs** of samples instead [Motiian et al., 2017a]

Group	domains	classes
$\mathcal{G}_1$	source – source	same
$\mathcal{G}_2$	source – target	same
$\mathcal{G}_3$	source – source	different
$\mathcal{G}_4$	source – target	different



DCD = domain-class discriminator

1. Pre-train g and c on the source
2. Freeze g, train the DCD to recognise the group of each pair (4-class cross-entropy)
3. Freeze the DCD, update g and c so that $\mathcal{G}_2$ is classified as $\mathcal{G}_1$ and $\mathcal{G}_4$ as $\mathcal{G}_3$:

$$L_{g,c} = -\gamma\, \mathbb{E}\big[\log D_{\mathcal{G}_1}(g(\mathcal{G}_2)) + \log D_{\mathcal{G}_3}(g(\mathcal{G}_4))\big] + L_C(c \circ g; S) + L_C(c \circ g; T)$$

- Confusing $\mathcal{G}_1/\mathcal{G}_2$ and $\mathcal{G}_3/\mathcal{G}_4$ = **domain confusion**; keeping $\{\mathcal{G}_1, \mathcal{G}_2\}$ apart from $\{\mathcal{G}_3, \mathcal{G}_4\}$ = **class separation**

Domain adaptation methods: summary

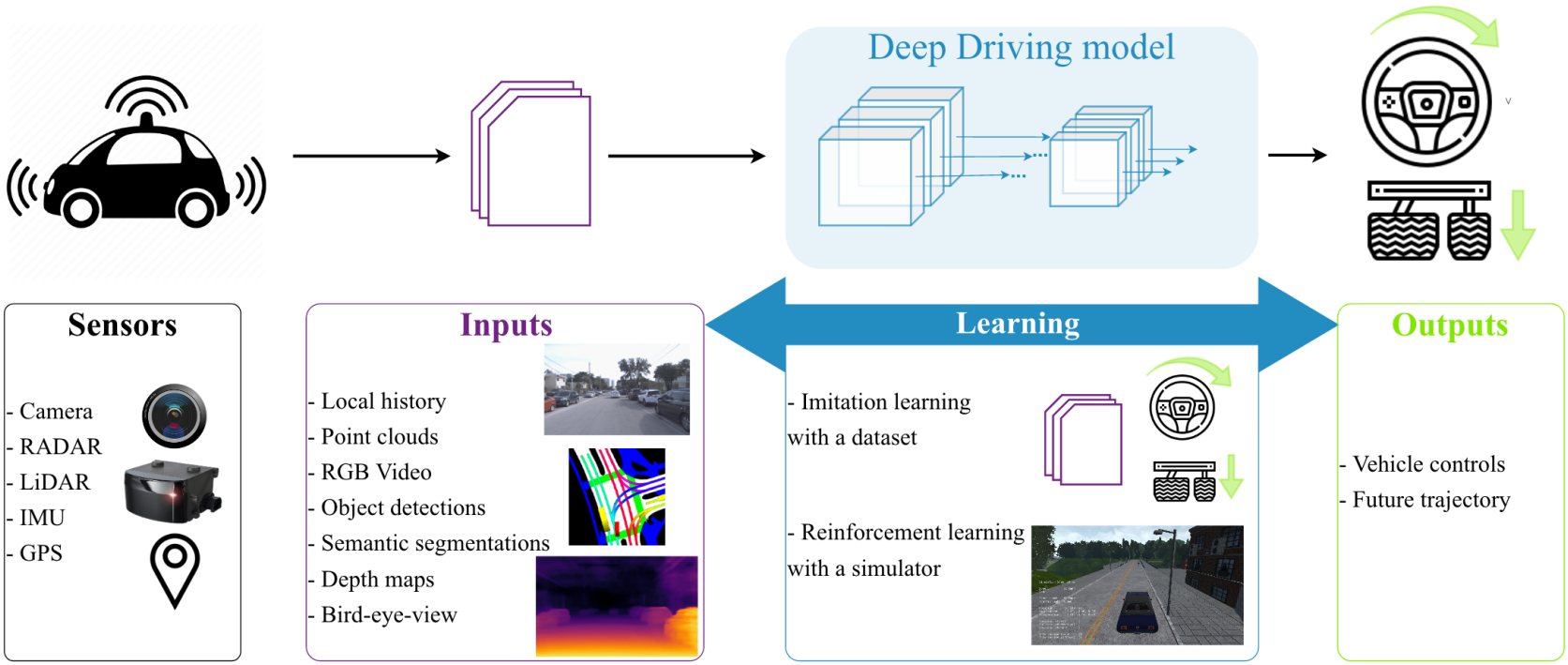
	target labels	what is aligned	how
DANN	none	feature marginals P_S^Z, P_T^Z	domain classifier + gradient reversal
ADVENT	none	entropy maps of the predictions (output space)	entropy loss, or discriminator on self-information maps
CCSA	few per class	class-conditional features, point-wise	contrastive loss on source–target pairs
FADA	few per class	class-conditional features	4-way discriminator on pairs

- Two families of alignment criteria: **adversarial** (DANN, AdvEnt, FADA) and **explicit distances / losses** (MinEnt, CCSA; also MMD, CORAL, optimal transport)
- **Without** target labels: only marginal or output statistics can be matched; class alignment relies on the classifier's own predictions
- **With** a few target labels: alignment becomes class-conditional, and a small labelled target set also solves the model selection problem

Outline

1. Transfer learning: recap
2. A general framework for transfer learning
3. Unsupervised Domain Adaptation
4. Supervised Domain Adaptation
5. Use case: domain adaptation for autonomous driving
6. Going further: domain shift after 2019

Context: neural network-based autonomous driving system framework



Domain gap

- Different, though related, input data distributions

Source domain $\rightarrow$ **Target domain**

- Different weather, light, location, sensor's spec/setup





Domain gap = distribution shift between training and deployment data

Domain gap

- Different, though related, input data distributions

Source domain $\rightarrow$ **Target domain**

- Different weather, light, location, sensor's spec/setup





Domain gap = distribution shift between training and deployment data

Domain gap

- Different, though related, input data distributions

Source domain $\rightarrow$ **Target domain**

- Different weather, light, location, sensor's spec/setup





Domain gap = distribution shift between training and deployment data

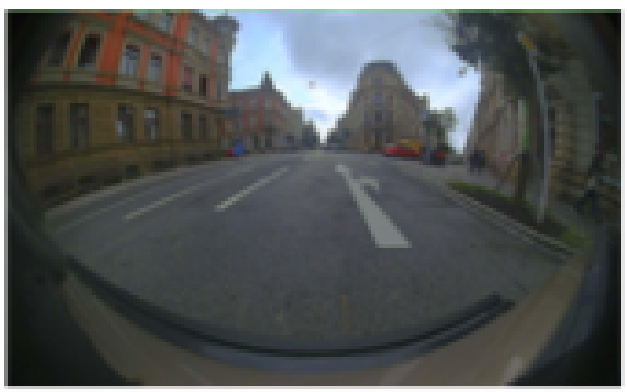
Domain gap

- Different, though related, input data distributions

Source domain $\rightarrow$ **Target domain**

- Different weather, light, location, sensor's spec/setup





Domain gap = distribution shift between training and deployment data

Domain gap

- Different, though related, input data distributions

Source domain $\rightarrow$ **Target domain**

- **Synthetic vs. real**



synthetic image (GTA5)



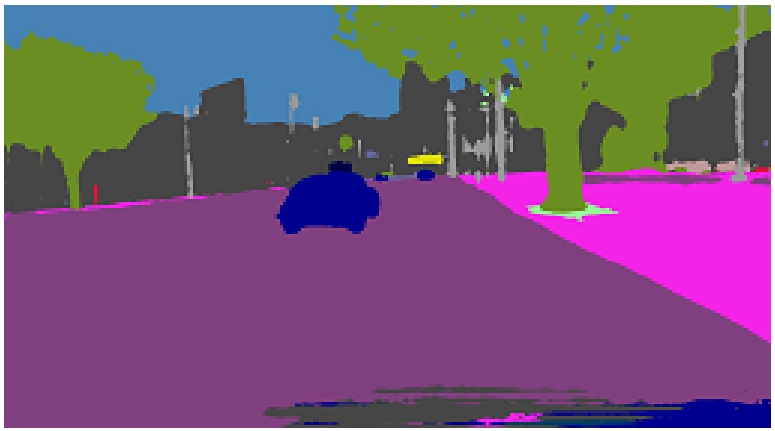
real image (Cityscapes)

Domain gap for visual segmentation

- Different, though related, input data distributions

  Source domain $\rightarrow$ **Target domain**

- **Synthetic vs. real**



synthetic: labels for free



real image (Cityscapes)



Domain gap for visual segmentation

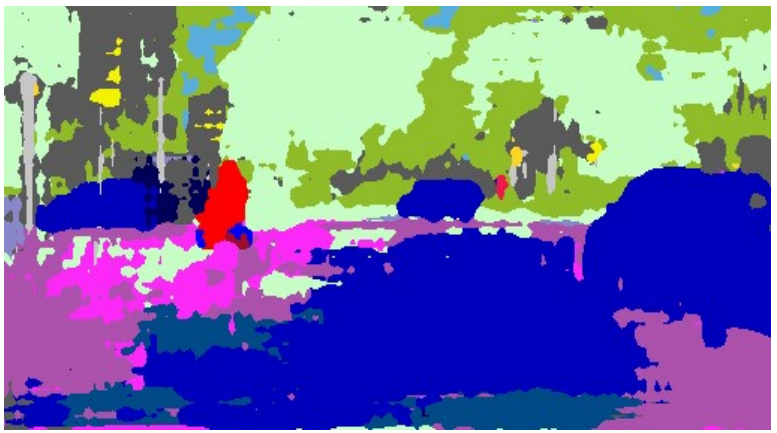
- Different, though related, input data distributions

  Source domain $\rightarrow$ **Target domain**

- **Synthetic vs. real**



synthetic: labels for free

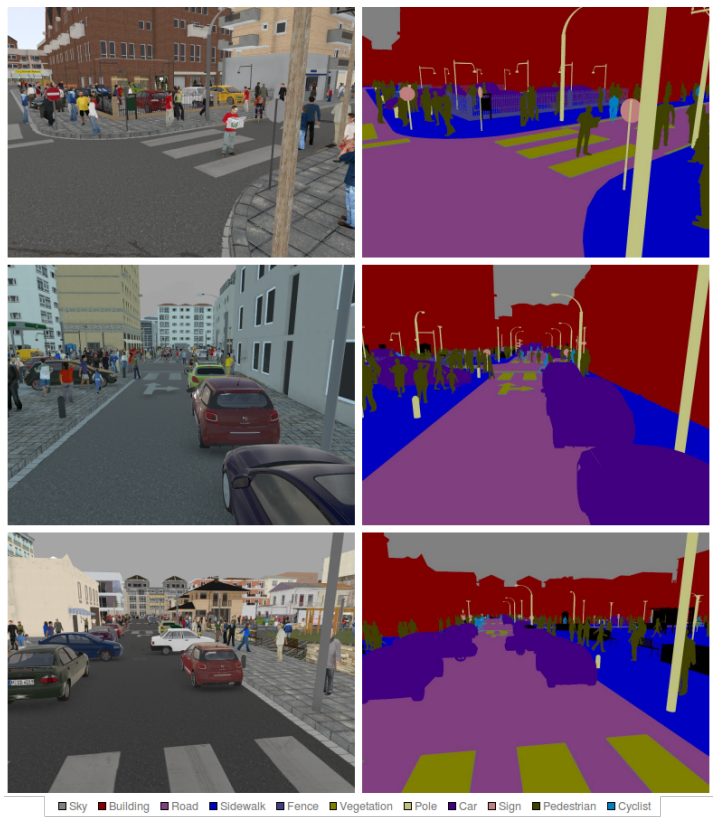


real: prediction of the source-only model



Unsupervised Domain Adaptation (UDA)

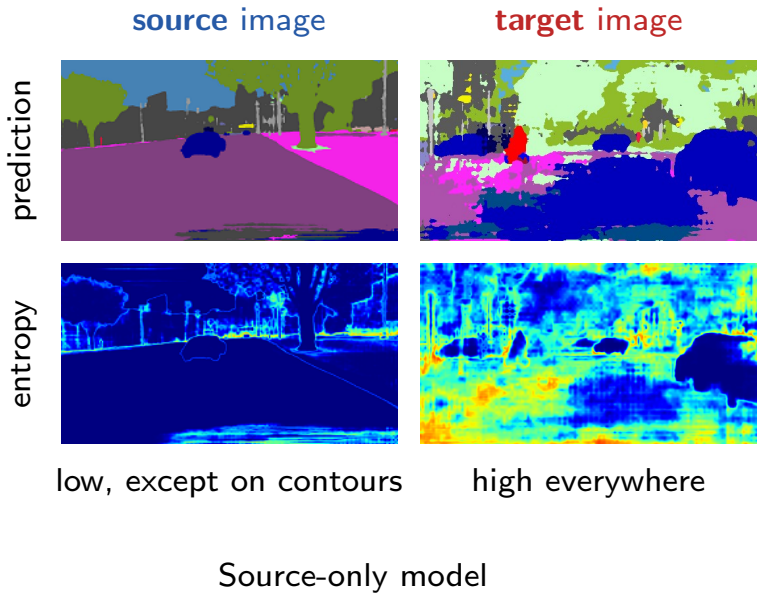
Labelled source domain data



Unlabelled target domain data



UDA for segmentation in detail (1/2): entropy minimisation



- Segmentation network F: soft-segmentation map $P_x \in [0,1]^{H \times W \times C}$
- **Observation:** over-confident on the source, under-confident on the target
- Pixel-wise **entropy map**

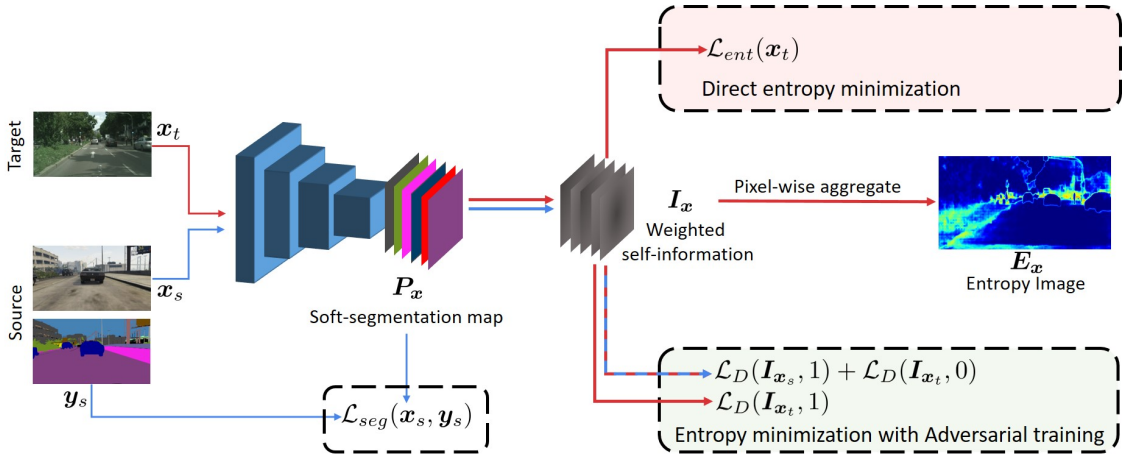
$$E_x^{(h,w)} = \frac{-1}{\log C} \sum_{c=1}^{C} P_x^{(h,w,c)} \log P_x^{(h,w,c)}$$

Direct entropy minimisation (MinEnt) [Vu et al., 2019]

$$\min_{\theta_F} \frac{1}{|\mathcal{X}_s|} \sum_{x_s} L_{seg}(x_s, y_s) + \frac{\lambda_{ent}}{|\mathcal{X}_t|} \sum_{x_t} \sum_{h,w} E_{x_t}^{(h,w)}$$

- Entropy as an unsupervised loss on the target [Grandvalet and Bengio, 2004]: pushes decision boundaries away from dense regions; a soft form of self-training with pseudo-labels

UDA for segmentation in detail (2/2): adversarial entropy minimisation



- MinEnt treats pixels independently. **AdvEnt**: align the *distribution* of weighted self-information maps $I_x = -P_x \cdot \log P_x$ between domains with a fully-convolutional discriminator D

$$\min_{\theta_D} \frac{1}{|\mathcal{X}_s|} \sum_{x_s} L_D(I_{x_s}, 1) + \frac{1}{|\mathcal{X}_t|} \sum_{x_t} L_D(I_{x_t}, 0)$$

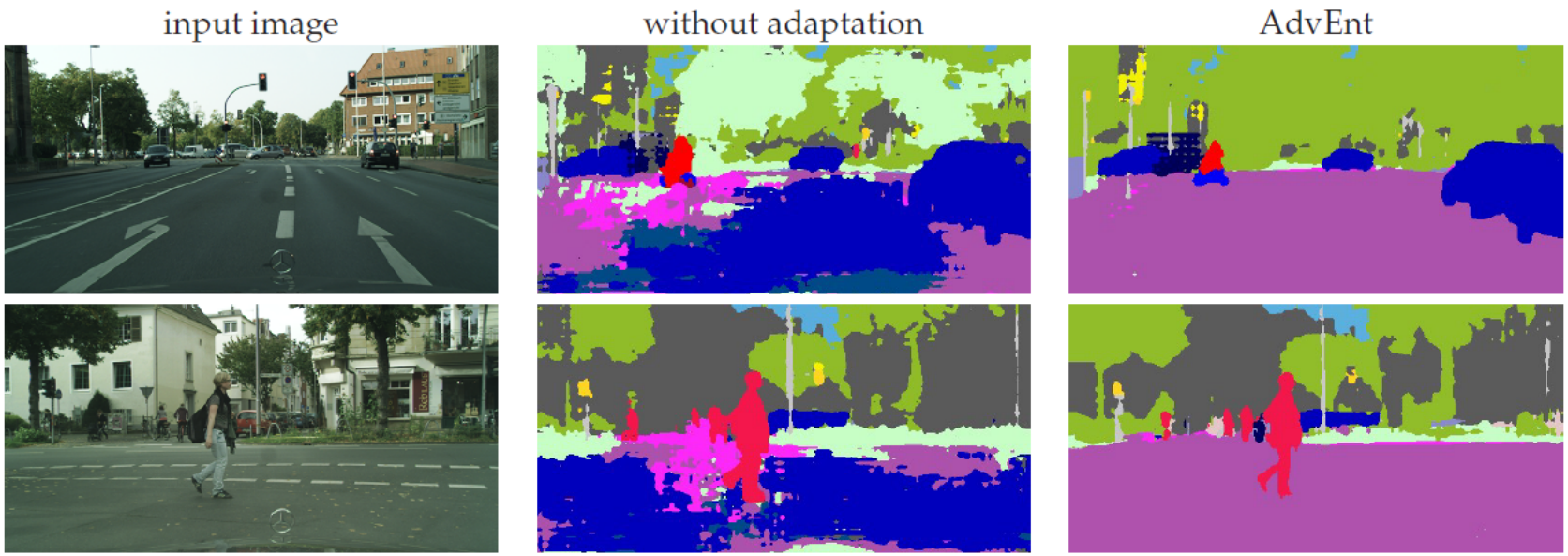
D: source (1) or target (0)?

$$\min_{\theta_F} \frac{1}{|\mathcal{X}_s|} \sum_{x_s} L_{seg}(x_s, y_s) + \frac{\lambda_{adv}}{|\mathcal{X}_t|} \sum_{x_t} L_D(I_{x_t}, 1)$$

F: make target maps look like source

- Same adversarial game as DANN, played in the **output space**: the layout of a street scene (road below, sky above) is shared by both domains, unlike low-level appearance

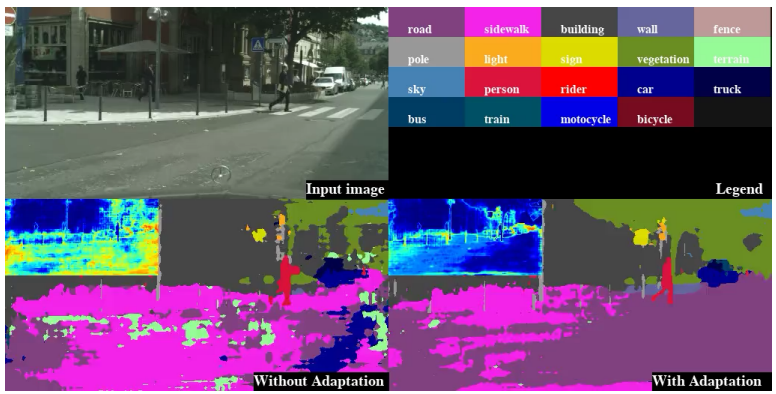
Qualitative results



UDA results with Adversarial Entropy Minimization (ADVENT)

GTA5 → Cityscapes

Method	mIoU (%)
FCNs in the wild [Hoffman et al., 2016]	27.1
Curriculum learning [Zhang et al., 2017]	29.0
CyCADA [Hoffman et al., 2018]	34.8
Adversarial adaptation on output [Tsai et al., 2018]	42.4
ADVENT [Vu et al., 2019]	**44.8**



▶ demo video (media/advent_demo.mp4)

Extension: zero-shot + domain adaptation

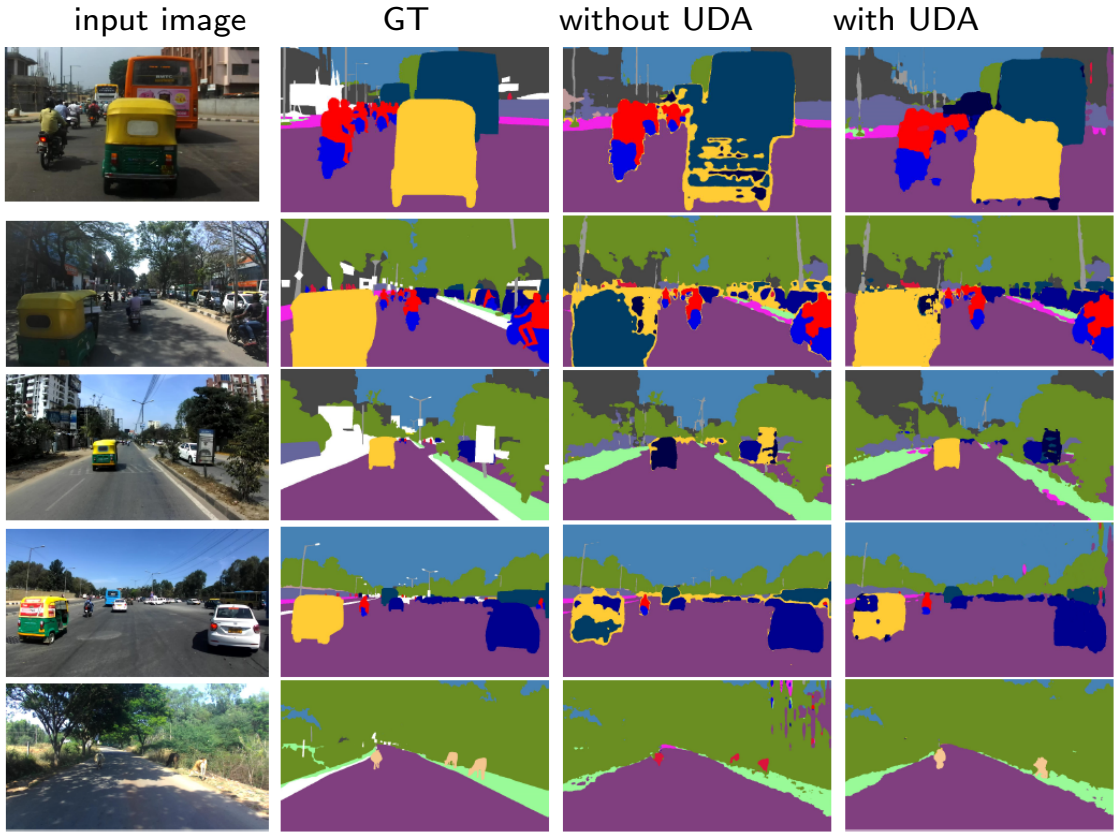


Fig. 2: **Additional qualitative results on IDD**. Similar arrangement as in Figure 1. Private target classes: tuk-tuk, animal. Some shared classes: truck, road, side walk, car, person, motorbike, tree, building.

- Target domain contains **private classes** never seen in the source (*e.g.* tuk-tuk, animal)
- Zero-shot segmentation (in transductive form) combined with self-training UDA [Bucher et al., 2020]

Transfer Learning – Overview

Source data (not directly related to the task)

Target data	**labelled**	**unlabeled**
labelled	Fine-tuning Multitask learning	*Not considered here*
unlabeled	Domain adaptation adversarial training Zero-shot learning	*Not considered here*

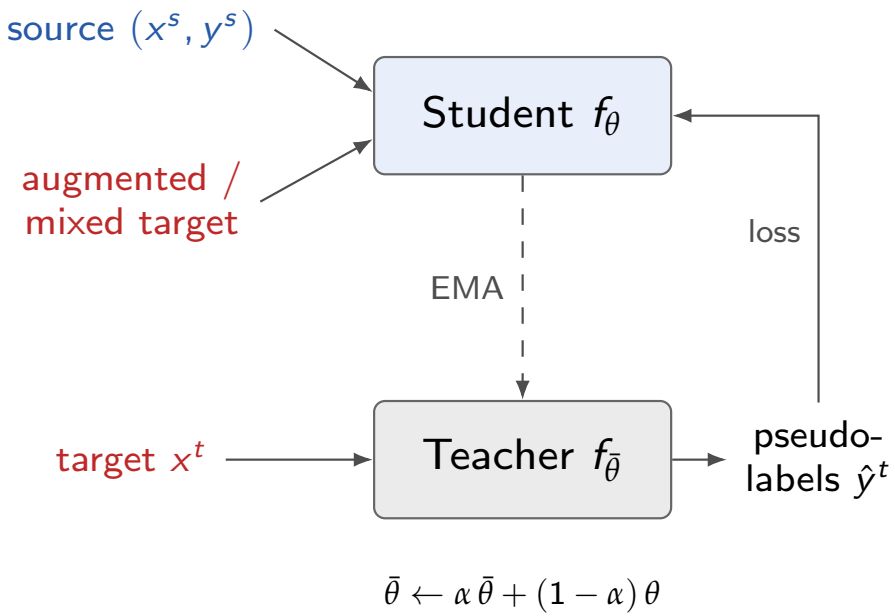
Outline

1. Transfer learning: recap
2. A general framework for transfer learning
3. Unsupervised Domain Adaptation
4. Supervised Domain Adaptation
5. Use case: domain adaptation for autonomous driving
6. Going further: domain shift after 2019

What changed since ADVENT (2019)?

1. **Method: from adversarial alignment to self-training**
   - Pseudo-labels on the target + mean teacher + strong augmentation
   - Transformer backbones instead of ResNet/DeepLab
2. **Setting: fewer assumptions on what is available**
   - No source data at adaptation time (*source-free*)
   - Adapt on the fly, during inference (*test-time adaptation*)
   - No target data at all (*domain generalization*)
3. **Starting point: foundation models**
   - Pre-training on 10^8–10^9 images already covers many "target" domains
   - The question moves from *how to align two domains* to *how to fine-tune without losing robustness*
   - Language gives a handle on domains we have no image of

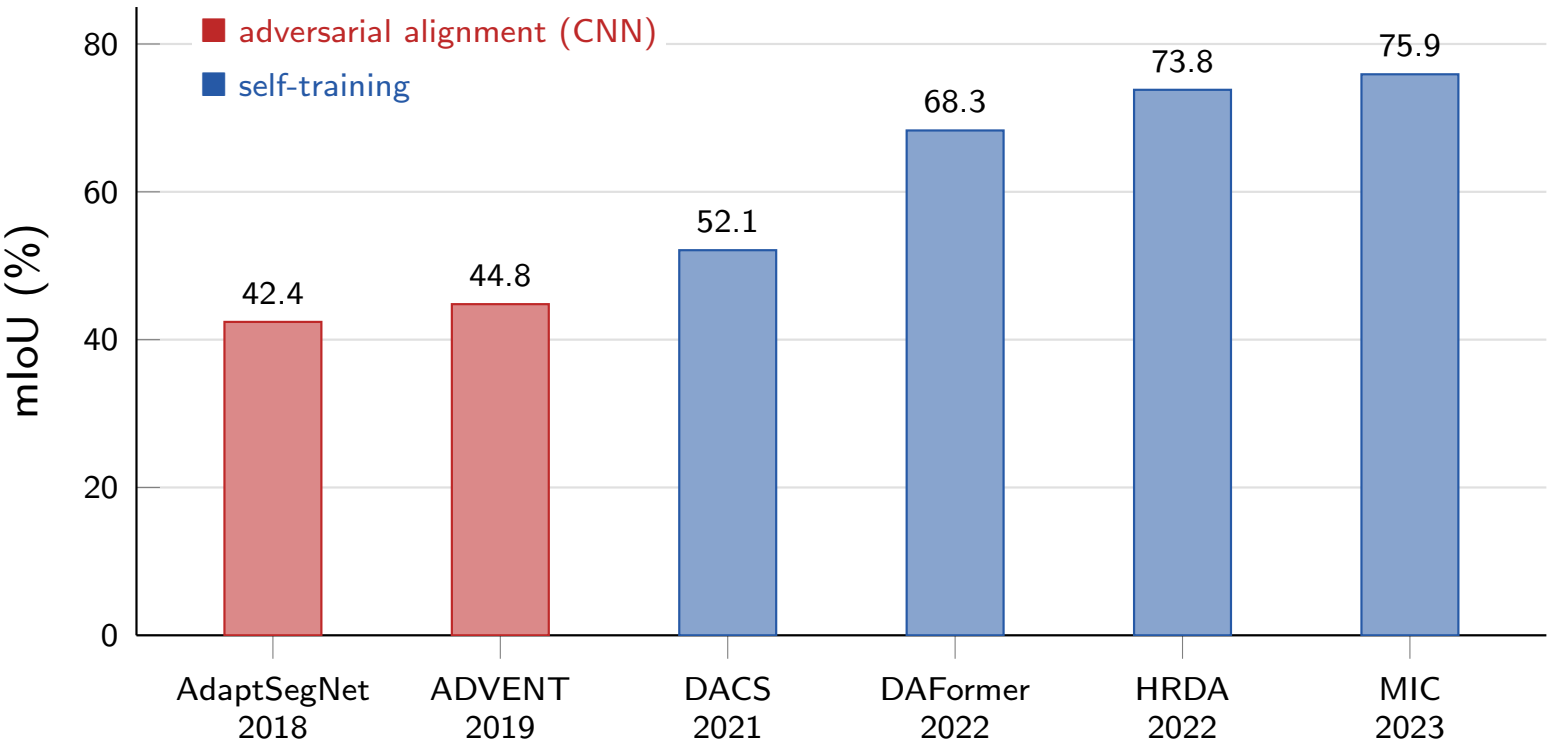
1/ Self-training for UDA



- Teacher = moving average of the student
  [Tarvainen and Valpola, 2017] $\Rightarrow$ stable pseudo-labels
- **DACS** [Tranheden et al., 2021]: paste source objects onto target images
- **DAFormer** [Hoyer et al., 2022a]: transformer backbone, rare class sampling
- **HRDA** [Hoyer et al., 2022b]: high-resolution crops + low-resolution context
- **MIC** [Hoyer et al., 2023]: mask target patches, predict them from context

$\Rightarrow$ no discriminator any more: the gap is closed through the **labels**, not the features

1/ Self-training for UDA: GTA5 → Cityscapes



- +30 points in four years on the benchmark of the previous slides
- The largest single jump (DACS → DAFormer) comes mostly from the **backbone and its pre-training**

2/ Relaxing the setting: test-time adaptation

	source data	target data	target labels	
Fine-tuning	–	train set	✓	
UDA	✓	train set	✗	DANN, ADVENT, MIC
Source-free DA	✗	train set	✗	SHOT [Liang et al., 2020]
Test-time adaptation	✗	**test stream**	✗	TENT [Wang et al., 2021]
Domain generalization	✓	✗	✗	

Why? In deployment:

- source data cannot be shipped (size, privacy, licence)
- the target domain is unknown and *changes*: day, dusk, rain, tunnel...

TENT: for each test batch, minimise the **entropy** of the predictions

$$\min_{\gamma,\beta} \; -\sum_{c} p_\theta(c \mid x^t) \, \log p_\theta(c \mid x^t)$$

updating only the normalisation parameters (γ, β)

- Same criterion as ADVENT, without source data and without discriminator
- **Limits:** error accumulation, collapse on small or non-i.i.d. batches; continual shift [Wang et al., 2022]

3/ Foundation models: is there still a gap to adapt?

- **Large-scale pre-training is the strongest "adaptation" method**
  - CLIP [Radford et al., 2021]: zero-shot classifier far more stable under distribution shift (sketches, renditions, new acquisitions) than an ImageNet-supervised model of the same accuracy
  - DINOv2 [Oquab et al., 2024]: frozen features + linear head transfer across domains

- **Back to the first part of this lecture: how do we fine-tune?**
  - Full fine-tuning improves in-distribution accuracy but can *distort* pre-trained features and lose robustness out of distribution [Kumar et al., 2022]
  - Linear probing then fine-tuning (LP-FT); interpolating zero-shot and fine-tuned weights (WiSE-FT) [Wortsman et al., 2022]
  - Parameter-efficient fine-tuning: frozen backbone + a few trainable tokens / low-rank updates

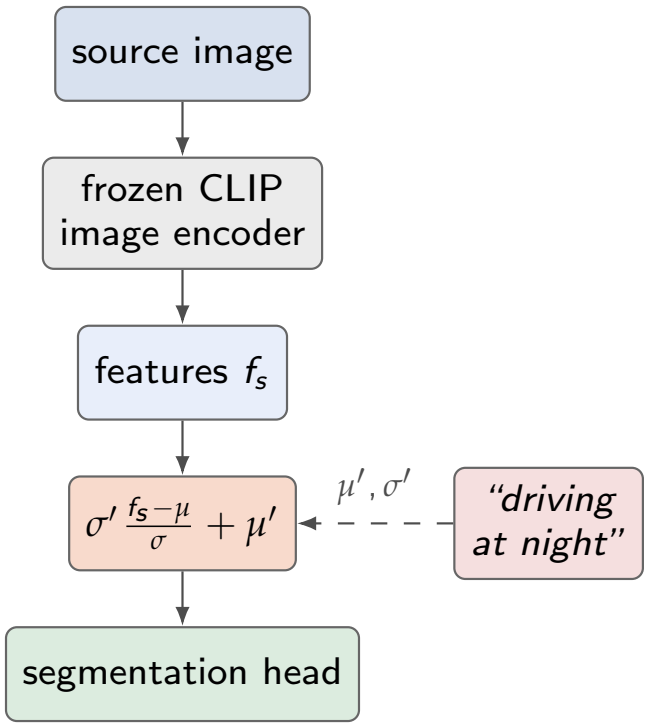
- **Driving scenes:** a frozen DINOv2 with $\approx 1\%$ extra trainable parameters, trained on synthetic GTA5 only, generalises to real datasets without seeing a single target image (Rein [Wei et al., 2024])

$\Rightarrow$ domain *generalization* with a foundation model competes with domain *adaptation*

3/ Language as a bridge between domains

Prompt-driven zero-shot adaptation
(PØDA [Fahes et al., 2023])

- The target domain is only *described*: *"driving at night"*, *"driving in snow"*
- CLIP image encoder as frozen backbone
- Shift the statistics (μ, σ) of source features so that their CLIP embedding moves towards the text embedding of the prompt
- Fine-tune the segmentation head on these stylised features



$\Rightarrow$ no target image, not even unlabelled

- Same idea as the zero-shot + DA extension, with the domain itself given by language

Take-away

	2015–2019	**Today**
Backbone	CNN pre-trained on ImageNet-1k	ViT foundation model (CLIP, DINOv2)
Closing the gap	align feature / output distributions with a discriminator	pseudo-labels, consistency, masking; or nothing but a good pre-training
Target data	unlabelled set, known in advance	test stream, a text prompt, or none
What is trained	the whole network	a head, normalisation layers, a few tokens
Benchmark	GTA5 $\rightarrow$ Cityscapes	adverse conditions (night, fog, rain, snow) [Sakaridis et al., 2021], continual shift

- The ideas of this lecture are still there: **entropy minimisation**, **frozen features**, **small learning rate on the backbone**
- What changed is the scale of the source: the "source domain" is now most of the web

References I



Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. (2010).

A theory of learning from different domains.
Machine Learning, 79.



Bucher, M., Vu, T.-H., Cord, M., and Pérez, P. (2020).

BUDA: Boundless unsupervised domain adaptation in semantic segmentation.
arXiv preprint arXiv:2004.01130.



Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. (2017).

Optimal transport for domain adaptation.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(9).



Fahes, M., Vu, T.-H., Bursuc, A., Pérez, P., and de Charette, R. (2023).

PØDA: Prompt-driven zero-shot domain adaptation.
In *IEEE/CVF International Conference on Computer Vision (ICCV)*.



Ganin, Y. and Lempitsky, V. (2015).

Unsupervised domain adaptation by backpropagation.
In *International Conference on Machine Learning (ICML)*.



Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016).

Domain-adversarial training of neural networks.
Journal of Machine Learning Research, 17.



Grandvalet, Y. and Bengio, Y. (2004).

Semi-supervised learning by entropy minimization.
In *Advances in Neural Information Processing Systems (NeurIPS)*.



Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A. A., and Darrell, T. (2018).

CyCADA: Cycle-consistent adversarial domain adaptation.
In *International Conference on Machine Learning (ICML)*.

References II



Hoffman, J., Wang, D., Yu, F., and Darrell, T. (2016).

FCNs in the wild: Pixel-level adversarial and constraint-based adaptation.
arXiv preprint arXiv:1612.02649.



Hoyer, L., Dai, D., and Van Gool, L. (2022a).

DAFormer: Improving network architectures and training strategies for domain-adaptive semantic segmentation.
In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.



Hoyer, L., Dai, D., and Van Gool, L. (2022b).

HRDA: Context-aware high-resolution domain-adaptive semantic segmentation.
In *European Conference on Computer Vision (ECCV)*.



Hoyer, L., Dai, D., Wang, H., and Van Gool, L. (2023).

MIC: Masked image consistency for context-enhanced domain adaptation.
In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.



Kumar, A., Raghunathan, A., Jones, R., Ma, T., and Liang, P. (2022).

Fine-tuning can distort pretrained features and underperform out-of-distribution.
In *International Conference on Learning Representations (ICLR)*.



Li, X., Grandvalet, Y., and Davoine, F. (2018).

Explicit inductive bias for transfer learning with convolutional networks.
In *International Conference on Machine Learning (ICML)*.



Liang, J., Hu, D., and Feng, J. (2020).

Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation.
In *International Conference on Machine Learning (ICML)*.



Long, M., Cao, Y., Wang, J., and Jordan, M. I. (2015).

Learning transferable features with deep adaptation networks.
In *International Conference on Machine Learning (ICML)*.

References III



Motiian, S., Jones, Q., Iranmanesh, S. M., and Doretto, G. (2017a).

Few-shot adversarial domain adaptation.
In *Advances in Neural Information Processing Systems (NeurIPS)*.



Motiian, S., Piccirilli, M., Adjeroh, D. A., and Doretto, G. (2017b).

Unified deep supervised domain adaptation and generalization.
In *IEEE International Conference on Computer Vision (ICCV)*.



Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., et al. (2024).

DINOv2: Learning robust visual features without supervision.
Transactions on Machine Learning Research (TMLR).



Pan, S. J. and Yang, Q. (2010).

A survey on transfer learning.
IEEE Transactions on Knowledge and Data Engineering, 22(10).



Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021).

Learning transferable visual models from natural language supervision.
In *International Conference on Machine Learning (ICML)*.



Sakaridis, C., Dai, D., and Van Gool, L. (2021).

ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding.
In *IEEE/CVF International Conference on Computer Vision (ICCV)*.



Shimodaira, H. (2000).

Improving predictive inference under covariate shift by weighting the log-likelihood function.
Journal of Statistical Planning and Inference, 90(2).



Sun, B. and Saenko, K. (2016).

Deep CORAL: Correlation alignment for deep domain adaptation.
In *ECCV Workshops*.

References IV



Tarvainen, A. and Valpola, H. (2017).

Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results.
In *Advances in Neural Information Processing Systems (NeurIPS)*.



Tranheden, W., Olsson, V., Pinto, J., and Svensson, L. (2021).

DACS: Domain adaptation via cross-domain mixed sampling.
In *IEEE Winter Conference on Applications of Computer Vision (WACV)*.



Tsai, Y.-H., Hung, W.-C., Schulter, S., Sohn, K., Yang, M.-H., and Chandraker, M. (2018).

Learning to adapt structured output space for semantic segmentation.
In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.



Vu, T.-H., Jain, H., Bucher, M., Cord, M., and Pérez, P. (2019).

ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation.
In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.



Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. (2021).

Tent: Fully test-time adaptation by entropy minimization.
In *International Conference on Learning Representations (ICLR)*.



Wang, Q., Fink, O., Van Gool, L., and Dai, D. (2022).

Continual test-time domain adaptation.
In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.



Wei, Z., Chen, L., Jin, Y., Ma, X., Liu, T., Ling, P., Wang, B., Chen, H., and Zheng, J. (2024).

Stronger, fewer, & superior: Harnessing vision foundation models for domain generalized semantic segmentation.
In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.



Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Kornblith, S., Roelofs, R., et al. (2022).

Robust fine-tuning of zero-shot models.
In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

References V



Zeiler, M. D. and Fergus, R. (2014).

Visualizing and understanding convolutional networks.
In *European Conference on Computer Vision (ECCV)*.



Zhang, Y., David, P., and Gong, B. (2017).

Curriculum domain adaptation for semantic segmentation of urban scenes.
In *IEEE International Conference on Computer Vision (ICCV)*.



Zhao, H., Tachet des Combes, R., Zhang, K., and Gordon, G. J. (2019).

On learning invariant representations for domain adaptation.
In *International Conference on Machine Learning (ICML)*.