An overview of Domain Adaptation concepts and techniques

Dino Ienco (dino.ienco@inrae.fr)
Research Director, PhD, HDR
INRAE, INRIA, EVERGREEN, UMR TETIS
Evergreen team - https://team.inria.fr/evergreen/

Giuseppe Guarino (giuseppe.guarino@inrae.fr)
Post-Doc Fellowship
INRAE, INRIA, EVERGREEN, UMR TETIS

Outline

Motivation

(Unsupervised) Domain Adaptation

- Instance Re-weighting approaches
- Feature Alignment
- Generative Approaches
- Self-Training strategies

Semi-Supervised DA

Beyond Classical UDA

- Heterogeneous DA
- Partial/Open/Universal DA
- Multi-Source DA
- Source-Free DA

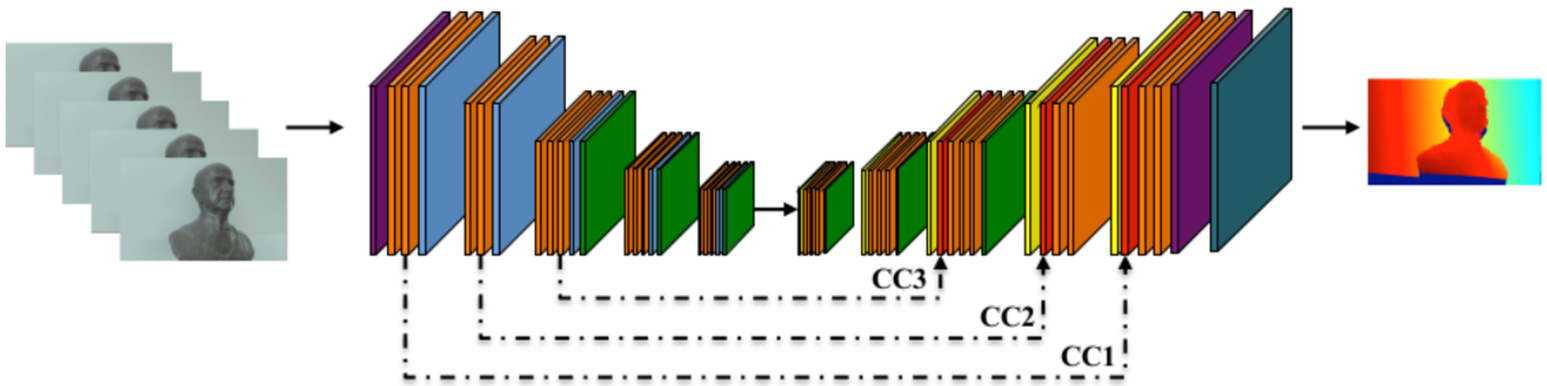
Test-Time Adaptation

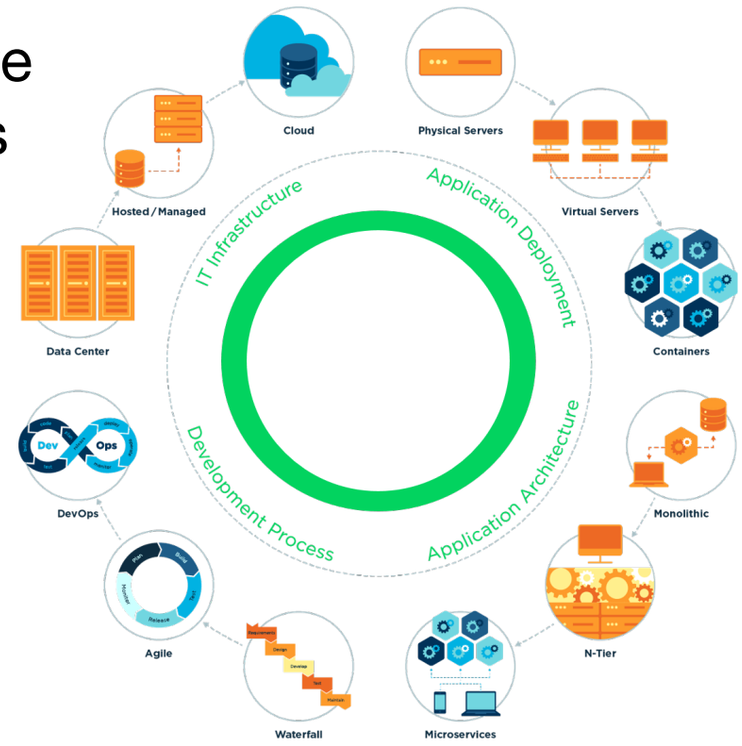
Foundation Model & DA

Conclusions

Motivation

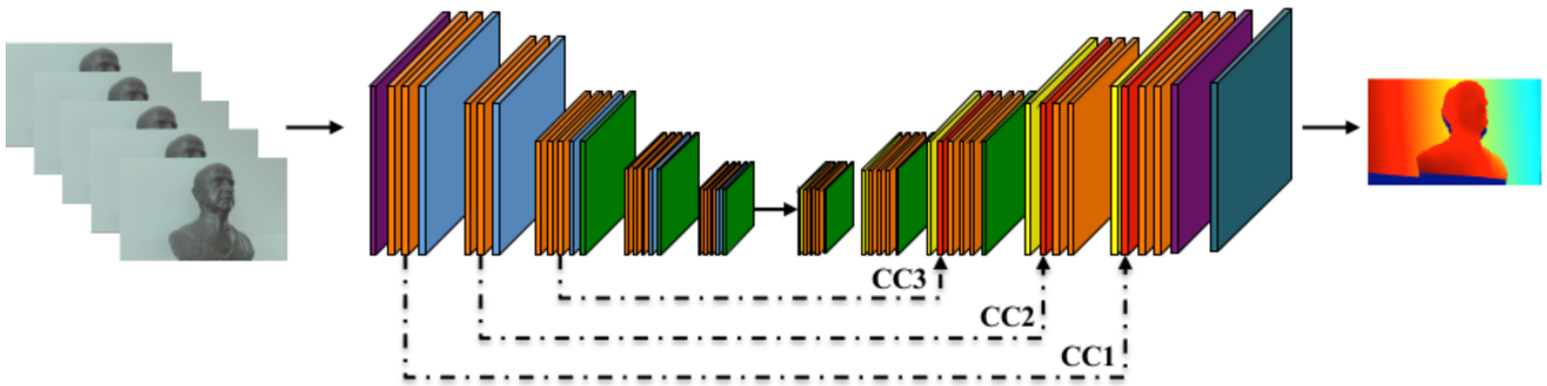
Machine Learning techniques (e.g. Deep Neural Networks) have demonstrated today impressive results in many application fields

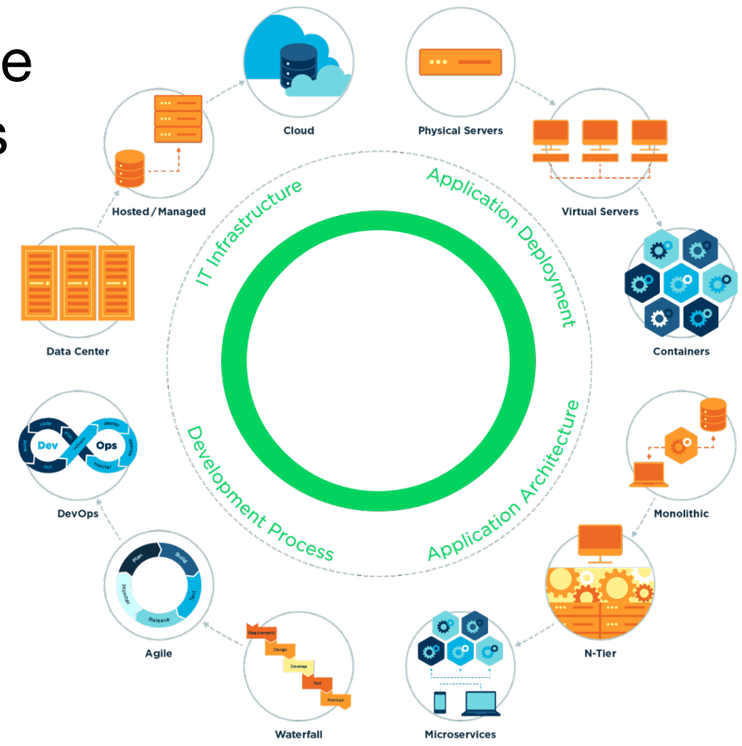




Motivation

Machine Learning techniques (e.g. Deep Neural Networks) have demonstrated today impressive results in many application fields





The available DATA is at the centre of the current era:

- A **large diversity** of devices is continually sensing the environment.
- This results in large volume of data collected for the same task.
- The abundance of information poses **new challenges for** data-driven **AI (ML and CV)** techniques that should work seamlessly across different scenarios/settings

Motivation

Images data



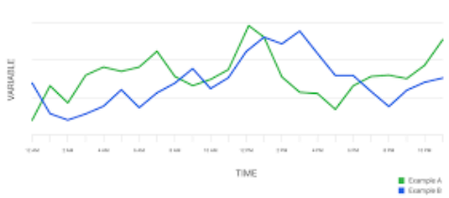
Sound data



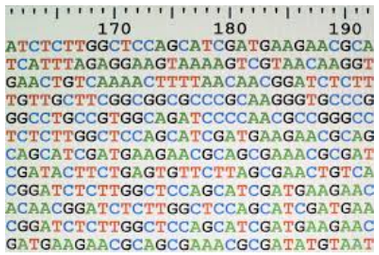
Text Data



Time Series



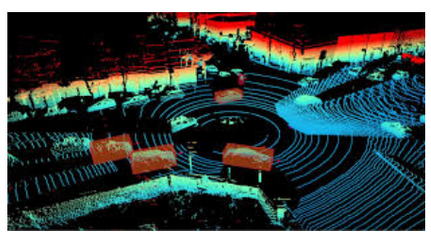
Genomic data



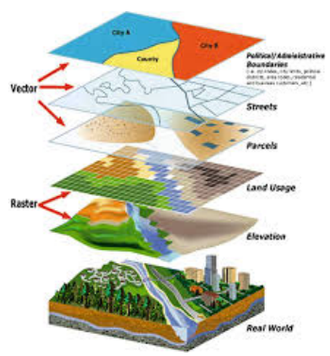
Social Media



Point Cloud (3D) data



Geospatial Data



Video data



Tabular data

Date collected	Plot	Species	Sex	Weight
1/9/78	1	DM	M	40
1/9/78	1	DM	F	36
1/9/78	1	DS	F	135
1/20/78	1	DM	F	39
1/20/78	2	DM	M	43
1/20/78	2	DS	F	144
3/13/78	2	DM	F	51
3/13/78	2	DM	F	44
3/13/78	2	DS	F	146

Motivation

Images data



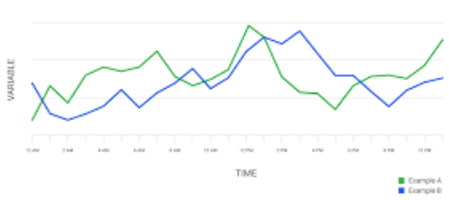
Sound data



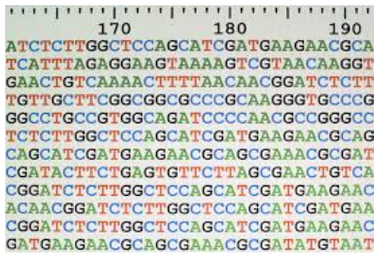
Text Data



Time Series



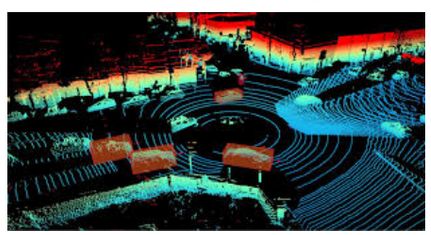
Genomic data



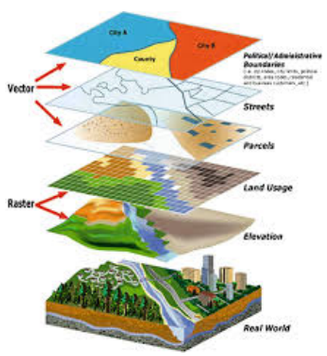
Social Media



Point Cloud (3D) data



Geospatial Data



Video data



Tabular data

Date collected	Plot	Species	Sex	Weight
1/9/78	1	DM	M	40
1/9/78	1	DM	F	36
1/9/78	1	DS	F	135
1/20/78	1	DM	F	39
1/20/78	2	DM	M	43
1/20/78	2	DS	F	144
3/13/78	2	DM	F	51
3/13/78	2	DM	F	44
3/13/78	2	DS	F	146

Domains

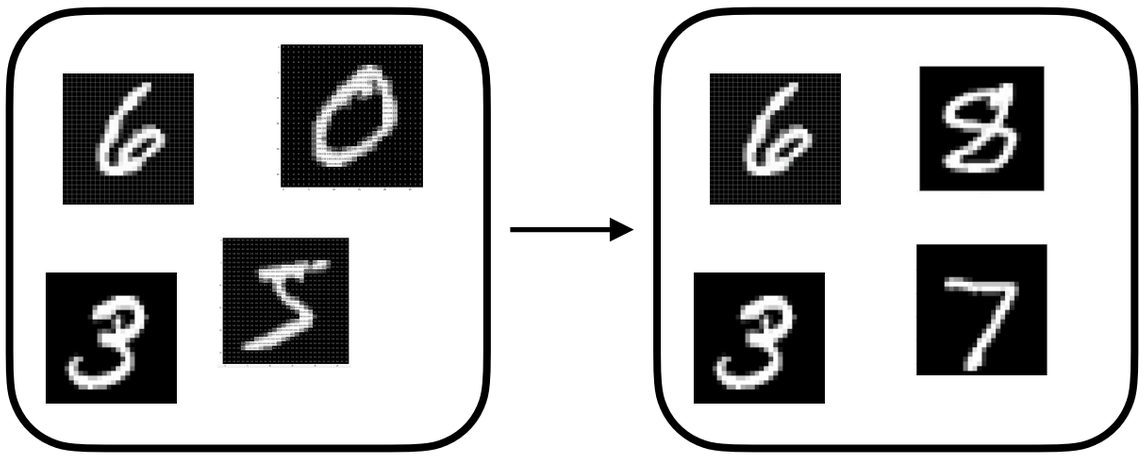
- Medical
- Biological
- Remote Sensing
- Finance
- Human-Computer Interaction
- Robotics
- Autonomous driving

Motivation

Standard Machine Learning techniques make the (strong) assumption that:
- Training and Test data follow the same data distribution

Motivation

Standard Machine Learning techniques make the (strong) assumption that:
- Training and Test data follow the same data distribution

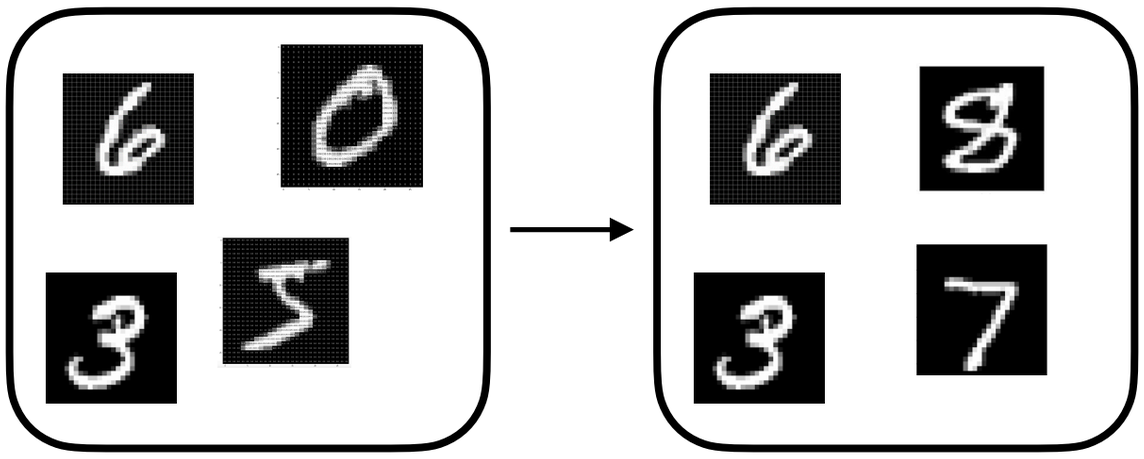


Training Data

Test Data

Motivation

Standard Machine Learning techniques make the (strong) assumption that:
- Training and Test data follow the same data distribution



Training Data

Test Data



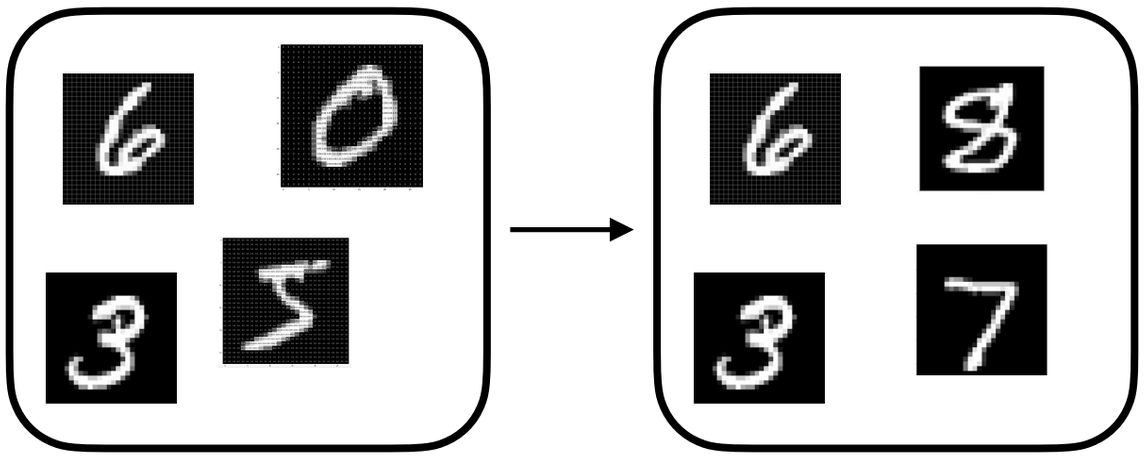
Training Data



Test Data

Motivation

Standard Machine Learning techniques make the (strong) assumption that:
- Training and Test data follow the same data distribution



Training Data

Test Data



Training Data



Test Data



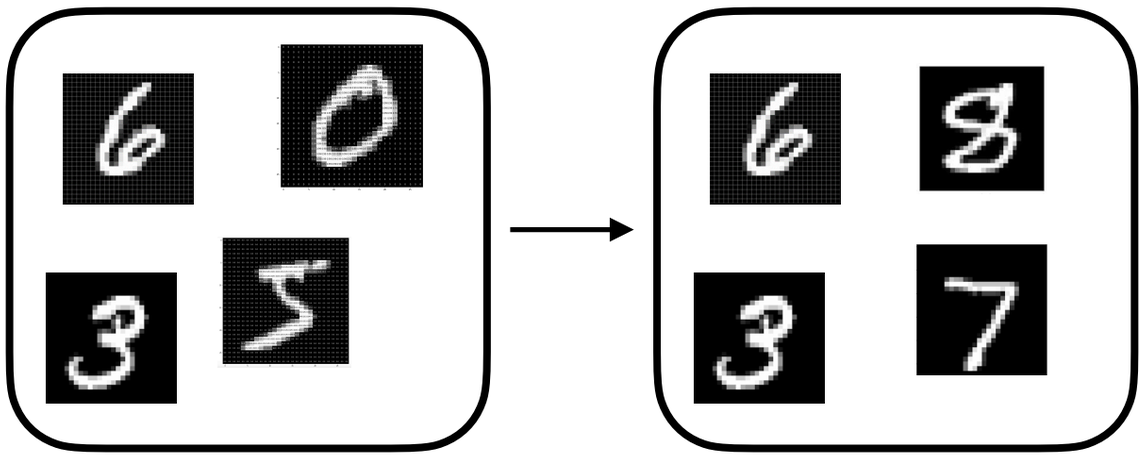
Training Data (from Hospital 1)



Test Data (from Hospital 1)

Motivation

Standard Machine Learning techniques make the (strong) assumption that:
- Training and Test data follow the same data distribution



Training Data

Test Data



Training Data



Test Data



Training Data (from Hospital 1)



Test Data (from Hospital 1)

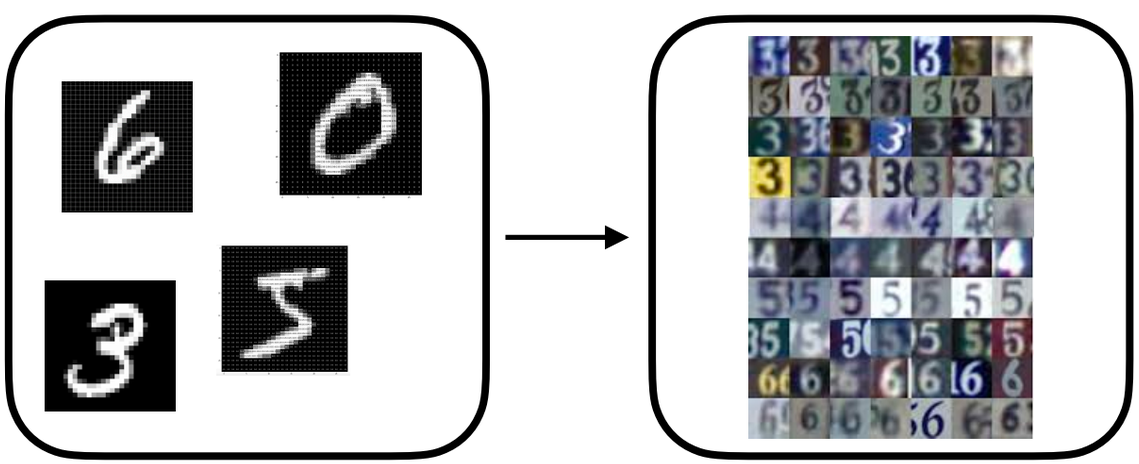
Main Assumption
Training and Test come **from the same distributions**

Motivation

In real world scenarios, the **assumption** in which training and test data come from the same distribution **is often invalid**

Motivation

In real world scenarios, the **assumption** in which training and test data come from the same distribution **is often invalid**

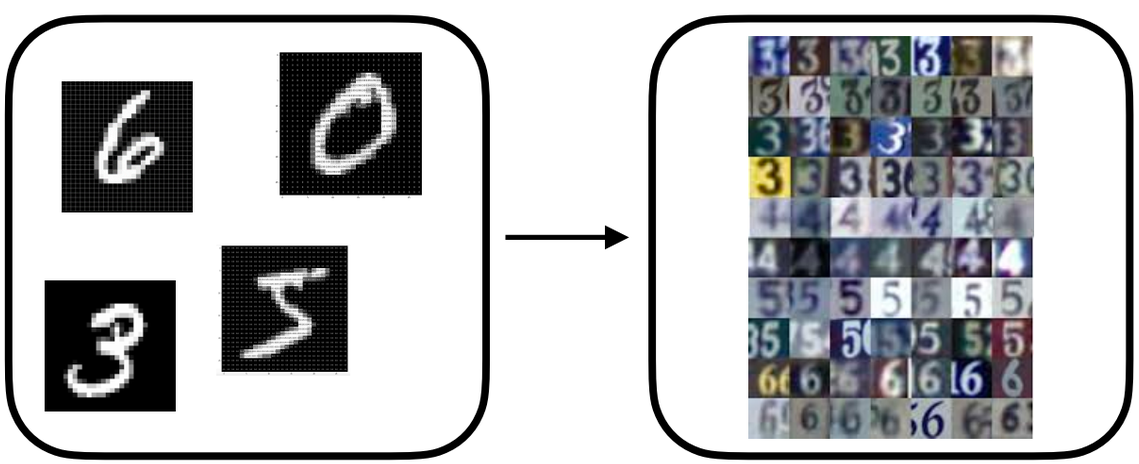


Training Data

Test Data

Motivation

In real world scenarios, the **assumption** in which training and test data come from the same distribution **is often invalid**

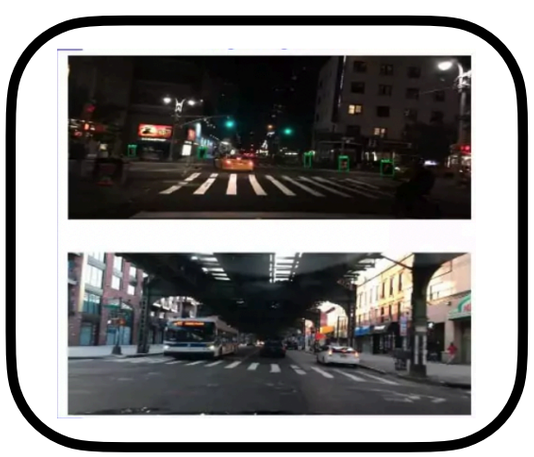


Training Data

Test Data



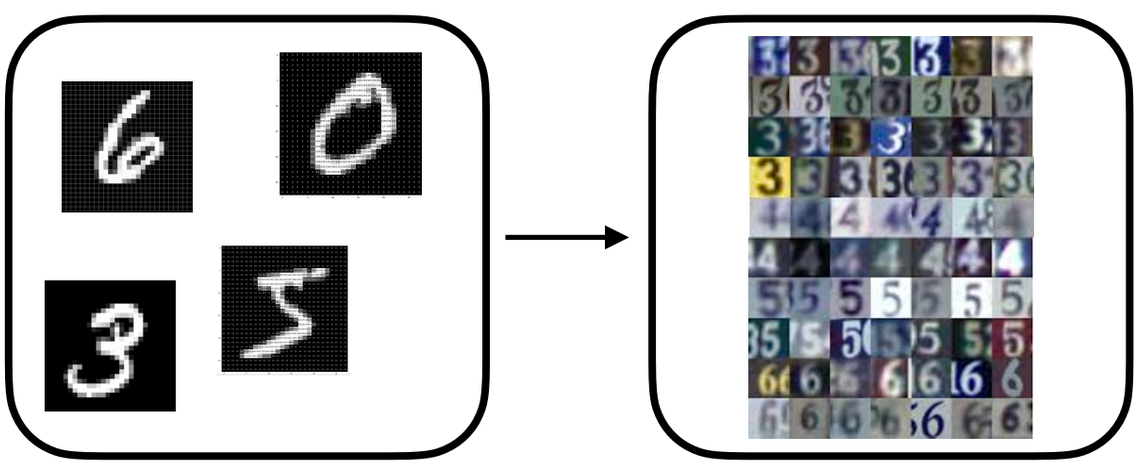
Training Data



Test Data

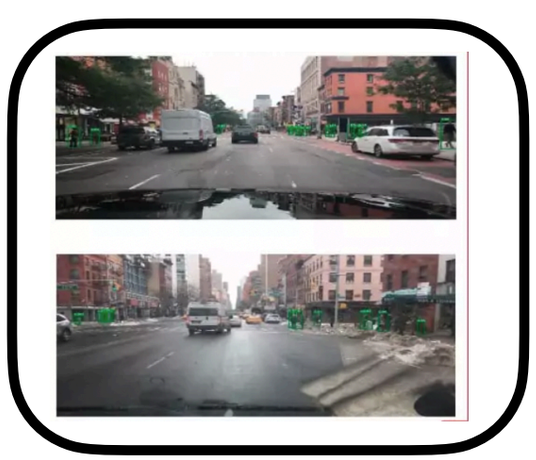
Motivation

In real world scenarios, the **assumption** in which training and test data come from the same distribution **is often invalid**

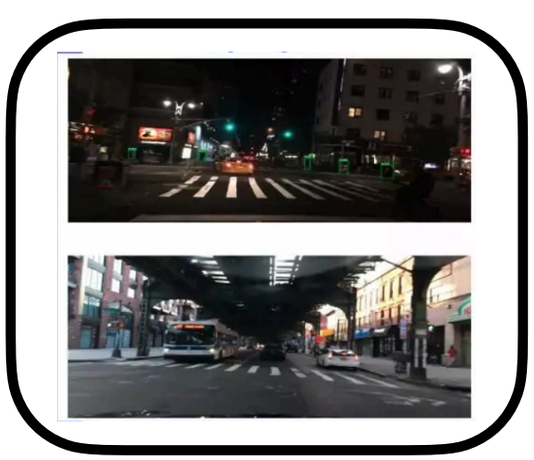


Training Data

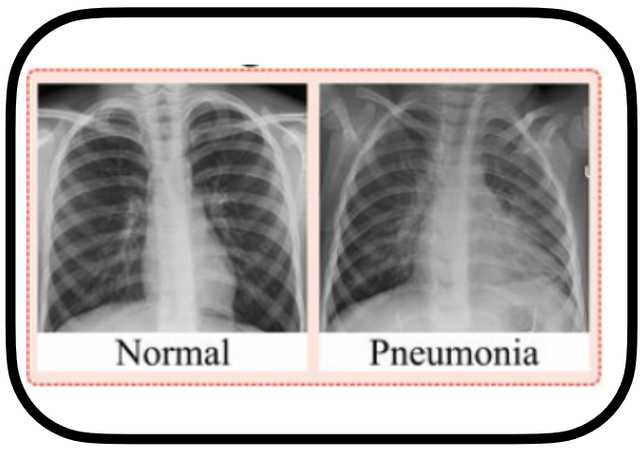
Test Data



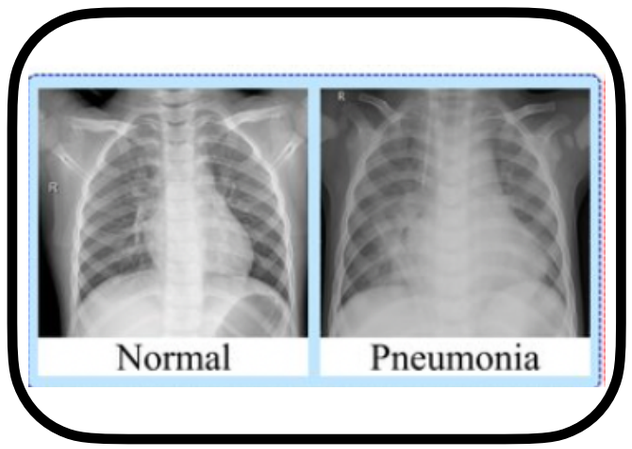
Training Data



Test Data



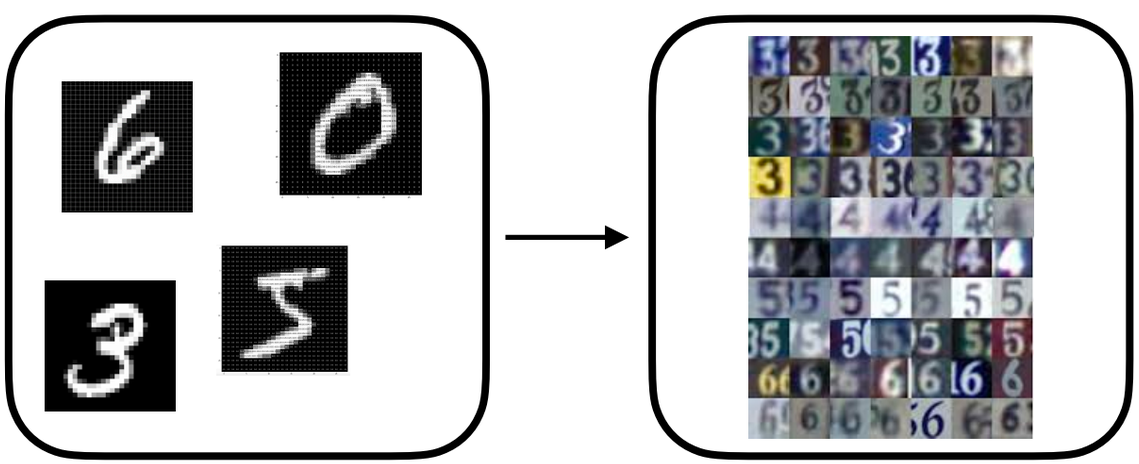
Training Data



Test Data

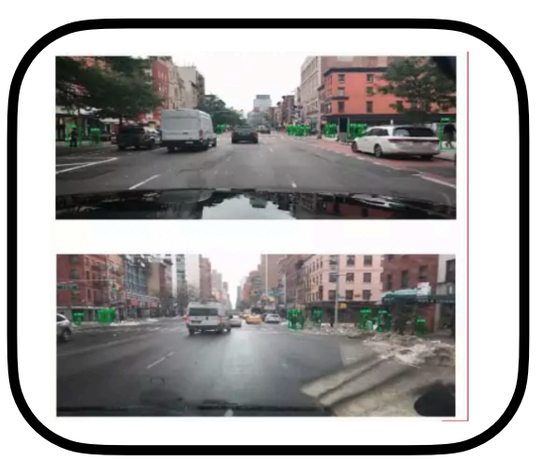
Motivation

In real world scenarios, the **assumption** in which training and test data come from the same distribution **is often invalid**

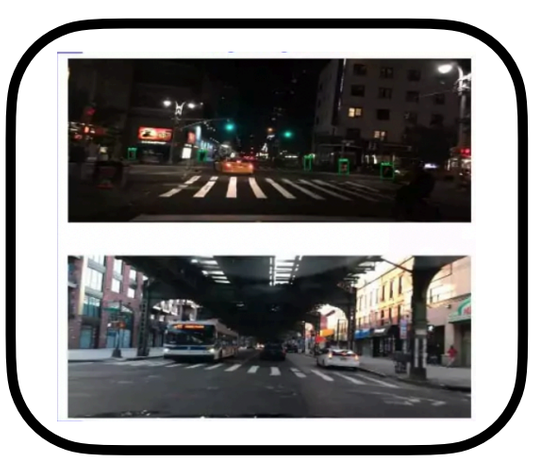


Training Data

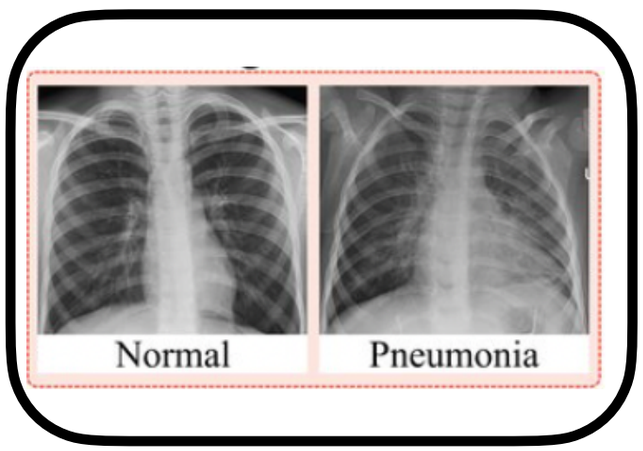
Test Data



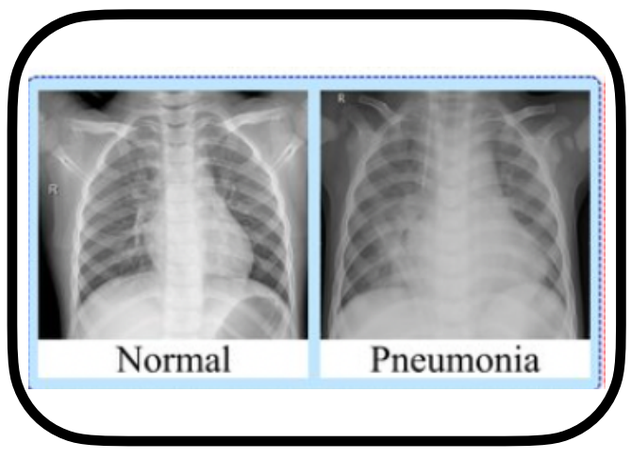
Training Data



Test Data



Training Data



Test Data

Training and Test cover the same task but come **from different distributions**

Motivation

Same task, different world — **why does the model fail?**



Training Data







Training distribution → **Test distribution**

Motivation

Same task, different world — **why does the model fail?**



Training Data







Training distribution → **Test distribution**

The **model never changed** but the **world has changed**
Same task (detect the cat) but the input distribution P(X) shifted.

Motivation

Same task, different world — **why does the model fail?**



Training Data







Training distribution → **Test distribution**

The **model never changed** but the **world has changed**
Same task (detect the cat) but the input distribution P(X) shifted.

The concept remains the same but:

- **Context shift** (Day to Night)
- **Acquisition shift** (Different sensor)
- **Synthetic gap** (Sim to Real or vice versa)

The concept is the same but the model did not learn the concept by the appearance of the concept

Motivation

So why not just re-label everything?

Motivation

So why not just re-label everything?

Labeled data is a **scarce** and **expensive resource**



Motivation

So why not just re-label everything?



Labeled data is a **scarce** and **expensive resource**

💸 **Annotation cost:** medical labels require domain experts; linguistic annotation requires native speakers; each label can cost money and/or hours

⏳ **Data collection inertia:** curated datasets represent years of community effort (ImageNet took ~3 years; clinical cohorts take decades)

✅ **Task invariance:** the underlying task is identical; a scene in a simulated environment (e.g. GTA5) is semantically the same as one on a real street

Motivation

So why not just re-label everything?



Labeled data is a **scarce** and **expensive resource**

💸 **Annotation cost:** medical labels require domain experts; linguistic annotation requires native speakers; each label can cost money and/or hours

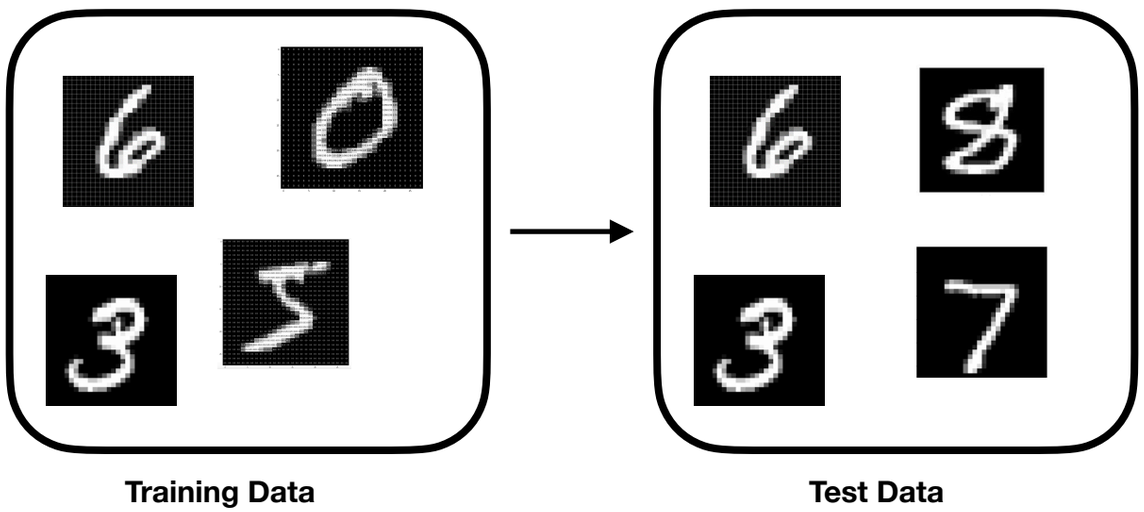
⏳ **Data collection inertia:** curated datasets represent years of community effort (ImageNet took ~3 years; clinical cohorts take decades)

✅ **Task invariance:** the underlying task is identical; a scene in a simulated environment (e.g. GTA5) is semantically the same as one on a real street

Source data is available. Target labels are not. The question is:
Can we bridge the gap by reusing available labelled data collected under a different distribution?

Domain Adaptation [Wilson20]

Domain Adaptation is the field of machine learning that studies how to **adapt a model** from a source domain to a target domain.

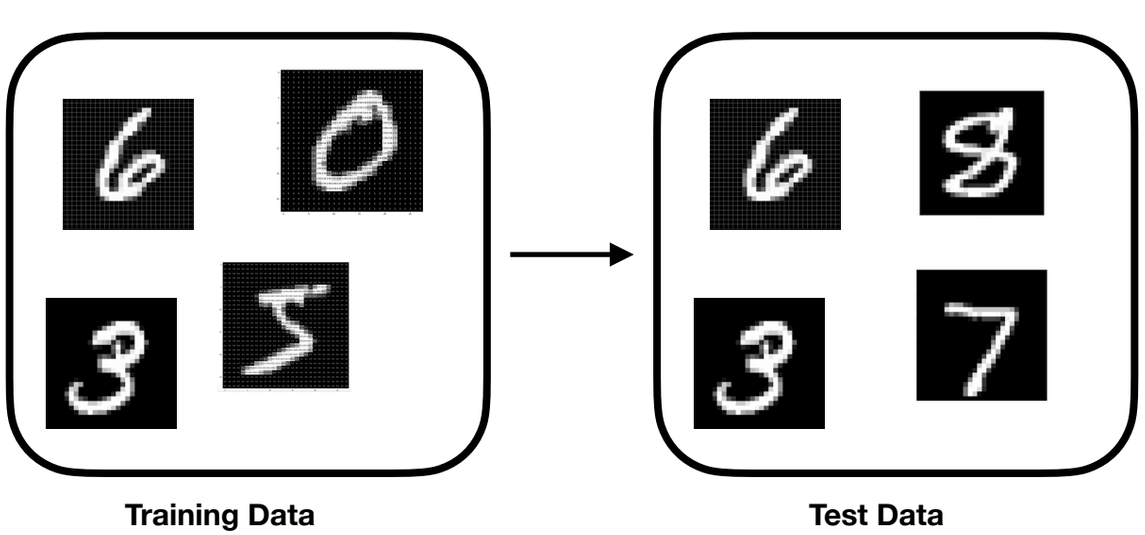


Training Data **Test Data**

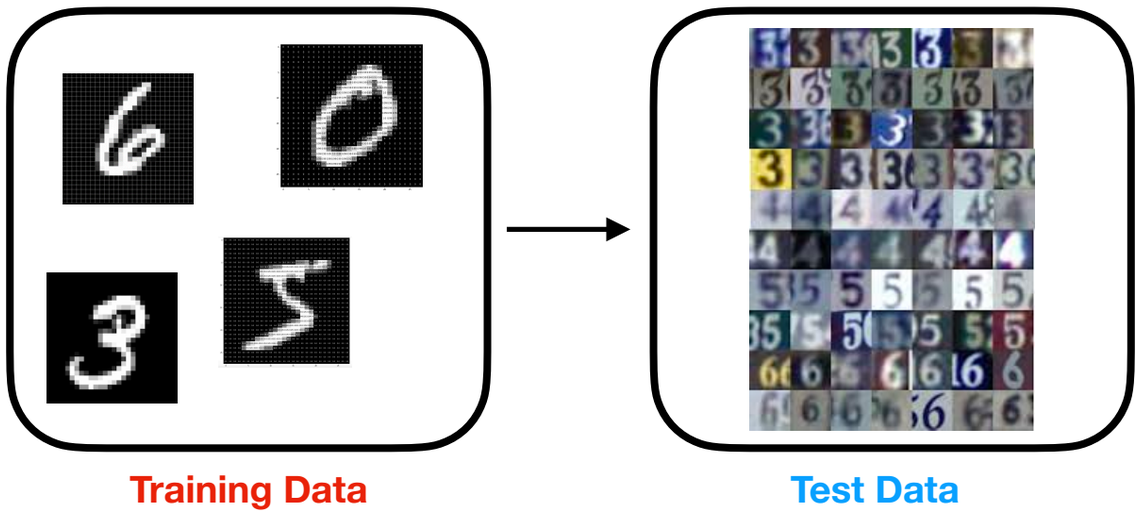
Training and Test
from same domain distributions

Domain Adaptation [Wilson20]

Domain Adaptation is the field of machine learning that studies how to **adapt a model** from a source domain to a target domain.



Training and Test
from same domain distributions

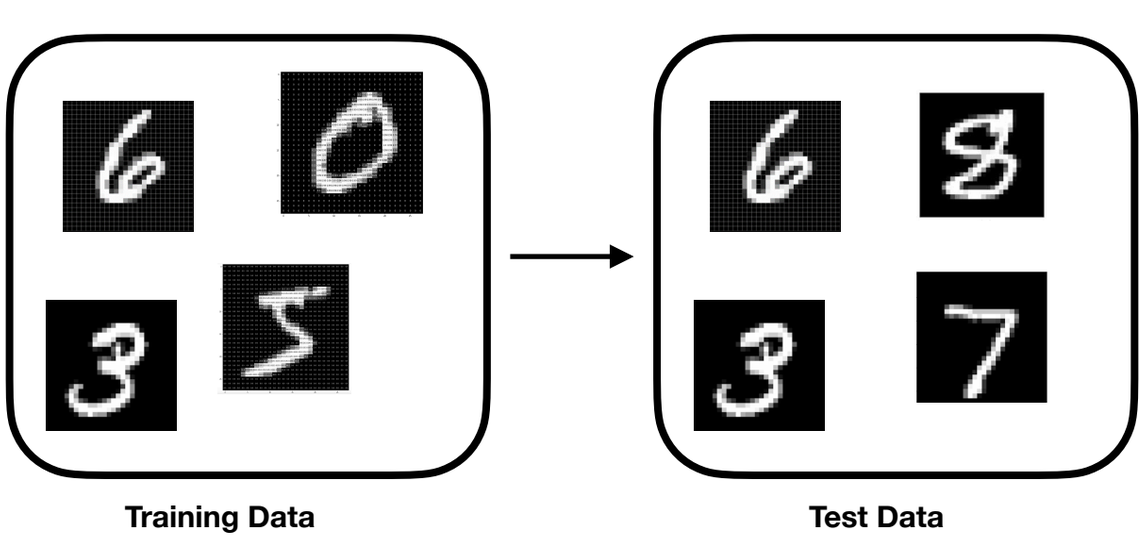


Training and Test
from different domain distributions (gap/shift)

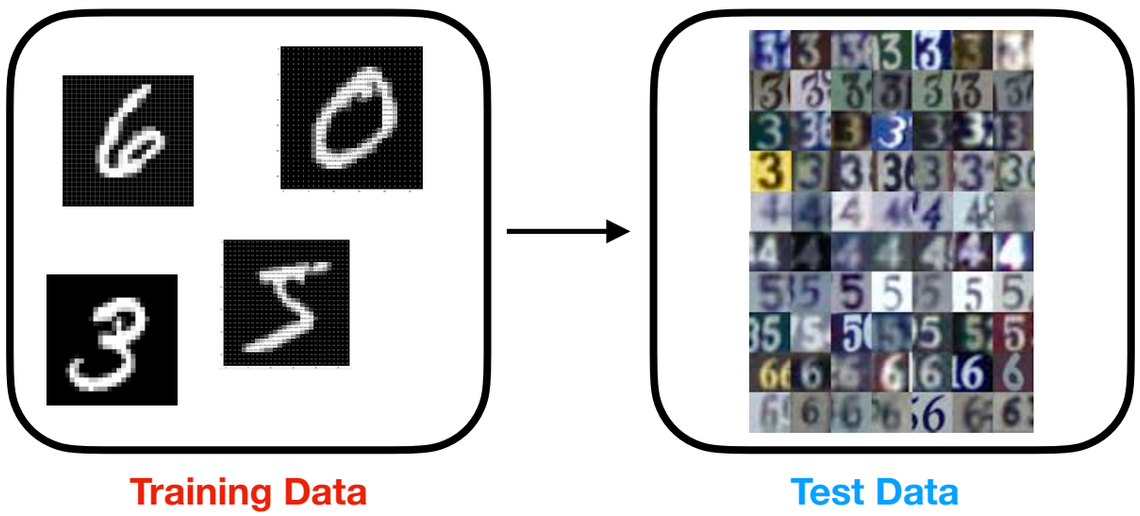
[Wilson20] G. Wilson, D. J. Cook: **A Survey of Unsupervised Deep Domain Adaptation**. ACM Trans. Intell. Syst. Technol. 11(5): 51:1-51:46 (2020)

Domain Adaptation [Wilson20]

Domain Adaptation is the field of machine learning that studies how to **adapt a model** from a source domain to a target domain.



Training and Test
from same domain distributions



Training and Test
from different domain distributions (gap/shift)

- Distribution from Training (Source) and Test (Target) domains can be different
- In this case standard supervised ML approaches fail to generalise, thus …
- Needs for methods dealing with distribution shift => Domain Adaptation [Wilson20]

[Wilson20] G. Wilson, D. J. Cook: **A Survey of Unsupervised Deep Domain Adaptation**. ACM Trans. Intell. Syst. Technol. 11(5): 51:1-51:46 (2020)

Domain Adaptation

Standard **Machine Learning** assumption

- Joint distribution between data (X) and label (Y)
- Ps is the Source Domain
- Pt is the Target Domain

$$P_s(X, Y) = P_t(X, Y)$$

Domain Adaptation

Standard **Machine Learning** assumption

- Joint distribution between data (X) and label (Y)
- Ps is the Source Domain
- Pt is the Target Domain

$$P_s(X, Y) = P_t(X, Y)$$

Domain Adaptation assumption

- Joint distribution between data (X) and label (Y)
- Ps is the Source Domain
- Pt is the Target Domain

$$P_s(X, Y) \neq P_t(X, Y)$$

Domain Adaptation

Standard **Machine Learning** assumption

- Joint distribution between data (X) and label (Y)
- Ps is the Source Domain
- Pt is the Target Domain

$$P_s(X, Y) = P_t(X, Y)$$

Domain Adaptation assumption

- Joint distribution between data (X) and label (Y)
- Ps is the Source Domain
- Pt is the Target Domain

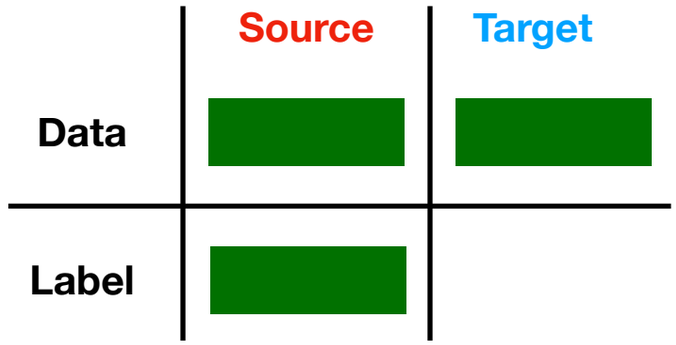
$$P_s(X, Y) \neq P_t(X, Y)$$

Different **Domain Adaptation settings/scenarios**

- Unsupervised Domain Adaptation (**UDA**)
- Semi-Supervised Domain Adaptation (**SSDA**)
- Supervised Domain Adaptation (**SDA**)

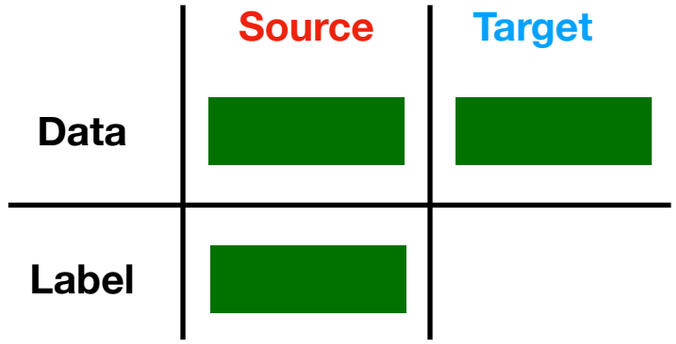
Domain Adaptation

Unsupervised Domain Adaptation (**UDA**)

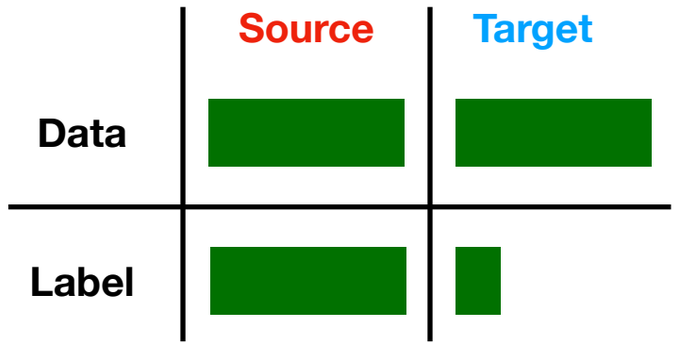


Domain Adaptation

Unsupervised Domain Adaptation (**UDA**)

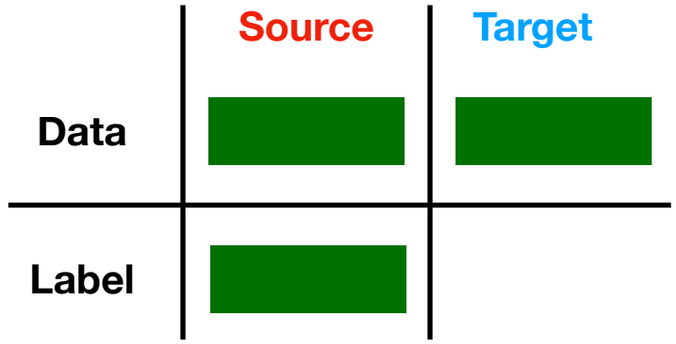


Semi-Supervised Domain Adaptation (**SSDA**)

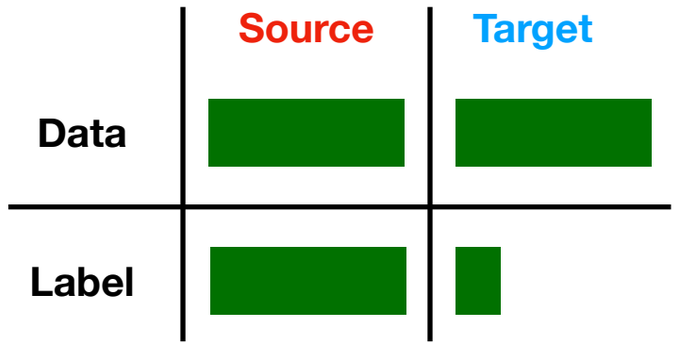


Domain Adaptation

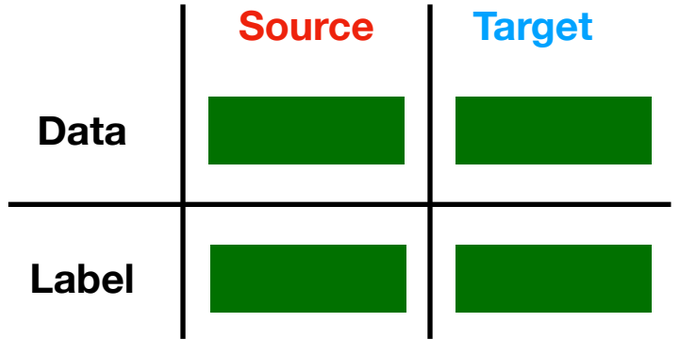
Unsupervised Domain Adaptation (**UDA**)



Semi-Supervised Domain Adaptation (**SSDA**)

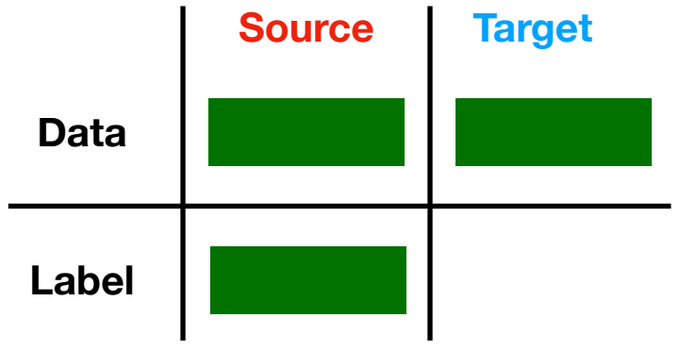


Supervised Domain Adaptation (**SDA**)



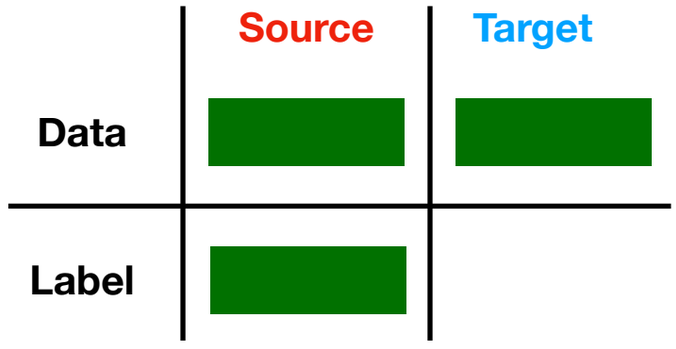
Domain Adaptation

Unsupervised Domain Adaptation (**UDA**)



Domain Adaptation

Unsupervised Domain Adaptation (**UDA**)



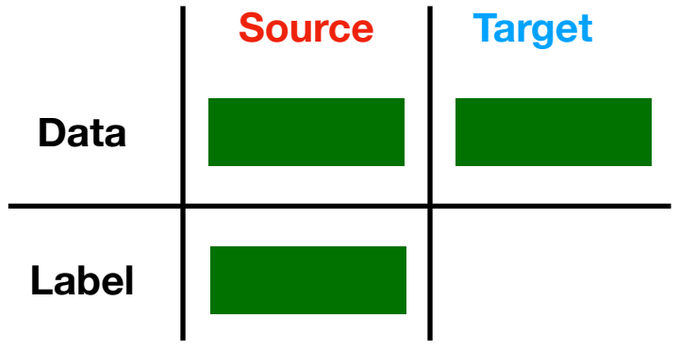
Under (classical) **UDA assumptions**:

- Source and Target domains share the same (homogeneous) representation space (e.g. RGB images, X-Ray images, time series with same format, tabular data with same structure, …)

- The label space is the same, this means that the set of classes between Source and Target domains is exactly the same

Domain Adaptation

Unsupervised Domain Adaptation (**UDA**)

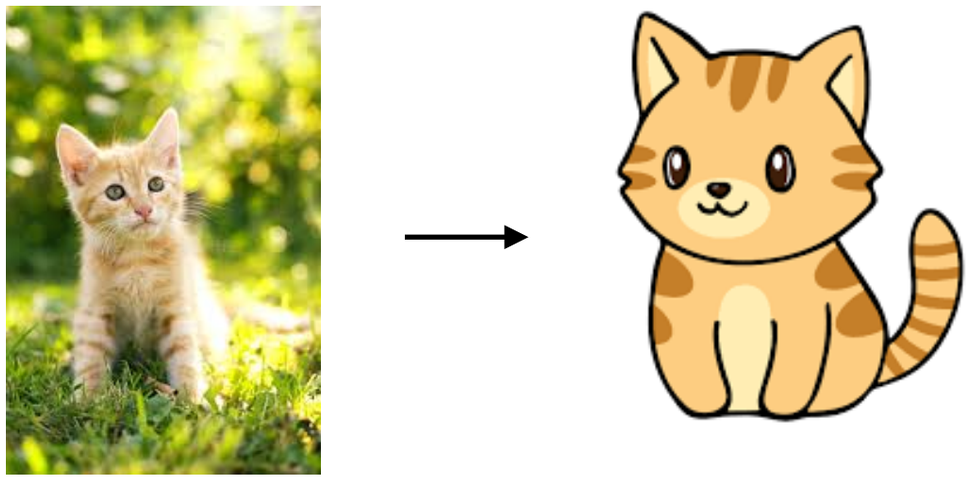


Under (classical) **UDA assumptions**:

- Source and Target domains share the same (homogeneous) representation space (e.g. RGB images, X-Ray images, time series with same format, tabular data with same structure, …)

- The label space is the same, this means that the set of classes between Source and Target domains is exactly the same

The **distribution shift** is related to the representation of the concept (appearance, contextual information, acquisition sensor/ condition, …) and not to the concept.



Source Domain

Target Domain

Unsupervised Domain Adaptation

- Instance reweighing approaches
- Feature / Distribution Alignment
- Generative Approach to fill domain gap
- Self-Training Strategies

Instance Reweighting approaches

Instance Reweighting approaches [1]

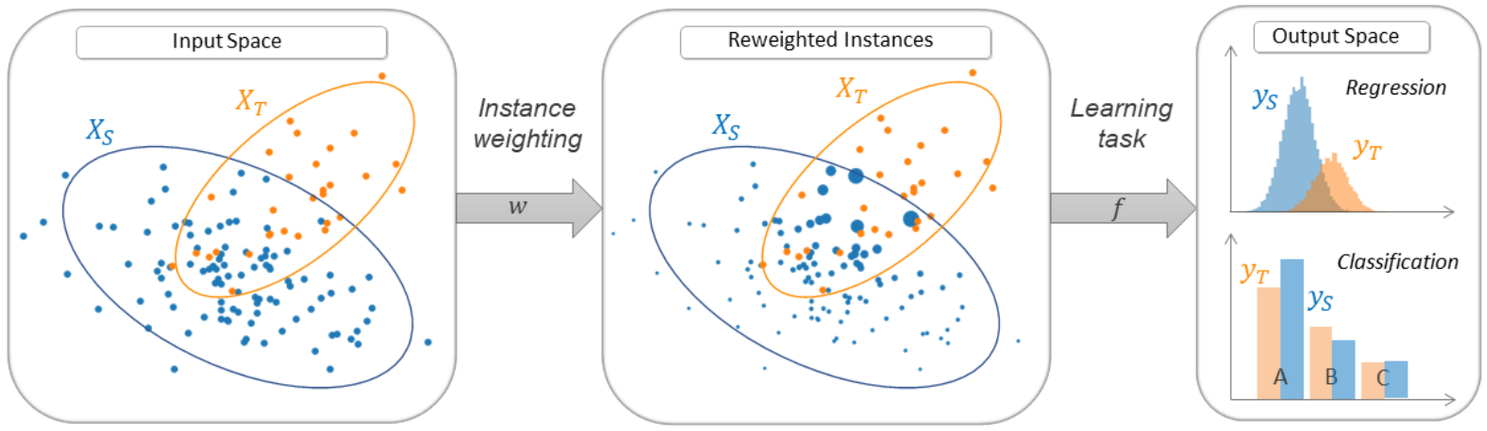
Reweight source samples so that the weighted source distribution matches the target [2].
Based on **quadratic programming optimisation** procedure.
Estimate **a reweighing coefficient** for each source sample
Introduced as a variation of the SVM (Support Vector Machine) machine learning approach.

[1] Sugiyama, Masashi, and Motoaki Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.
[2] *Gretton, Arthur, et al. "Covariate shift by kernel mean matching." Dataset shift in machine learning 3.4 (2009): 5*
[3] https://adapt-python.github.io/adapt/contents.html#adapt-instance-based

Instance Reweighting approaches [1]

Reweight source samples so that the weighted source distribution matches the target [2].
Based on **quadratic programming optimisation** procedure.
Estimate **a reweighing coefficient** for each source sample
Introduced as a variation of the SVM (Support Vector Machine) machine learning approach.

Image from the ADAPT-PYTHON Library [3]

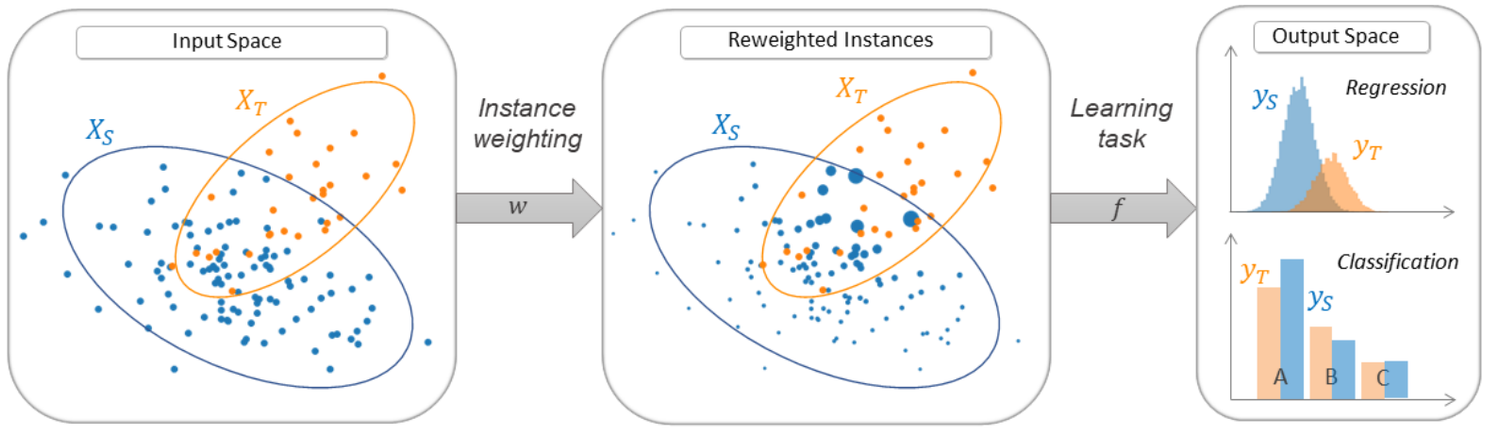


[1] Sugiyama, Masashi, and Motoaki Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.
[2] *Gretton, Arthur, et al. "Covariate shift by kernel mean matching." Dataset shift in machine learning 3.4 (2009): 5*
[3] https://adapt-python.github.io/adapt/contents.html#adapt-instance-based

Instance Reweighting approaches [1]

Reweight source samples so that the weighted source distribution matches the target [2].
Based on **quadratic programming optimisation** procedure.
Estimate **a reweighing coefficient** for each source sample
Introduced as a variation of the SVM (Support Vector Machine) machine learning approach.

Image from the ADAPT-PYTHON Library [3]



Some of the approaches belonging to this family [3]:

- Kernel Mean Matching (KMM) / The reweigh coefficient is estimated via kernel approach based on Maximum Mean Discrepancy (MMD).

- Kullback-Leibler Importance Estimation Procedure (KLIEP) / The reweigh coefficient is estimated by minimising KL divergence between source and target distributions

[1] Sugiyama, Masashi, and Motoaki Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.
[2] *Gretton, Arthur, et al. "Covariate shift by kernel mean matching." Dataset shift in machine learning 3.4 (2009): 5*
[3] *https://adapt-python.github.io/adapt/contents.html#adapt-instance-based*

Instance Reweighting approaches [1]

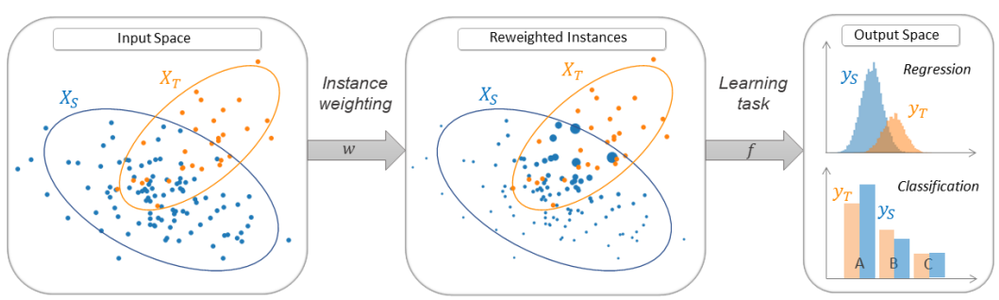
Instance reweighting approaches work **estimating the density ratio between Target and Source domains**:

- **Affected by the curse of dimensionality** for the density ratio estimation (complex/ difficult to estimate in high dimensional spaces)

- **Work** mainly in **input space** conversely to modern approach that learn new representations on which analysis are performed

- Source/Target **support mismatch induced by possible target domain regions not covered** by source samples (cannot modify the weight of an instance that does not exist)

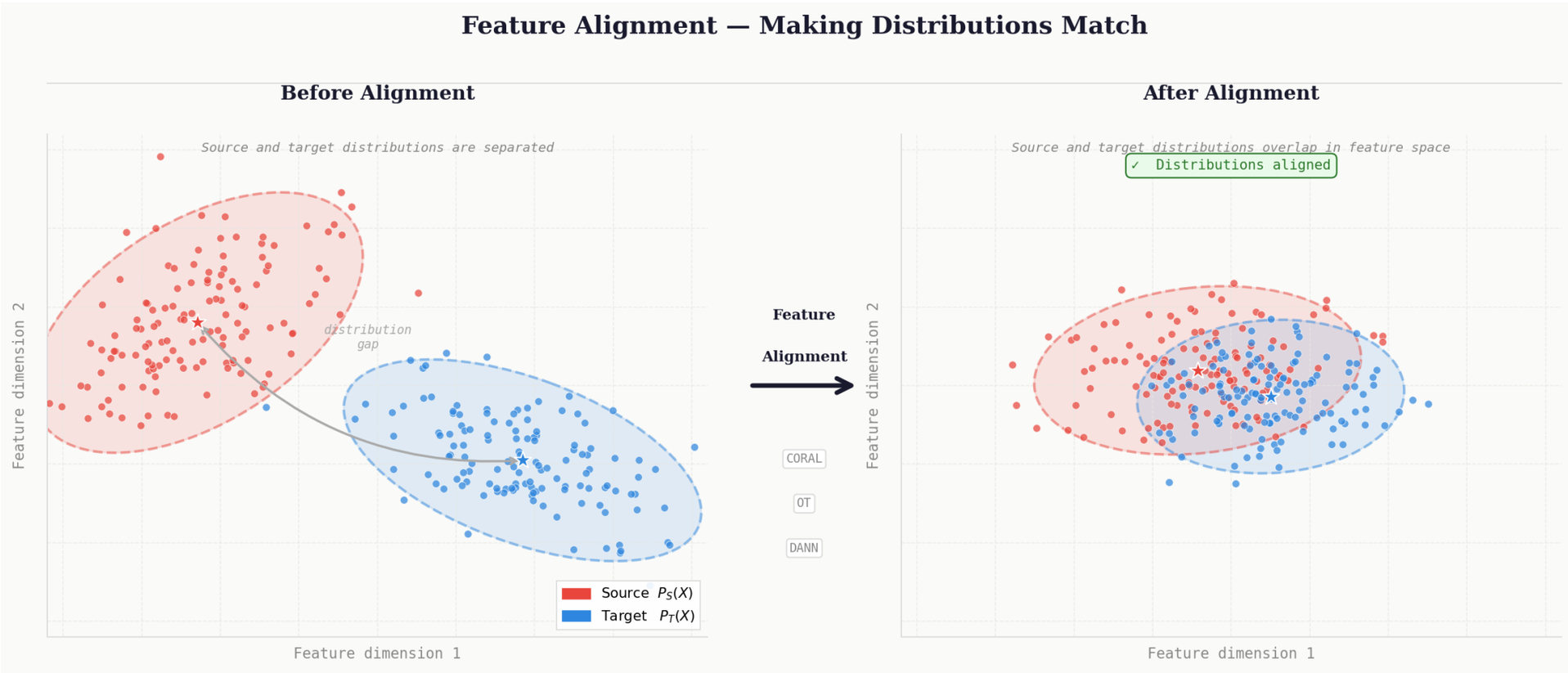
- Few Source samples weight with **too big importance** (few source examples get enormous weight while most get nearly zero)



Feature / Distribution Alignment

Feature / Distribution Alignment

Instead of reweighting samples, **align the feature distributions directly** in a learned representation space.



Different strategies:

- **Statistics-based**
- **Optimal Transport**
- **Adversarial**

Feature / Distribution Alignment

CORAL [1]

The main hypothesis is that the two domains **follow approximatively a Gaussian** distribution

Align **the second-order statistics** (covariance matrices) of source and target features

Minimize the Frobenius norm between source and target covariance matrices

[1] Sun, Baochen, and Kate Saenko. "Deep coral: Correlation alignment for deep domain adaptation." *European conference on computer vision*. Cham: Springer International Publishing, 2016.

Feature / Distribution Alignment

CORAL [1]

The main hypothesis is that the two domains **follow approximatively a Gaussian** distribution

Align **the second-order statistics** (covariance matrices) of source and target features

Minimize the Frobenius norm between source and target covariance matrices

$$L_{CORAL} = \frac{1}{4d^2}||C_s - C_t||_F^2$$

- Where Cs and Ct are the covariance matrices of source and target features, d is the feature dimension.
- **Covariance captures feature correlations**, aligning it standardizes the spread and orientation of both distributions.
- **Simple solution** to implement.
- It only aligns second order statistics, **high-order structure are ignored.**

[1] Sun, Baochen, and Kate Saenko. "Deep coral: Correlation alignment for deep domain adaptation." *European conference on computer vision*. Cham: Springer International Publishing, 2016.

Feature / Distribution Alignment

From shallow to Deep alignment

[1] Damodaran, Bharath Bhushan, et al. "Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation." *Proceedings of the European conference on computer vision (ECCV)*. 2018.
[2] Sun, Baochen, and Kate Saenko. "Deep coral: Correlation alignment for deep domain adaptation." *European conference on computer vision*. Cham: Springer International Publishing, 2016.

Feature / Distribution Alignment

From shallow to Deep alignment

Optimal Transport

- OT finds the **minimum cost plan** to transport the source distribution onto the target (Wasserstein distance)

- More principled than covariance alignment, captures the full geometry of both distributions, **not just second-order statistics** [1]

- In practice: **more theoretically grounded than CORAL**, but **significantly more expensive** to compute at scale

[1] Damodaran, Bharath Bhushan, et al. "Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation." *Proceedings of the European conference on computer vision (ECCV)*. 2018.
[2] Sun, Baochen, and Kate Saenko. "Deep coral: Correlation alignment for deep domain adaptation." *European conference on computer vision*. Cham: Springer International Publishing, 2016.

Feature / Distribution Alignment

From shallow to Deep alignment

Optimal Transport

- OT finds the **minimum cost plan** to transport the source distribution onto the target (Wasserstein distance)

- More principled than covariance alignment, captures the full geometry of both distributions, **not just second-order statistics** [1]

- In practice: **more theoretically grounded than CORAL**, but **significantly more expensive** to compute at scale

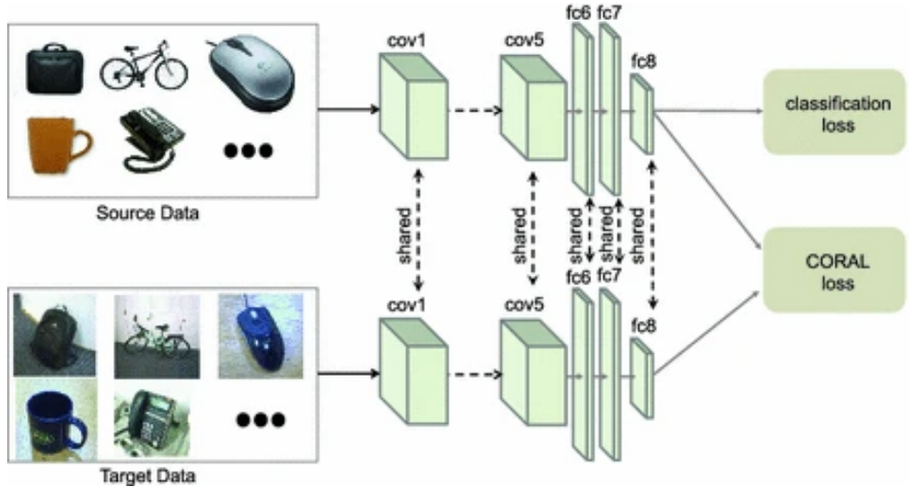
Deep Coral

- Conversely from standard CORAL, **the features are now learned end-to-end**

- **Trade off** (lambda) between the two losses

- **Still only second-order** alignment

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda \mathcal{L}_{\text{CORAL}}$$



[1] Damodaran, Bharath Bhushan, et al. "Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation." *Proceedings of the European conference on computer vision (ECCV)*. 2018.

[2] Sun, Baochen, and Kate Saenko. "Deep coral: Correlation alignment for deep domain adaptation." *European conference on computer vision*. Cham: Springer International Publishing, 2016.

Feature / Distribution Alignment

ADDA [1]: Adversarial Discriminative Domain Adaptation

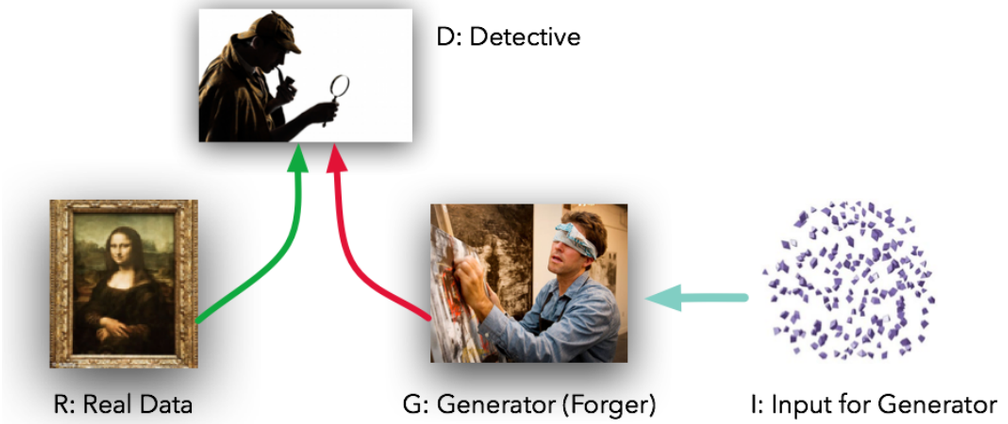
[1] Tzeng, Eric, et al. "Adversarial discriminative domain adaptation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

Feature / Distribution Alignment

ADDA [1]: Adversarial Discriminative Domain Adaptation

Key features:

- An encoder per domain
- **Sequential training** process (two-stage training)
- CGAN (Conditional Generative Adversarial Network) mechanism



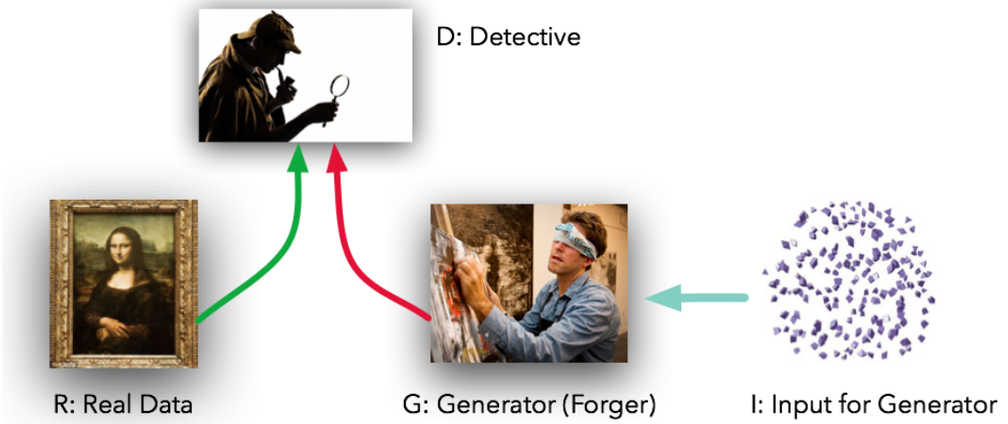
[1] Tzeng, Eric, et al. "Adversarial discriminative domain adaptation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

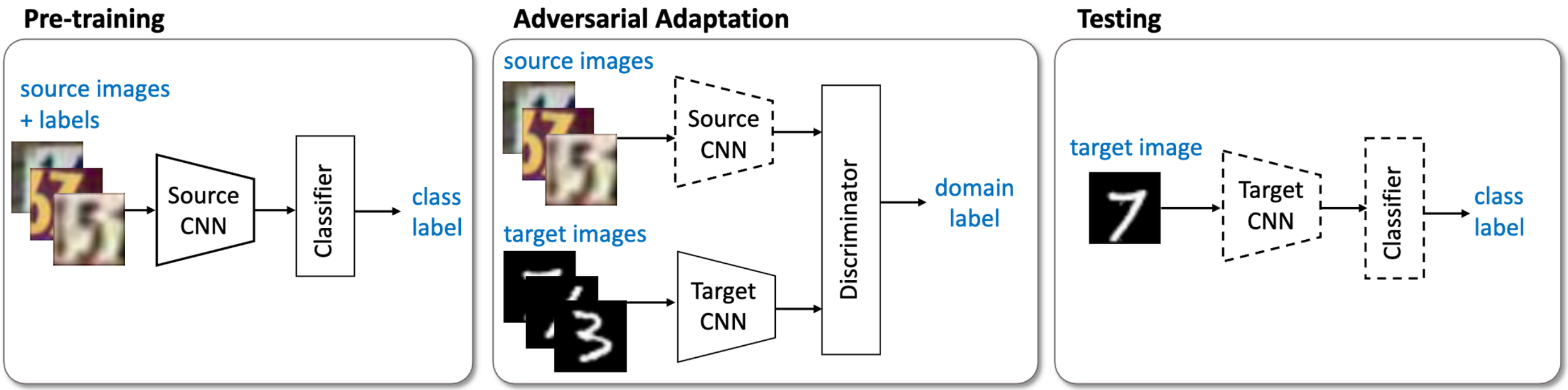
Feature / Distribution Alignment

ADDA [1]: Adversarial Discriminative Domain Adaptation

Key features:

- An encoder per domain
- **Sequential training** process (two-stage training)
- CGAN (Conditional Generative Adversarial Network) mechanism





[1] Tzeng, Eric, et al. "Adversarial discriminative domain adaptation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

Feature / Distribution Alignment

DANN [1]: Domain Adversarial Training for Neural Network

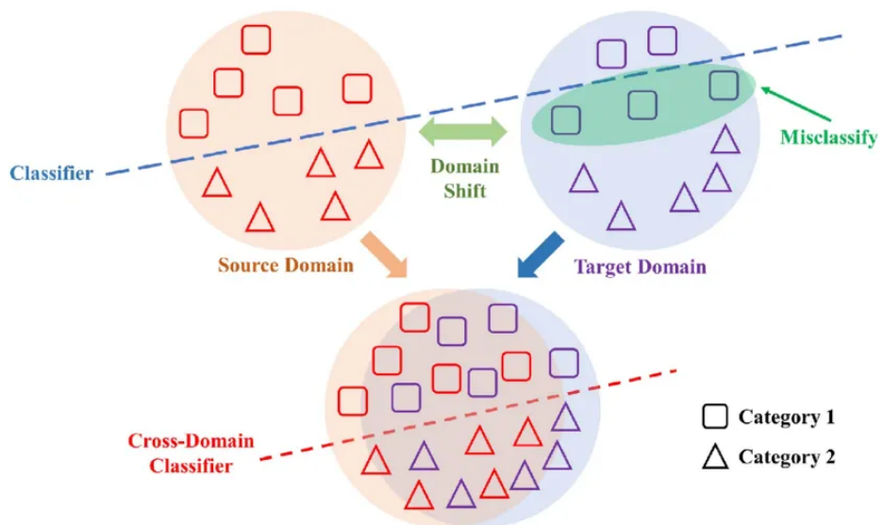
[1] Ganin, Yaroslav, et al. "Domain-adversarial training of neural networks." *Journal of machine learning research* 17.59 (2016): 1-35.

Feature / Distribution Alignment

DANN [1]: Domain Adversarial Training for Neural Network

Key features:

- A single encoder for both Source and Target domains
- An **End-to-End** training strategy
- **Adversarial learning** based strategy
- **Three** main **components**: Encoder, Classification Head, Domain Head



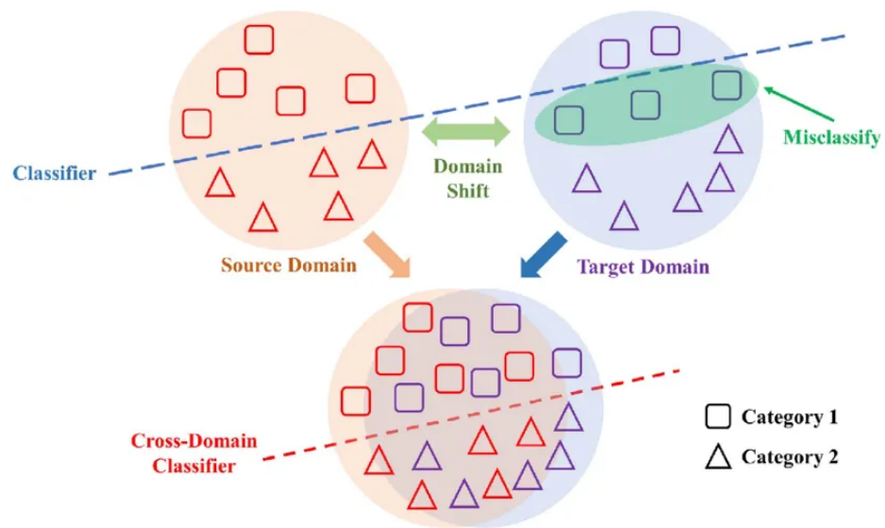
[1] Ganin, Yaroslav, et al. "Domain-adversarial training of neural networks." *Journal of machine learning research* 17.59 (2016): 1-35.

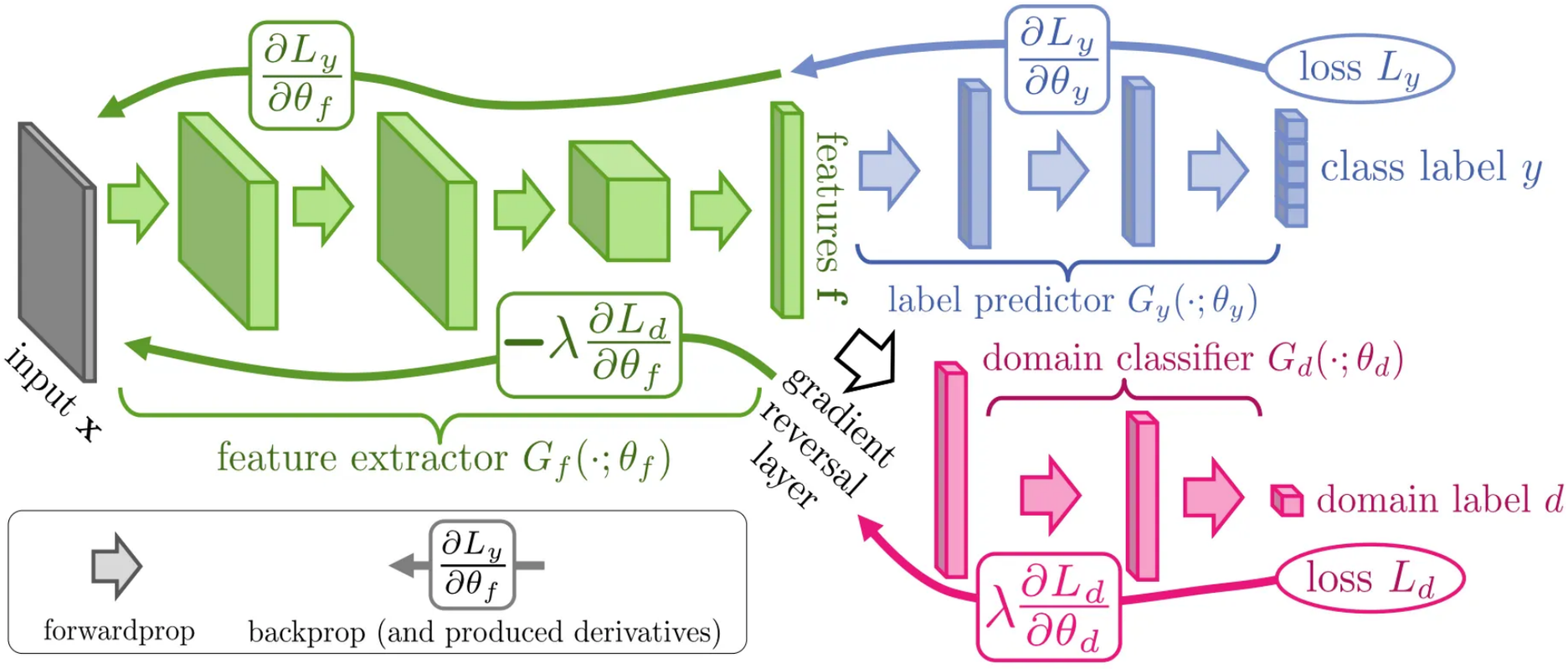
Feature / Distribution Alignment

DANN [1]: Domain Adversarial Training for Neural Network

Key features:

- A single encoder for both Source and Target domains
- An **End-to-End** training strategy
- **Adversarial learning** based strategy
- **Three** main **components**: Encoder, Classification Head, Domain Head





[1] Ganin, Yaroslav, et al. "Domain-adversarial training of neural networks." *Journal of machine learning research* 17.59 (2016): 1-35.

Feature / Distribution Alignment

DANN [1]: Domain Adversarial Training for Neural Network

The main trick - **Gradient Reversal Layer**:

- Compute standard gradient for source sample on the Classification Head
- Compute gradient on the Domain Head (Source vs Target)
- Reverse the gradient sign, for the encoder parameters update, coming from the Domain Head

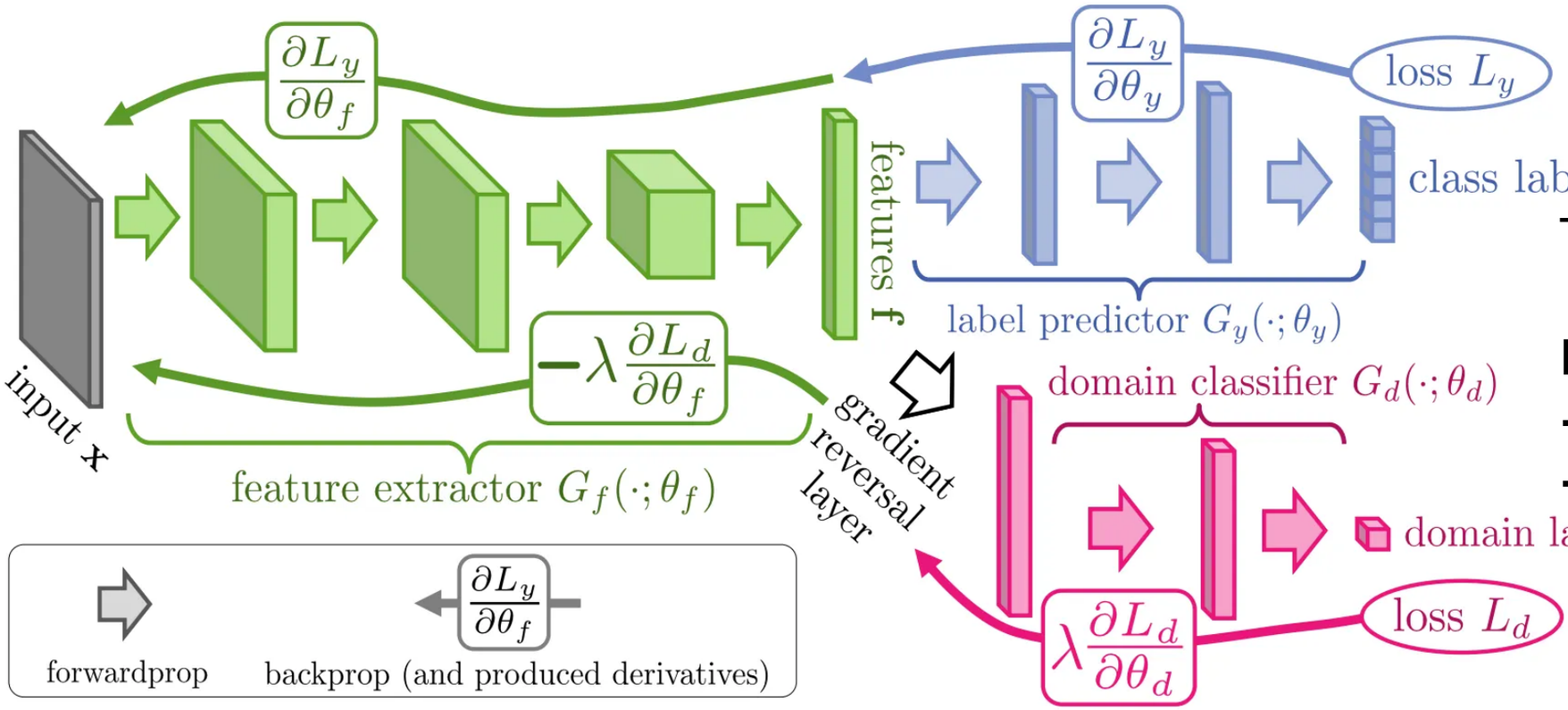
Feature / Distribution Alignment

DANN [1]: Domain Adversarial Training for Neural Network

The main trick - **Gradient Reversal Layer**:

- Compute standard gradient for source sample on the Classification Head
- Compute gradient on the Domain Head (Source vs Target)
- Reverse the gradient sign, for the encoder parameters update, coming from the Domain Head

$$\mathcal{L} = \mathcal{L}_c + \lambda \mathcal{L}_d$$



The main objective:

Learn features
- **Well suited** for the task
- **Invariant** between domains

Generative approach to fill the domain Gap

Generative Approach I2I translation

CyCADA [1]: CyCADA: Cycle-Consistent Adversarial Domain Adaptation

Key features:

- As CycleGAN, It leverages a two levels adaptation strategy:
    1. **It translates** images from one domain to the other
    2. It uses **CGAN** approach to align extracted features
- Originally introduced for any image classification task, then mainly adopted for Semantic Segmentations

[1] Hoffman, Judy, et al. "Cycada: Cycle-consistent adversarial domain adaptation." *ICML* 2018.

Generative Approach I2I translation

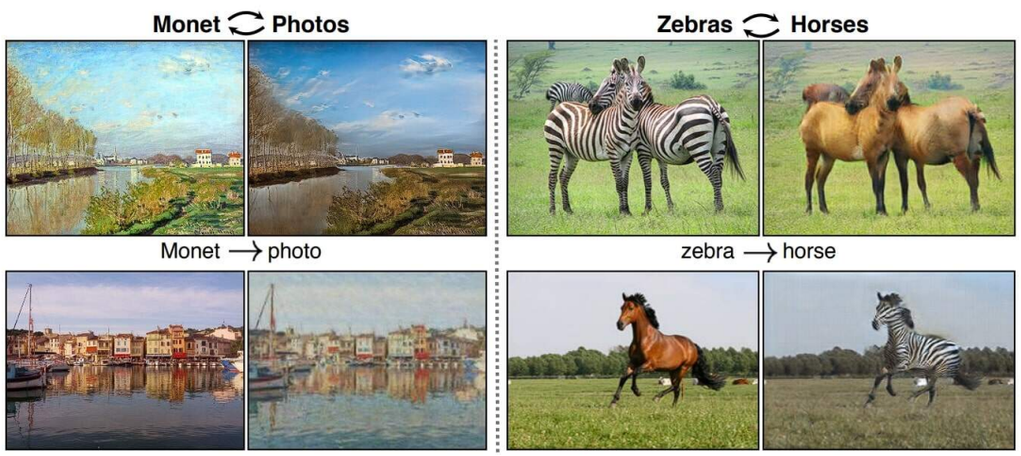
CyCADA [1]: CyCADA: Cycle-Consistent Adversarial Domain Adaptation

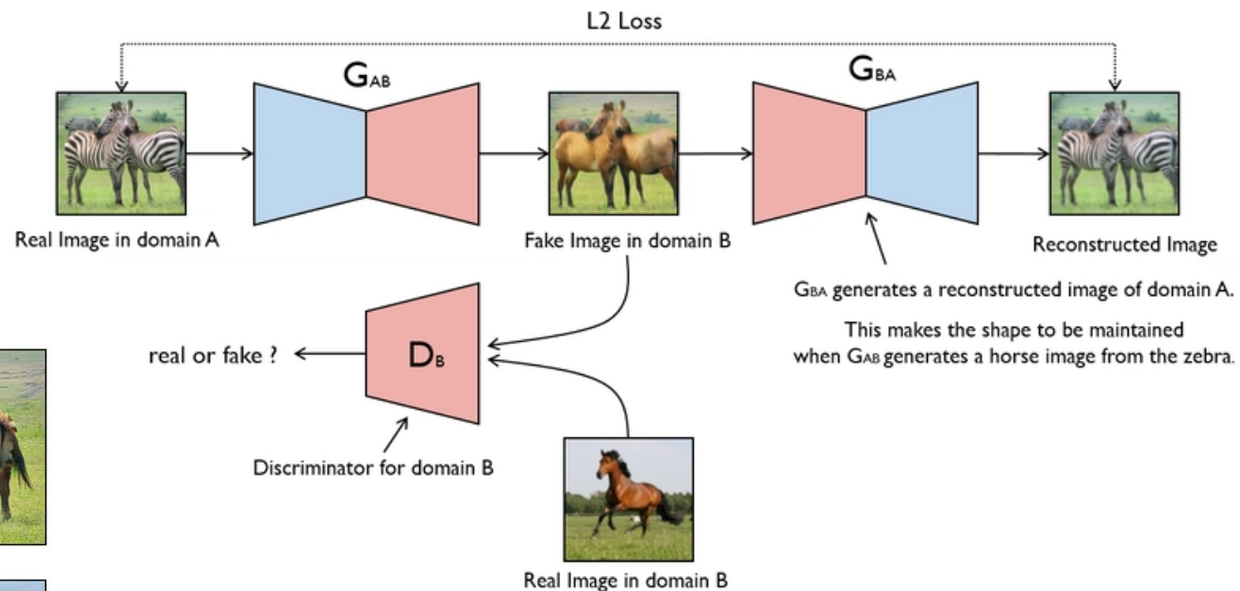
Key features:

- As CycleGAN, It leverages a two levels adaptation strategy:
    1. **It translates** images from one domain to the other
    2. It uses **CGAN** approach to align extracted features
- Originally introduced for any image classification task, then mainly adopted for Semantic Segmentations

Cycle GAN:

- Introduced for Image-To-Image Translation
- In DA: Images **from the source domain** are translated **into target domain** images





[1] Hoffman, Judy, et al. "Cycada: Cycle-consistent adversarial domain adaptation." *ICML* 2018.

Generative Approach I2I translation

CyCADA [1]: CyCADA: Cycle-Consistent Adversarial Domain Adaptation

It leverages a two levels adaptation strategy:

1. **Cycle GAN** to align input images
2. **CGAN** to align extracted features

$$
\begin{aligned}
\mathcal{L}_{\text{CyCADA}}\ & (f_T, X_S, X_T, Y_S, G_{S\to T}, G_{T\to S}, D_S, D_T) \\
& = \mathcal{L}_{\text{task}}(f_T, G_{S\to T}(X_S), Y_S) \\
& + \mathcal{L}_{\text{GAN}}(G_{S\to T}, D_T, X_T, X_S) + \mathcal{L}_{\text{GAN}}(G_{T\to S}, D_S, X_S, X_T) \\
& + \mathcal{L}_{\text{GAN}}(f_T, D_{\text{feat}}, f_S(G_{S\to T}(X_S)), X_T) \\
& + \mathcal{L}_{\text{cyc}}(G_{S\to T}, G_{T\to S}, X_S, X_T) + \mathcal{L}_{\text{sem}}(G_{S\to T}, G_{T\to S}, X_S, X_T, f_S).
\end{aligned}
$$

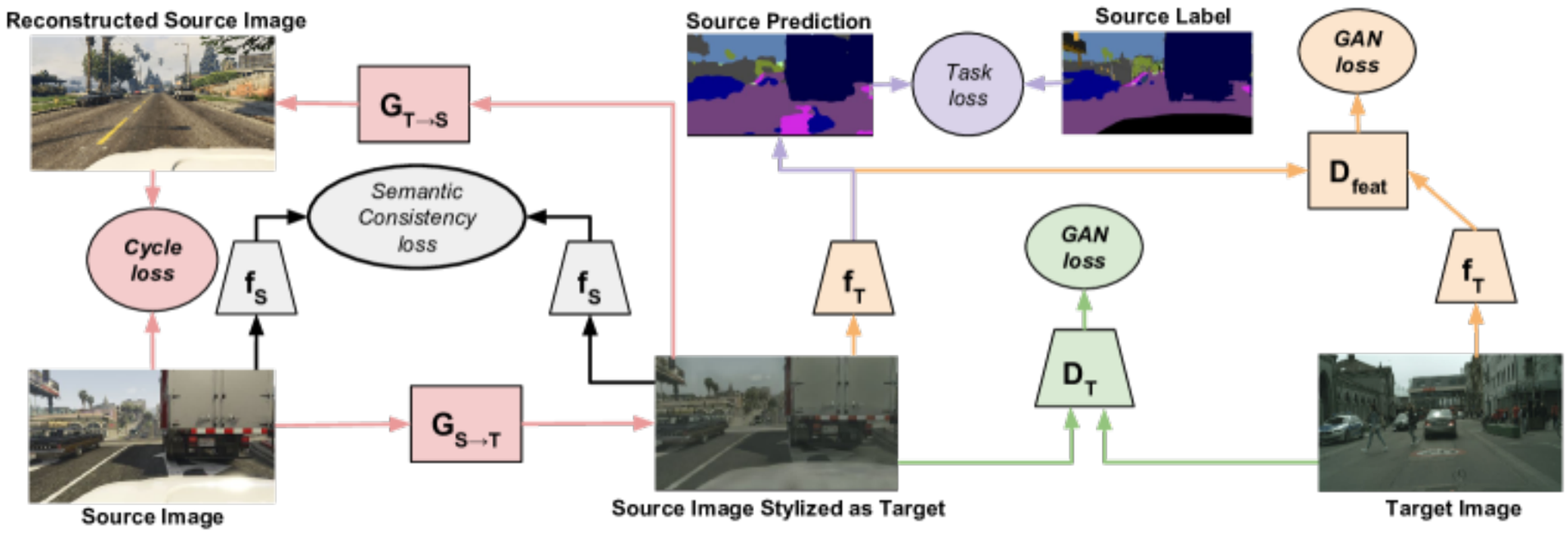
Generative Approach I2I translation

CyCADA [1]: CyCADA: Cycle-Consistent Adversarial Domain Adaptation

It leverages a two levels adaptation strategy:

1. **Cycle GAN** to align input images
2. **CGAN** to align extracted features

$$\mathcal{L}_{\text{CyCADA}}\ (f_T, X_S, X_T, Y_S, G_{S\to T}, G_{T\to S}, D_S, D_T)$$
$$= \mathcal{L}_{\text{task}}(f_T, G_{S\to T}(X_S), Y_S)$$
$$+\mathcal{L}_{\text{GAN}}(G_{S\to T}, D_T, X_T, X_S) + \mathcal{L}_{\text{GAN}}(G_{T\to S}, D_S, X_S, X_T)$$
$$+\mathcal{L}_{\text{GAN}}(f_T, D_{\text{feat}}, f_S(G_{S\to T}(X_S)), X_T)$$
$$+\mathcal{L}_{\text{cyc}}(G_{S\to T}, G_{T\to S}, X_S, X_T) + \mathcal{L}_{\text{sem}}(G_{S\to T}, G_{T\to S}, X_S, X_T, f_S).$$



Self-Training Strategies

Self-Training Strategies

Self-Training: Leveraging Unlabelled Target Data

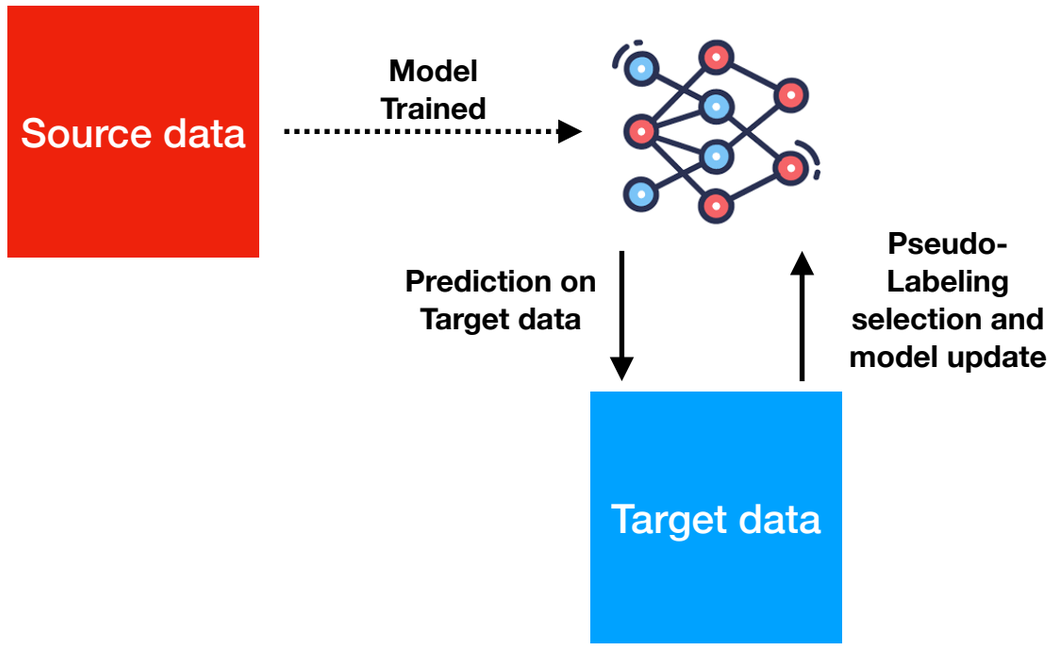
Self-Training Strategies

Self-Training: Leveraging Unlabelled Target Data

Standard pipeline for **Self-Training UDA**:

- Train model on labelled Source data
- Predict pseudo-labels on unlabelled Target data
- Filter by confidence threshold τ
- Retrain on source + confident target pseudo-labels

→ **repeat until converge or predefined # epochs**



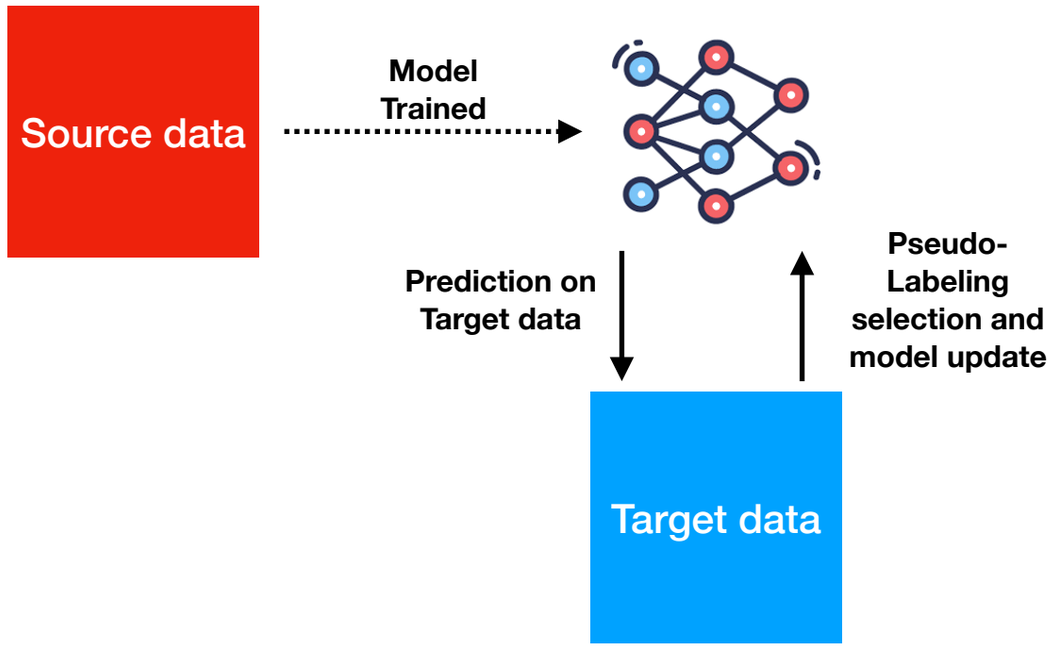
Self-Training Strategies

Self-Training: Leveraging Unlabelled Target Data

Standard pipeline for **Self-Training UDA**:

- Train model on labelled Source data
- Predict pseudo-labels on unlabelled Target data
- Filter by confidence threshold τ
- Retrain on source + confident target pseudo-labels

→ **repeat until converge or predefined # epochs**



Key elements:

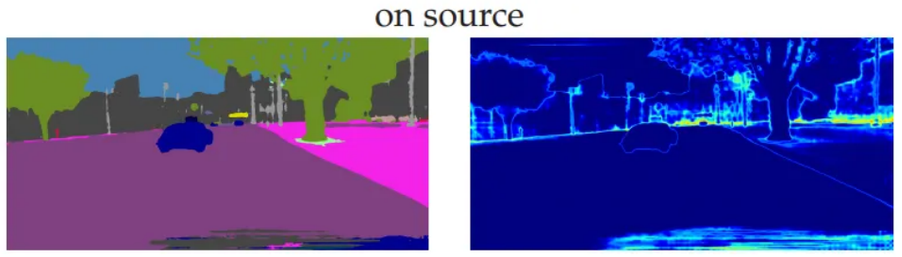
- Unlike feature alignment, self-training operates **on the target directly** (supervisory signal where there was none)
- **The key assumption**: if the model is confident, it is probably right
- The key risk: **confirmation bias:** early errors get reinforced through retraining
- Confidence threshold τ controls the trade-off b/w quantity and quality of pseudo-labels

Self-Training Strategies

ADVENT [1]: ADVENT: Adversarial Entropy Minimization for Domain Adaptation in Semantic Segmentation

Advent:

- Minimize the Entropy on the target prediction (roughly equivalent as training on pseudo-labels
- Include an adversarial strategy on the entropy maps s.t. entropy maps from Source and Target should have similar appearance



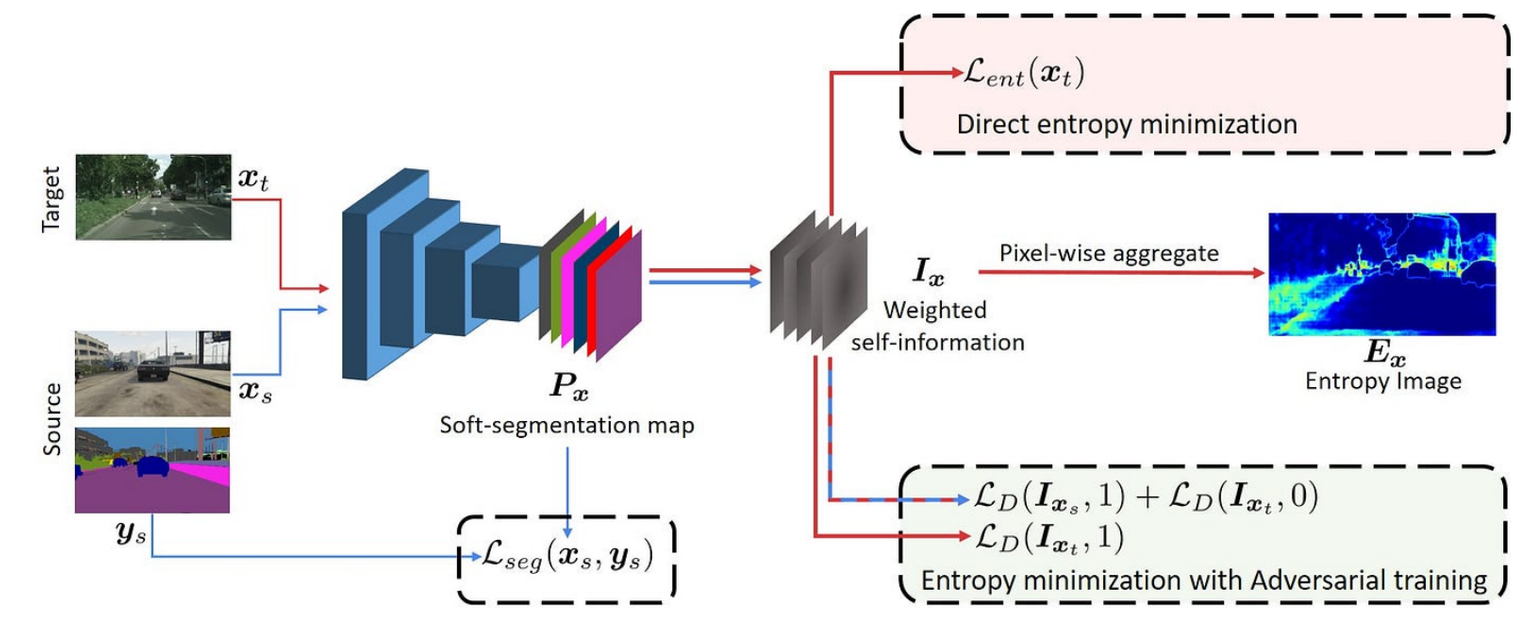
[1] Vu, Tuan-Hung, et al. "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation." *IEEE CVPR*. 2019.

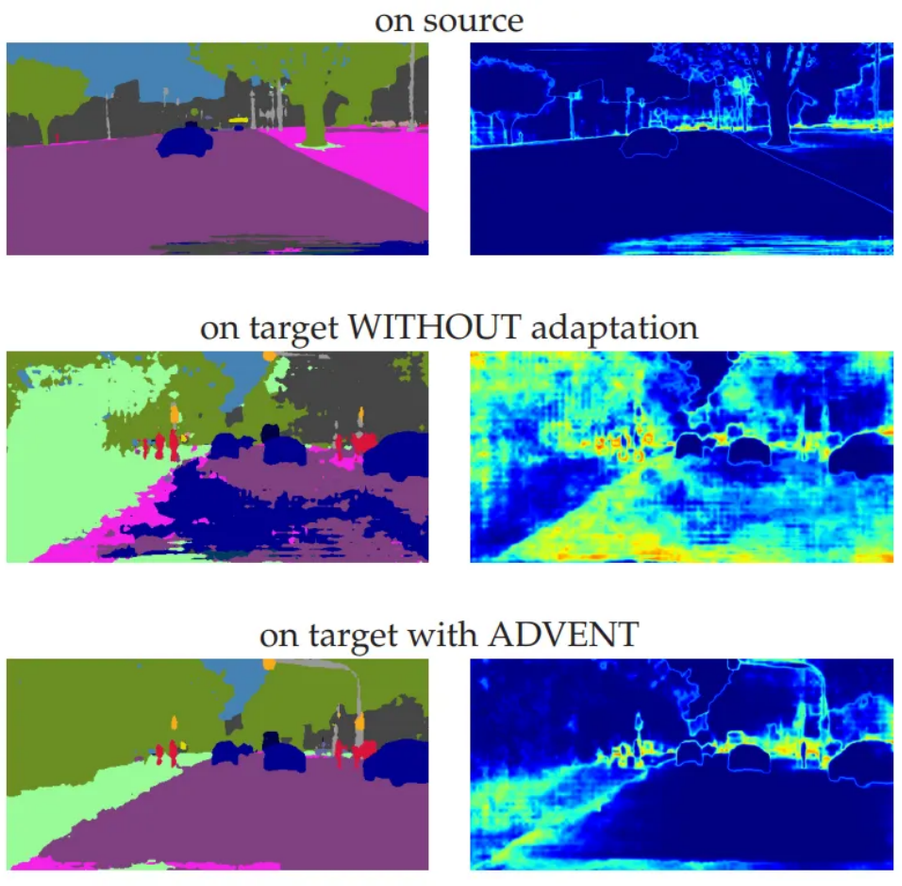
Self-Training Strategies

ADVENT [1]: ADVENT: Adversarial Entropy Minimization for Domain Adaptation in Semantic Segmentation

Advent:

- Minimize the Entropy on the target prediction (roughly equivalent as training on pseudo-labels
- Include an adversarial strategy on the entropy maps s.t. entropy maps from Source and Target should have similar appearance

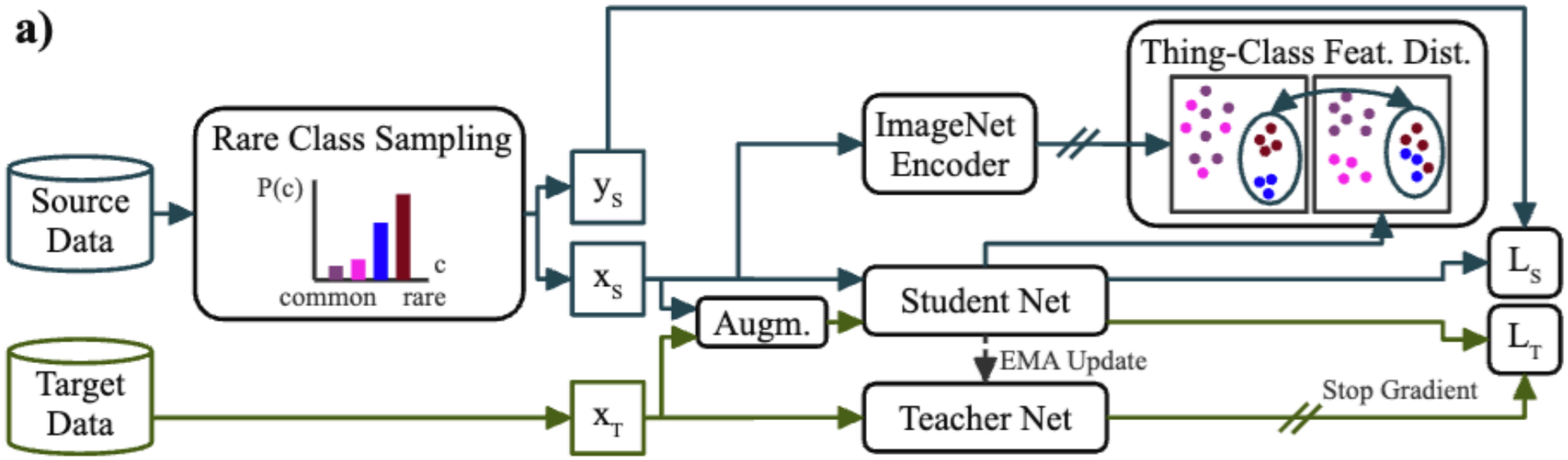




[1] Vu, Tuan-Hung, et al. "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation." *IEEE CVPR*. 2019.

Self-Training Strategies

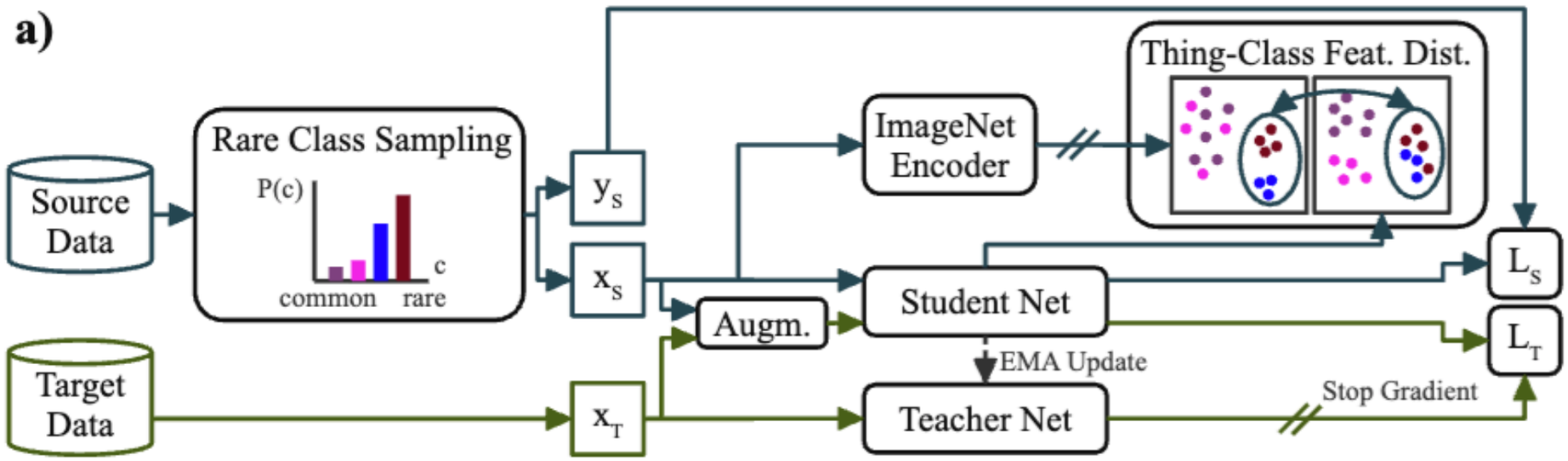
DAformer [1]: Daformer: Improving network architecture and training strategies for domain-adaptive semantic segmentation



[1] Hoyer, Lukas, Dengxin Dai, and Luc Van Gool. "Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation." *IEEE CVPR*. 2022.

Self-Training Strategies

DAformer [1]: Daformer: Improving network architecture and training strategies for domain-adaptive semantic segmentation



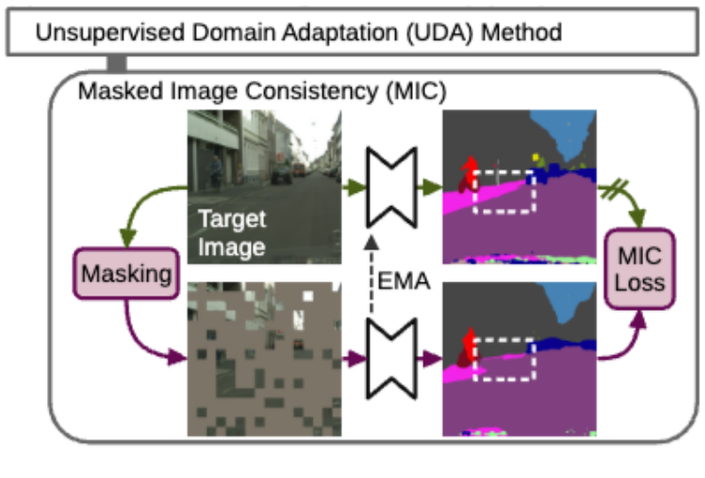
Daformer:

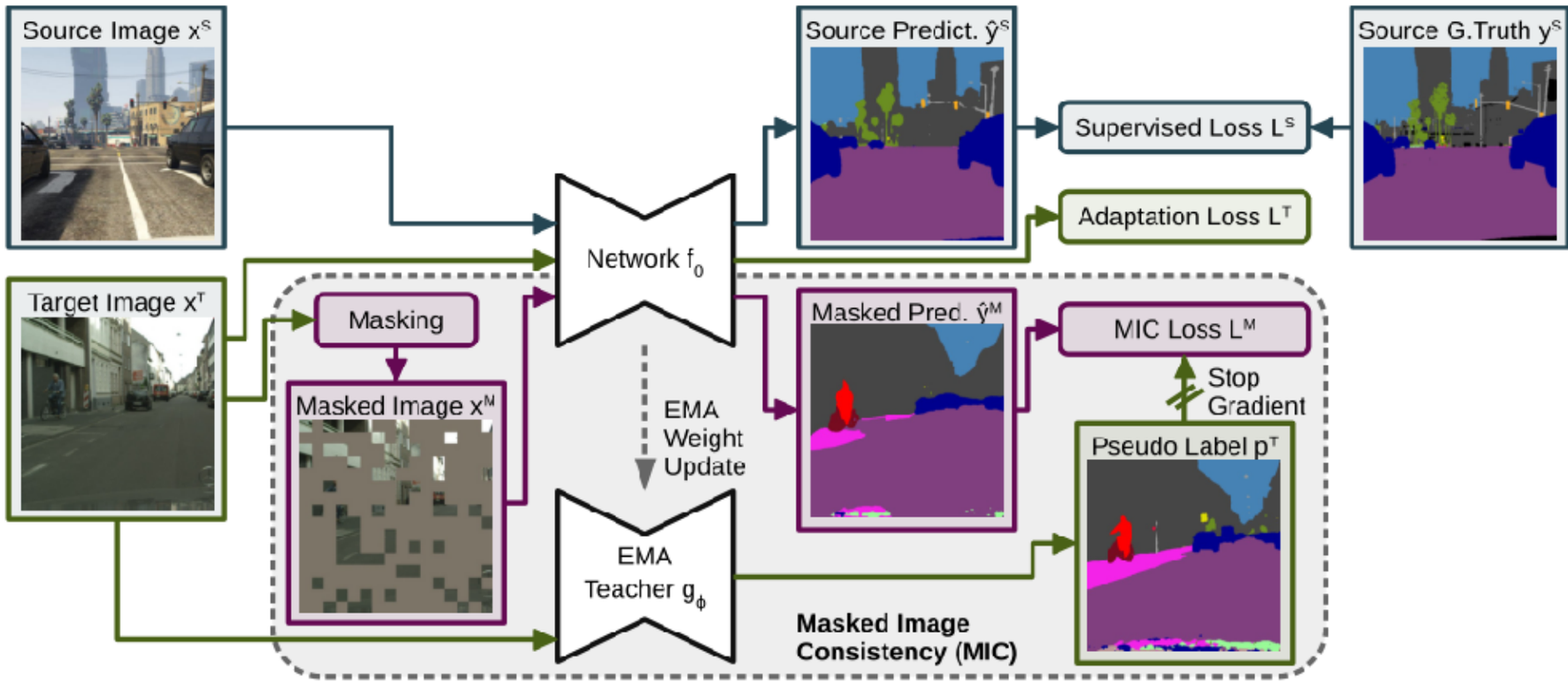
- Tailored for **Semantic Segmentation** tasks
- First framework using **Transformer for UDA SemSeg** (MiT-B5 from SegFormer)
- It implements a **"Rare Class Sampling" strategy**: explicitly oversamples samples with rare classes
- It leverages a **Teacher/Student mechanism** to generate **pseudo-labels** on Target samples
- It uses a **pretrained backbone on ImageNet** (RGB) data

[1] Hoyer, Lukas, Dengxin Dai, and Luc Van Gool. "Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation." *IEEE CVPR*. 2022.

Self-Training Strategies

MIC [1]: MIC: Masked Image Consistency for Context-Enhanced Domain Adaptation

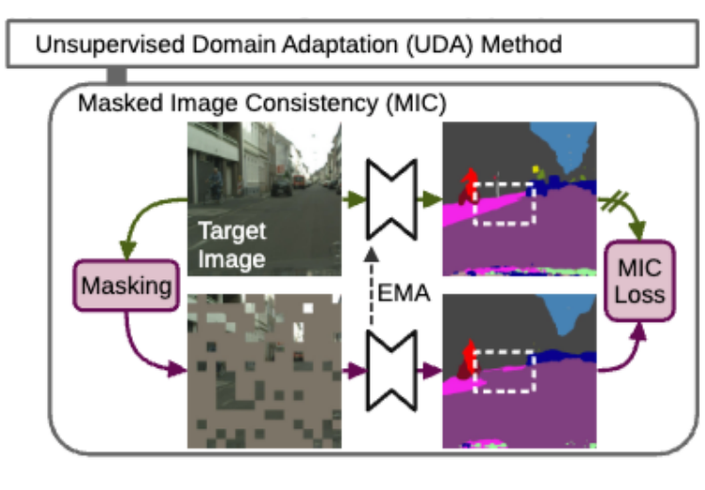


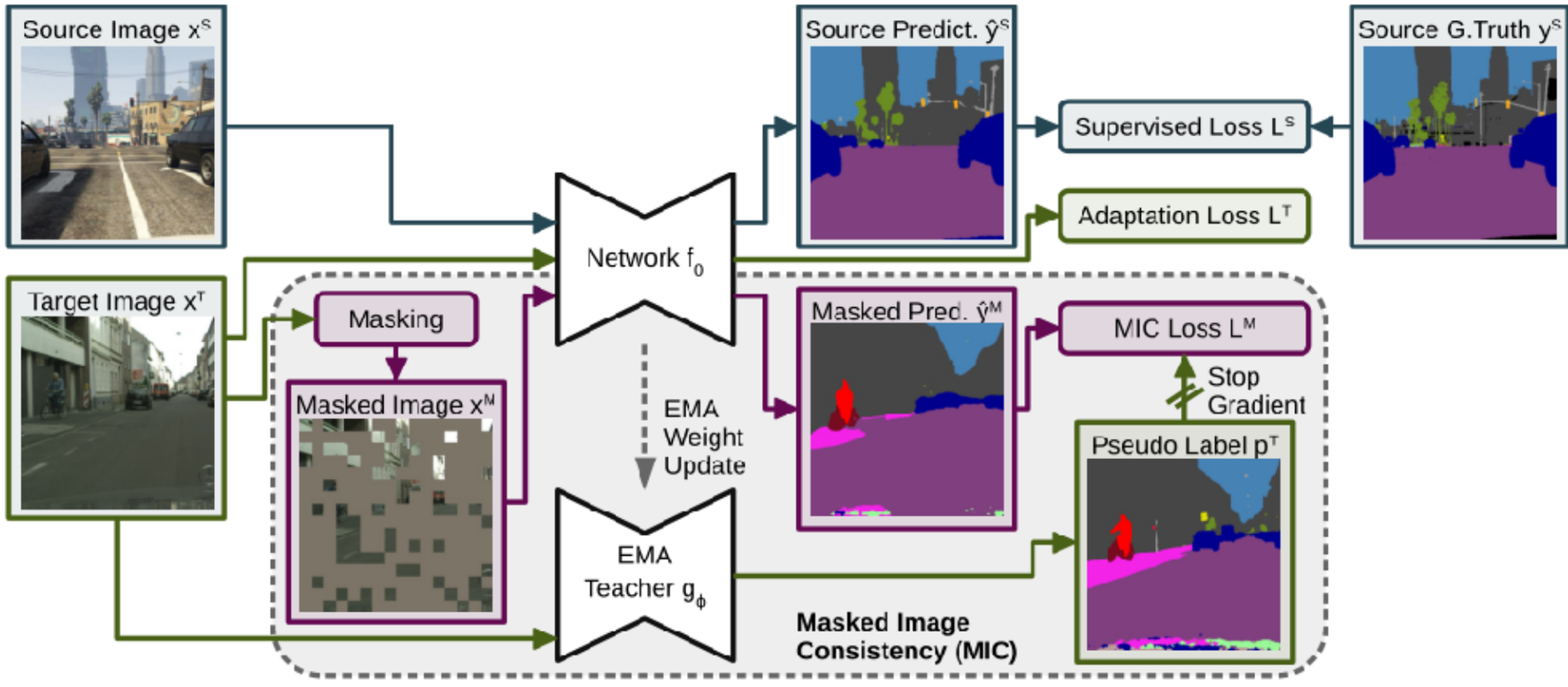


[1] Hoyer, Lukas, et al. "MIC: Masked image consistency for context-enhanced domain adaptation." *IEEE CVPR*. 2023.

Self-Training Strategies

MIC [1]: MIC: Masked Image Consistency for Context-Enhanced Domain Adaptation





MIC:

- Is a direct **extension of DAFormer** (the same research group)
- Still tailored for **Semantic Segmentation tasks**
- It leverages **Masked Image Modelling** to enforce feature alignment on the Target domain
- The model is forced to use **global structural context** (instead of texture) to make its prediction
- It has **connection with Self-Supervised Learning** (loosely related to MAE)
- It is a **plug-and-play strategy** that can be used for many other SemSeg+UDA approaches

[1] Hoyer, Lukas, et al. "MIC: Masked image consistency for context-enhanced domain adaptation." *IEEE CVPR*. 2023.

Unsupervised Domain Adaptation

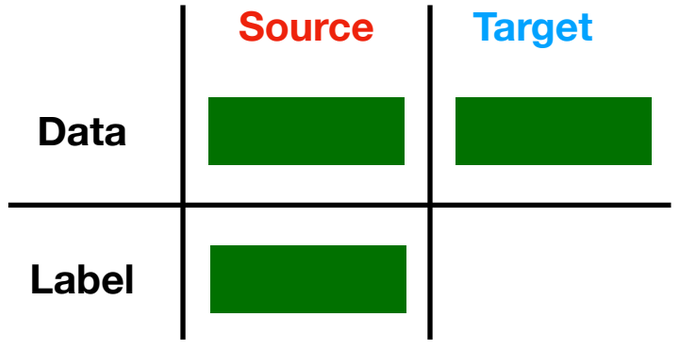
	Instance Reweighting	Feature Alignment	Generative	Self-Training
Core mechanism	Reweight source samples to match target distribution	Minimise distance between source and target feature distributions	Translate images across domains at pixel level	Generate pseudo-labels on target, retrain iteratively
Alignment level	Sample space	Feature space	Pixel space	Prediction space
Representative methods	KMM, KLIEP	CORAL, DANN, ADDA	CyCADA, CycleGAN	ADVENT, DAFormer, MIC
Source data at adapt. time	✓ Required	✓ Required	✓ Required	△ Method-dependent
Needs adversarial training	✗ No	Partial (DANN, ADDA)	✓ Yes (GAN)	✗ No
Theoretical grounding	★★★ Strong (covariate shift theory)	★★ Moderate (Ben-David bound)	★ Weak (empirical)	★★ Moderate (entropy minimisation)
Scales to high dimensions	✗ Poor	✓ Good	✓ Good	✓ Good
Computational cost	🟢 Low	🟡 Medium	🔴 High	🟡 Medium
Main weakness	Breaks in high dimensions, support mismatch	No guarantee features remain discriminative	Training instability, high compute	Confirmation bias, class imbalance
Best suited for	Low-dim, tabular, shallow models	General purpose, image classification	Large visual domain gaps	Segmentation, dense prediction

Unsupervised Domain Adaptation

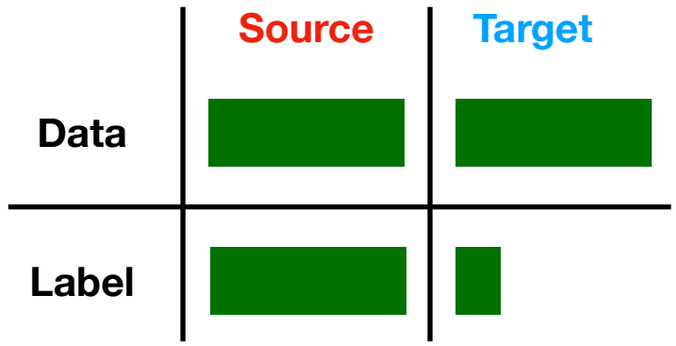
Family	Take-home
Instance reweighting	Theoretically clean but practically limited to low dimensions
Feature alignment	The workhorse of deep UDA, robust and general
Generative	Powerful for large visual gaps but expensive and unstable
Self-training	The most recent approaches, scales to dense tasks

Domain Adaptation

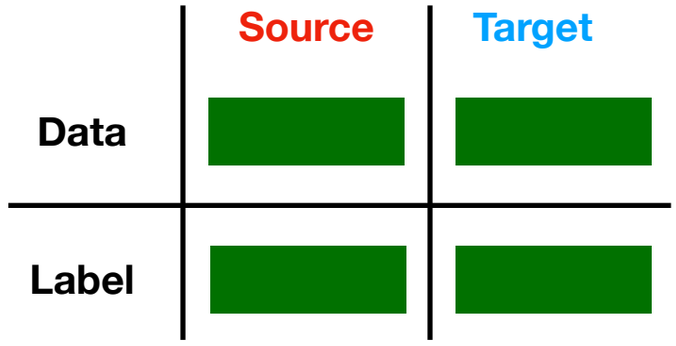
Unsupervised Domain Adaptation (**UDA**)



Semi-Supervised Domain Adaptation (**SSDA**)



Supervised Domain Adaptation (**SDA**)



Semi-Supervised Domain Adaptation

The key elements to support **Semi-Supervised Domain Adaptation** (SSDA) [1]:

- Source domain fully labeled, target domain has **a small number of labeled samples** (typically 1, 3, or 5 per class) + a large unlabeled target pool.

- **Why it matters**: even a handful of target labels dramatically changes what's possible, the model can now anchor its representation to the actual target distribution.

[1] Saito, Kuniaki, et al. "Semi-supervised domain adaptation via minimax entropy." *Proceedings of the IEEE/CVF international conference on computer vision*. 2019.
[2] Van Engelen, Jesper E., and Holger H. Hoos. "A survey on semi-supervised learning." *Machine learning* 109.2 (2020): 373-440.

Semi-Supervised Domain Adaptation

The key elements to support **Semi-Supervised Domain Adaptation** (SSDA) [1]:

- Source domain fully labeled, target domain has **a small number of labeled samples** (typically 1, 3, or 5 per class) + a large unlabeled target pool.

- **Why it matters**: even a handful of target labels dramatically changes what's possible, the model can now anchor its representation to the actual target distribution.

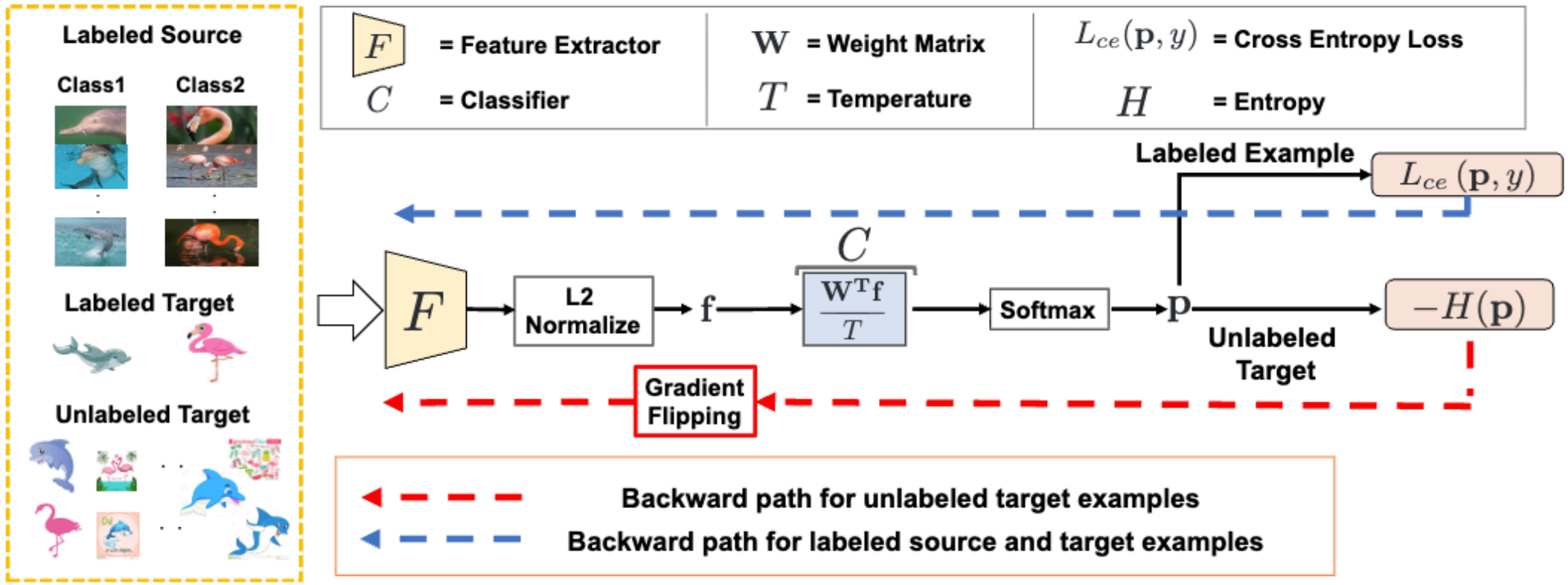
- **Main question:** How to best exploit the few target labels without overfitting, while still leveraging the large unlabeled target pool (question tightly connected with general Semi-Supervised Learning [2]).

- Contrast with UDA (no target labels) and SDA (fully labeled target): **SSDA sits in the most practically realistic scenarios** (we can almost always assume to have very few target labelled samples).

[1] Saito, Kuniaki, et al. "Semi-supervised domain adaptation via minimax entropy." *Proceedings of the IEEE/CVF international conference on computer vision*. 2019.
[2] Van Engelen, Jesper E., and Holger H. Hoos. "A survey on semi-supervised learning." *Machine learning* 109.2 (2020): 373-440.

Semi-Supervised Domain Adaptation

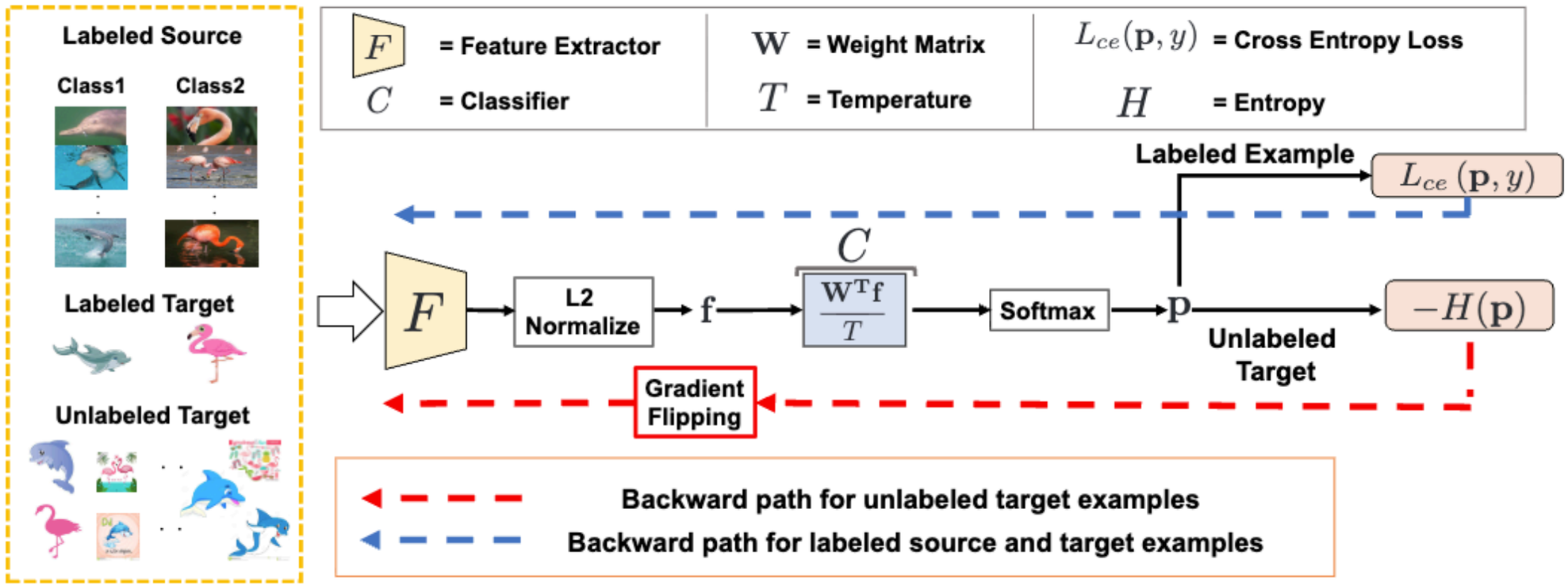
MME [1]: Semi-supervised domain adaptation via minimax entropy



[1] Saito, Kuniaki, et al. "Semi-supervised domain adaptation via minimax entropy." *Proceedings of the IEEE/CVF international conference on computer vision*. 2019.

Semi-Supervised Domain Adaptation

MME [1]: Semi-supervised domain adaptation via minimax entropy



MME Ideas:

- Minimax Entropy approach that **adversarially optimizes an adaptive few-shot model.**
- Inspired by Few-Shot Learning (**class prototypes** to make decision).
- Adaptation is achieved by alternately **maximizing** the cond. entropy of unlabeled target data w.r.t. the **classifier** and **minimizing** it w.r.t. the **feature encoder**.
- The entropy maximisation part plays as a **regulariser on the prototype geometry**.
- **All labelled** (source and target) data are **combined together** (preventing degenerate solutions) while unsupervised loss are used on the remaining (unlabelled) target data

[1] Saito, Kuniaki, et al. "Semi-supervised domain adaptation via minimax entropy." *Proceedings of the IEEE/CVF international conference on computer vision*. 2019.

Semi-Supervised Domain Adaptation

CDAC [1]: Semi-supervised domain adaptation with source label adaptation

[1] Yu, Yu-Chu, and Hsuan-Tien Lin. "Semi-supervised domain adaptation with source label adaptation." *CVPR*. 2023.

Semi-Supervised Domain Adaptation

CDAC [1]: Semi-supervised domain adaptation with source label adaptation

CDAC extends MME by adding two ingredients:

- **Contrastive learning** to enforce semantic consistency between labeled and unlabeled target samples. Target labelled examples considered as reliable anchors while all the unlabelled target data are associated with the pseudo-labels.
- **Adaptive pseudo-labeling** to progressively assign reliable class labels to unlabeled target samples and use them in the contrastive mechanism. A **class-specific adaptive threshold** is used to cope with class specificity.

[1] Yu, Yu-Chu, and Hsuan-Tien Lin. "Semi-supervised domain adaptation with source label adaptation." *CVPR*. 2023.

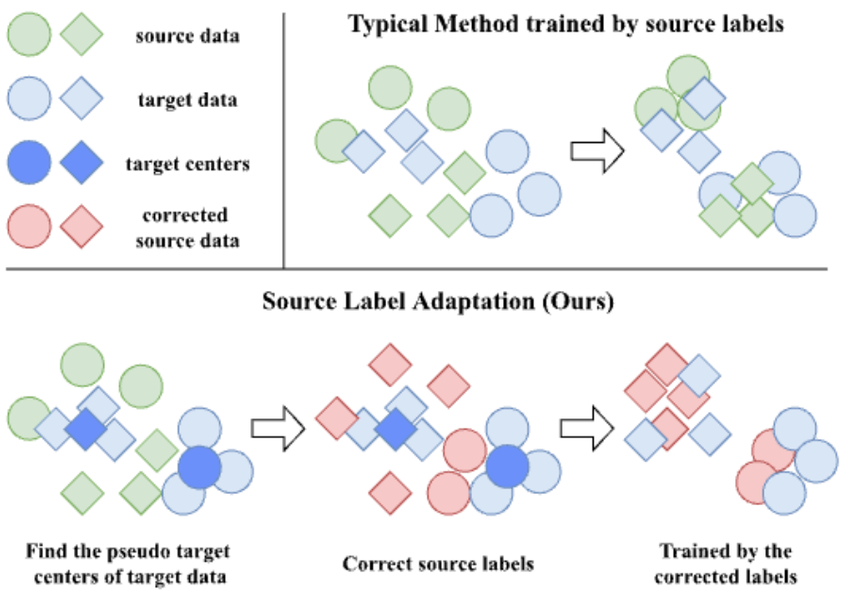
Semi-Supervised Domain Adaptation

CDAC [1]: Semi-supervised domain adaptation with source label adaptation

CDAC extends MME by adding two ingredients:

- **Contrastive learning** to enforce semantic consistency between labeled and unlabeled target samples. Target labelled examples considered as reliable anchors while all the unlabelled target data are associated with the pseudo-labels.
- **Adaptive pseudo-labeling** to progressively assign reliable class labels to unlabeled target samples and use them in the contrastive mechanism. A **class-specific adaptive threshold** is used to cope with class specificity.

- A **connection with Noisy Label Learning** (NLL) is also provided for CDAC
- The manuscript shows that standard **NLL** approaches **cannot directly deal with SSDA** tasks.



[1] Yu, Yu-Chu, and Hsuan-Tien Lin. "Semi-supervised domain adaptation with source label adaptation." *CVPR*. 2023.

Beyond classical UDA

What happens when we drop each assumption one by one?

Some classical UDA assumptions:

- **A1 — Shared feature space**: source and target live in the same representation space
- **A2 — Shared label space**: source and target have exactly the same set of classes
- **A3 — Only a single labelled source domain:** multiple (labelled) source domain can exist
- **A4 — Source data always available:** scenarios in which only the model and target data are available due to data restrictions / limitations

Beyond classical UDA

What happens when we drop each assumption one by one?

Some classical UDA assumptions:

- **A1 — Shared feature space**: source and target live in the same representation space
- **A2 — Shared label space**: source and target have exactly the same set of classes
- **A3 — Only a single labelled source domain:** multiple (labelled) source domain can exist
- **A4 — Source data always available:** scenarios in which only the model and target data are available due to data restrictions / limitations

The methods covered so far (CORAL, DANN, CyCADA, DAFormer…) **all respect these assumptions**.

But in practice:

- A radiologist labeling chest X-rays may **annotate 20 pathology classes**, **but the target hospital only cares about 5** or about an overlapping set of classes + additional ones.
- We can access **multiple labelled datasets** for a task but spanning different data properties.
- For the same task, we can have two different sensors (optical vs radar) that produce **fundamentally different feature spaces**

Beyond classical UDA

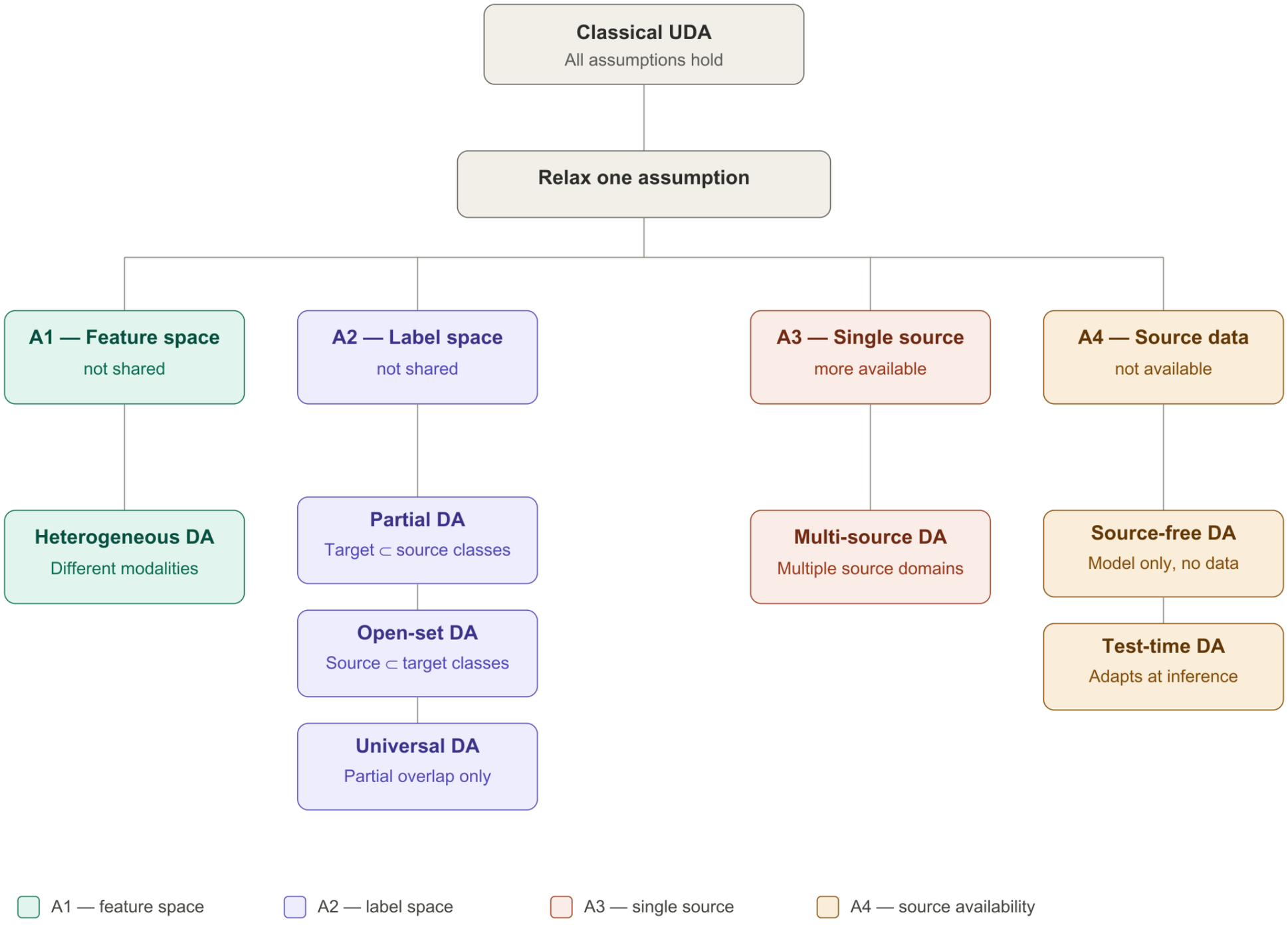
A1 — Shared feature space - Heterogeneous Domain Adaptation

A2 — Shared feature space - Partial DA, Open-Set DA, Universal DA

A3 — Only a single labelled source domain - Multi-Source Domain Adaptation

A4 — Source data always available - Source Free DA, Test-Time DA

Beyond classical UDA



Heterogeneous DA

Heterogeneous DA

Classical DA assumes source and target share the same feature space.

Heterogeneous DA relaxes this: source and target can have **different modalities, dimensionalities, or feature representations [1]**

Examples natural to your audience:

- Text → Image
- Optical image → SAR image
- LiDAR (3D points) → RGB
- Image → Audio

[1] Liu, Feng, Guangquan Zhang, and Jie Lu. "Heterogeneous domain adaptation: An unsupervised approach." *IEEE TNNLS* 31.12 (2020): 5588-5602.

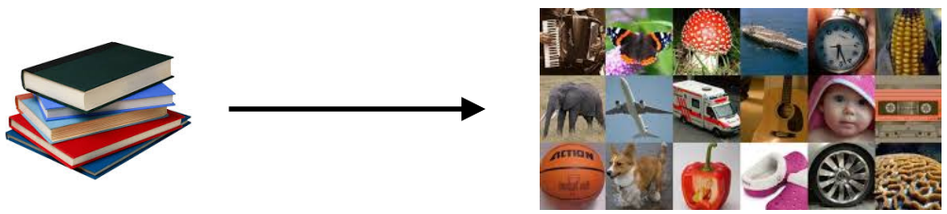
Heterogeneous DA

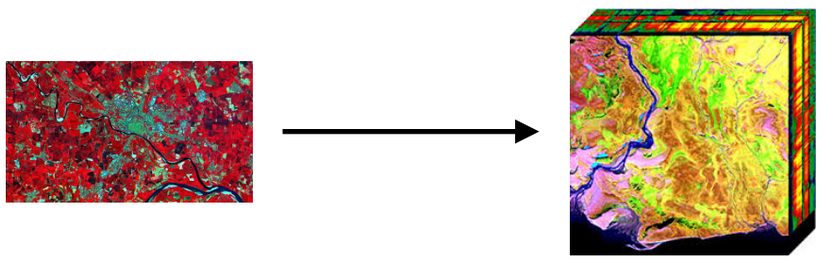
Classical DA assumes source and target share the same feature space.

Heterogeneous DA relaxes this: source and target can have **different modalities, dimensionalities, or feature representations [1]**

Examples natural to your audience:

- Text → Image
- Optical image → SAR image
- LiDAR (3D points) → RGB
- Image → Audio





[1] Liu, Feng, Guangquan Zhang, and Jie Lu. "Heterogeneous domain adaptation: An unsupervised approach." *IEEE TNNLS* 31.12 (2020): 5588-5602.

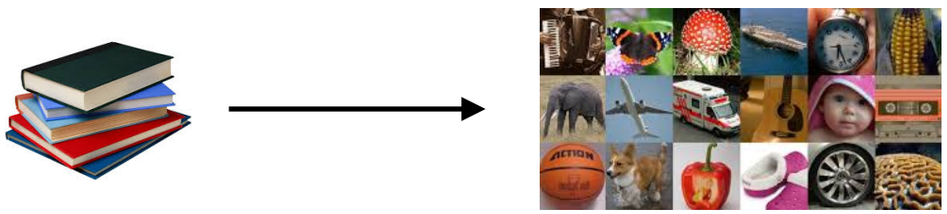
Heterogeneous DA

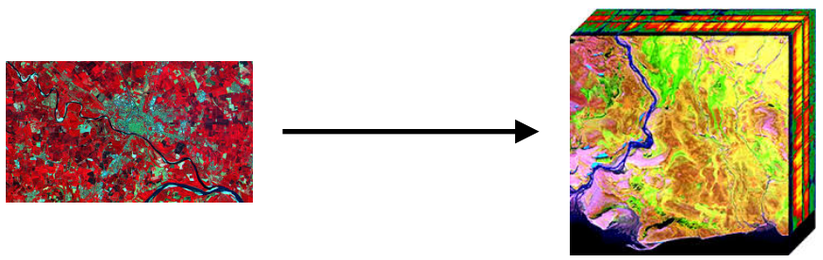
Classical DA assumes source and target share the same feature space.

Heterogeneous DA relaxes this: source and target can have **different modalities, dimensionalities, or feature representations [1]**

Examples natural to your audience:

- Text → Image
- Optical image → SAR image
- LiDAR (3D points) → RGB
- Image → Audio





Main Question:

How can we align two domains when there is no shared feature space ?

[1] Liu, Feng, Guangquan Zhang, and Jie Lu. "Heterogeneous domain adaptation: An unsupervised approach." *IEEE TNNLS* 31.12 (2020): 5588-5602.

Heterogeneous DA

Two emerging strategies for HDA

Heterogeneous DA

Two emerging strategies for HDA

Strategy 1 — Leverage pretrained backbones

- When a **rich pretrained model exists** for one or both modalities (e.g. CLIP, DINO, ImageNet-pretrained CNNs for RGB; BERT/RoBERTa for text)

- The adaptation problem reduces to **bridging two already-rich embedding spaces** rather than learning everything from scratch

- The pretrained backbone does the heavy lifting; a **lightweight alignment module** (projection head, adapter, or prompt tuning) **handles** the cross-modal **gap**

Heterogeneous DA

Two emerging strategies for HDA

Strategy 1 — Leverage pretrained backbones

- When a **rich pretrained model exists** for one or both modalities (e.g. CLIP, DINO, ImageNet-pretrained CNNs for RGB; BERT/RoBERTa for text)

- The adaptation problem reduces to **bridging two already-rich embedding spaces** rather than learning everything from scratch

- The pretrained backbone does the heavy lifting; a **lightweight alignment module** (projection head, adapter, or prompt tuning) **handles** the cross-modal **gap**

Strategy 2 — End-to-end training

- When **no pretrained backbone exists** for one or both modalities, you cannot rely on off-the-shelf representations

- The model must **jointly learn the feature extractors and the alignment** from scratch, in a single end-to-end training loop

- **More complex**, high research gap to fill

Different label Space for DA

Different label Space for DA

Classical UDA assumption:

source and target domain share **exactly the same label space**

$$\mathcal{Y}_s = \mathcal{Y}_t$$

Different label Space for DA

Classical UDA assumption:

source and target domain share **exactly the same label space**

$$\mathcal{Y}_s = \mathcal{Y}_t$$

In practice this is **frequently violated**:

- A model trained on ImageNet (1000 classes) is deployed in a medical context that only uses 10 of them

- A sentiment classifier trained on product reviews encounters customer feedback containing topics never seen in training

- A remote sensing model trained on a land-use dataset from one region encounters new land cover types in a different geography

Different label Space for DA

Classical UDA assumption:

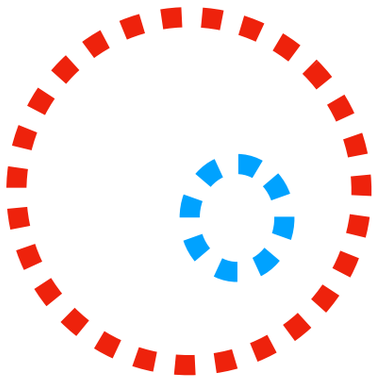
source and target domain share **exactly the same label space**

$$\mathcal{Y}_s = \mathcal{Y}_t$$

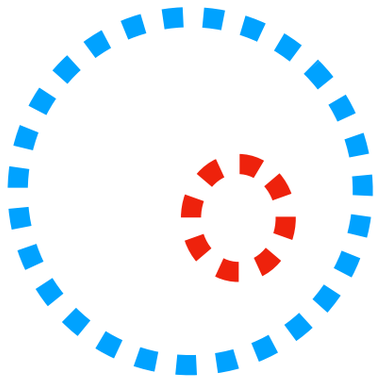
In practice this is **frequently violated**:

- A model trained on ImageNet (1000 classes) is deployed in a medical context that only uses 10 of them
- A sentiment classifier trained on product reviews encounters customer feedback containing topics never seen in training
- A remote sensing model trained on a land-use dataset from one region encounters new land cover types in a different geography

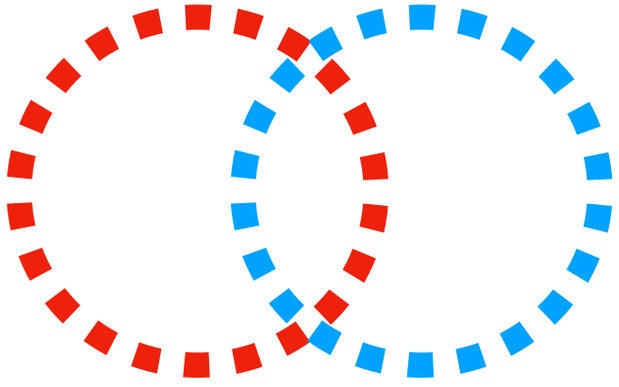
Partial DA



Open-Set DA



Universal DA



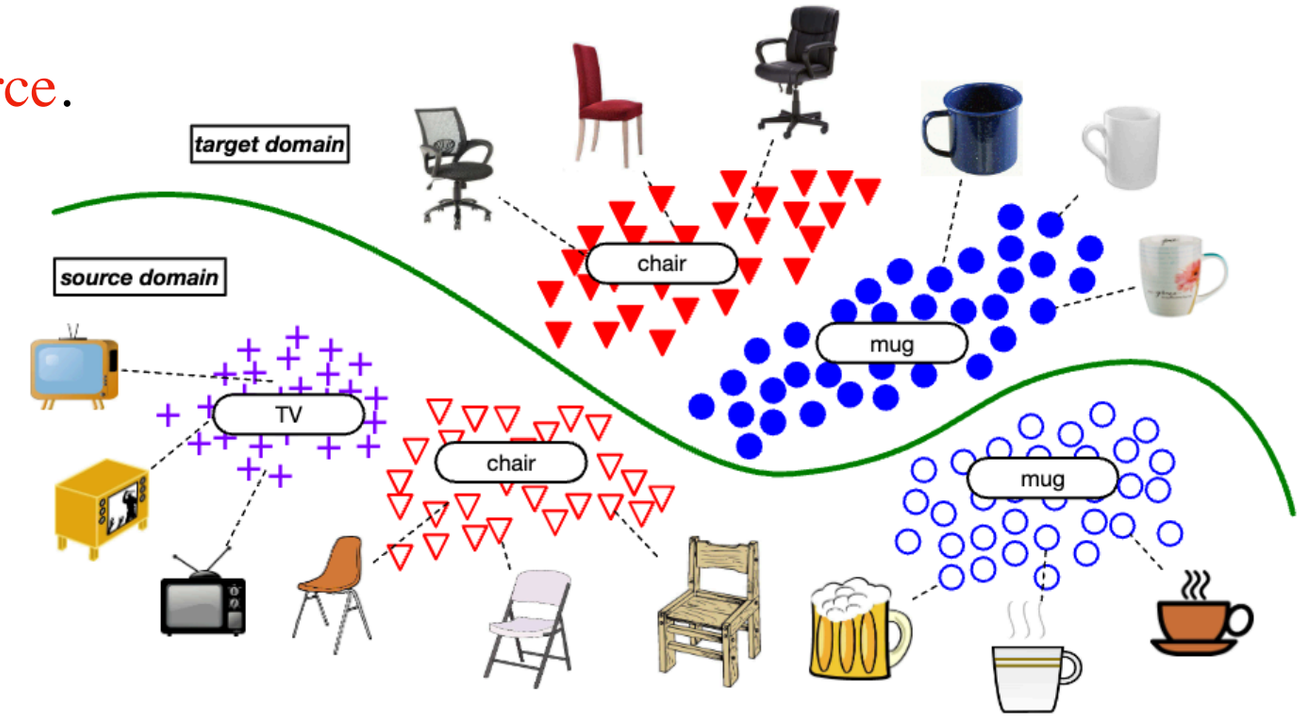
Partial DA

Setting:

The target label space is a strict subset of the source.

The target domain only cares about some of the source classes, and the rest are irrelevant.

$$\mathcal{Y}_t \subset \mathcal{Y}_s$$



[1] Cao, Zhangjie, et al. "Partial adversarial domain adaptation." *Proceedings of the European conference on computer vision (ECCV)*. 2018.
[2] *Cao, Zhangjie, et al. "Learning to transfer examples for partial domain adaptation." CVPR 2019.*

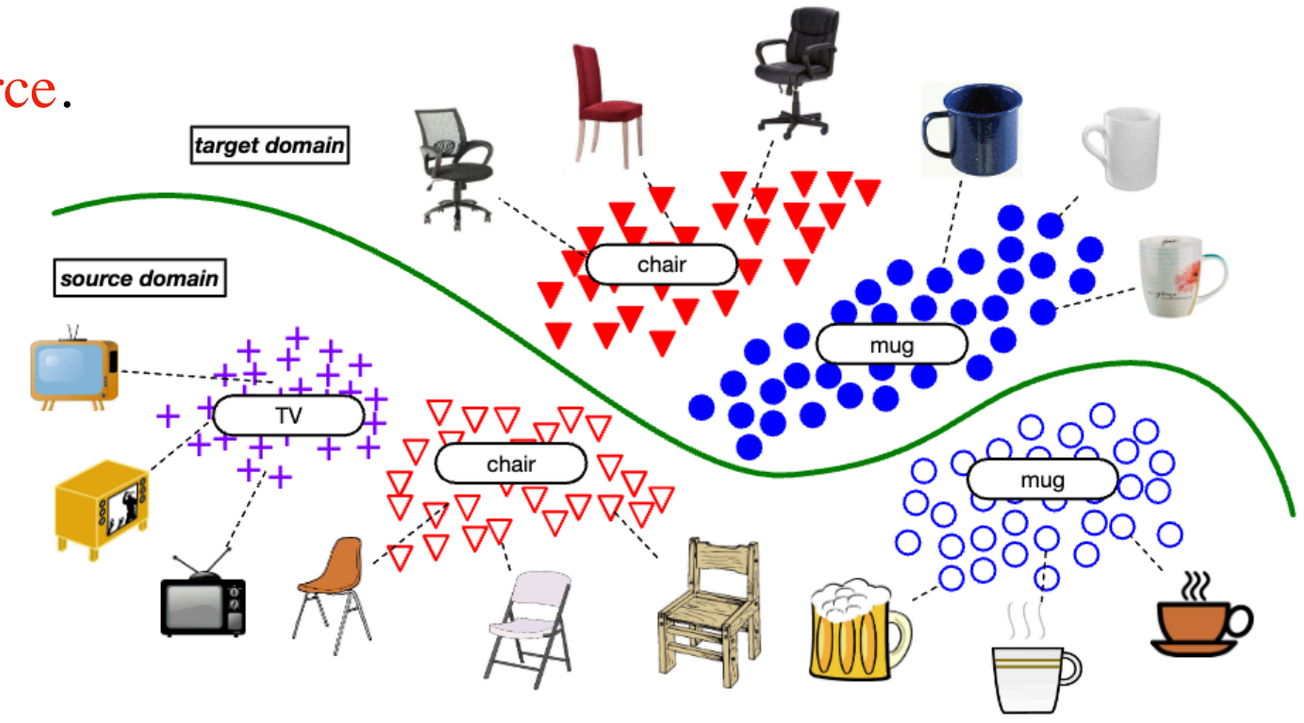
Partial DA

Setting:

The target label space is a strict subset of the source.

The target domain only cares about some of the source classes, and the rest are irrelevant.

$$\mathcal{Y}_t \subset \mathcal{Y}_s$$



Why standard UDA fails here: Standard alignment methods (DANN, CORAL…) try to align the full source distribution with the target, creating negative transfer.
-> **The model gets confused by classes that** simply **do not exist in the target.**

[1] Cao, Zhangjie, et al. "Partial adversarial domain adaptation." *Proceedings of the European conference on computer vision (ECCV)*. 2018.
[2] *Cao, Zhangjie, et al. "Learning to transfer examples for partial domain adaptation." CVPR 2019.*

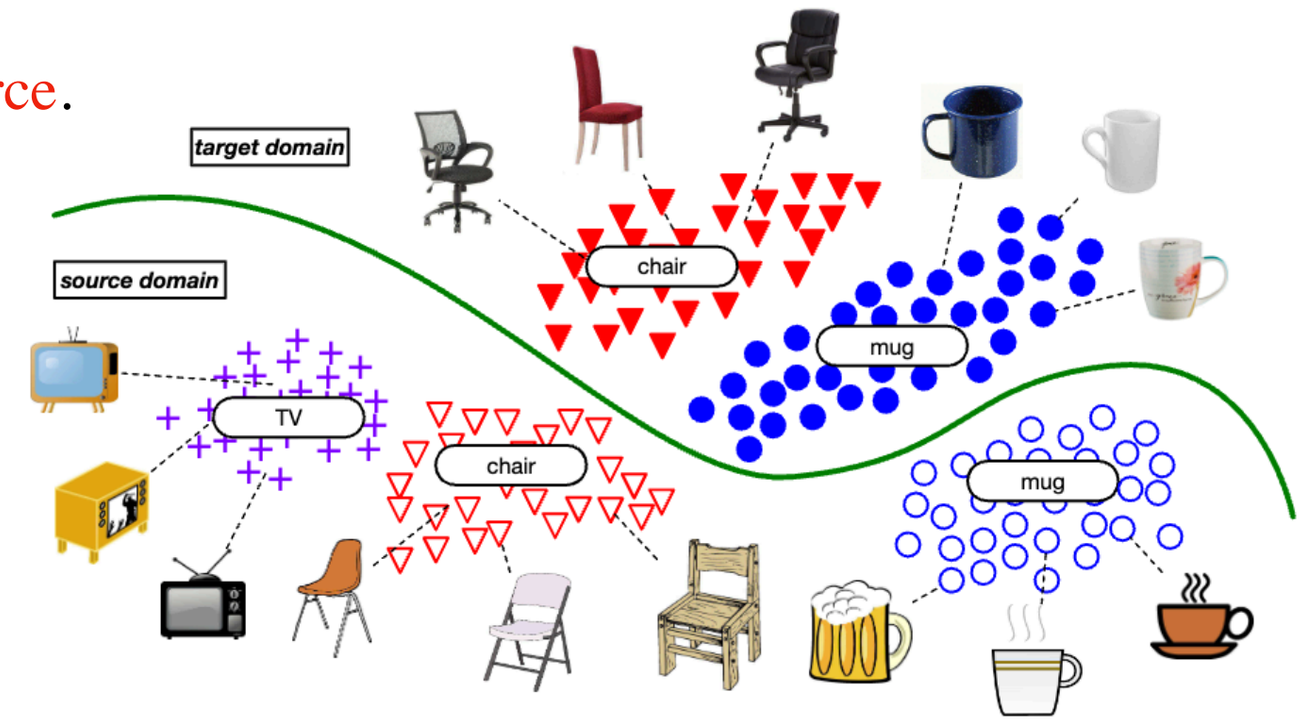
Partial DA

Setting:

The target label space is a strict subset of the source.

The target domain only cares about some of the source classes, and the rest are irrelevant.

$$\mathcal{Y}_t \subset \mathcal{Y}_s$$



Why standard UDA fails here: Standard alignment methods (DANN, CORAL…) try to align the full source distribution with the target, creating negative transfer.
-> **The model gets confused by classes that** simply **do not exist in the target.**

The key idea [1,2]

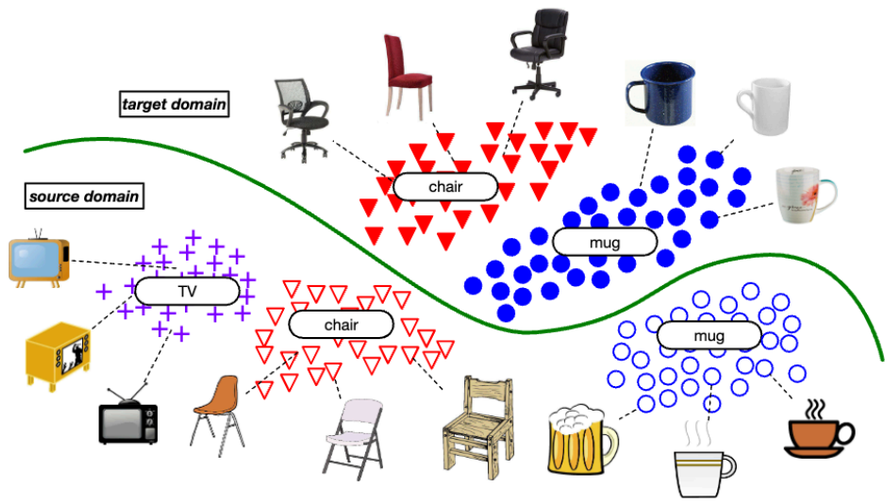
Down-weight irrelevant source classes: Rather than aligning everything, estimate which source classes are likely to appear in the target and reduce the influence of the others.

[1] Cao, Zhangjie, et al. "Partial adversarial domain adaptation." *Proceedings of the European conference on computer vision (ECCV)*. 2018.
[2] *Cao, Zhangjie, et al. "Learning to transfer examples for partial domain adaptation." CVPR 2019.*

Partial DA

PADA: Partial Adversarial Domain Adaptation [1]

- For each source class k, **compute its relevance weight wk** as the average prediction probability that unlabelled target samples belong to class k.
- If the model never predicts class k on target data, **wk** is decreased and the importance of source samples of class k is reduced in both classification and domain alignment loss.

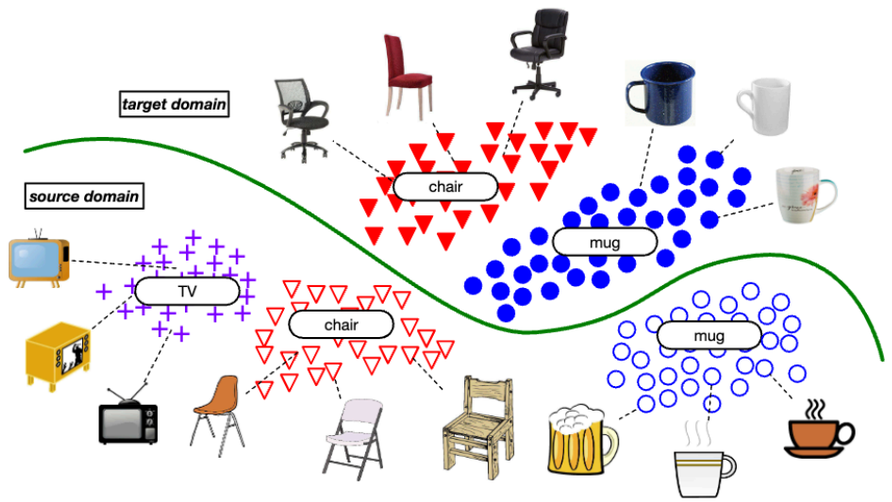


[1] Cao, Zhangjie, et al. "Partial adversarial domain adaptation." *Proceedings of the European conference on computer vision (ECCV)*. 2018.
[2] *Cao, Zhangjie, et al. "Learning to transfer examples for partial domain adaptation." CVPR 2019.*

Partial DA

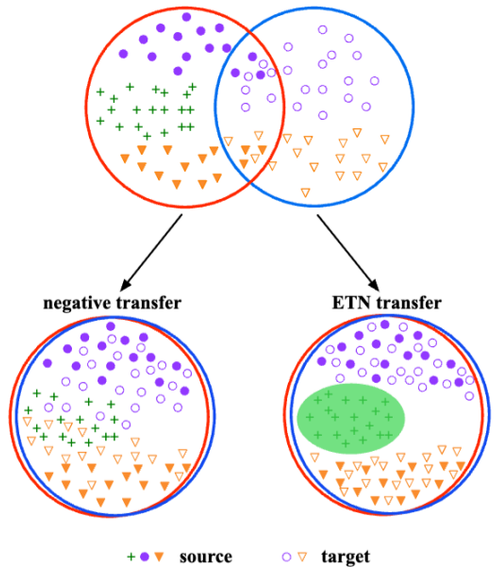
PADA: Partial Adversarial Domain Adaptation [1]

- For each source class k, **compute its relevance weight wk** as the average prediction probability that unlabelled target samples belong to class k.
- If the model never predicts class k on target data, **wk** is decreased and the importance of source samples of class k is reduced in both classification and domain alignment loss.



ETN: Learning to transfer examples for partial domain adaptation [2]

- Explicitly learning a transferability score per source sample via an aux discriminator
- Source samples from shared (source/target) classes get high transferability and contribute fully to alignment.
- Source samples from source-only classes get low transferability and are progressively excluded.

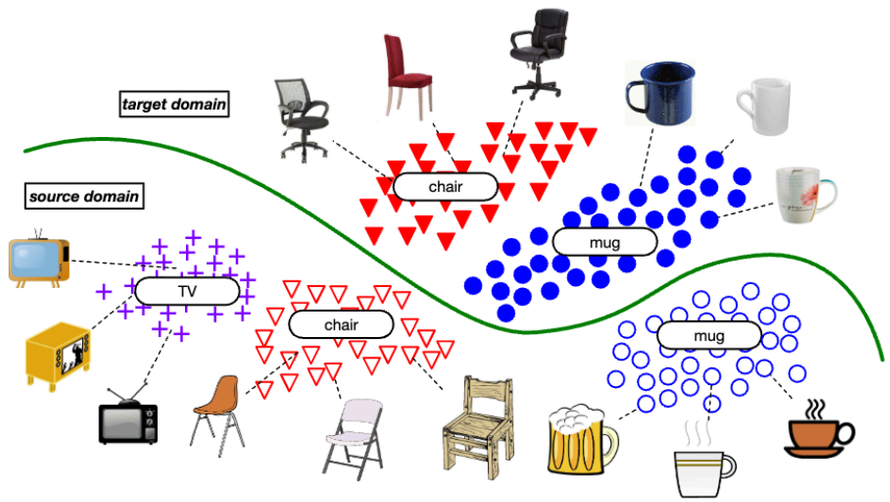


[1] Cao, Zhangjie, et al. "Partial adversarial domain adaptation." *Proceedings of the European conference on computer vision (ECCV)*. 2018.
[2] *Cao, Zhangjie, et al. "Learning to transfer examples for partial domain adaptation." CVPR 2019.*

Partial DA

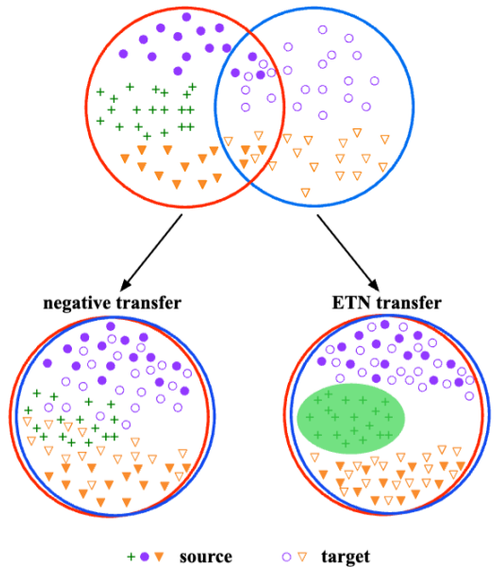
PADA: Partial Adversarial Domain Adaptation [1]

- For each source class k, **compute its relevance weight wk** as the average prediction probability that unlabelled target samples belong to class k.
- If the model never predicts class k on target data, **wk** is decreased and the importance of source samples of class k is reduced in both classification and domain alignment loss.



ETN: Learning to transfer examples for partial domain adaptation [2]

- Explicitly learning a transferability score per source sample via an aux discriminator
- Source samples from shared (source/target) classes get high transferability and contribute fully to alignment.
- Source samples from source-only classes get low transferability and are progressively excluded.



In Partial DA, the ratio is **selective alignment**:
Identify and **down-weight source classes/examples** without counterpart in the target.

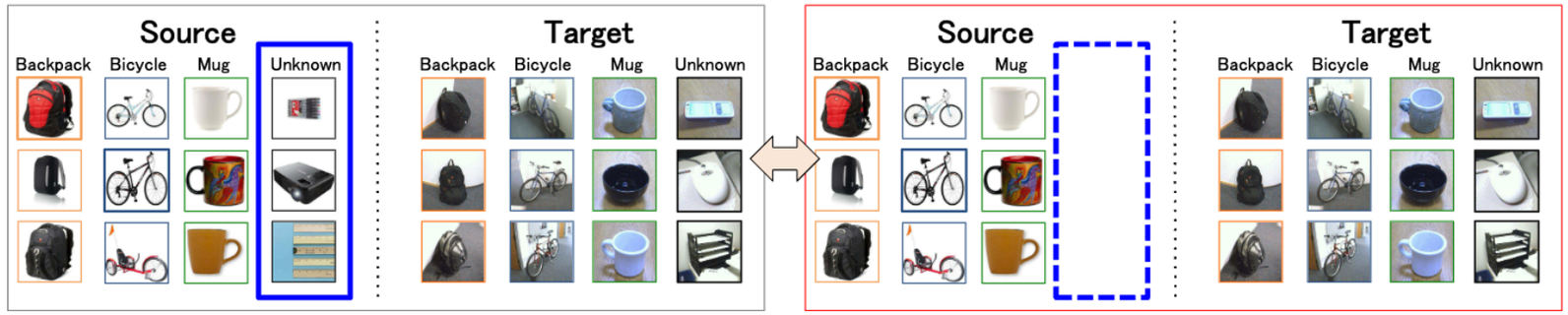
[1] Cao, Zhangjie, et al. "Partial adversarial domain adaptation." *Proceedings of the European conference on computer vision (ECCV)*. 2018.
[2] *Cao, Zhangjie, et al. "Learning to transfer examples for partial domain adaptation." CVPR 2019.*

Open-Set DA

Setting:

- The target label space is a strict superset of the source.
- The target domain cares about the source classes, plus additional ones.

$$\mathcal{Y}_s \subset \mathcal{Y}_t$$



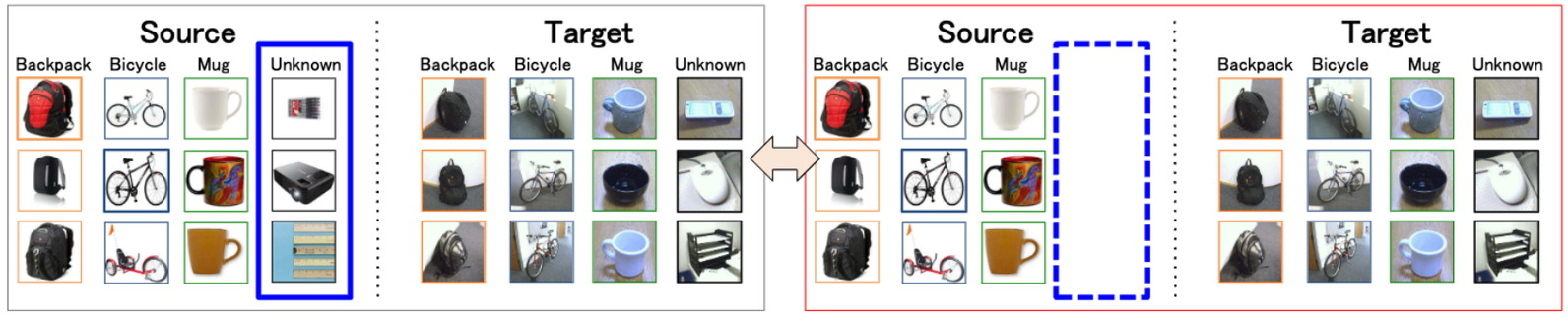
[1] Saito, Kuniaki, et al. "Open set domain adaptation by backpropagation." *Proceedings of the European conference on computer vision (ECCV)*. 2018.

Open-Set DA

Setting:

- The target label space is a strict superset of the source.
- The target domain cares about the source classes, plus additional ones.

$$\mathcal{Y}_s \subset \mathcal{Y}_t$$



The target domain contains **unknown classes** that were never seen during source training.
The model must correctly **classify known** classes AND **detect and reject unknown** ones.

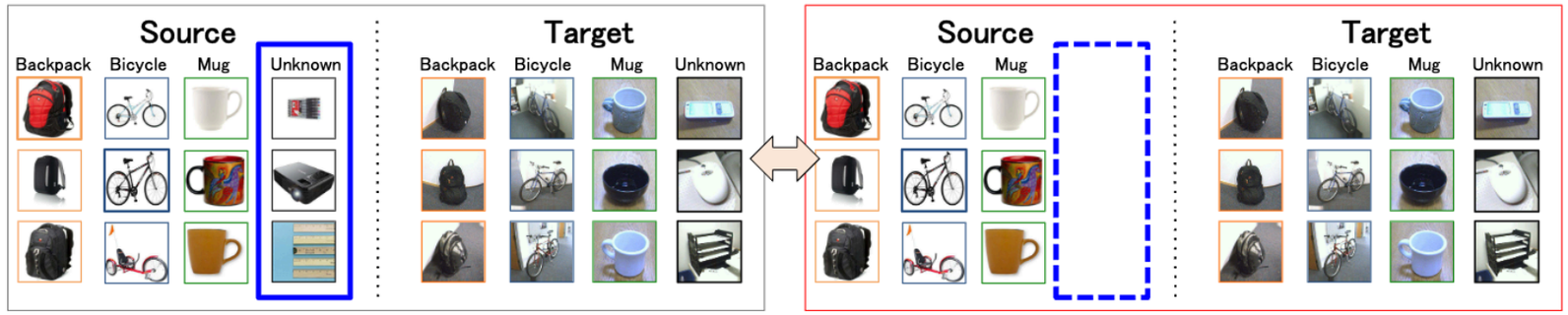
[1] Saito, Kuniaki, et al. "Open set domain adaptation by backpropagation." *Proceedings of the European conference on computer vision (ECCV)*. 2018.

Open-Set DA

Setting:

- The target label space is a strict superset of the source.
- The target domain cares about the source classes, plus additional ones.

$$\mathcal{Y}_s \subset \mathcal{Y}_t$$



The target domain contains **unknown classes** that were never seen during source training.
The model must correctly **classify known** classes AND **detect and reject unknown** ones.

Why standard UDA fails here:
Standard methods have **no mechanism for "I don't know”**. Every target sample gets forced into one of the known source classes. Unknown target samples get misclassified with high confidence

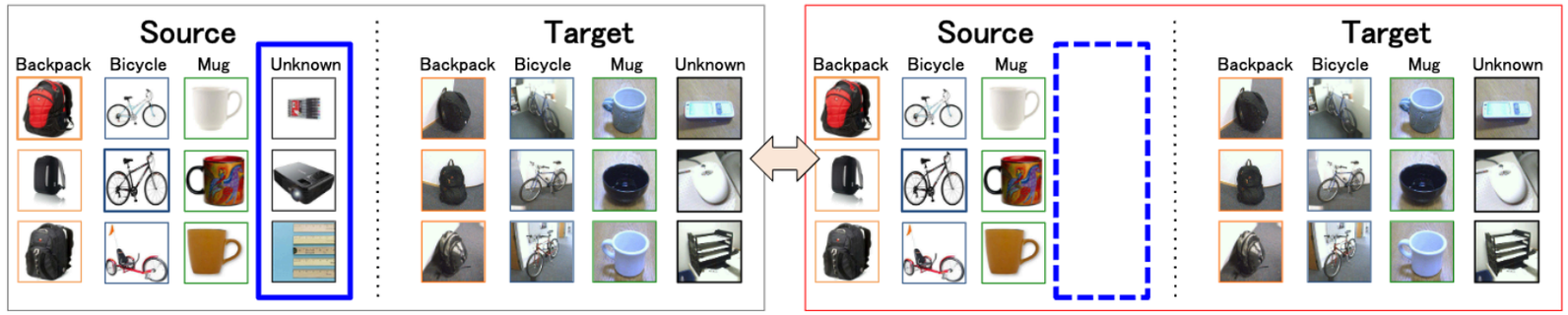
[1] Saito, Kuniaki, et al. "Open set domain adaptation by backpropagation." *Proceedings of the European conference on computer vision (ECCV)*. 2018.

Open-Set DA

Setting:

- The target label space is a strict superset of the source.
- The target domain cares about the source classes, plus additional ones.

$$\mathcal{Y}_s \subset \mathcal{Y}_t$$



The target domain contains **unknown classes** that were never seen during source training.
The model must correctly **classify known** classes AND **detect and reject unknown** ones.

Why standard UDA fails here:
Standard methods have **no mechanism for "I don't know”**. Every target sample gets forced into one of the known source classes. Unknown target samples get misclassified with high confidence

The key idea [1]

Unknown class as a reject option: The model needs a threshold or explicit mechanism to say “*this target sample does not belong to any known class*”.

[1] Saito, Kuniaki, et al. "Open set domain adaptation by backpropagation." *Proceedings of the European conference on computer vision (ECCV)*. 2018.

Open-Set DA

OSBP [1]: Open set domain adaptation by backpropagation

- Extends the adversarial DA framework with an explicit **unknown class threshold**

- The feature extractor is trained to **push** unknown target samples **away from all known class boundaries** rather than toward any of them

- A target sample is classified as "unknown" if its maximum class probability falls below a learned threshold

- The adversarial objective is modified: rather than making source and target indistinguishable, it only aligns the *known* portion of the target distribution

[1] Saito, Kuniaki, et al. "Open set domain adaptation by backpropagation." *Proceedings of the European conference on computer vision (ECCV)*. 2018.

Open-Set DA

OSBP [1]: Open set domain adaptation by backpropagation

- Extends the adversarial DA framework with an explicit **unknown class threshold**
- The feature extractor is trained to **push** unknown target samples **away from all known class boundaries** rather than toward any of them
- A target sample is classified as "unknown" if its maximum class probability falls below a learned threshold
- The adversarial objective is modified: rather than making source and target indistinguishable, it only aligns the *known* portion of the target distribution

Connection with Out-of-Distribution detection:

Open-Set DA and OOD detection are related

- **Open-Set DA** uses the unlabeled target data during training to **learn the rejection boundary**
- **OOD** detection typically operates **purely at test time** on a frozen model.

[1] Saito, Kuniaki, et al. "Open set domain adaptation by backpropagation." *Proceedings of the European conference on computer vision (ECCV)*. 2018.

Open-Set DA

OSBP [1]: Open set domain adaptation by backpropagation

- Extends the adversarial DA framework with an explicit **unknown class threshold**

- The feature extractor is trained to **push** unknown target samples **away from all known class boundaries** rather than toward any of them

- A target sample is classified as "unknown" if its maximum class probability falls below a learned threshold

- The adversarial objective is modified: rather than making source and target indistinguishable, it only aligns the *known* portion of the target distribution

Connection with Out-of-Distribution detection:

Open-Set DA and OOD detection are related

- **Open-Set DA** uses the unlabeled target data during training to **learn the rejection boundary**
- **OOD** detection typically operates **purely at test time** on a frozen model.

In Open-Set DA, the ratio is **selective rejection**:
Align known classes while pushing unknown target samples away from all decision boundaries.

[1] Saito, Kuniaki, et al. "Open set domain adaptation by backpropagation." *Proceedings of the European conference on computer vision (ECCV)*. 2018.

Universal DA

Setting:

- No assumption is made about the relationship between source and target label space
- **Both domains have private classes** (unknown to the other), plus a shared label space

(The most general case, it subsumes both Partial and Open-Set DA)

$$\mathcal{Y}_s \cap \mathcal{Y}_t \neq \emptyset$$

[1] You, Kaichao, et al. "Universal domain adaptation." Proceedings of the IEEE/CVF CVPR. 2019.
[2] Saito, Kuniaki, et al. "Universal domain adaptation through self supervision." *Advances in neural information processing systems* 33 (2020): 16282-16292.

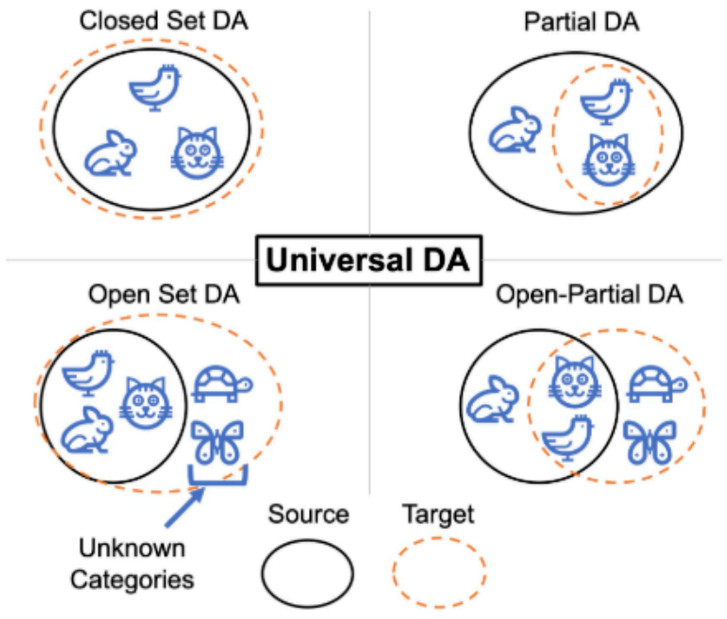
Universal DA

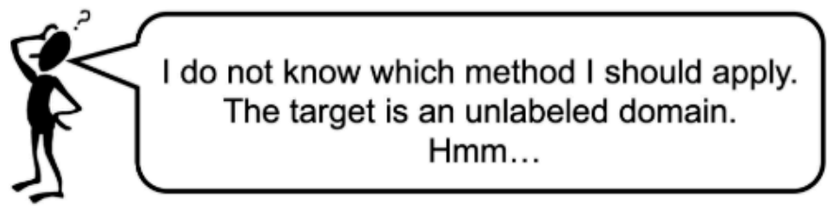
Setting:

- No assumption is made about the relationship between source and target label space
- **Both domains have private classes** (unknown to the other), plus a shared label space

(The most general case, it subsumes both Partial and Open-Set DA)

$$\mathcal{Y}_s \cap \mathcal{Y}_t \neq \emptyset$$





[1] You, Kaichao, et al. "Universal domain adaptation." Proceedings of the IEEE/CVF CVPR. 2019.
[2] Saito, Kuniaki, et al. "Universal domain adaptation through self supervision." *Advances in neural information processing systems* 33 (2020): 16282-16292.

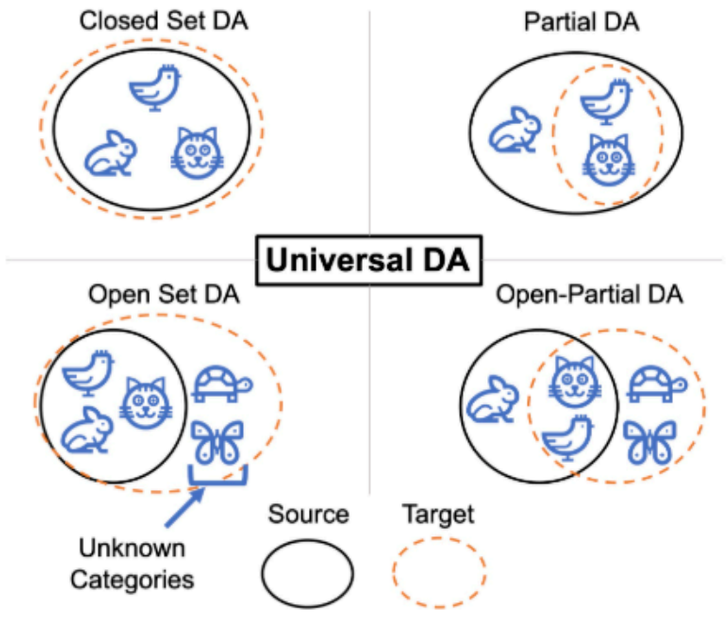
Universal DA

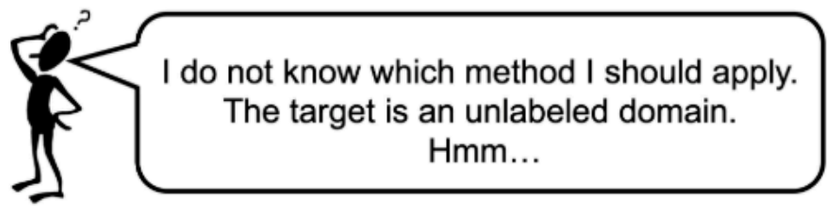
Setting:

- No assumption is made about the relationship between source and target label space
- **Both domains have private classes** (unknown to the other), plus a shared label space

(The most general case, it subsumes both Partial and Open-Set DA)

$$\mathcal{Y}_s \cap \mathcal{Y}_t \neq \emptyset$$





The key challenge — double uncertainty [1,2]:

- Which target samples belong to the shared space? (as in Open-Set DA)
- Which source classes should be down-weighted? (as in Partial DA)
- **Both questions** must be answered **simultaneously**, w/o any label information on the target side

[1] You, Kaichao, et al. "Universal domain adaptation." Proceedings of the IEEE/CVF CVPR. 2019.
[2] Saito, Kuniaki, et al. "Universal domain adaptation through self supervision." *Advances in neural information processing systems* 33 (2020): 16282-16292.

Universal DA

UAN [1]: Universal domain adaptation

[1] You, Kaichao, et al. "Universal domain adaptation." Proceedings of the IEEE/CVF CVPR. 2019.

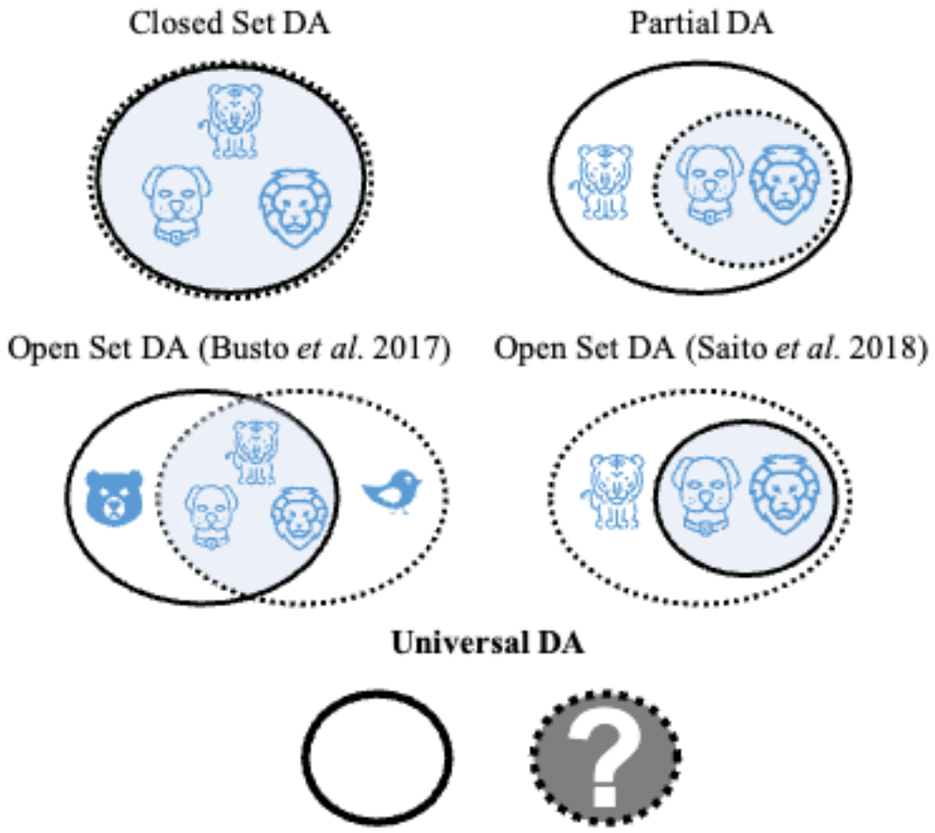
Universal DA

UAN [1]: Universal domain adaptation

Assigns each target sample two scores computed jointly:

- A **domain similarity score** d(x): how similar is this sample to source samples in feature space?
  High similarity → likely shared class

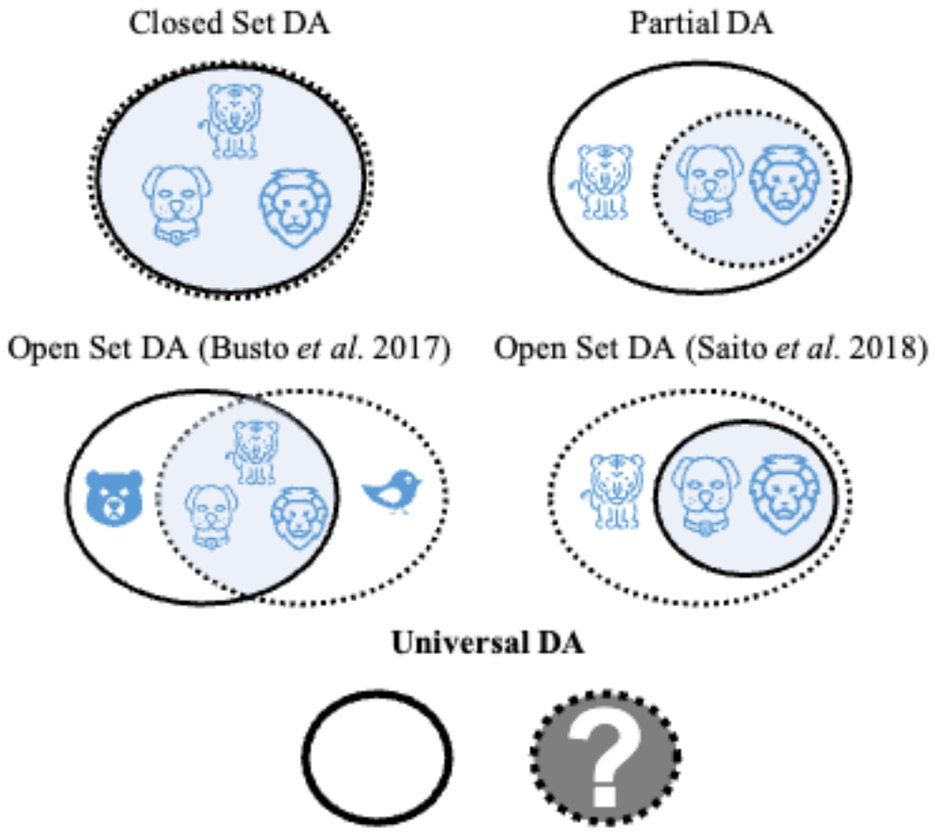
- A **prediction uncertainty score** u(x): how uncertain is the classifier about this sample?
  High uncertainty → likely unknown class



[1] You, Kaichao, et al. "Universal domain adaptation." Proceedings of the IEEE/CVF CVPR. 2019.

Universal DA

UAN [1]: Universal domain adaptation



Assigns each target sample two scores computed jointly:

- A **domain similarity score** d(x): how similar is this sample to source samples in feature space?
  High similarity → likely shared class

- A **prediction uncertainty score** u(x): how uncertain is the classifier about this sample?
  High uncertainty → likely unknown class

Their combination w(x) = d(x) * (1 - u(x)) gives a transferability weight per target sample:

- **High w**: confidently in the shared space → **used for alignment**

- **Low w**: likely unknown class → **treat as "unknown"**, exclude from alignment

No assumption about label space relationship is hardcoded — the method discovers it from data

[1] You, Kaichao, et al. "Universal domain adaptation." Proceedings of the IEEE/CVF CVPR. 2019.

Different label Space for DA

	Partial DA	Open-Set DA	Universal DA
Label assumption	$\mathcal{Y}_t \subset \mathcal{Y}_s$	$\mathcal{Y}_s \subset \mathcal{Y}_t$	No assumption
Key mechanism	Down-weight irrelevant source classes	Reject unknown target samples	Discover shared space dynamically
Representative method	**PADA, ETN**	**OSBP**	**UAN**
Subsumes the others?			

Multi-Source DA

Multi-Source DA

Multi-Source Domain Adaptation [1] **relaxes** the common assumption **access to only one source domain** by allowing multiple source domains,

$$\{\mathcal{D}_{s_1}, \mathcal{D}_{s_2}, \ldots, \mathcal{D}_{s_n}\}$$

each with its own distribution, all sharing the same label space with the target.

[1] Lu, Yuwu, et al. "Multiple adaptation network for multi-source and multi-target domain adaptation." *IEEE Transactions on Multimedia* 27 (2025): 5731-5745.
[2] Zhang, Huiying, et al. "Class-aware multi-source domain adaptation algorithm for medical image analysis using reweighted matrix matching strategy." *PLoS One* 20.7 (2025): e0323676.

[2]

Multi-Source DA

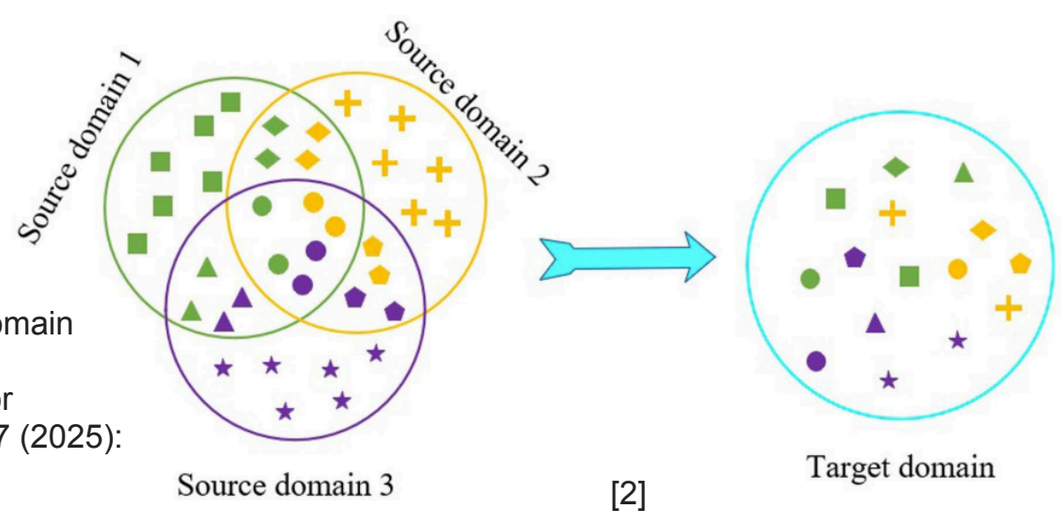
Multi-Source Domain Adaptation [1] **relaxes** the common assumption **access to only one source domain** by allowing multiple source domains,

$$\{\mathcal{D}_{s_1}, \mathcal{D}_{s_2}, \ldots, \mathcal{D}_{s_n}\}$$

each with its own distribution, all sharing the same label space with the target.

Why multiple sources arise naturally in practice:

- In remote sensing: the same land-cover classification task with **data collected from different sensors, seasons, or geographic regions**
- In medical imaging: the same pathology classification task with **data from different hospitals, scanners, or acquisition protocols**
- In autonomous driving: the same object detection task with **data collected in different cities, weather conditions, or times of day**



[2]

[1] Lu, Yuwu, et al. "Multiple adaptation network for multi-source and multi-target domain adaptation." *IEEE Transactions on Multimedia* 27 (2025): 5731-5745.
[2] Zhang, Huiying, et al. "Class-aware multi-source domain adaptation algorithm for medical image analysis using reweighted matrix matching strategy." *PLoS One* 20.7 (2025): e0323676.

Multi-Source DA

The naive baseline … and why it fails: The simplest approach is to **pool all source domains** into a single large source and apply standard UDA. This fails because:

- It ignores **inter-source diversity,** some sources are much closer to the target than others
- Distant sources can actively **hurt** adaptation by introducing conflicting feature distributions
- The **pooled distribution may not resemble any single source or the target,** it creates a new artificial distribution that is harder to align

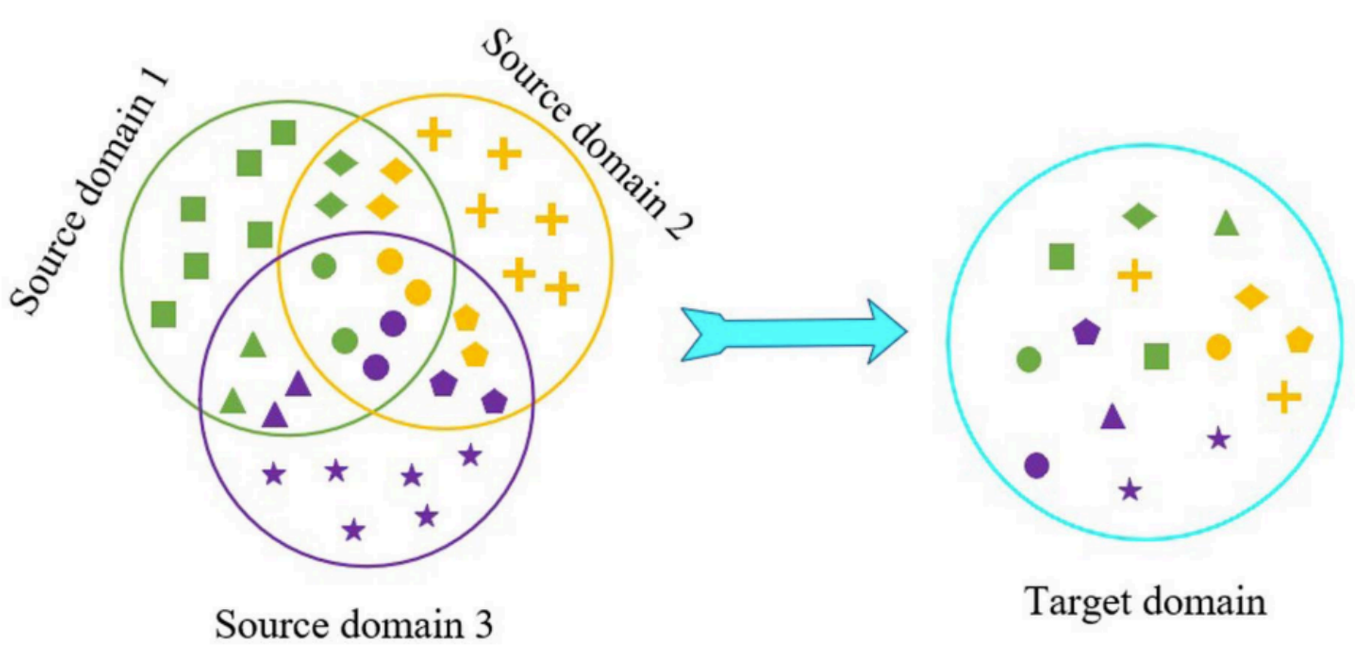
Multi-Source DA

The naive baseline … and why it fails: The simplest approach is to **pool all source domains** into a single large source and apply standard UDA. This fails because:

- It ignores **inter-source diversity,** some sources are much closer to the target than others
- Distant sources can actively **hurt** adaptation by introducing conflicting feature distributions
- The **pooled distribution may not resemble any single source or the target,** it creates a new artificial distribution that is harder to align

The key question

Should you **align each source to the target** independently, or learn a **joint multi-source alignment**, and how do you weight the contribution of each source?



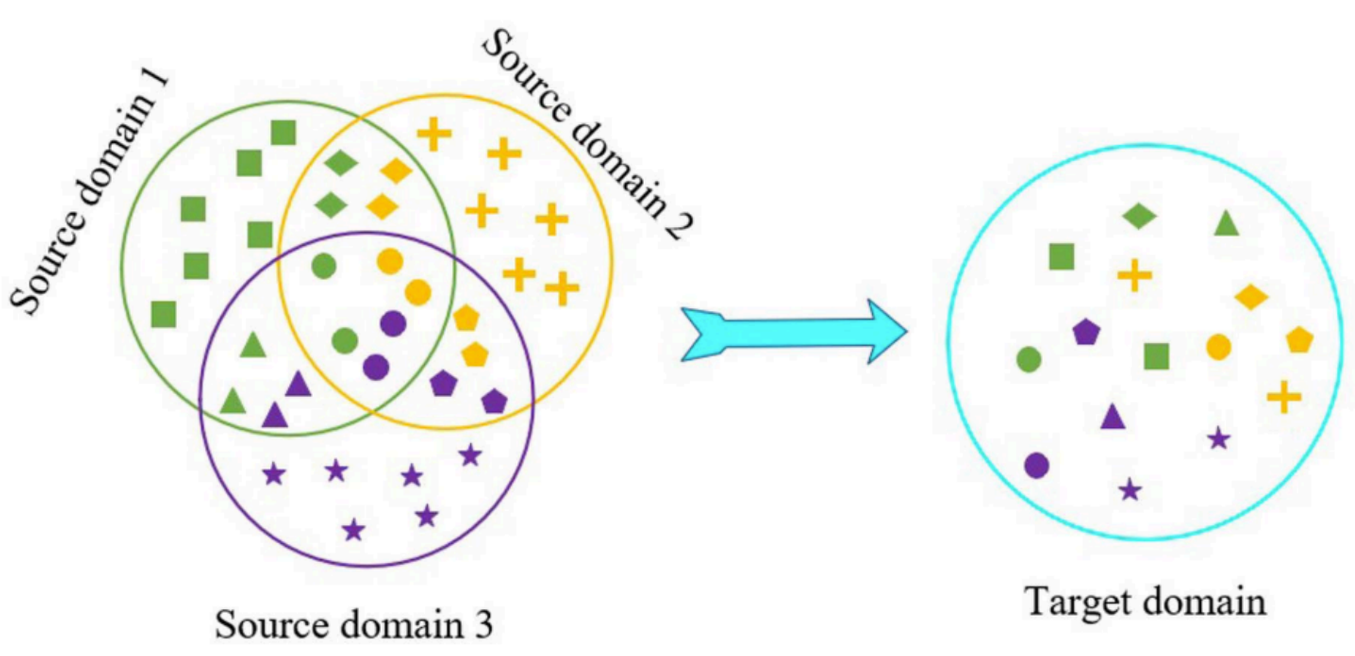
Multi-Source DA

The naive baseline … and why it fails: The simplest approach is to **pool all source domains** into a single large source and apply standard UDA. This fails because:

- It ignores **inter-source diversity,** some sources are much closer to the target than others
- Distant sources can actively **hurt** adaptation by introducing conflicting feature distributions
- The **pooled distribution may not resemble any single source or the target,** it creates a new artificial distribution that is harder to align

The key question

Should you **align each source to the target** independently, or learn a **joint multi-source alignment**, and how do you weight the contribution of each source?



Two design choices naturally emerge:

- **How do you weight** the contribution of each source domain? (uniform, learned, …)
- **At what level** do you perform alignment? (input space, feature space, prediction space)

Multi-Source DA

The first fundamental question every Multi-Source DA method must answer is:

Are all sources equally useful for the target?

[1] Ahmed, Sk Miraj, et al. "Unsupervised multi-source domain adaptation without access to source data." *Proceedings of the IEEE/CVF CVPR*. 2021.
[2] Zhao, Han, et al. "Adversarial multiple source domain adaptation." *Advances in neural information processing systems* 31 (2018).

Multi-Source DA

The first fundamental question every Multi-Source DA method must answer is:

Are all sources equally useful for the target?

Uniform weighting — the baseline

- **All sources contribute equally**: i) by simple pooling or ii) by applying the same alignment loss
- **Simple and parameter-free**, but ignores that some sources are much closer to the target
- Can **actively hurt if one source is very distant from the target**

[1] Ahmed, Sk Miraj, et al. "Unsupervised multi-source domain adaptation without access to source data." *Proceedings of the IEEE/CVF CVPR*. 2021.
[2] Zhao, Han, et al. "Adversarial multiple source domain adaptation." *Advances in neural information processing systems* 31 (2018).

Multi-Source DA

The first fundamental question every Multi-Source DA method must answer is:

Are all sources equally useful for the target?

Uniform weighting — the baseline

- **All sources contribute equally**: i) by simple pooling or ii) by applying the same alignment loss
- **Simple and parameter-free**, but ignores that some sources are much closer to the target
- Can **actively hurt if one source is very distant from the target**

Learned weighting — proximity-based

- Assign a **relevance weight** wk to each source k based on its estimated proximity to the target
- **DECISION** [1] estimates wk **from prediction agreement on unlabeled target samples**: sources whose predictions are consistent with the target get higher weight
- **MDAN** [2] learns per-source weights **via multiple dom. discriminators** equipped with GRL

[1] Ahmed, Sk Miraj, et al. "Unsupervised multi-source domain adaptation without access to source data." *Proceedings of the IEEE/CVF CVPR*. 2021.
[2] Zhao, Han, et al. "Adversarial multiple source domain adaptation." *Advances in neural information processing systems* 31 (2018).

Multi-Source DA

The second fundamental question is: **where in the model does the multi-source alignment happen?**

[1] Peng, Xingchao, et al. "Moment matching for multi-source domain adaptation." *Proceedings of the IEEE/CVF ICCV*. 2019.
[2] Zhao, Han, et al. "Adversarial multiple source domain adaptation." *Advances in neural information processing systems* 31 (2018).

Multi-Source DA

The second fundamental question is: **where in the model does the multi-source alignment happen?**

Input / pixel level

- **Translate source domain images** into the target style before extracting features
- Each source gets its own style transfer module (e.g. CycleGAN per source pair)
- **Alignment happens before the feature extractor** sees the data
- **Expensive when the number of sources is large**: requires K separate translation models

[1] Peng, Xingchao, et al. "Moment matching for multi-source domain adaptation." *Proceedings of the IEEE/CVF ICCV*. 2019.
[2] Zhao, Han, et al. "Adversarial multiple source domain adaptation." *Advances in neural information processing systems* 31 (2018).

Multi-Source DA

The second fundamental question is: **where in the model does the multi-source alignment happen?**

Input / pixel level

- **Translate source domain images** into the target style before extracting features
- Each source gets its own style transfer module (e.g. CycleGAN per source pair)
- **Alignment happens before the feature extractor** sees the data
- **Expensive when the number of sources is large**: requires K separate translation models

Feature level

- **Learn a shared latent space** where all source distributions and the target distribution overlap
- Most common approach: alignment via adversarial training, moment matching, or optimal transport
- **M³SDA** [1] aligns higher-order moments between **every pair** of domains simultaneously, both source-source and source-target, forcing a truly shared feature space
- **MDAN** [2] uses one adversarial discriminator per source to push the feature extractor toward a space that is simultaneously invariant to all source-target gaps
- **The risk: over-alignment** can wash out source-specific discriminative signal that would have been useful for the target

[1] Peng, Xingchao, et al. "Moment matching for multi-source domain adaptation." *Proceedings of the IEEE/CVF ICCV*. 2019.
[2] Zhao, Han, et al. "Adversarial multiple source domain adaptation." *Advances in neural information processing systems* 31 (2018).

Multi-Source DA

The second fundamental question is: **where in the model does the multi-source alignment happen?**

Input / pixel level

- **Translate source domain images** into the target style before extracting features
- Each source gets its own style transfer module (e.g. CycleGAN per source pair)
- **Alignment happens before the feature extractor** sees the data
- **Expensive when the number of sources is large**: requires K separate translation models

Feature level

- **Learn a shared latent space** where all source distributions and the target distribution overlap
- Most common approach: alignment via adversarial training, moment matching, or optimal transport
- **M³SDA** [1] aligns higher-order moments between **every pair** of domains simultaneously, both source-source and source-target, forcing a truly shared feature space
- **MDAN** [2] uses one adversarial discriminator per source to push the feature extractor toward a space that is simultaneously invariant to all source-target gaps
- **The risk: over-alignment** can wash out source-specific discriminative signal that would have been useful for the target

Prediction level

- **Each source trains its own classifier independently**, and the final prediction is an **ensemble**
- **No explicit feature-level alignment,** the combination happens only at the output
- **Simple and modular** but **ignores** the potential benefit of a **shared feature representation**

[1] Peng, Xingchao, et al. "Moment matching for multi-source domain adaptation." *Proceedings of the IEEE/CVF ICCV*. 2019.
[2] Zhao, Han, et al. "Adversarial multiple source domain adaptation." *Advances in neural information processing systems* 31 (2018).

Multi-Source DA

The interaction between the two decisions:

The two design choices are not independent, they interact:

	Uniform weighting	**Learned weighting**
Feature level	M^3SDA	MDAN
Prediction level	Simple ensemble	DECISION-style weighted ensemble

Multi-Source DA

The interaction between the two decisions:

The two design choices are not independent, they interact:

	Uniform weighting	**Learned weighting**
Feature level	M^3SDA	MDAN
Prediction level	Simple ensemble	DECISION-style weighted ensemble

Take-home messages:

- There is **no universally best combination**: the right choice depends on how heterogeneous the sources are and how much data is available
- Feature-level alignment with **learned weighting is the most powerful** but most complex combination
- When sources are very heterogeneous, **aggressive feature-level alignment can be detrimental**, prediction-level ensembling is safer

Source-Free Domain Adaptation

Source-Free DA

Most of the DA methods assume source data is available during adaptation.
In many scenarios this is not the case:

- **Privacy regulations** (GDPR): medical imaging data, financial records, or personal data collected to train a source model legally cannot be shared with a third party for adaptation purposes

- **Proprietary data**: a company trains a model on internal data and deploys it to a client, sharing the training data is commercially or legally prohibited

- **Data volume**: source datasets can be enormous (millions of images), transferring them to the adaptation site is impractical

- **Continual deployment**: a model is deployed in the field and needs to adapt to new conditions without phoning home to retrieve training data

[1] Li, Jingjing, et al. "A comprehensive survey on source-free domain adaptation." *IEEE TPAMI* 46.8 (2024): 5743-5762.

Source-Free DA

Most of the DA methods assume source data is available during adaptation.
In many scenarios this is not the case:

- **Privacy regulations** (GDPR): medical imaging data, financial records, or personal data collected to train a source model legally cannot be shared with a third party for adaptation purposes

- **Proprietary data**: a company trains a model on internal data and deploys it to a client, sharing the training data is commercially or legally prohibited

- **Data volume**: source datasets can be enormous (millions of images), transferring them to the adaptation site is impractical

- **Continual deployment**: a model is deployed in the field and needs to adapt to new conditions without phoning home to retrieve training data

Necessity to cope with adaptation scenarios in which **source data are no more accessible** or available:

Source-Free Domain Adaptation [1]

[1] Li, Jingjing, et al. "A comprehensive survey on source-free domain adaptation." *IEEE TPAMI* 46.8 (2024): 5743-5762.

Source-Free DA

The formal setting:

- **At training time**: a source model F is trained on labeled source data in the usual way
- **At adaptation time**: only the pretrained source model F is available, the source data is gone
- The target domain provides unlabeled data for adaptation
- **The goal**: adapt F to perform well on the target data without ever accessing the source data again

Source-Free DA

The formal setting:

- **At training time**: a source model F is trained on labeled source data in the usual way
- **At adaptation time**: only the pretrained source model F is available, the source data is gone
- The target domain provides unlabeled data for adaptation
- **The goal**: adapt F to perform well on the target data without ever accessing the source data again

The fundamental challenge without access to source data:

- Cannot compute any source-target divergence
- Cannot re-run any alignment loss
- Cannot sample source batches for contrastive or adversarial training.

The only information about the source domain that survives is **encoded in the weights of F.**

Source-Free DA

The formal setting:

- **At training time**: a source model F is trained on labeled source data in the usual way
- **At adaptation time**: only the pretrained source model F is available, the source data is gone
- The target domain provides unlabeled data for adaptation
- **The goal**: adapt F to perform well on the target data without ever accessing the source data again

The fundamental challenge without access to source data:

- Cannot compute any source-target divergence
- Cannot re-run any alignment loss
- Cannot sample source batches for contrastive or adversarial training.

The only information about the source domain that survives is **encoded in the weights of F.**

Two main families of approaches:

- Self-training on the target
- Source data generation / reconstruction

Source-Free DA

SHOT [1]: Source hypothesis transfer for unsupervised domain adaptation

[1] Liang, Jian, Dapeng Hu, and Jiashi Feng. "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation." *ICML*, 2020.
[2] Chopra, Shivang, et al. "Source-free domain adaptation with diffusion-guided source data generation." *arXiv preprint arXiv:2402.04929* (2024).

Source-Free DA

SHOT [1]: Source hypothesis transfer for unsupervised domain adaptation

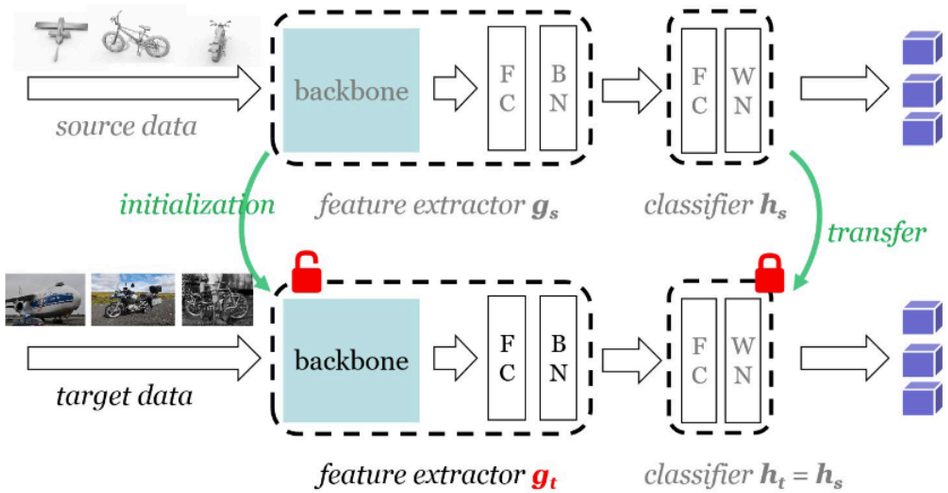
Use the source model to generate pseudo-labels on target data, then fine-tune the model on those pseudo-labels, without ever needing source data

Freezes the **classifier head** of the source model

Fine-tunes only the **feature extractor** on the target domain using two complementary objectives:

- **Maximize the mutual information** between input and predictions on target data

- **Assigns pseudo-labels** via nearest centroid in feature space and fine-tunes on them



Simple, effective, and widely adopted as a baseline

[1] Liang, Jian, Dapeng Hu, and Jiashi Feng. "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation." *ICML*, 2020.
[2] Chopra, Shivang, et al. "Source-free domain adaptation with diffusion-guided source data generation." *arXiv preprint arXiv:2402.04929* (2024).

Source-Free DA

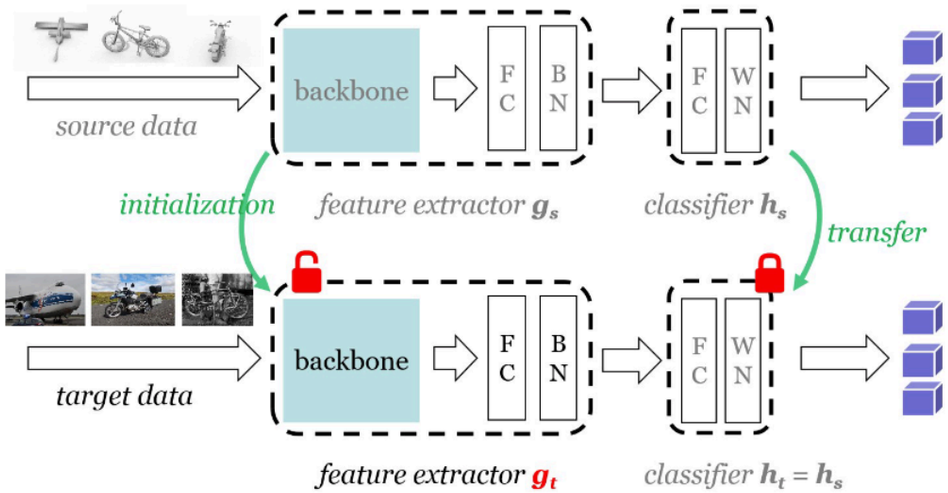
SHOT [1]: Source hypothesis transfer for unsupervised domain adaptation

Use the source model to generate pseudo-labels on target data, then fine-tune the model on those pseudo-labels, without ever needing source data

Freezes the **classifier head** of the source model

Fine-tunes only the **feature extractor** on the target domain using two complementary objectives:

- **Maximize the mutual information** between input and predictions on target data
- **Assigns pseudo-labels** via nearest centroid in feature space and fine-tunes on them



Simple, effective, and widely adopted as a baseline

Source data generation / reconstruction approaches also exist [2]

[1] Liang, Jian, Dapeng Hu, and Jiashi Feng. "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation." *ICML*, 2020.
[2] Chopra, Shivang, et al. "Source-free domain adaptation with diffusion-guided source data generation." *arXiv preprint arXiv:2402.04929* (2024).

Test-Time (Domain) Adaptation

Test-Time Adaptation

Classical UDA

Source data available

Source-Free DA

Only pretrained mode (on source data) available

Test-Time Adaptation

Adaptation only on the fly on the new data

Test-Time Adaptation

Classical UDA	Source-Free DA	Test-Time Adaptation
Source data available	Only pretrained mode (on source data) available	Adaptation only on the fly on the new data

What makes TTA fundamentally different from SFDA

In SFDA there is still a dedicated **adaptation phase,** the whole unlabeled target data are used for an adaptation procedure, and then deploy.

Test-Time Adaptation

Classical UDA	Source-Free DA	Test-Time Adaptation
Source data available	Only pretrained mode (on source data) available	Adaptation only on the fly on the new data

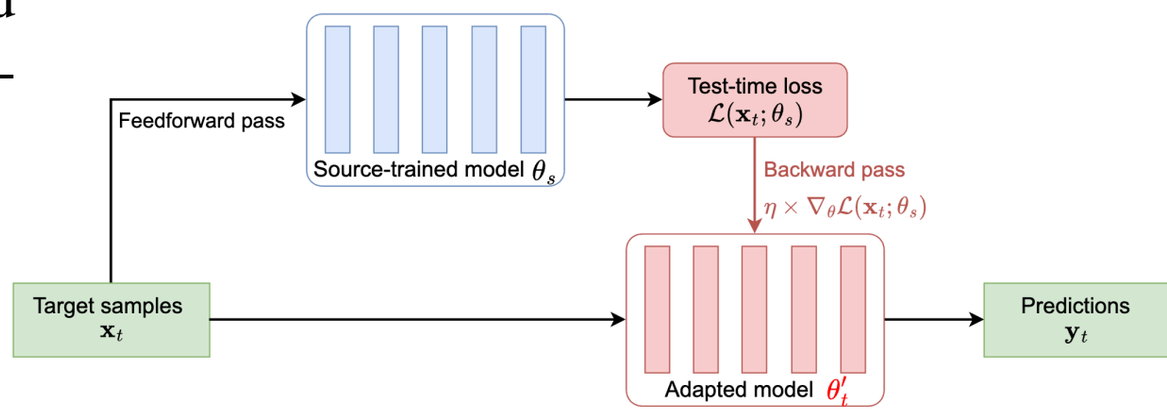
What makes TTA fundamentally different from SFDA

In SFDA there is still a dedicated **adaptation phase,** the whole unlabeled target data are used for an adaptation procedure, and then deploy.

TTA removes even this:

- **No separate adaptation phase,** the model adapts and produces predictions **simultaneously**, on test data mini-batches
- No labels are ever available, neither source nor target
- The model must be useful **immediately,** no way to collect enough data for a full adaptation loop

Xiao, Zehao, and Cees GM Snoek. "Beyond model adaptation at test time: A survey." *arXiv preprint arXiv:2411.03687* (2024).



Test-Time Adaptation

Why TTA is practically important [1]:

- **Deployment constraints**: in many real systems there is no opportunity or infrastructure to run a dedicated adaptation pipeline before serving predictions

- **Shifting conditions at inference time**: a deployed model encounters changing conditions, weather, sensor drift, domain change, and must handle them without interruption

- **Privacy at inference time**: sending test data back to a central server for adaptation may not be permitted, the model must adapt locally and on the fly

[1] Liang, Jian, Ran He, and Tieniu Tan. "A comprehensive survey on test-time adaptation under distribution shifts." *Int. Journal of Computer Vision* 133.1 (2025): 31-64.

Test-Time Adaptation

Why TTA is practically important [1]:

- **Deployment constraints**: in many real systems there is no opportunity or infrastructure to run a dedicated adaptation pipeline before serving predictions

- **Shifting conditions at inference time**: a deployed model encounters changing conditions, weather, sensor drift, domain change, and must handle them without interruption

- **Privacy at inference time**: sending test data back to a central server for adaptation may not be permitted, the model must adapt locally and on the fly

The key trade-off in TTA:

- Adapting too aggressively on a mini-batch risks **overfitting to noise** or a non-representative batch.
- Adapting too conservatively means the **distribution shift is never corrected**.

TTA methods live in this trade-off scenario.

[1] Liang, Jian, Ran He, and Tieniu Tan. "A comprehensive survey on test-time adaptation under distribution shifts." *Int. Journal of Computer Vision* 133.1 (2025): 31-64.

Test-Time Adaptation

Examples of approaches

[1] Schneider, Steffen, et al. "Improving robustness against common corruptions by covariate shift adaptation." *NeurIPS* 33 (2020): 11539-11551.
[2] Wang, Dequan, et al. "Tent: Fully Test-Time Adaptation by Entropy Minimization." *ICLR. 2021*
[3] Sun, Yu, et al. "Test-time training with self-supervision for generalization under distribution shifts." *ICML*, 2020.
[4] Niu, Shuaicheng, et al. "Towards Stable Test-time Adaptation in Dynamic Wild World." *ICLR. 2023*

Test-Time Adaptation

Examples of approaches

Normalization statistics adaptation: Adapt BN layers (as they store running statistics - mean, standard deviation, ..) computed on source data during training.

Examples:

- **Test-Time Normalization** (TTN) [1]: replaces the stored source BN statistics with statistics estimated from the current test mini-batch. No gradient
- **TENT** [2]: Keeps the BN statistics adaptation and additionally optimizes some BN parameters via gradient descent on an entropy minimization objective on each test mini-batch

[1] Schneider, Steffen, et al. "Improving robustness against common corruptions by covariate shift adaptation." *NeurIPS* 33 (2020): 11539-11551.
[2] Wang, Dequan, et al. "Tent: Fully Test-Time Adaptation by Entropy Minimization." *ICLR. 2021*
[3] Sun, Yu, et al. "Test-time training with self-supervision for generalization under distribution shifts." *ICML*, 2020.
[4] Niu, Shuaicheng, et al. "Towards Stable Test-time Adaptation in Dynamic Wild World." *ICLR. 2023*

Test-Time Adaptation

Examples of approaches

Normalization statistics adaptation: Adapt BN layers (as they store running statistics - mean, standard deviation, ..) computed on source data during training.

Examples:

- **Test-Time Normalization** (TTN) [1]: replaces the stored source BN statistics with statistics estimated from the current test mini-batch. No gradient
- **TENT** [2]: Keeps the BN statistics adaptation and additionally optimizes some BN parameters via gradient descent on an entropy minimization objective on each test mini-batch

Model parameters adaptation: Adapt a more large part of network parameters.

Examples:

- **Test-Time Training** (TTT) [3]: during training, a self-supervised auxiliary task is trained alongside the main task. At test time, the model updates the shared feature extractor by minimizing the self-supervised auxiliary loss on each incoming test mini-batch
- **SAR (Sharpness-Aware and Reliable entropy minimization)** [4]: uses sharpness-aware minimization to find flat entropy loss minima, making the adapted model more robust to

[1] Schneider, Steffen, et al. "Improving robustness against common corruptions by covariate shift adaptation." *NeurIPS* 33 (2020): 11539-11551.
[2] Wang, Dequan, et al. "Tent: Fully Test-Time Adaptation by Entropy Minimization." *ICLR. 2021*
[3] Sun, Yu, et al. "Test-time training with self-supervision for generalization under distribution shifts." *ICML*, 2020.
[4] Niu, Shuaicheng, et al. "Towards Stable Test-time Adaptation in Dynamic Wild World." *ICLR. 2023*

Test-Time Adaptation

	UDA	SFDA	TTA
Source data at adapt. time	Yes	No	No
Dedicated adaptation phase	Yes	Yes	No
Target distribution	Fixed	Fixed	Fixed
Adaptation granularity	Dataset	Dataset	Mini-batch
Risk of forgetting	Low	Low–Medium	Medium
Comp. cost at training time	High	Medium	Low
Comp. cost at test time	Low	Low	Medium
Representative methods	DANN, DAFormer	SHOT, DM-SFDA	TENT, TTT, SAR

Test-Time Adaptation

	UDA	SFDA	TTA
Source data at adapt. time	Yes	No	No
Dedicated adaptation phase	Yes	Yes	No
Target distribution	Fixed	Fixed	Fixed
Adaptation granularity	Dataset	Dataset	Mini-batch
Risk of forgetting	Low	Low–Medium	Medium
Comp. cost at training time	High	Medium	Low
Comp. cost at test time	Low	Low	Medium
Representative methods	DANN, DAFormer	SHOT, DM-SFDA	TENT, TTT, SAR

- TTA is the **most deployment-realistic** DA paradigm

- **TTA** has mini-batch sensitivity, error accumulation, and confirmation bias. **Genuine open challenges**

- **Normalization-based methods** (TENT) **are practical starting point**, simple/light, and effective for moderate distrib. shifts

- The field has **not yet converged on robust solutions for large distribution shifts** or very small mini-batch sizes (Open challenges)

Foundation Model and DA

Foundation Model & DA

How foundation models (FM) change the DA landscape

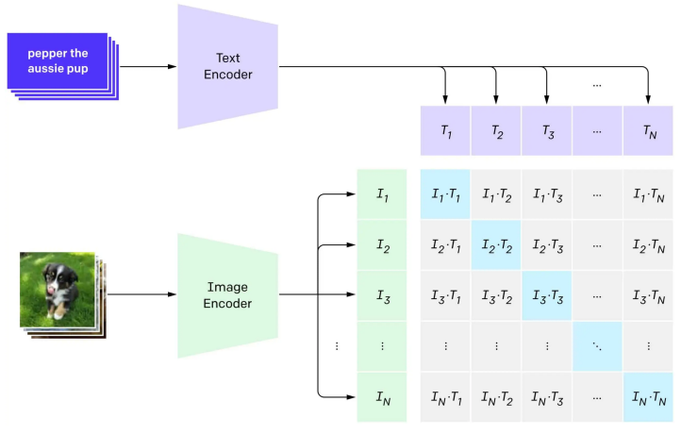
- What foundation models bring to DA: rich pretrained representations that **should be already partially domain-invariant**

- **Zero-shot or few-shot transfer** from FM often matches or exceeds carefully tuned UDA methods on classical benchmarks

- **Classical DA benchmarks** (Office-31, Office-Home) are arguably **"solved"** by foundation models, the field needs harder problems

- Examples of FM: CLIP, DINO, DINOv2, DINOv3, SAM for vision or vision-language







Foundation Model & DA

Adapting foundation models: the new DA problem

- Fine-tuning a FM on a source domain and then adapting to a target is now the dominant paradigm
- Lightweight adaptation strategies: full fine-tuning (expensive), **adapter tuning**, **prompt tuning**
- Since the pretrained backbone already carries rich semantic knowledge, **SFDA with FM can have some differences from SFDA with task-specific models**
- **FM as a universal bridge between modalities** (CLIP as a semantic anchor between image and text domains) - Heterogeneous DA (but mainly on the major modalities where FM are available)

Foundation Model & DA

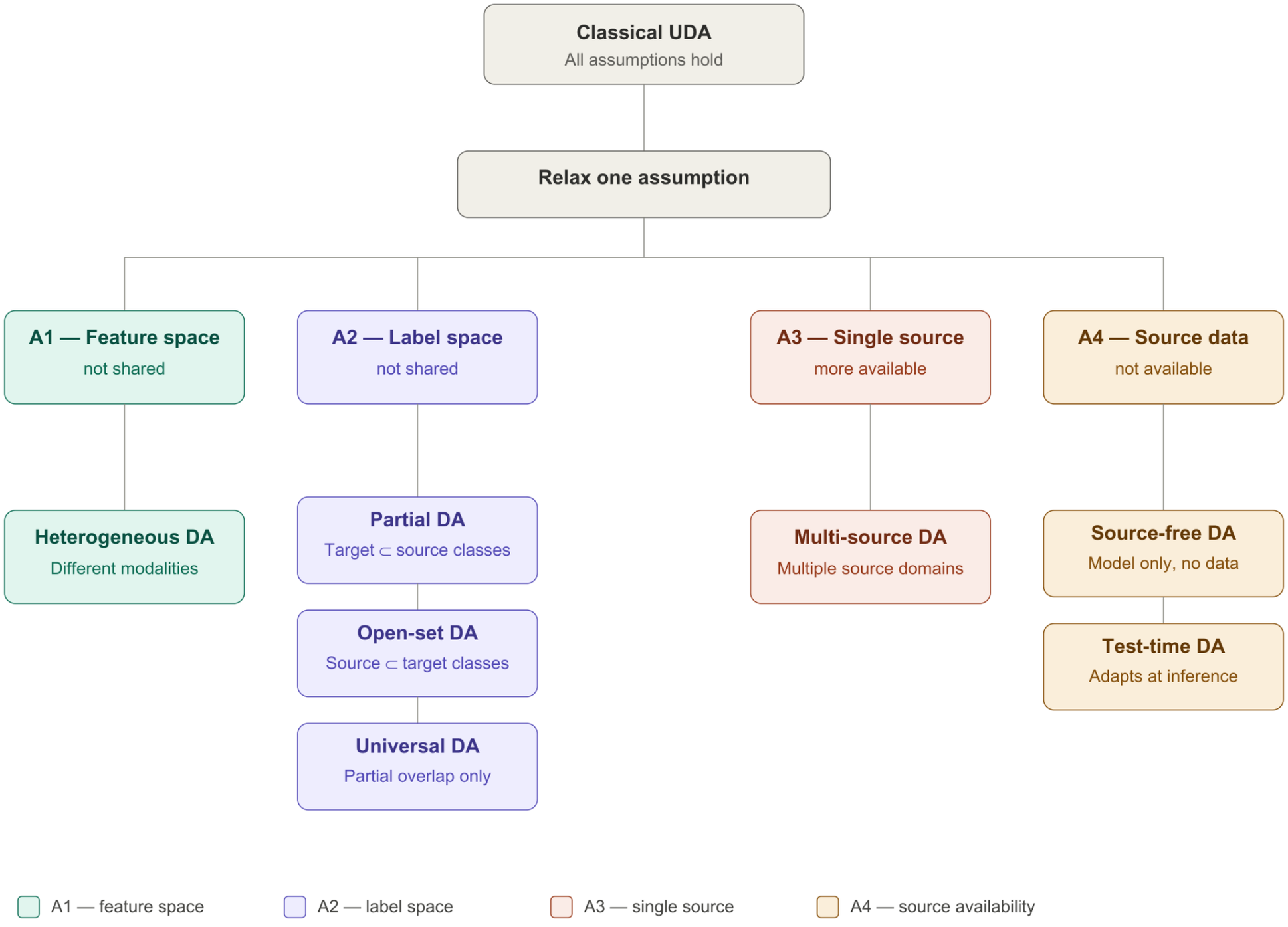
Adapting foundation models: the new DA problem

- Fine-tuning a FM on a source domain and then adapting to a target is now the dominant paradigm
- Lightweight adaptation strategies: full fine-tuning (expensive), **adapter tuning**, **prompt tuning**
- Since the pretrained backbone already carries rich semantic knowledge, **SFDA with FM can have some differences from SFDA with task-specific models**
- **FM as a universal bridge between modalities** (CLIP as a semantic anchor between image and text domains) - Heterogeneous DA (but mainly on the major modalities where FM are available)

Recent directions and Open Challenges

- **CLIP-based DA**: using CLIP's shared image-text embedding space as a domain-invariant anchor
- **Prompt tuning for DA**: Learning domain-adaptive prompts rather than adapting model weights
- **Foundation models for TTA**: using the frozen backbone as base during test-time adaptation
- Open problems: evaluation on **realistic benchmarks**, adaptation for **non-natural image modalities**, **computational cost** of foundation model adaptation at scale

Conclusions



Conclusions

Open Problems & Research Directions

Conclusions

Open Problems & Research Directions

Evaluation and benchmarks

- **Classical benchmarks** (Office-31, Office-Home, GTA5→Cityscapes) are increasingly **saturated**, FM achieve near-ceiling performance
- The field **needs harder, more realistic benchmarks** that reflect genuine deployment conditions: continual shift, label space uncertainty, heterogeneous modalities
- Evaluation under **multiple simultaneous assumption relaxations** is not largely unexplored, most papers relax one assumption at a time

Conclusions

Open Problems & Research Directions

Evaluation and benchmarks

- **Classical benchmarks** (Office-31, Office-Home, GTA5→Cityscapes) are increasingly **saturated**, FM achieve near-ceiling performance
- The field **needs harder, more realistic benchmarks** that reflect genuine deployment conditions: continual shift, label space uncertainty, heterogeneous modalities
- Evaluation under **multiple simultaneous assumption relaxations** is not largely unexplored, most papers relax one assumption at a time

Modality-specific challenges

- The vast majority of DA literature focuses on RGB images, methods for time series, hyperspectral data, SAR, point clouds, and multimodal data remain far less mature

- FM pretrained on natural images do not transfer cleanly to these modalities

Conclusions

Take-home messages

Conclusions

Take-home messages

Distribution shift is the norm, not the exception: any model deployed in the real world will encounter it

Conclusions

Take-home messages

Distribution shift is the norm, not the exception: any model deployed in the real world will encounter it

The family of methods matters more than the specific method: understanding when to use instance reweighting vs feature alignment vs self-training is more valuable than memorizing architectures

Conclusions

Take-home messages

Distribution shift is the norm, not the exception: any model deployed in the real world will encounter it

The family of methods matters more than the specific method: understanding when to use instance reweighting vs feature alignment vs self-training is more valuable than memorizing architectures

Every assumption relaxation opens a new research direction: Partial DA, Open-set DA, SFDA, TTA are all born from questioning one classical assumption

Conclusions

Take-home messages

Distribution shift is the norm, not the exception: any model deployed in the real world will encounter it

The family of methods matters more than the specific method: understanding when to use instance reweighting vs feature alignment vs self-training is more valuable than memorizing architectures

Every assumption relaxation opens a new research direction: Partial DA, Open-set DA, SFDA, TTA are all born from questioning one classical assumption

Foundation models change the playing field but do not end the game: they reduce classical gaps while introducing new ones

Conclusions

Take-home messages

Distribution shift is the norm, not the exception: any model deployed in the real world will encounter it

The family of methods matters more than the specific method: understanding when to use instance reweighting vs feature alignment vs self-training is more valuable than memorizing architectures

Every assumption relaxation opens a new research direction: Partial DA, Open-set DA, SFDA, TTA are all born from questioning one classical assumption

Foundation models change the playing field but do not end the game: they reduce classical gaps while introducing new ones

The best DA method is the one that matches your deployment constraints: source availability, label space knowledge, inference budget, and modality all determine the right choice

EVERGREEN Team

EVERGREEN Team: 5 permanents, 4 PhDs, 2 Post-doc, 2 Engineers, Visiting researchers (several per year)

Team Topics:

Machine Learning for EO data with applications in Agriculture & Environment



R. Interdonato



R. Gaetano



C. Fraga Dantas



D. Ienco





D. Marcos

EVERGREEN Team



EVERGREEN Team: 5 permanents, 4 PhDs, 2 Post-doc, 2 Engineers, Visiting researchers (several per year)

Team Topics:

Machine Learning for EO data with applications in Agriculture & Environment











R. Interdonato



R. Gaetano



C. Fraga Dantas



D. Ienco



D. Marcos

EVERGREEN Team



EVERGREEN Team: 5 permanents, 4 PhDs, 2 Post-doc, 2 Engineers, Visiting researchers (several per year)

Team Topics:

Machine Learning for EO data with applications in Agriculture & Environment











R. Interdonato



R. Gaetano



C. Fraga Dantas



D. Ienco



D. Marcos

EVERGREEN Team

- Advancing **multi-source/ multi-temporal** remote sensing analysis
- Enhancing **domain generalization** and **domain adaptation** for spatio-temporal data
- Exploring **multi-modal learning** approaches specifically tailored to spatio-temporal data



- Post-hoc and by-design **interpretability / explainability** approaches
- Knowledge-driven (**hybrid**) **machine learning** techniques for agro-env applications
- **Foundation models** geared toward **Earth observation** (i.e. remote sensing) **data**