

Shift Happens: Building Robust AI Models with Domain Adaptation

Aleksandra Ciprijanvoic

Fermilab



U.S. DEPARTMENT
of **ENERGY**

Fermi National Accelerator Laboratory is managed by
FermiForward for the U.S. Department of Energy Office of Science



Major challenges for AI applications in Science

As listed in The Dawes Review 10: The impact of deep learning for the analysis of galaxy surveys
[Huertas-Company & Lanusse 2022](#)

I want to train my AI model by I don't have enough trustworthy labels?



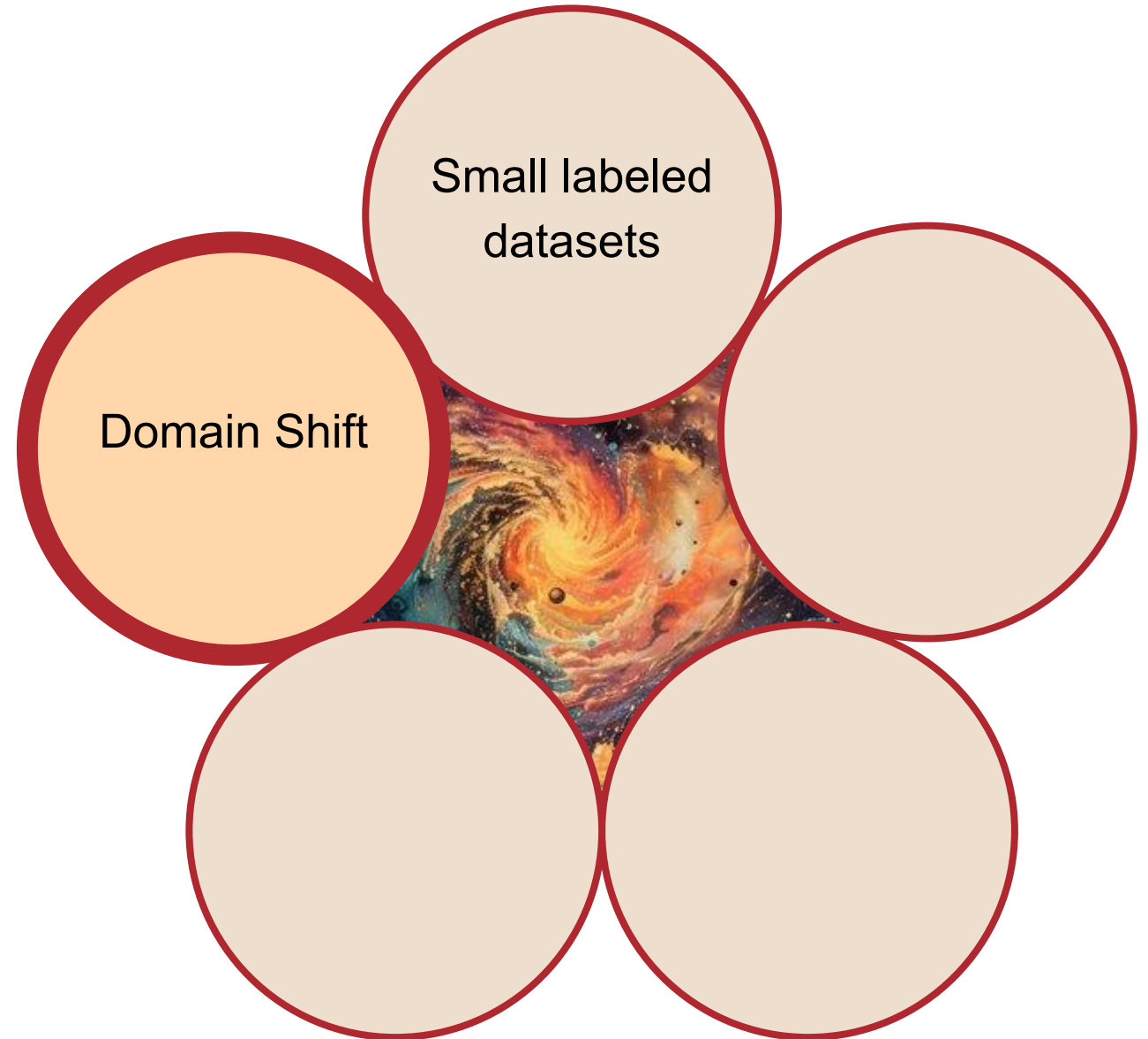


Major challenges for AI applications in Science

As listed in The Dawes Review 10: The impact of deep learning for the analysis of galaxy surveys
[Huertas-Company & Lanusse 2022](#)

I want to train my AI model by I don't have enough trustworthy labels?

Can I train on simulations and apply the model to my real data?





Major challenges for AI applications in Science

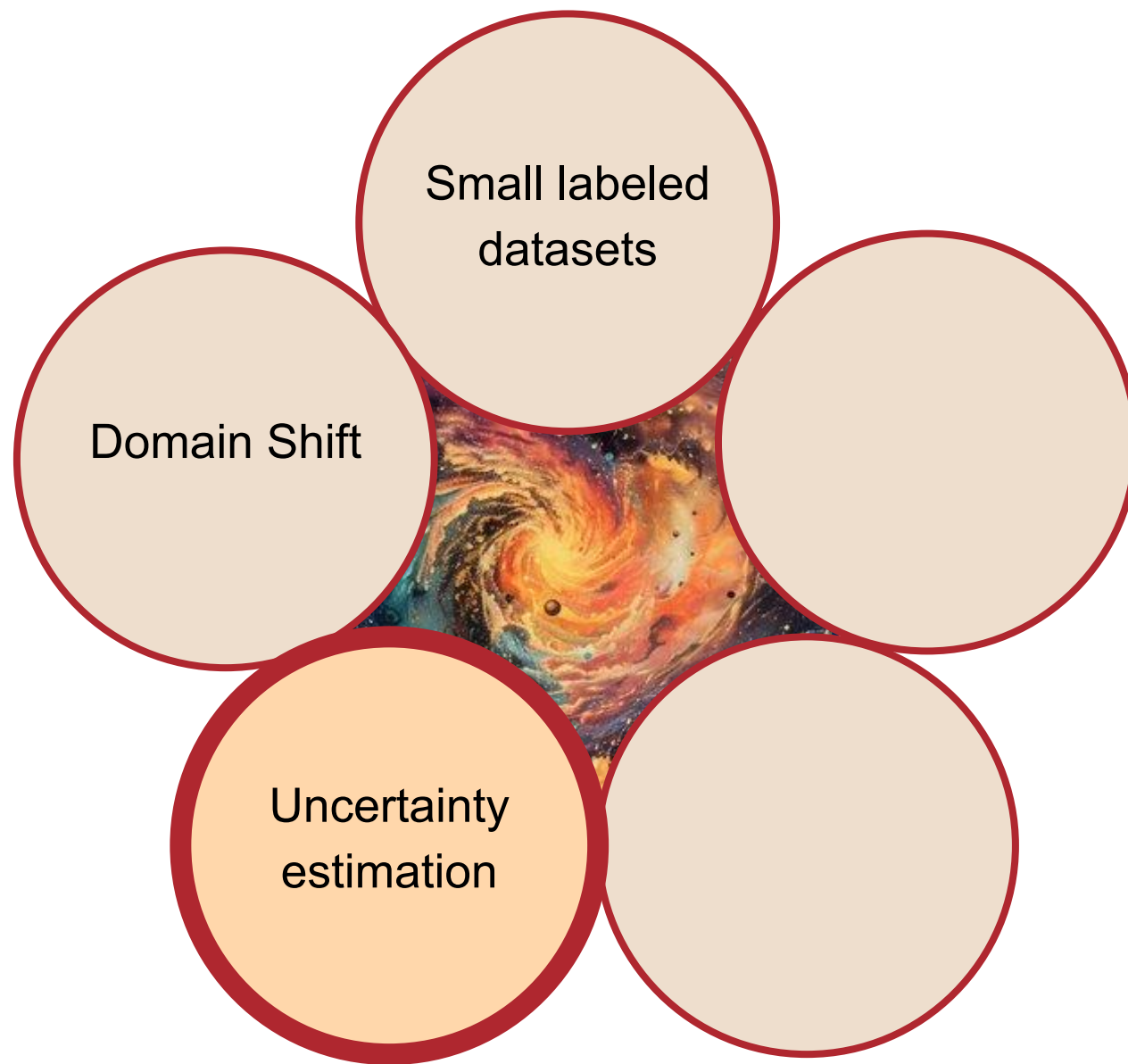
As listed in The Dawes Review 10: The impact of deep learning for the analysis of galaxy surveys

[Huertas-Company & Lanusse 2022](#)

I want to train my AI model by I don't have enough trustworthy labels?

Can I train on simulations and apply the model to my real data?

How well is my model going to perform?





Major challenges for AI applications in Science

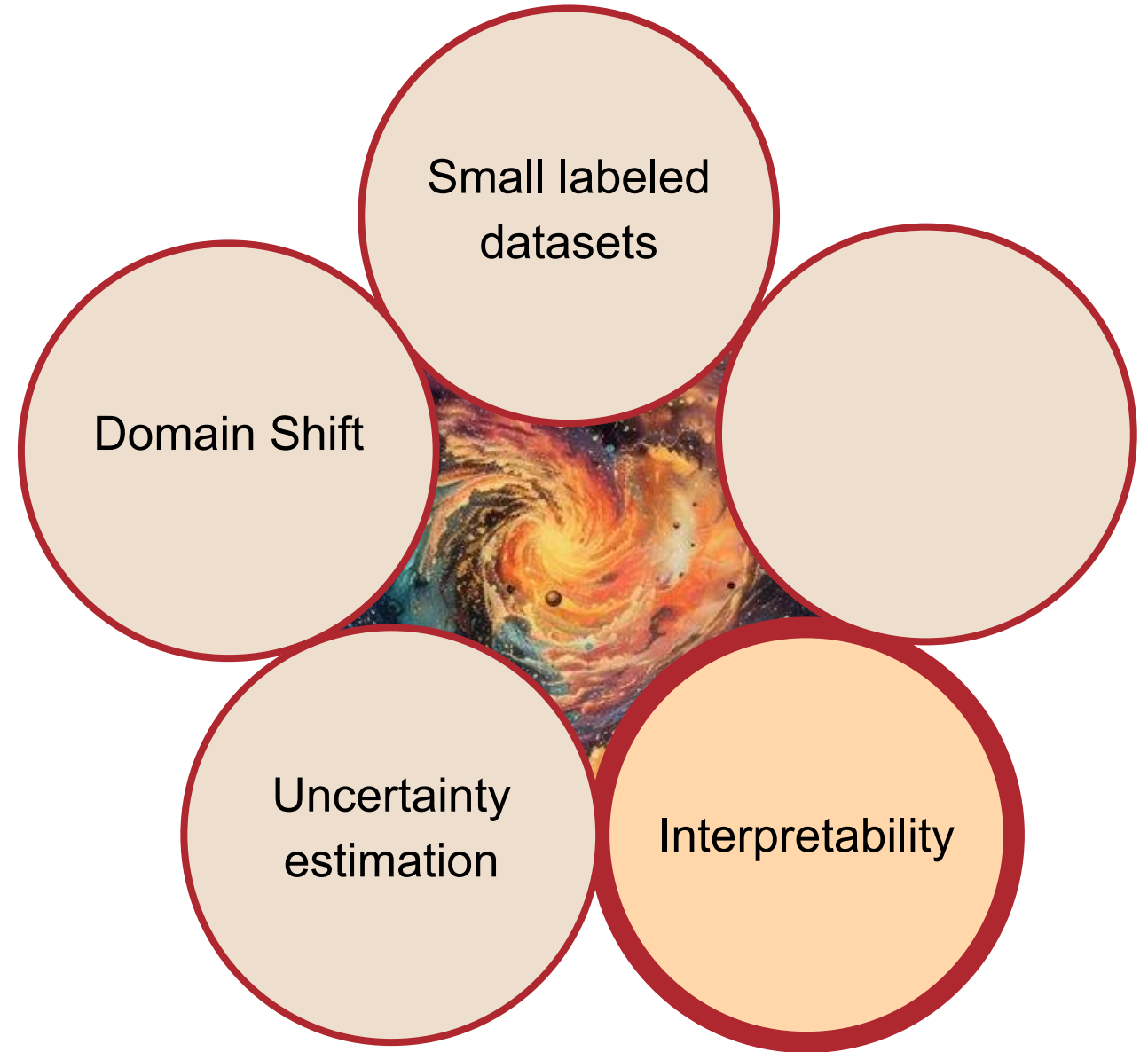
As listed in The Dawes Review 10: The impact of deep learning for the analysis of galaxy surveys
[Huertas-Company & Lanusse 2022](#)

I want to train my AI model by I don't have enough trustworthy labels?

Can I train on simulations and apply the model to my real data?

How well is my model going to perform?

Can I understand its performance?





Major challenges for AI applications in Science

As listed in The Dawes Review 10: The impact of deep learning for the analysis of galaxy surveys

[Huertas-Company & Lanusse 2022](#)

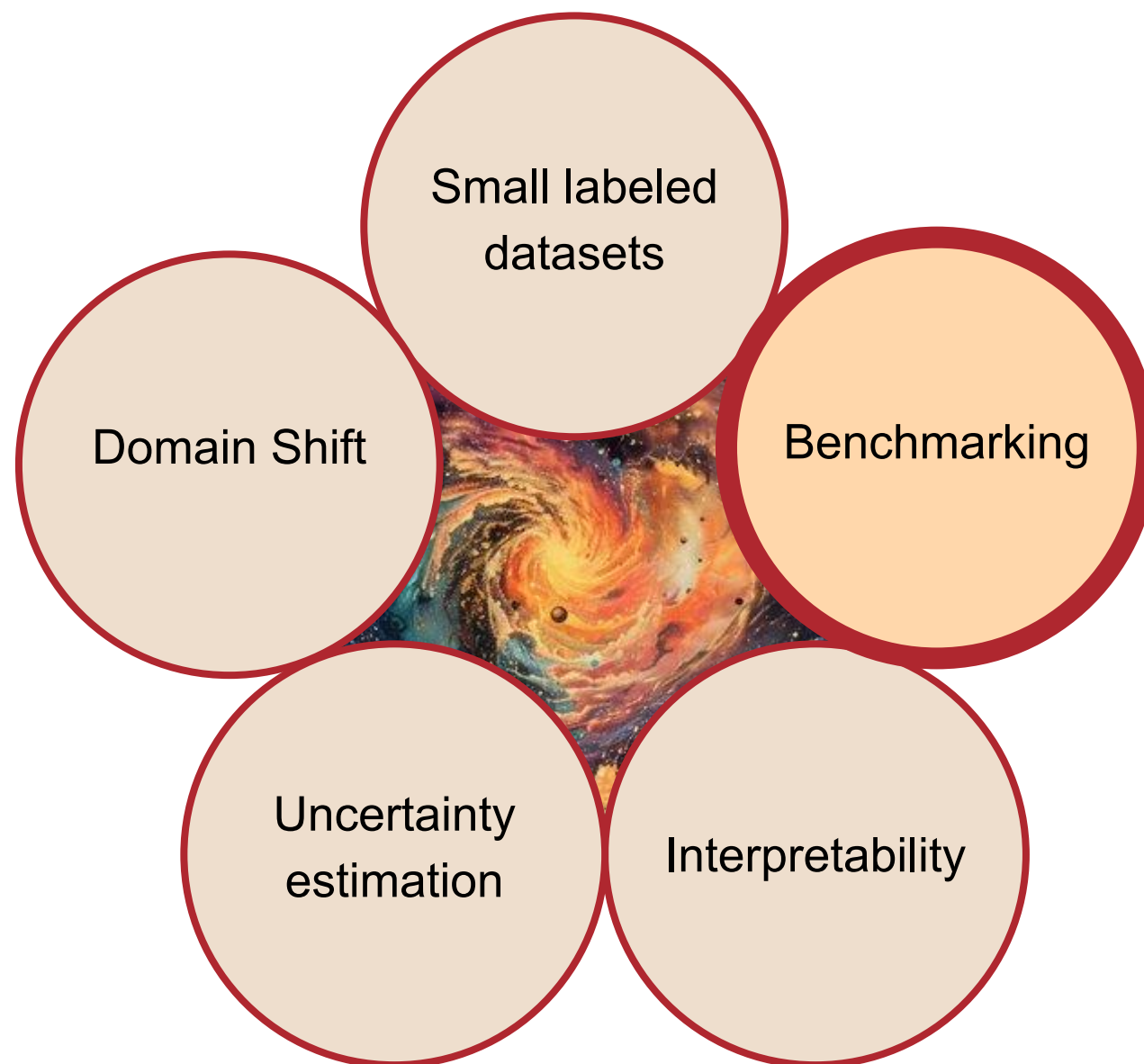
I want to train my AI model by I don't have enough trustworthy labels?

Can I train on simulations and apply the model to my real data?

How well is my model going to perform?

Can I understand its performance?

Can I compare my model with other researchers?





Major challenges for AI applications in Science

As listed in The Dawes Review 10: The impact of deep learning for the analysis of galaxy surveys
[Huertas-Company & Lanusse 2022](#)

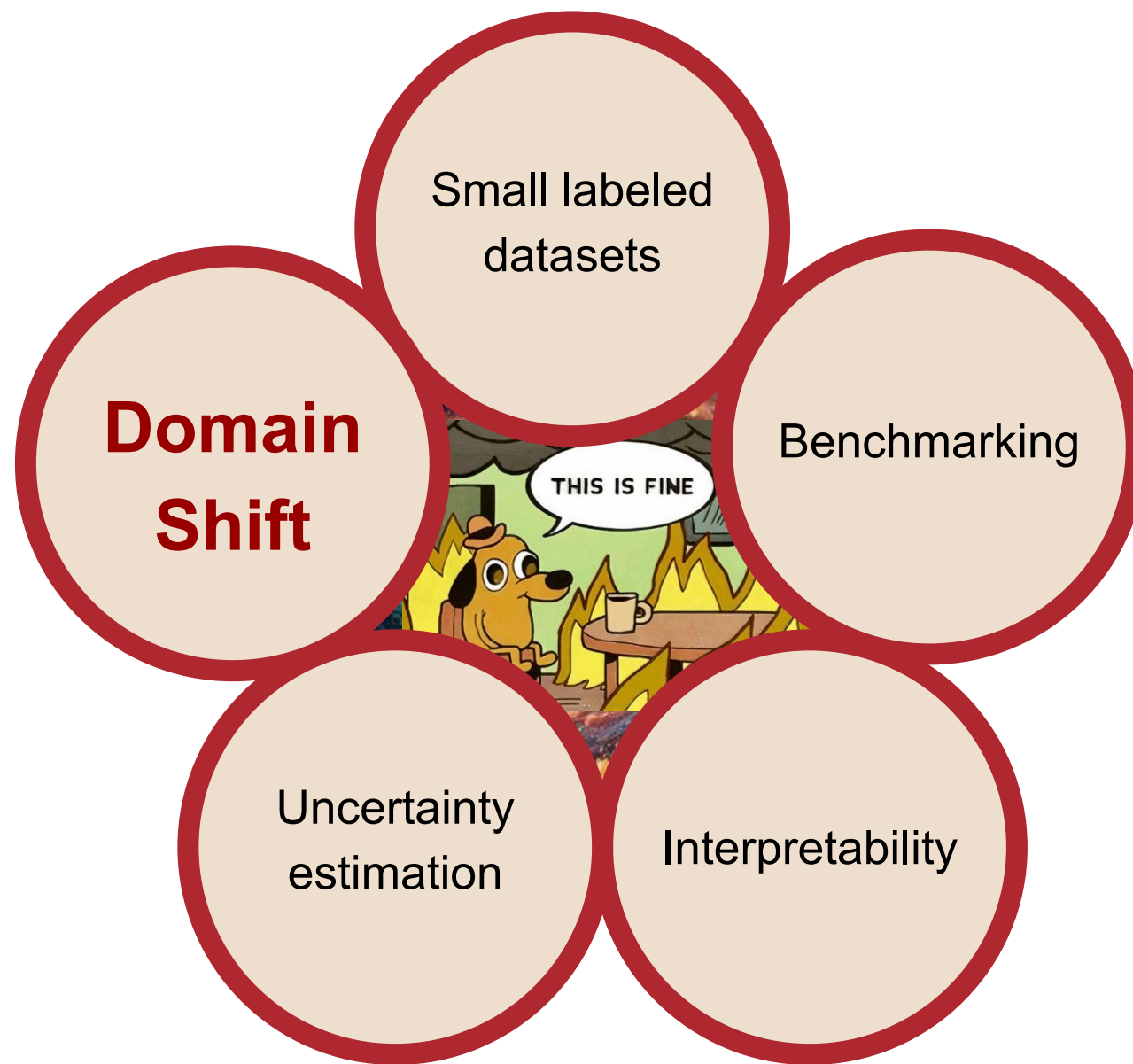
I want to train my AI model by I don't have enough trustworthy labels?

Can I train on simulations and apply the model to my real data?

How well is my model going to perform?

Can I understand its performance?

Can I compare my model with other researchers?





Talk Outline

Part 1: Domain Shift Problem

Part 2: Domain Adaptation Techniques

Part 3: Distance-based Methods

Part 4: Uncertainty Quantification



01

Domain Shift Problem



More Data - More problems

All areas of astro(physics) often need to create **model trained on simulated data, that also work on real data!**

DOMAIN SHIFT

Missing and unknown physics, wrong geometry, background levels

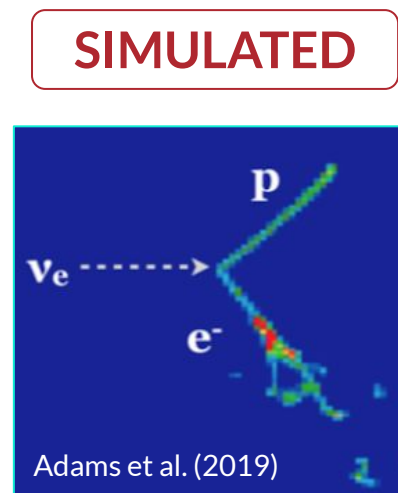
Computational constraints for simulations

Detector problems, transients, errors, data compression

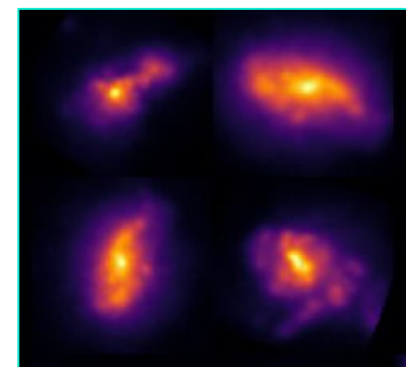
Imperfect addition of observational effects

Different detectors or telescopes

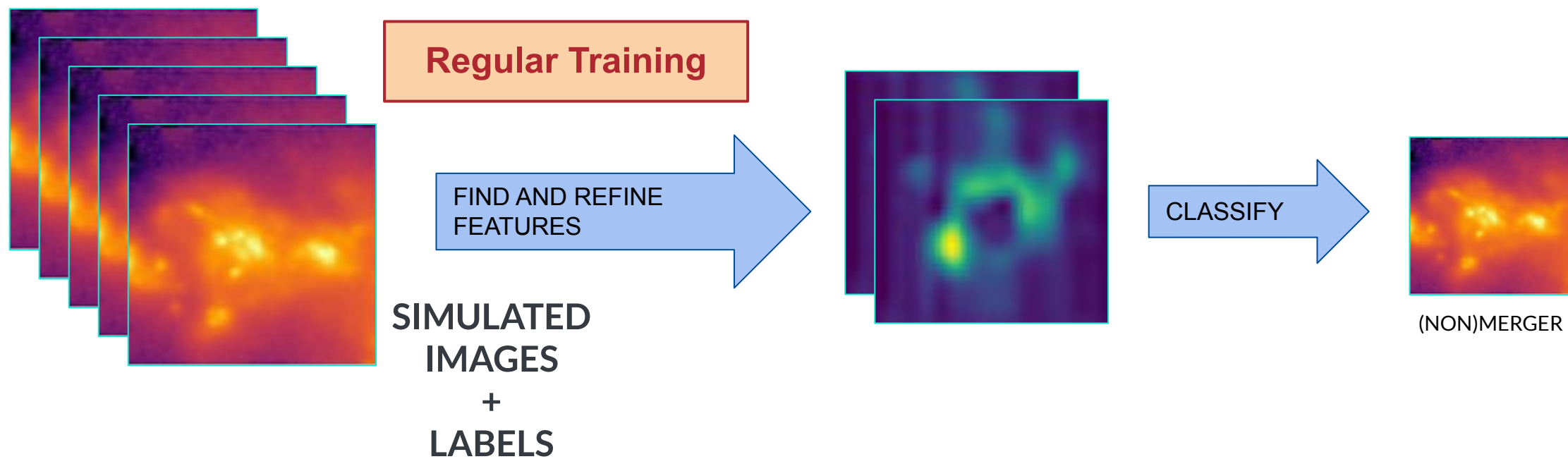
MicroBooNE
(neutrinos)



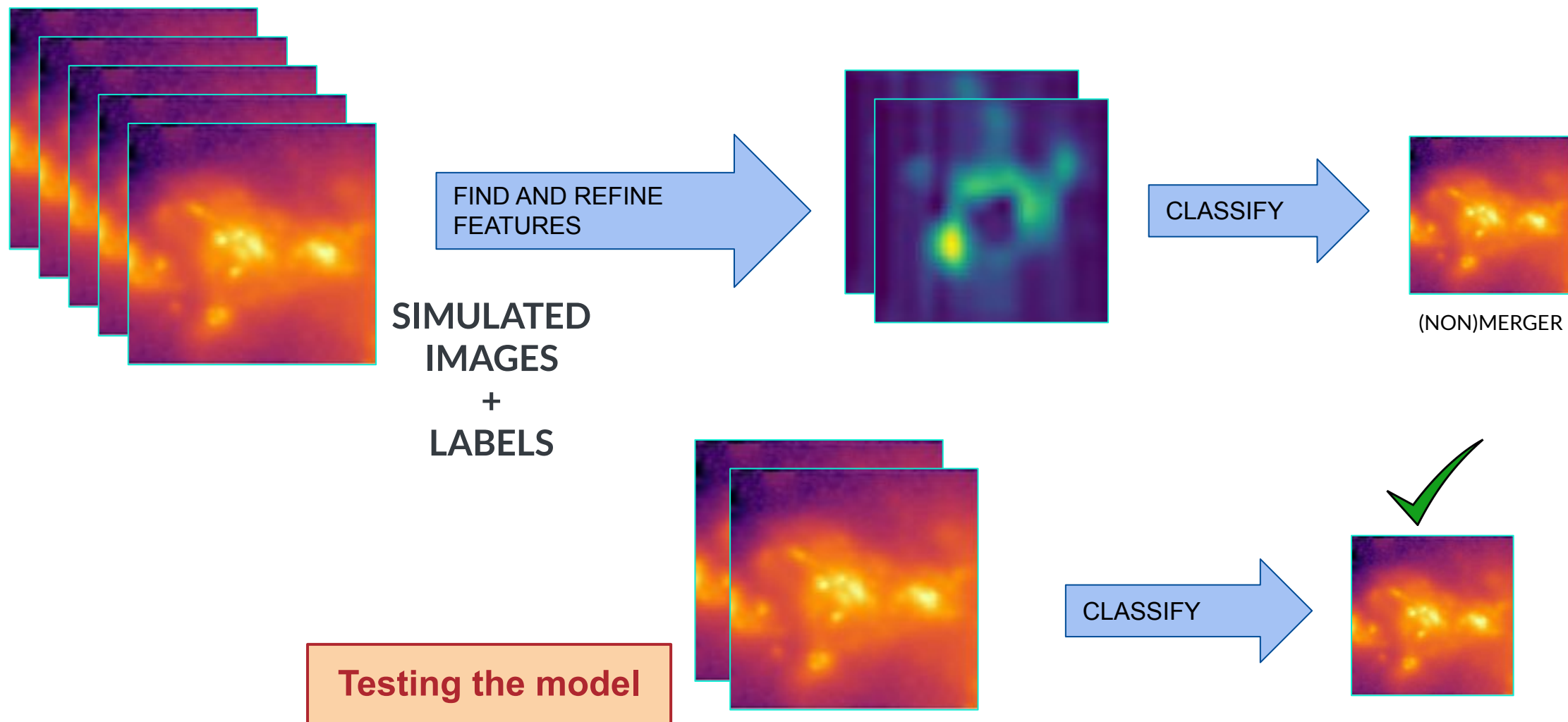
Illustris / Hubble
(merging galaxies)



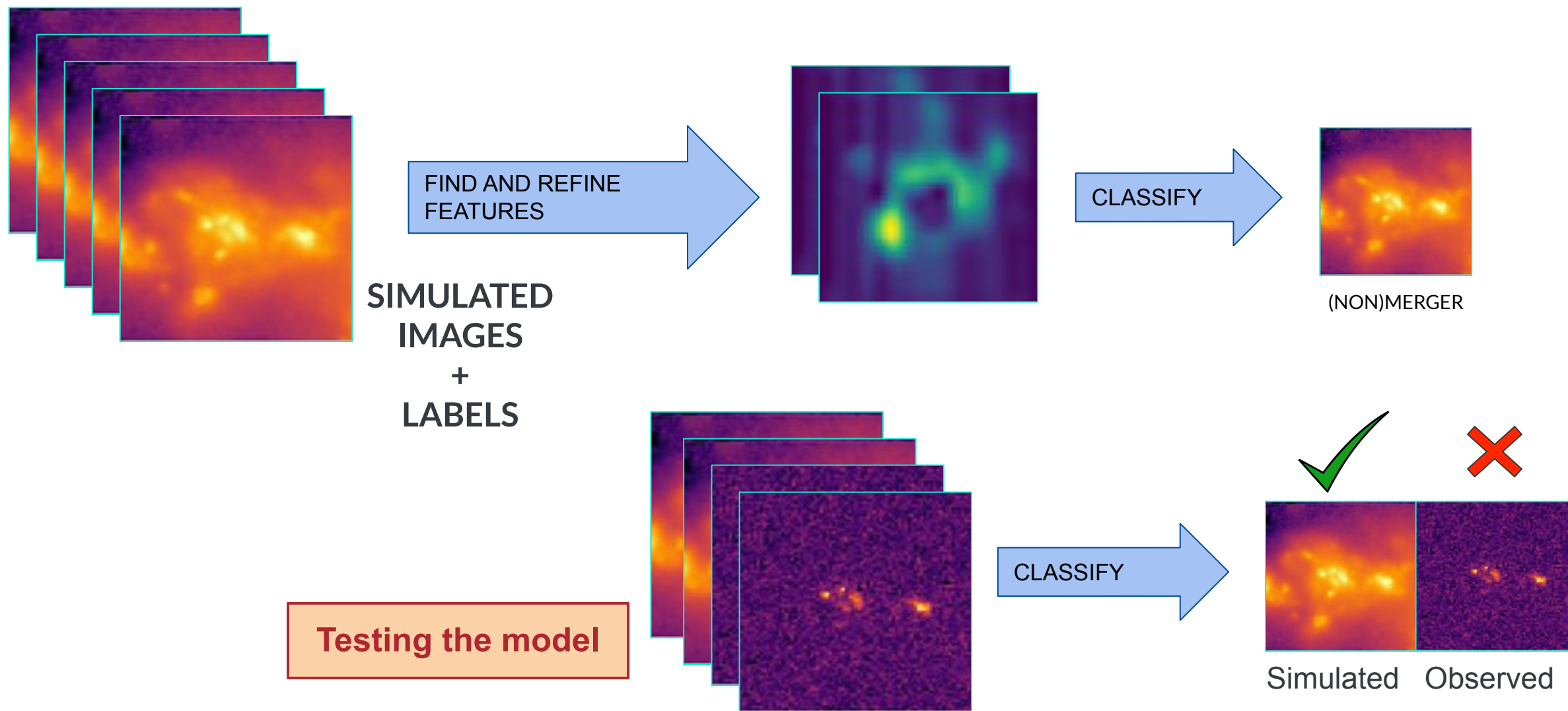
Traditional ML



Traditional ML



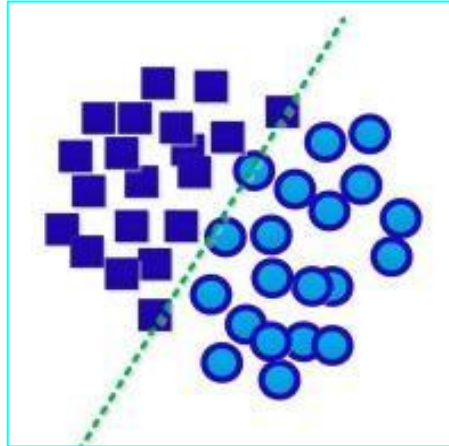
Traditional ML



Why do models fail?

Train the model
on source
dataset and find
the decision
boundary.

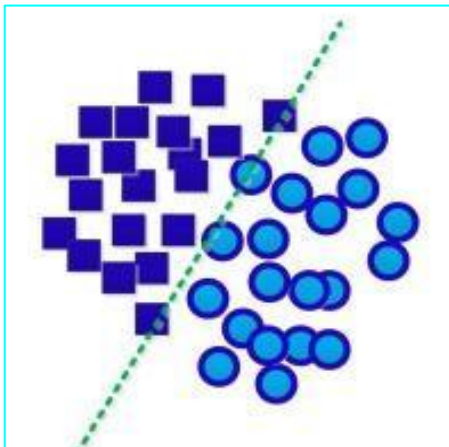
Source Domain



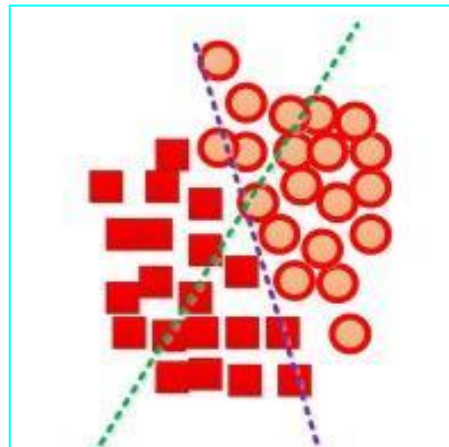
Why do models fail?

New domain is shifted,
learned decision
boundary doesn't
work.

Source Domain



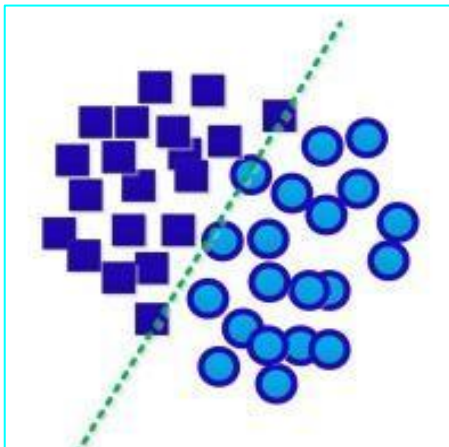
Target Domain



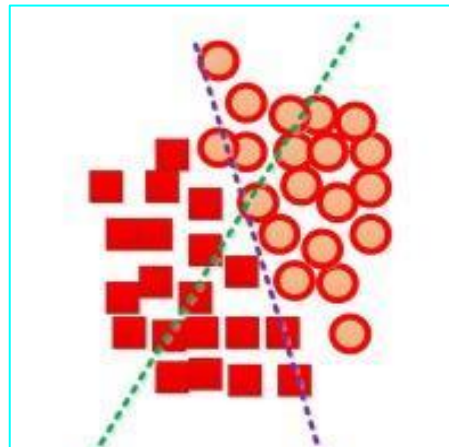
⚙️ Why do models fail?

We need to align
the data during
training!

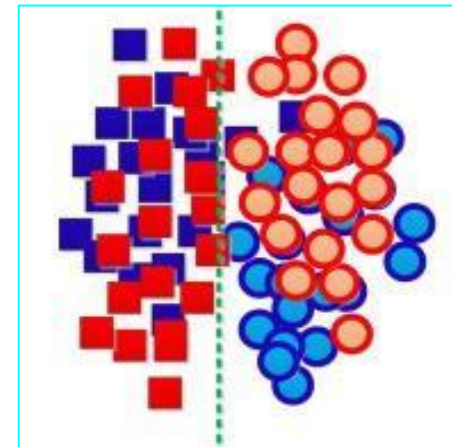
Source Domain



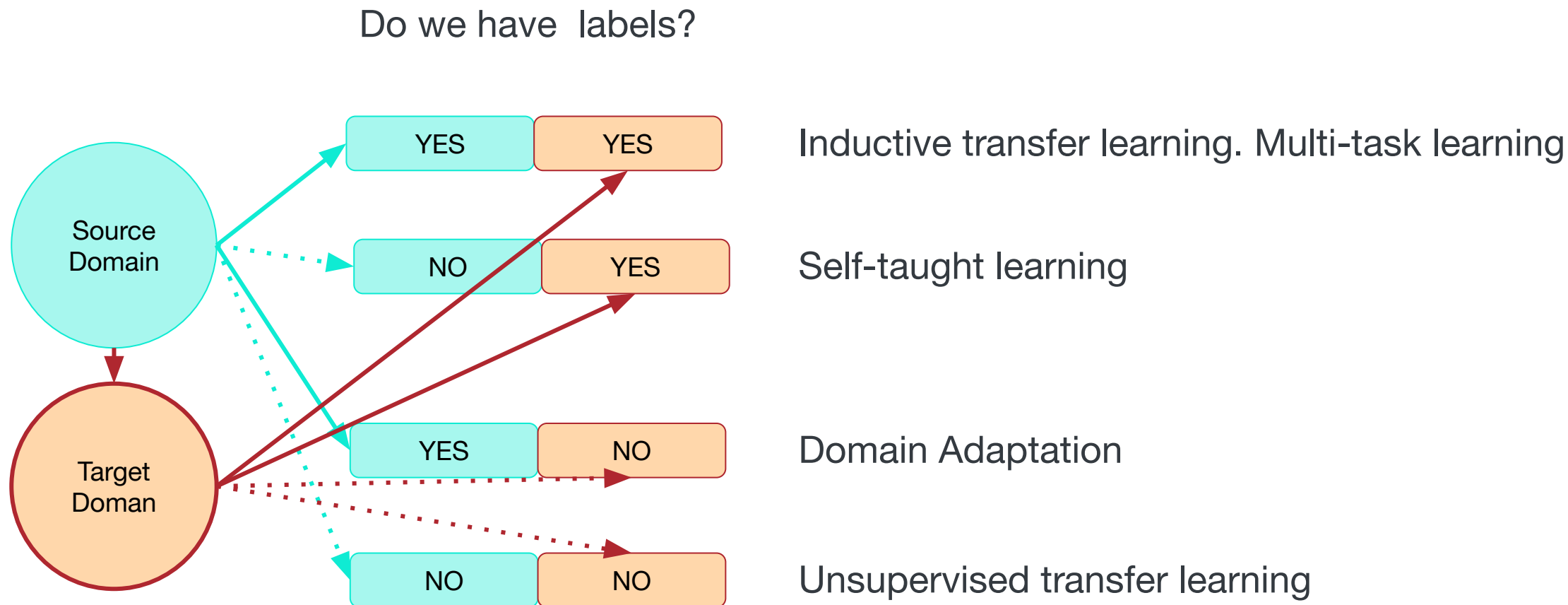
Target Domain



Domain Alignment

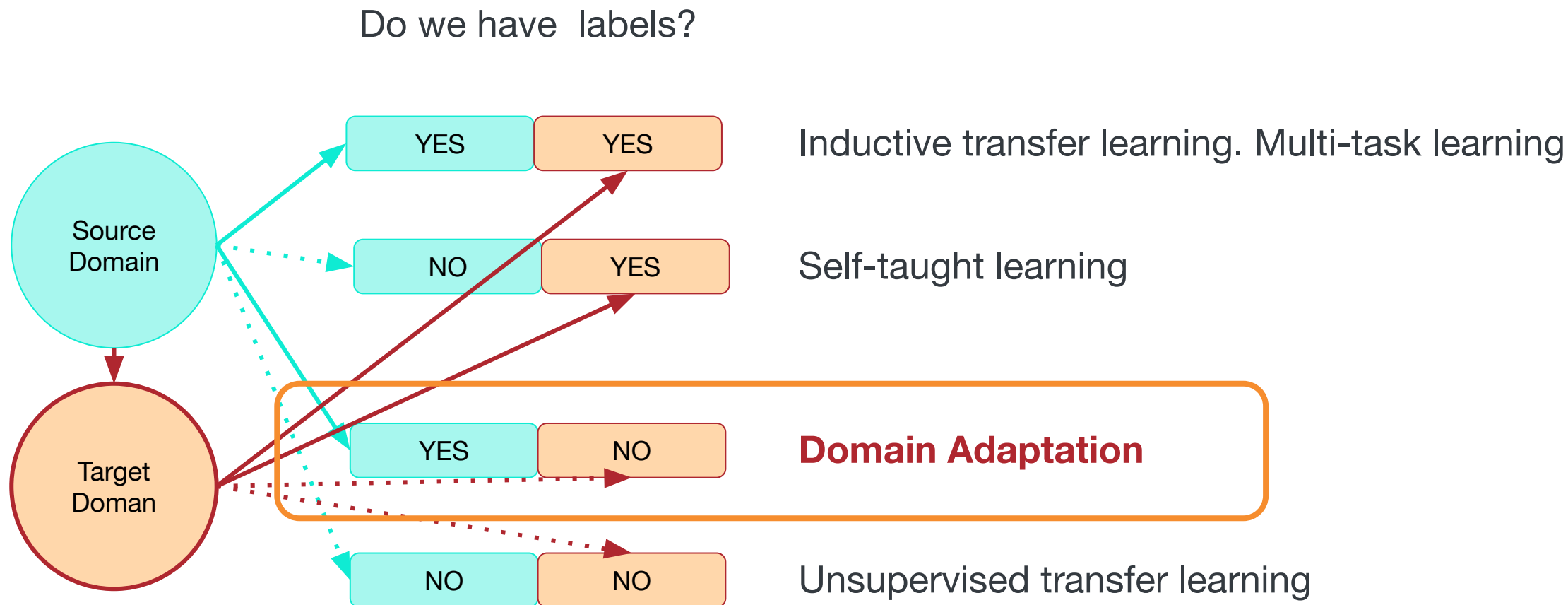


The Landscape of Multi-Dataset Learning



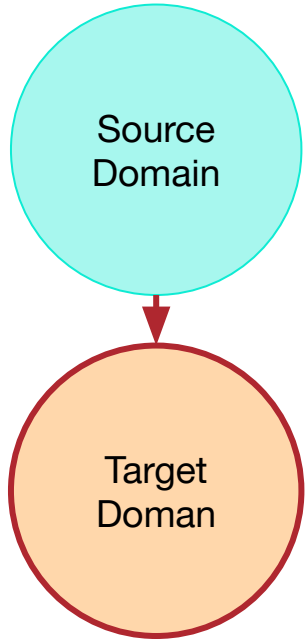


The Landscape of Multi-Dataset Learning





Datasets and Types of Shifts



$$\mathcal{D}_S = \{(x_i^S, y_i^S)\}_{i=1}^{n_S}$$

with n_S labelled samples, from joint distribution $P_S(x, y)$

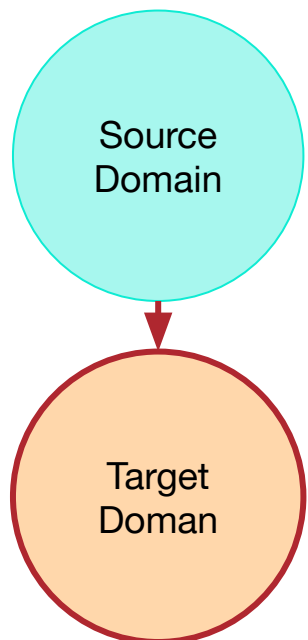
$$\mathcal{D}_T = \{x_i^T\}_{i=1}^{n_T}$$

with n_T unlabelled samples, from joint distribution $P_T(x)$

$$P_S(x, y) \neq P_T(x, y)$$



Datasets and Types of Shifts



$$\mathcal{D}_S = \{(x_i^S, y_i^S)\}_{i=1}^{n_S}$$

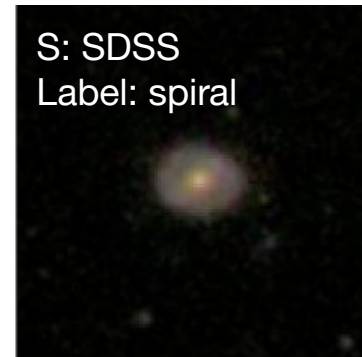
with n_S labelled samples, from joint distribution $P_S(x, y)$

$$\mathcal{D}_T = \{x_i^T\}_{i=1}^{n_T}$$

with n_T unlabelled samples, from joint distribution $P_T(x)$

$$P_S(x, y) \neq P_T(x, y)$$

S: SDSS
Label: spiral



T: SDSS
No Label

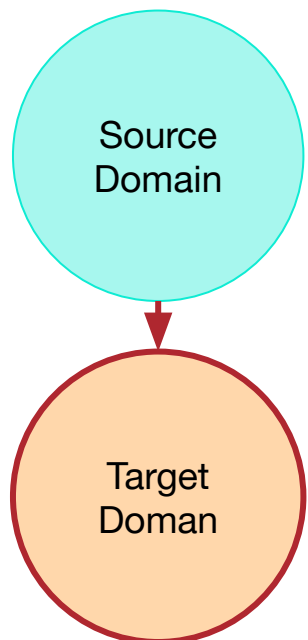


Label Shift:

$$P_s(Y) \neq P_t(Y) \quad \text{while} \quad P_s(X | Y) = P_t(X | Y)$$

Conditional distribution of data given a label stays the same, but the label distribution **changes** i.e., the frequency of classes differs across domains (e.g., class imbalance), but **each class "looks" the same in both domains.**

⚙ Datasets and Types of Shifts



$$\mathcal{D}_S = \{(x_i^S, y_i^S)\}_{i=1}^{n_S}$$

with n_S labelled samples, from joint distribution $P_S(x, y)$

$$\mathcal{D}_T = \{x_i^T\}_{i=1}^{n_T}$$

with n_T unlabelled samples, from joint distribution $P_T(x)$

$$P_S(x, y) \neq P_T(x, y)$$

S: SDSS
Label: spiral

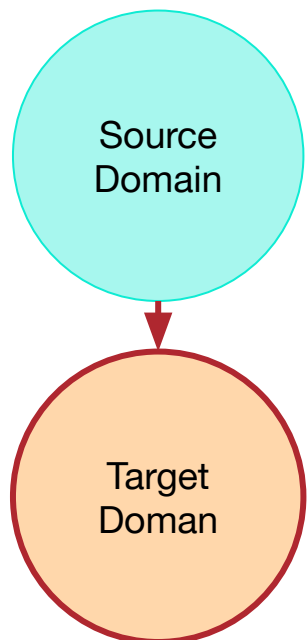
T: SDSS
No Label

(barred/unbarred
spiral)

Concept Shift: $P_s(X) = P_t(X)$, but $P_s(Y | X) \neq P_t(Y | X)$

The input distribution of data stays the same, but the mapping from input to label **changes** i.e., the same input might be labeled differently across domains (e.g., due to different labeling policies or evolving concepts).

⚙ Datasets and Types of Shifts



$$\mathcal{D}_S = \{(x_i^S, y_i^S)\}_{i=1}^{n_S}$$

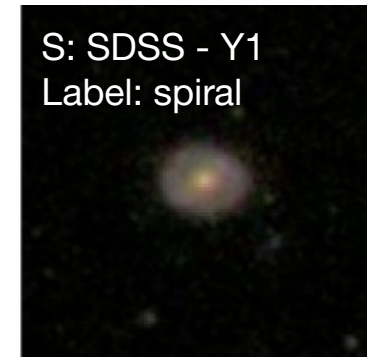
with n_S labelled samples, from joint distribution $P_S(x, y)$

$$\mathcal{D}_T = \{x_i^T\}_{i=1}^{n_T}$$

with n_T unlabelled samples, from joint distribution $P_T(x)$

$$P_S(x, y) \neq P_T(x, y)$$

S: SDSS - Y1
Label: spiral



T: SDSS - Y5
No Label

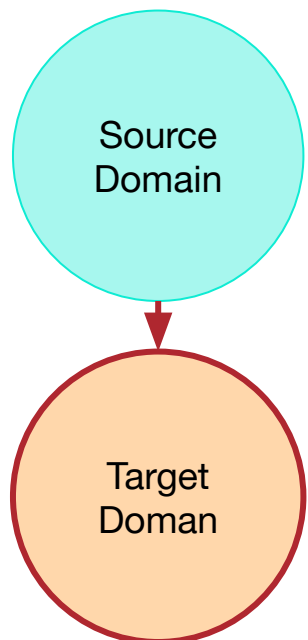


Conditional Shift: $P_S(Y) = P_T(Y)$ but $P_S(X | Y) \neq P_T(X | Y)$

The label distribution the same, but the mapping from input to label changes i.e., the same labeled corresponds to different data due to measurement differences.



Datasets and Types of Shifts



$$\mathcal{D}_S = \{(x_i^S, y_i^S)\}_{i=1}^{n_S}$$

with n_S labelled samples, from joint distribution $P_S(x, y)$

$$\mathcal{D}_T = \{x_i^T\}_{i=1}^{n_T}$$

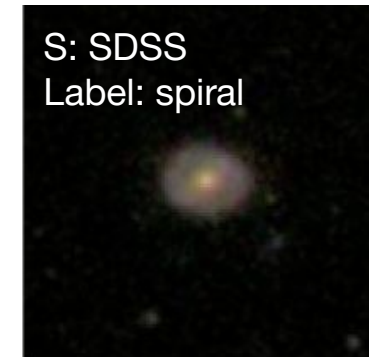
with n_T unlabelled samples, from joint distribution $P_T(x)$

$$P_S(x, y) \neq P_T(x, y)$$

Covariate Shift: $P_s(X) \neq P_t(X)$ but $P_s(Y | X) = P_t(Y | X)$

Data distribution changes (often not just image properties but class distribution as well for example e.g., different sensors, sampling bias), **but the conditional distribution of labels given the data remains the same.**

S: SDSS
Label: spiral

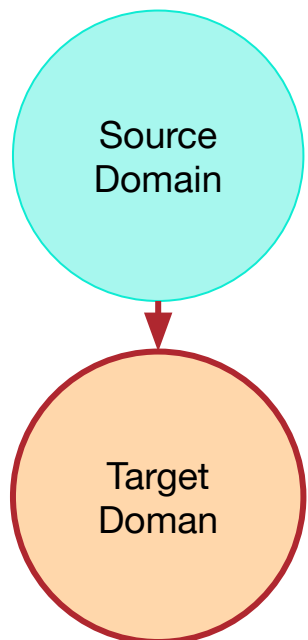


T: DESI
No Label





Datasets and Types of Shifts



$$\mathcal{D}_S = \{(x_i^S, y_i^S)\}_{i=1}^{n_S}$$

with n_S labelled samples, from joint distribution $P_S(x, y)$

$$\mathcal{D}_T = \{x_i^T\}_{i=1}^{n_T}$$

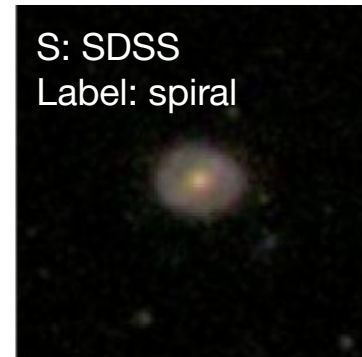
with n_T unlabelled samples, from joint distribution $P_T(x)$

$$P_S(x, y) \neq P_T(x, y)$$

Compound Shift:

Any potential combination of two or more shift!

S: SDSS
Label: spiral



T: DESI
No Label



Dataset Shift Quiz !

Scenario 1:

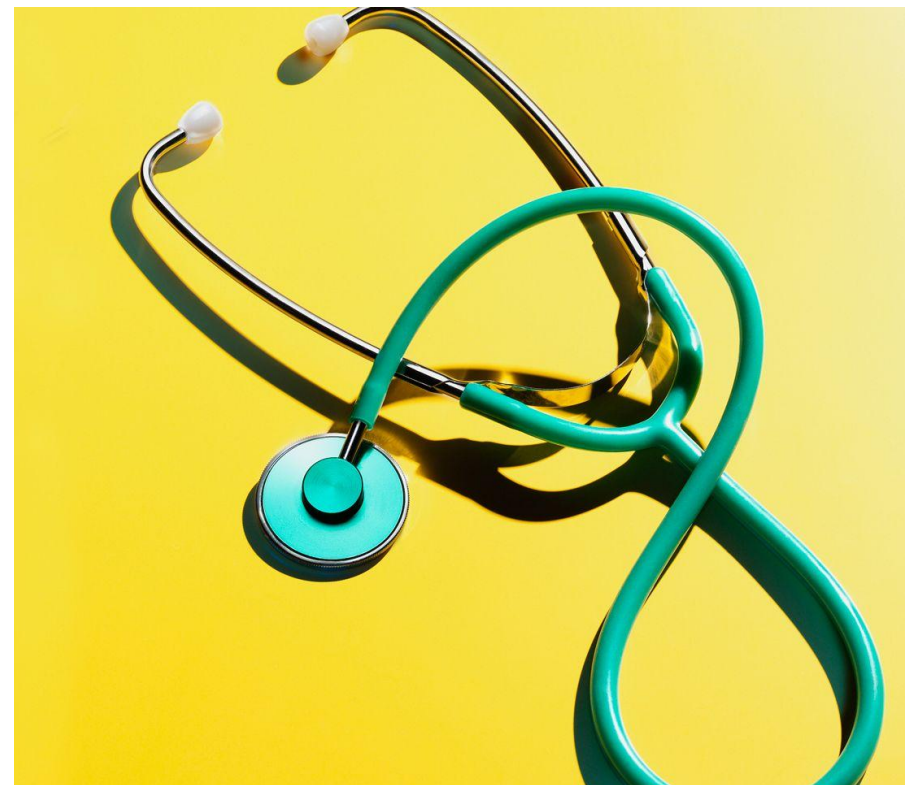
You trained a model to detect a rare disease in a hospital with a low prevalence rate (5%). It's now deployed in a different region where the prevalence is 30%. Patient features (symptoms, test values) behave similarly for each diagnosis.

Covariate

Concept / Conditional

Label

Answer:



Dataset Shift Quiz !

Scenario 1:

You trained a model to detect a rare disease in a hospital with a low prevalence rate (5%). It's now deployed in a different region where the prevalence is 30%. Patient features (symptoms, test values) behave similarly for each diagnosis.

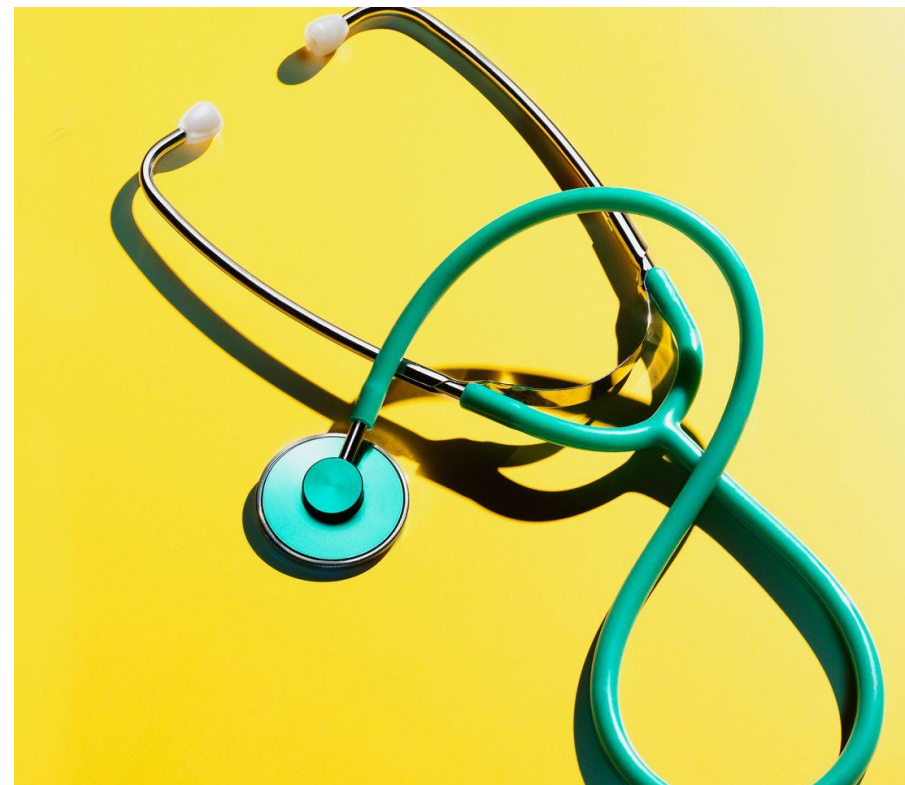
Covariate

Concept / Conditional

Label

Answer:

Label shift — $P(y)$ has changed (disease rate), but $P(x | y)$ is stable.



Dataset Shift Quiz !

Scenario 2:

Your spam classifier was trained on English-language emails. Now it's deployed to classify emails in both English and French. The spam-to-nospam ratio is about the same, and spam emails still tend to have similar structures (e.g., lots of links, certain keywords).

Covariate

Concept / Conditional

Label

Answer:



Dataset Shift Quiz !

Scenario 2:

Your spam classifier was trained on English-language emails. Now it's deployed to classify emails in both English and French. The spam-to-nospam ratio is about the same, and spam emails still tend to have similar structures (e.g., lots of links, certain keywords).

Covariate

Concept / Conditional

Label

Answer:

Covariate shift — the input distribution $P(x)$ changed (language), but the conditional $P(y | x)$ stayed relatively consistent.



Dataset Shift Quiz !

Scenario 3:

Your credit risk model was trained 5 years ago. Now, due to economic changes and inflation, people with low income are more likely to default than they used to. Income distribution remains unchanged.

Covariate

Concept / Conditional

Label

Answer:



Dataset Shift Quiz !

Scenario 3:

Your credit risk model was trained 5 years ago. Now, due to economic changes and inflation, people with low income are more likely to default than they used to. Income distribution remains unchanged.

Covariate

Concept / Conditional

Label

Answer:

Concept shift — the relationship $P(y | x)$ changed due to evolving economic factors even though $P(x)$ remained the same.



Dataset Shift Quiz !

Scenario 4:

You trained a model on a balanced dataset of merging and non-merging galaxies. At test time, your realistic catalog contains 90% of the non-merging galaxy images, and the remaining 10% are merging. The images for each galaxy still look the same.

Covariate

Concept / Conditional

Label

Answer:



Dataset Shift Quiz !

Scenario 4:

You trained a model on a balanced dataset of merging and non-merging galaxies. At test time, your realistic catalog contains 90% of the non-merging galaxy images, and the remaining 10% are merging. The images for each galaxy still look the same.

Covariate

Concept / Conditional

Label

Answer:

Label shift — the label distribution $P(y)$ changed (mergers are rare), but image features ($P(x | y)$) stayed the same.



Dataset Shift Quiz !

Scenario 5:

A remote sensing model is trained to classify crop types (e.g., wheat, corn, rice) using satellite imagery from **Region A** during a specific season. Later, the model is applied to **Region B**, where crop types and labels are the same, but **soil composition, weather patterns, and sunlight levels differ**, affecting how the crops look in satellite images

Covariate

Concept / Conditional

Label

Answer:



Dataset Shift Quiz !

Scenario 5:

A remote sensing model is trained to classify crop types (e.g., wheat, corn, rice) using satellite imagery from **Region A** during a specific season. Later, the model is applied to **Region B**, where crop types and labels are the same, but **soil composition, weather patterns, and sunlight levels differ**, affecting how the crops look in satellite images .

Covariate

Concept / Conditional

Label

Answer:

Covariate shift — the input distribution $P(\mathbf{x})$ (satellite pixel features) has changed due to environmental factors, but $P(\mathbf{y} | \mathbf{x})$ (how features map to crop types) is assumed to remain the same.



Useful Resources:

- [“A Brief Review of Domain Adaptation” Farahini et al. 2020.](#)
- [“Deep Visual Domain Adaptation: A Survey” Wand & Deng 2018.](#)
- [“Domain Adaptation for Visual Applications: A Comprehensive Survey” Csurka 2017.](#)

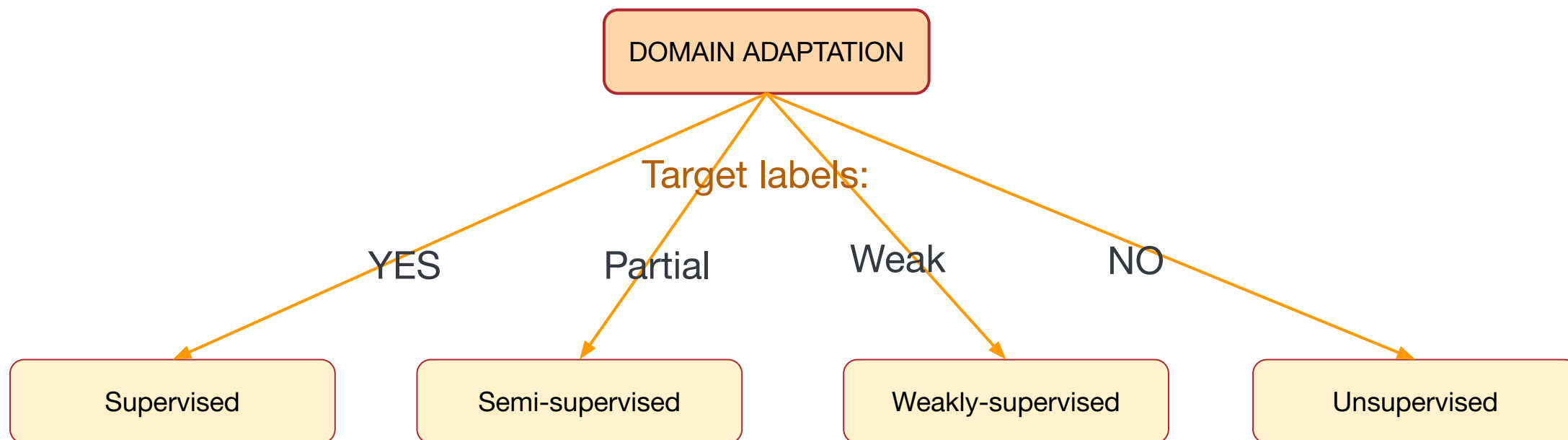
02

Domain Adaptation Techniques



Types of Domain Adaptation Approaches

There is a huge number of domain adaptation methods to suit different needs.

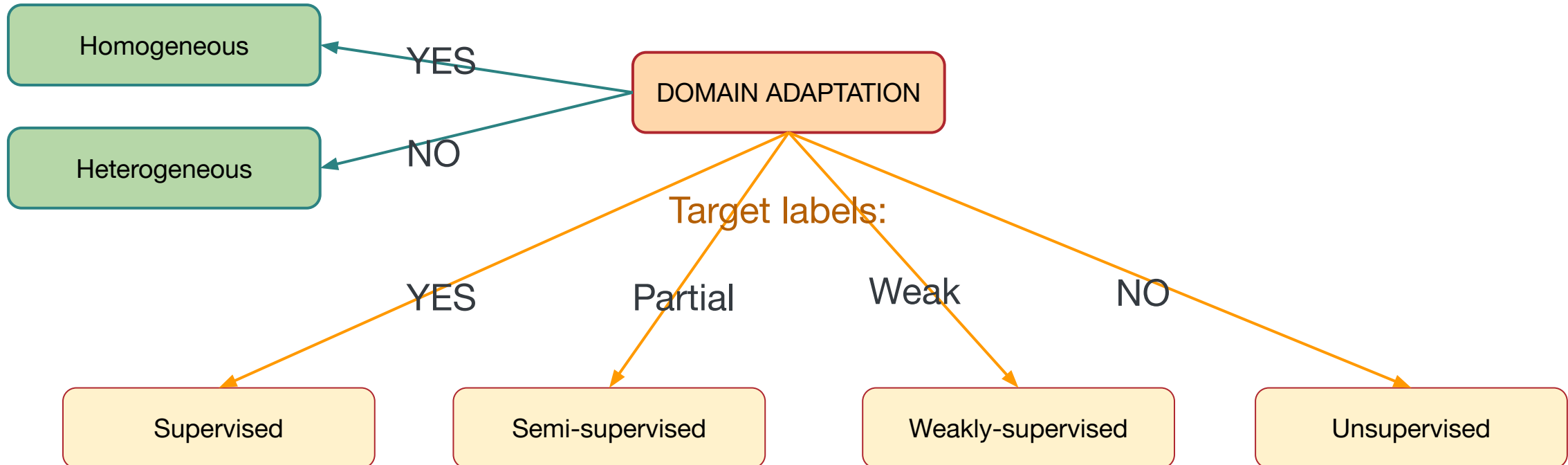




Types of Domain Adaptation Approaches

There is a huge number of domain adaptation methods to suit different needs.

Similarity of the
feature space:



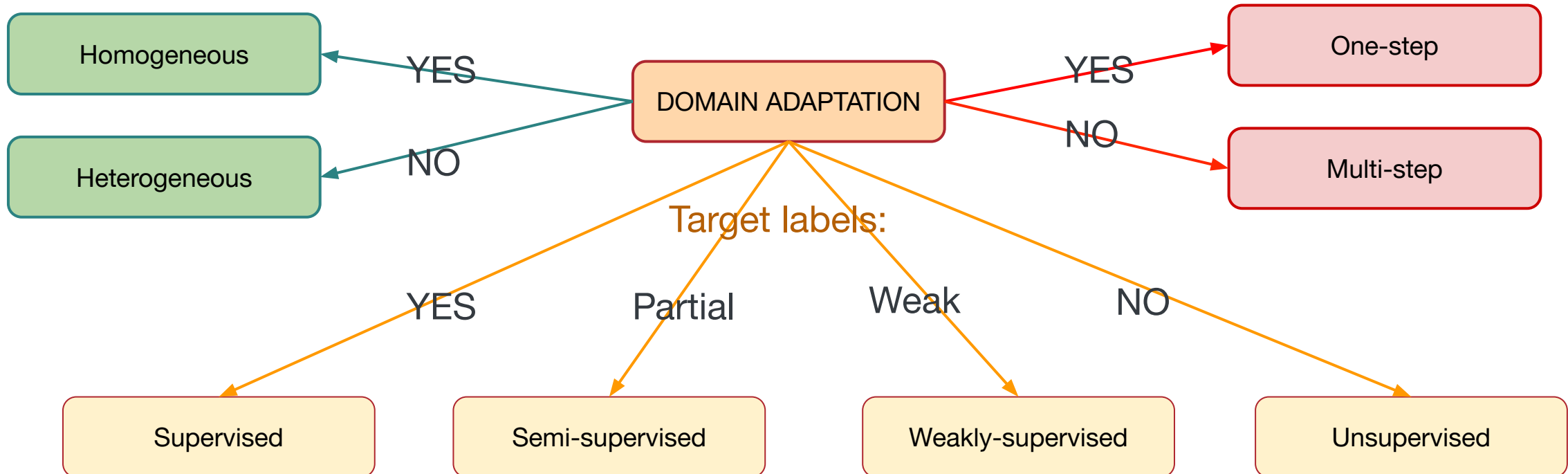


Types of Domain Adaptation Approaches

There is a huge number of domain adaptation methods to suit different needs.

Similarity of the
feature space:

Knowledge directly
transferred:

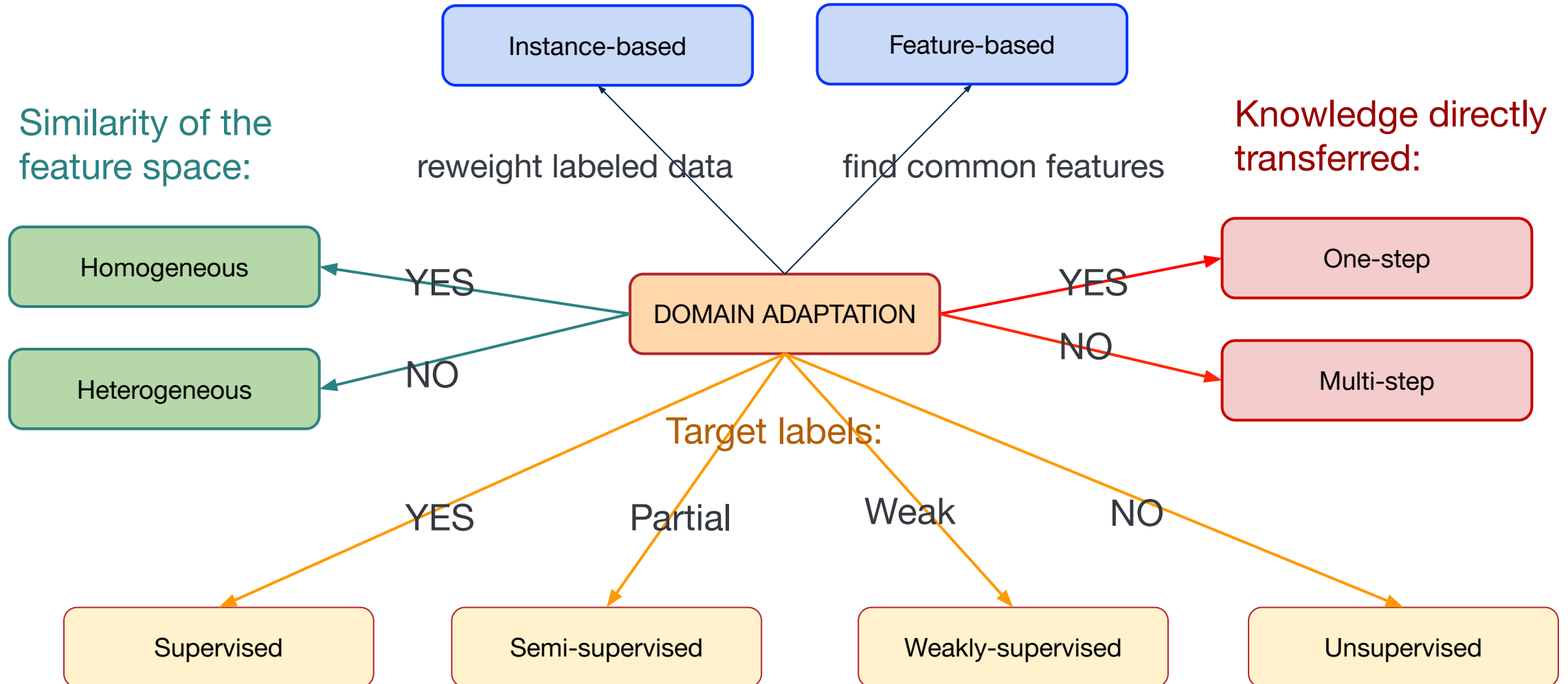




Types of Domain Adaptation Approaches

There is a huge number of domain adaptation methods to suit different needs.

Alignment approach:



Domain Adaptation Quiz !

Scenario 1:

You trained a cosmic ray classifier using labeled simulated data. You also have a small set of real cosmic ray events with noisy labels from an old detector. What is this setup best described as?

Supervised

Weakly supervised

Unsupervised

Answer:



⚙️ Domain Adaptation Quiz !

Scenario 1:

You trained a cosmic ray classifier using labeled simulated data. You also have a small set of real cosmic ray events with noisy labels from an old detector. What is this setup best described as?

Supervised

Weakly supervised

Unsupervised

Answer:

You have real data labels, but they are **noisy or imprecise** — so you need to use **weak supervision**. Supervised DA assumes accurate labels, while unsupervised DA assumes no labels. Weak supervision methods handle uncertainty, label correction, or probabilistic label modeling.





Domain Adaptation Quiz !

Scenario 2:

You align galaxy image features across domains by first performing PCA, then minimizing some domain adaptation loss. You then fine-tune a regressor on the aligned target features. What type of method is this?

One-step, instance based

Multi-step, feature based

Multi-step, instance based

Answer:



Domain Adaptation Quiz !

Scenario 2:

You align galaxy image features across domains by first performing PCA, then minimizing some domain adaptation loss. You then fine-tune a regressor on the aligned target features. What type of method is this?

One-step, instance based

Multi-step, feature based

Multi-step, instance based

Answer:

This approach first **transforms features** (PCA), then **aligns distributions** using a DA method, and finally adapts the model — making it **multi-step and feature-based**. Instance-based methods directly reweight or select examples data points.





Domain Adaptation Quiz !

Scenario 3:

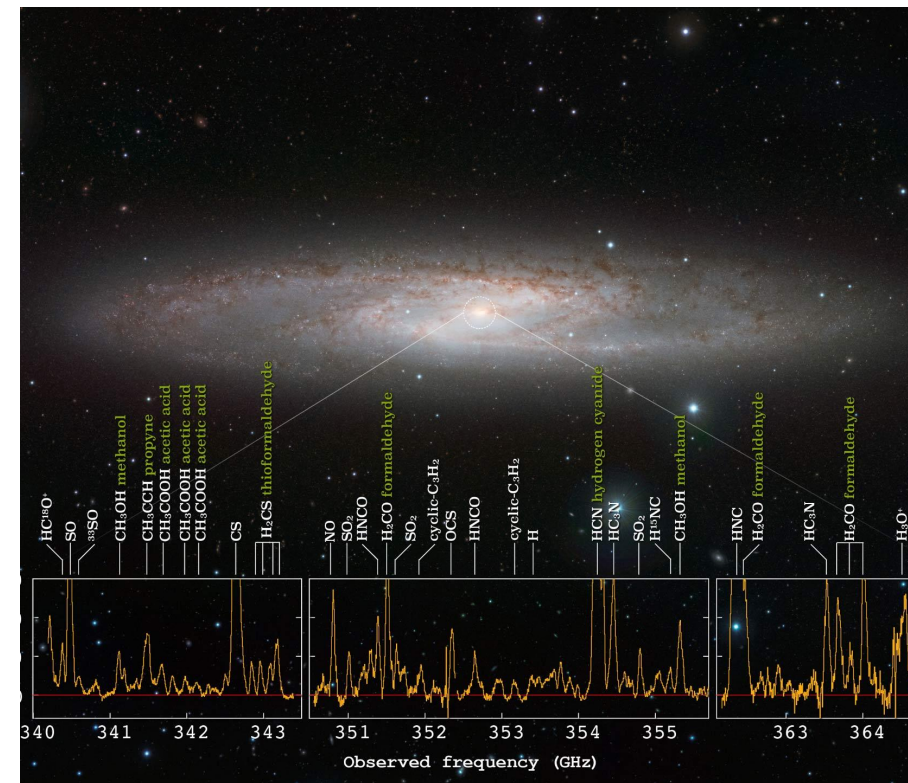
You trained a redshift estimation model using spectroscopic data (rich, high-dimensional spectra). You now want to apply it to photometric data (broadband magnitudes). What kind of domain adaptation challenge is this?

Unsupervised

Homogeneous

Heterogeneous

Answer:



Domain Adaptation Quiz !

Scenario 3:

You trained a redshift estimation model using spectroscopic data (rich, high-dimensional spectra). You now want to apply it to photometric data (broadband magnitudes). What kind of domain adaptation challenge is this?

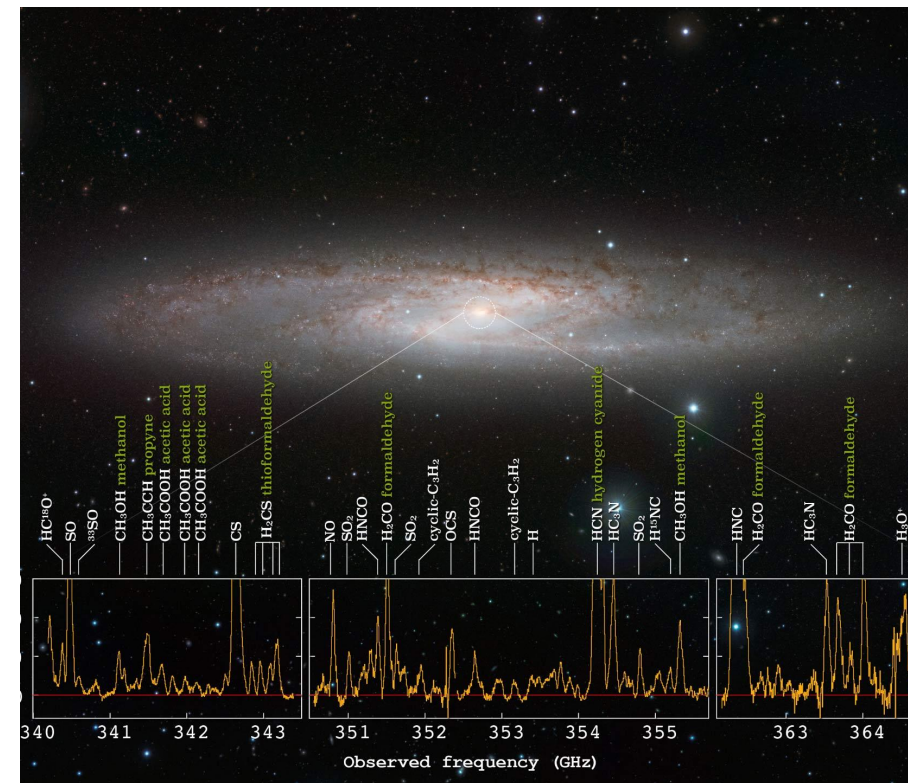
Unsupervised

Homogeneous

Heterogeneous

Answer:

The **input features differ in type and dimension** — spectra vs. photometry — so this is a **heterogeneous DA** problem. Homogeneous DA assumes the same input space. The key challenge here lies in mapping between incompatible feature spaces (e.g., using shared latent representations or cross-modal alignment).





Types of Domain Adaptation Approaches

There is a huge number of domain adaptation methods to suit different needs.

Alignment approach:

Instance-based

Feature-based

reweight labeled data

find common features

Similarity of the
feature space:

Homogeneous

YES

Heterogeneous

NO

DOMAIN ADAPTATION

Knowledge directly
transferred:

One-step

YES

NO

Multi-step

Target labels:

YES

Partial

Weak

NO

Supervised

Semi-supervised

Weakly-supervised

Unsupervised



Combining Datasets

DOMAIN ADAPTATION

Align data distributions in the latent space of the network by forcing the network to **find more robust domain-invariant features**.

Combining Datasets

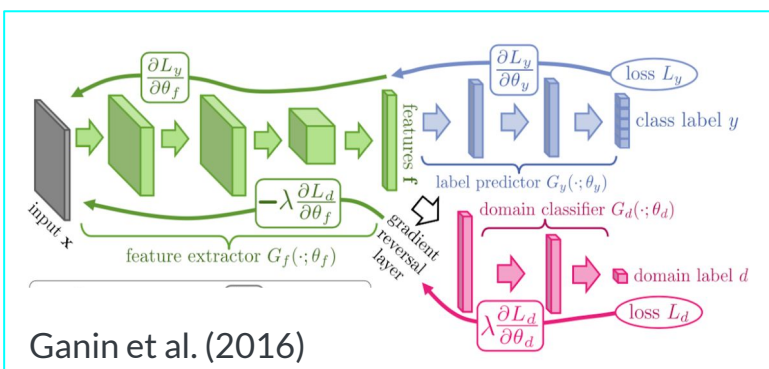
DOMAIN ADAPTATION

Align data distributions in the latent space of the network by forcing the network to **find more robust domain-invariant features**.

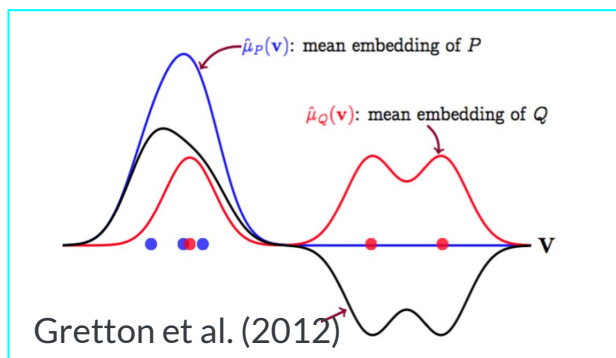
Adversarial

Distance-based
methods

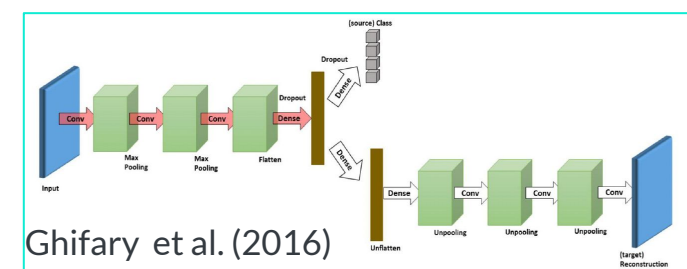
Reconstruction-based



Using domain discriminators to encourage domain confusion through an adversarial objective.



Minimize a distance metric between source and target data.



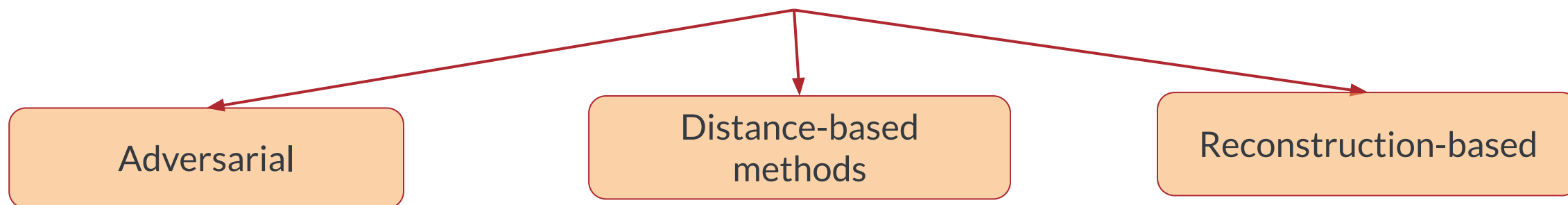
Data reconstruction as an auxiliary task to ensure feature invariance.



Combining Datasets

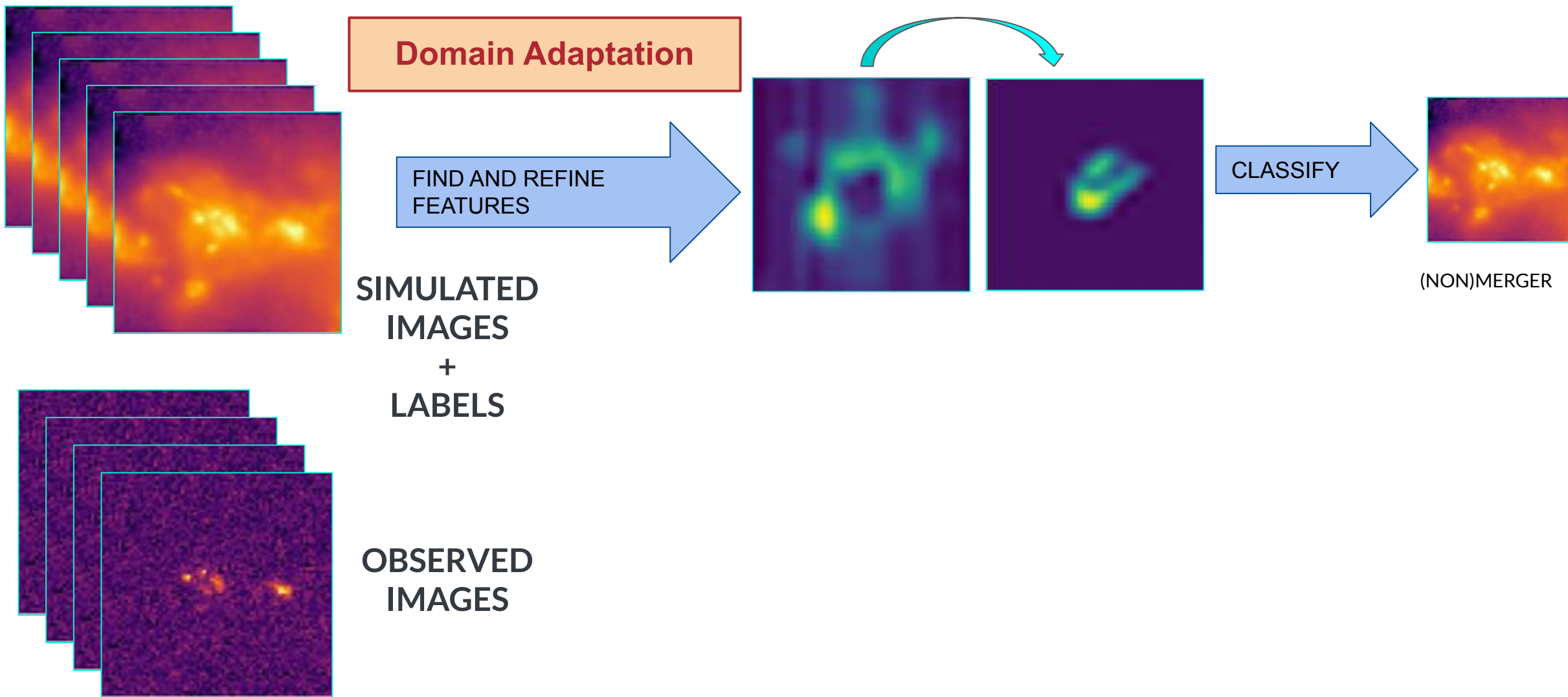
DOMAIN ADAPTATION

Align data distributions in the latent space of the network by forcing the network to **find more robust domain-invariant features**.

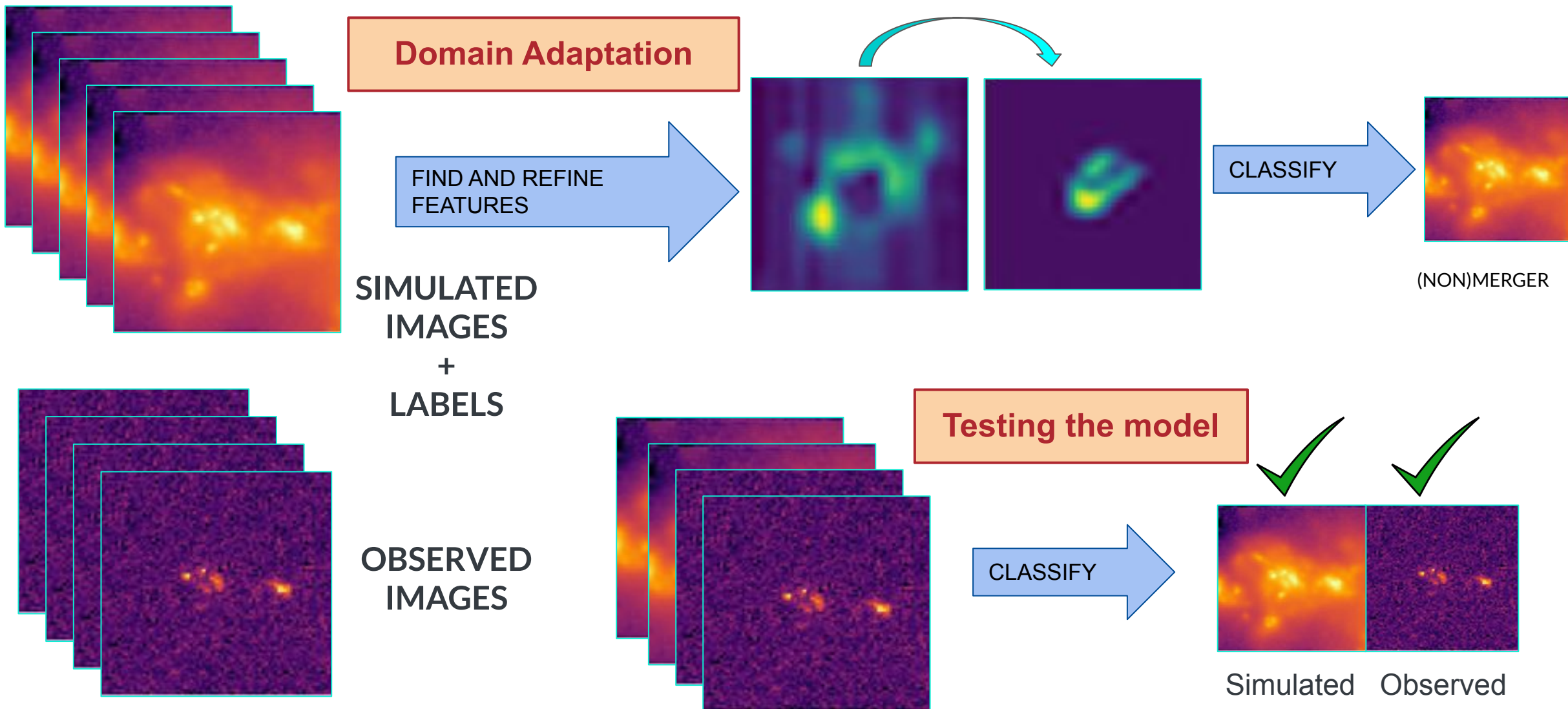


Works on **unlabeled target domain!**
Can be applied to **new data**, no need for scientists to label anything.

Training with Domain Adaptation



Training with Domain Adaptation



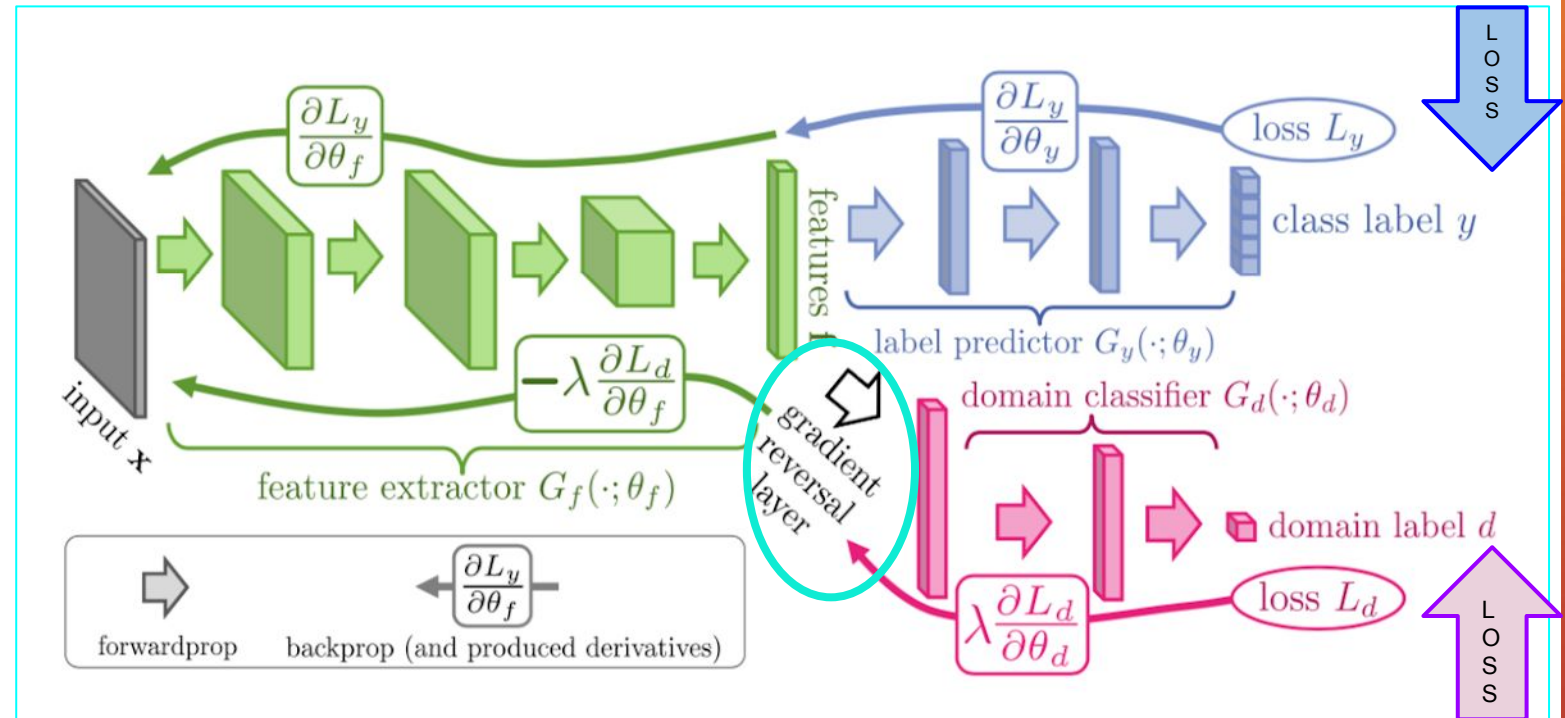


Adversarial Training

Domain Adversarial Neural Networks - DANNs

DANN - feature extractor + label predictor + domain classifier

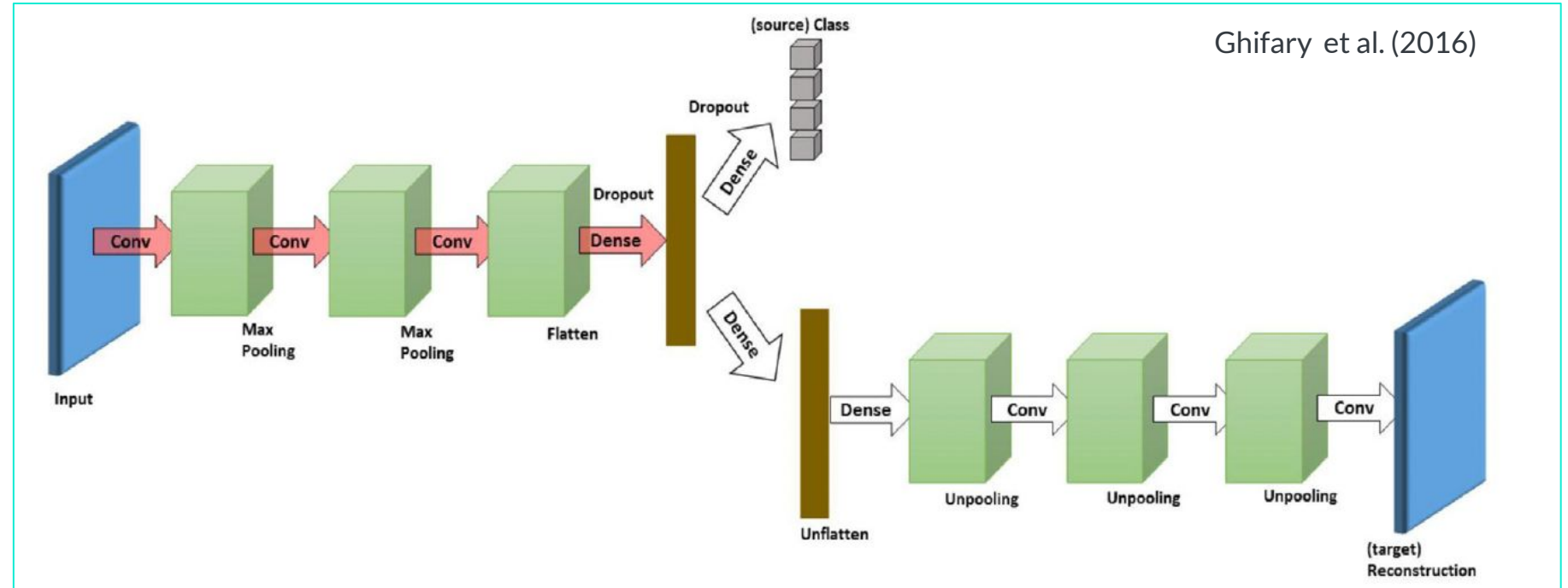
- **Gradient reversal layer** - multiplies the gradient by a negative constant during the backpropagation.
- Results in the extraction of **domain-invariant features**.
- Only source domain images are labeled during training.



Ganin et al. (2016)

Reconstruction-Based Training

Deep Reconstruction-Classification Networks - DRCN



Model jointly learns a shared encoding representation for two tasks:

- **supervised classification** of labeled source data
- **unsupervised reconstruction** of unlabeled target data.

The learnt representation not only preserves discriminability, but also encodes useful information from the target domain.

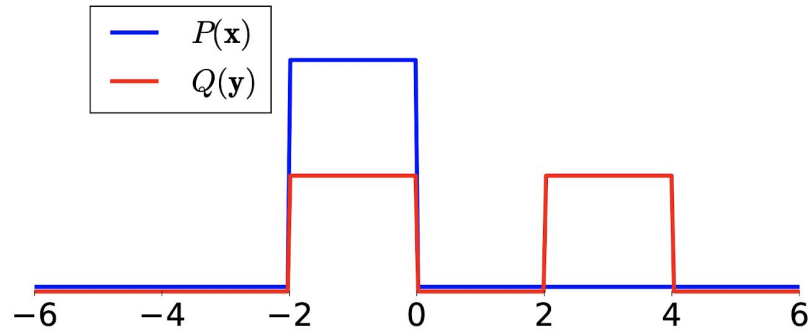


Distance-Based Training

Maximum Mean Discrepancy - MMD

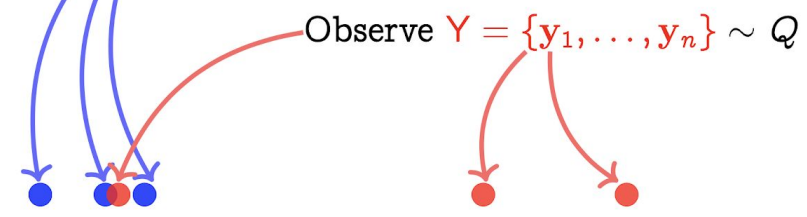
Smola et al. (2007)
Gretton et al. (2012)

Are P and Q different?



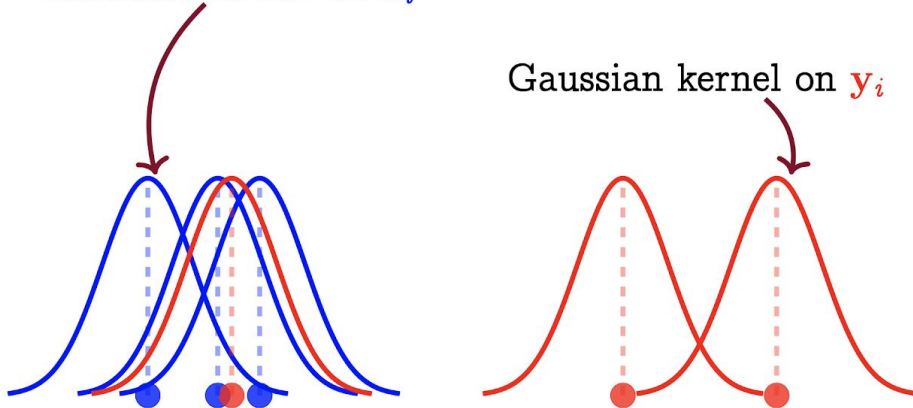
Observe $X = \{x_1, \dots, x_n\} \sim P$

Observe $Y = \{y_1, \dots, y_n\} \sim Q$



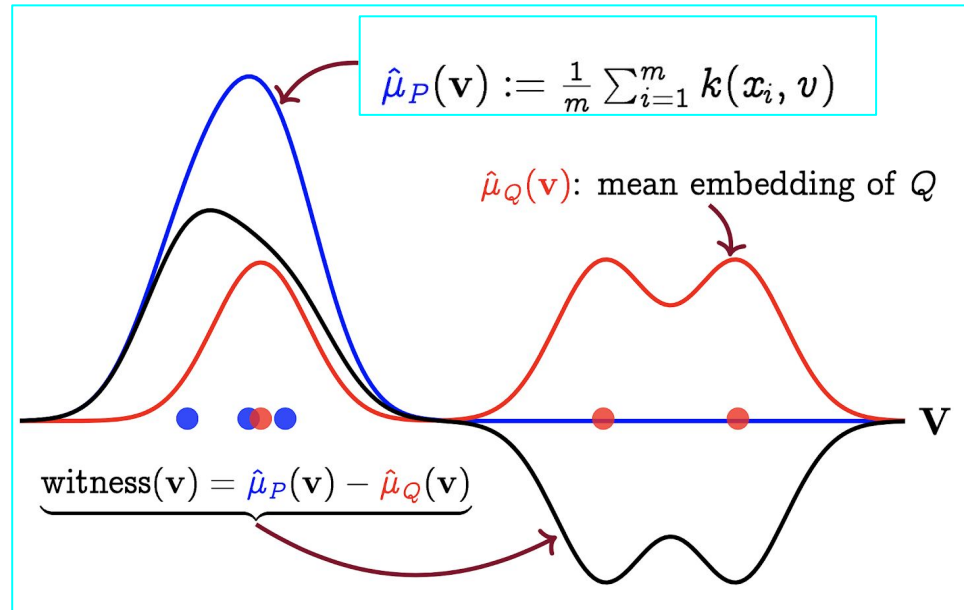
Gaussian kernel on x_i

Gaussian kernel on y_i



$$\hat{\mu}_P(\mathbf{v}) := \frac{1}{m} \sum_{i=1}^m k(x_i, \mathbf{v})$$

$\hat{\mu}_Q(\mathbf{v})$: mean embedding of Q

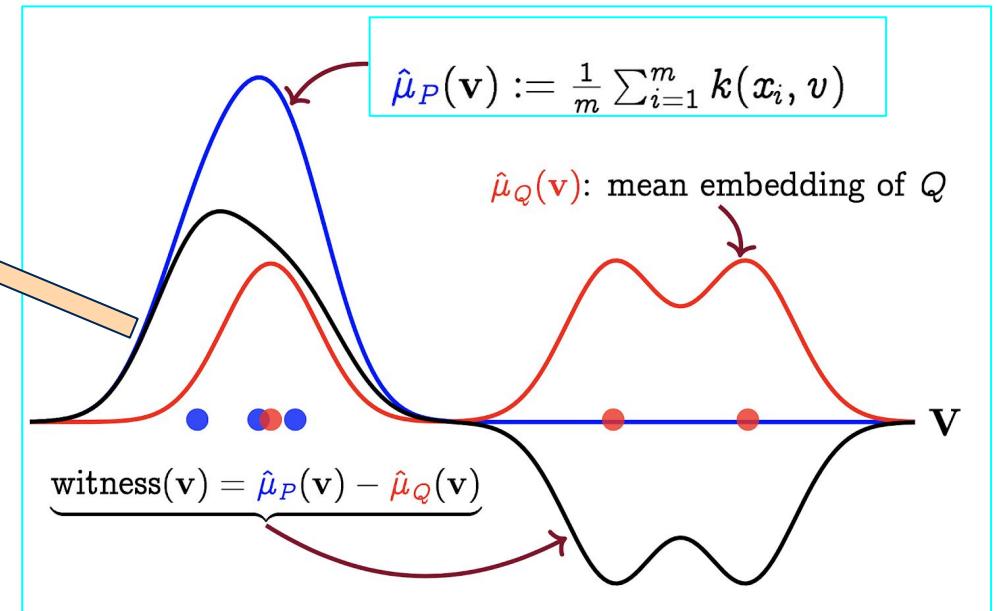
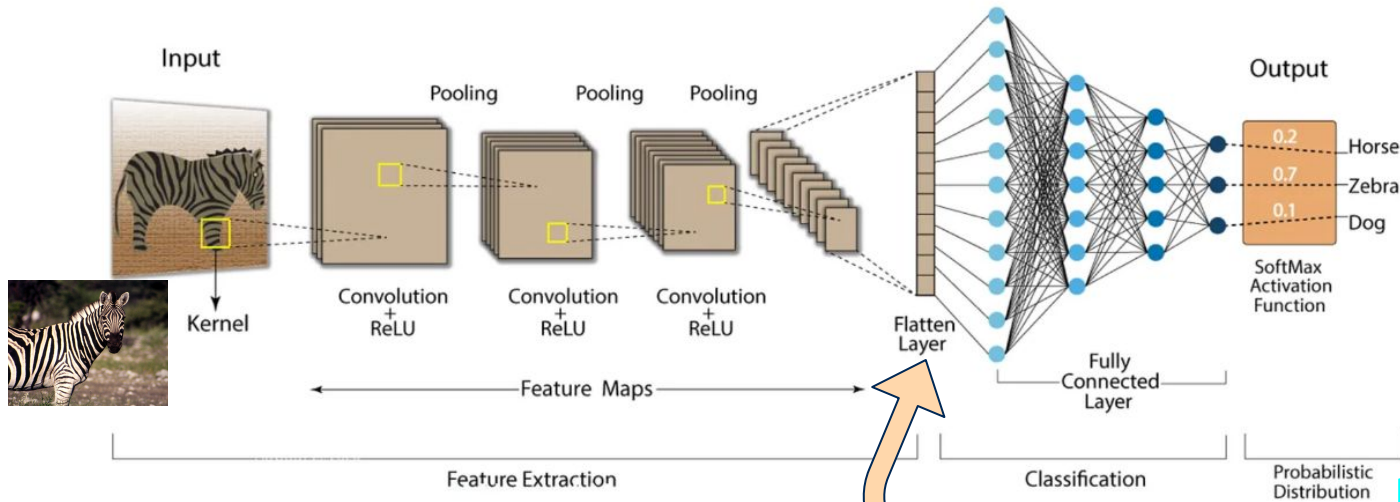




Distance-Based Training

Maximum Mean Discrepancy - MMD

Smola et al. (2007)
Gretton et al. (2012)

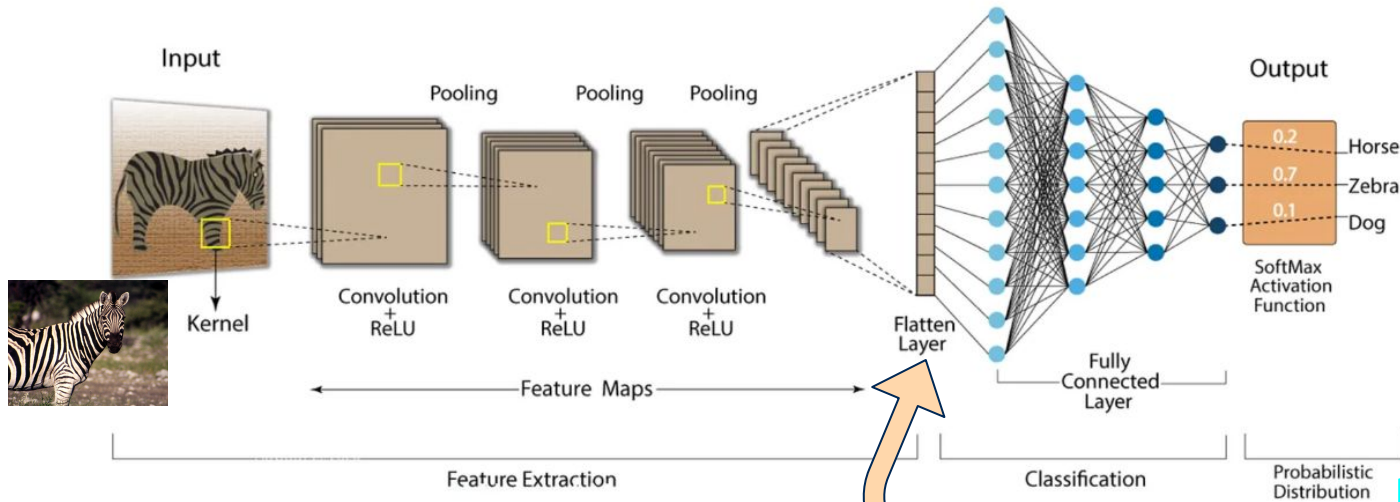




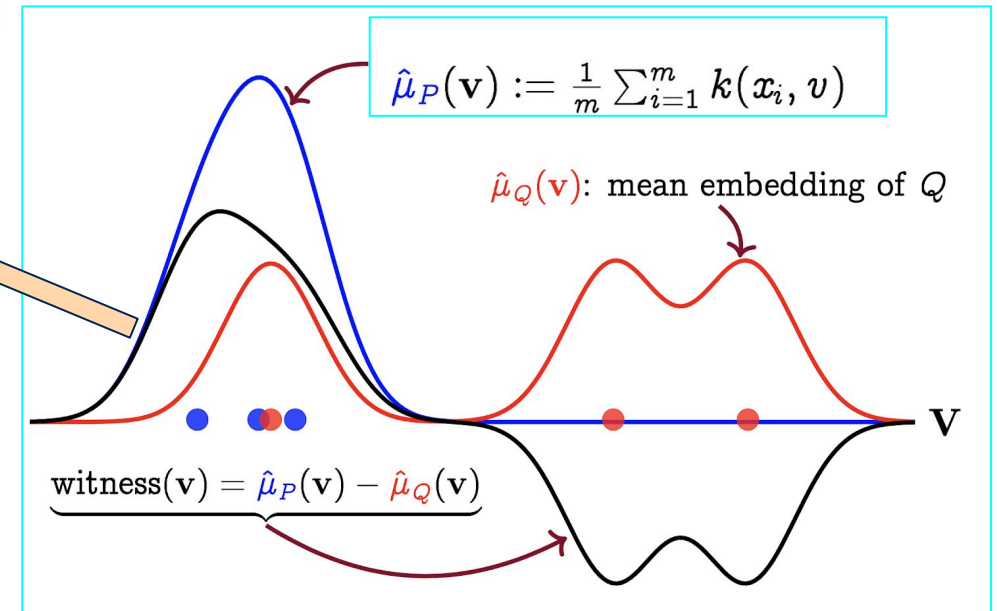
Distance-Based Training

Maximum Mean Discrepancy - MMD

Smola et al. (2007)
Gretton et al. (2012)



Domain adaptation method of your choice is applied to the latent embedding.
What happens then?...Stay tuned!





How do visualise what DA is doing?

Dimensionality reduction techniques

**We need to be able to examine the
multi-dimensional laten (embedding) space
i.e. go from ND to 2D or 3D.**

tSNE

Isomap

Umap



How do visualise what DA is doing?

Dimensionality reduction techniques

We need to be able to examine the
multi-dimensional laten (embedding) space
i.e. go from ND to 2D or 3D.

tSNE

Isomap

Umap



Original



Perplexity: 100
Step: 5,000

t-distributed Stochastic Neighbor Embedding

A nonlinear dimensionality reduction technique that preserves local similarities by modeling high-dimensional data into low-dimensional space using probability distributions. It excels at visualizing clusters but can distort global structure and is sensitive to parameters like perplexity. van der Maaten et al. 2008.



How do visualise what DA is doing?

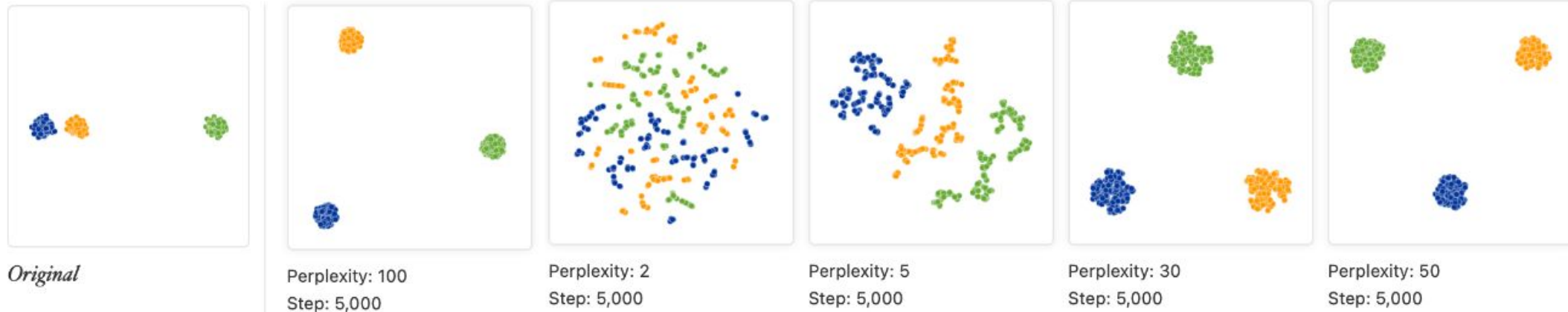
Dimensionality reduction techniques

We need to be able to examine the
multi-dimensional laten (embedding) space
i.e. go from ND to 2D or 3D.

tSNE

Isomap

Umap





How do visualise what DA is doing?

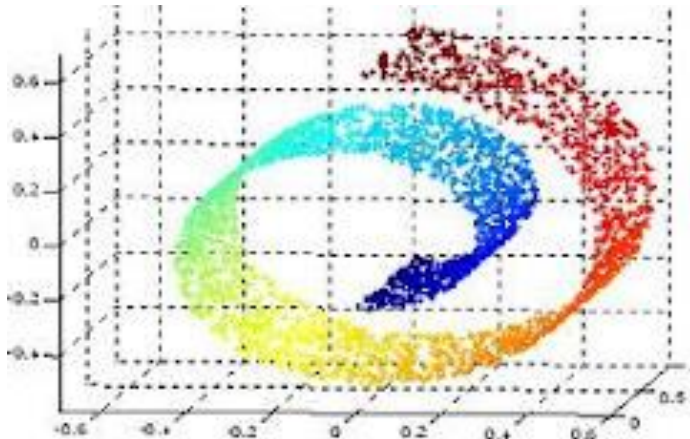
Dimensionality reduction techniques

**We need to be able to examine the
multi-dimensional laten (embedding) space
i.e. go from ND to 2D or 3D.**

tSNE

Isomap

Umap



Isometric Mapping

Isomap is a manifold learning method that preserves global geometry by computing geodesic distances along a graph connecting nearest neighbors. It is useful for capturing the underlying structure of data that lies on a nonlinear manifold. Tenenbaum et al. 2000.



How do visualise what DA is doing?

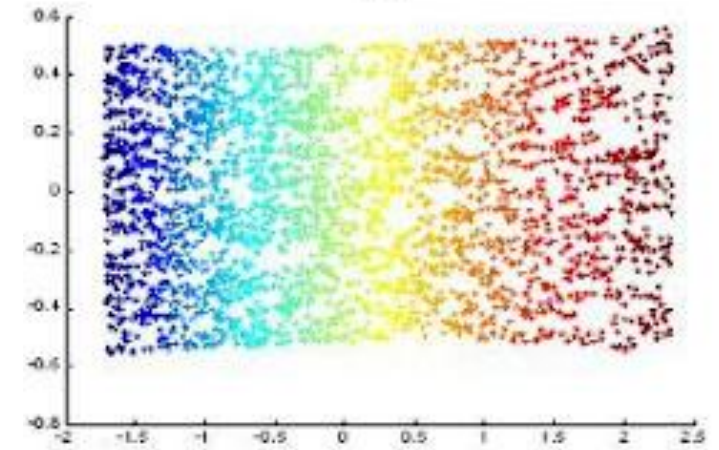
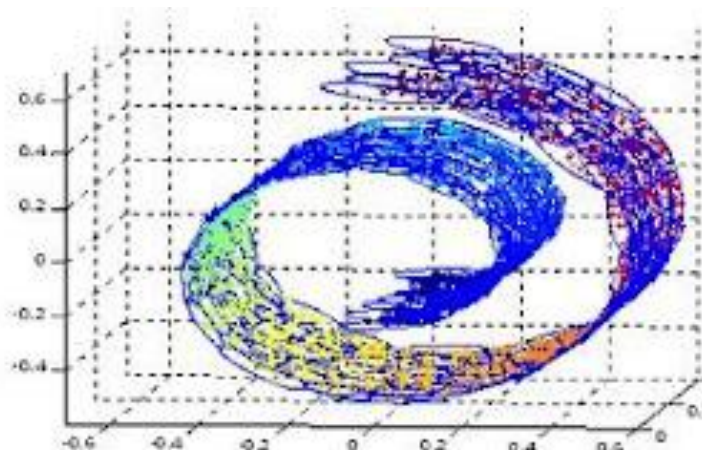
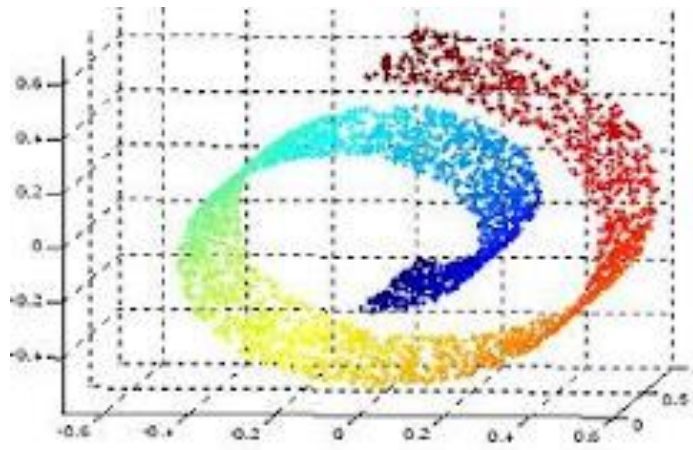
Dimensionality reduction techniques

We need to be able to examine the
multi-dimensional laten (embedding) space
i.e. go from ND to 2D or 3D.

tSNE

Isomap

Umap





How do visualise what DA is doing?

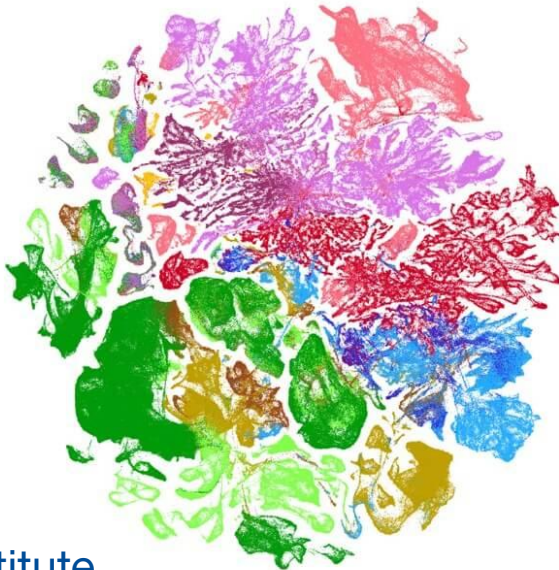
Dimensionality reduction techniques

We need to be able to examine the
multi-dimensional laten (embedding) space
i.e. go from ND to 2D or 3D.

tSNE

Isomap

Umap



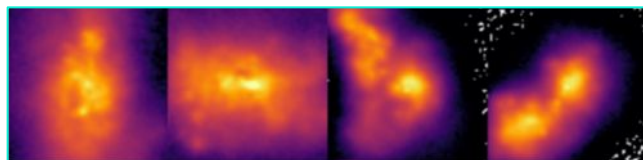
*~4 million single-cell
transcriptomes from
adult mouse brain
labeled by source
brain region
represented by a
UMAP (Yao Z. et al.
2023, bioRxiv).*

Uniform Manifold Approximation and Projection

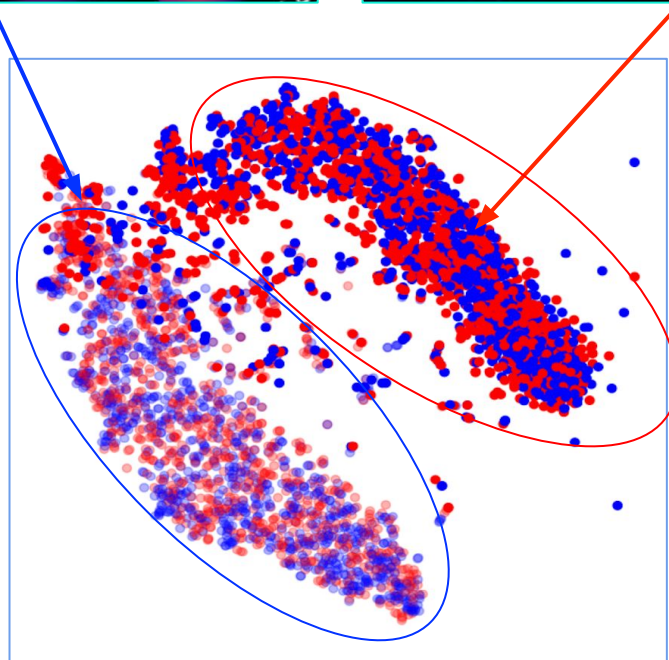
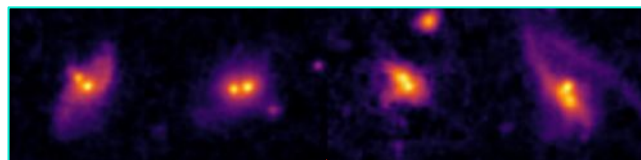
UMAP is a fast, scalable nonlinear dimensionality reduction technique that preserves both local and some global structure using fuzzy topological representations. It often outperforms t-SNE in preserving global relationships and is more suitable for large datasets. McInnes et al. 2020.

Latent Space Alignment

Source - Illustris



Target - SDSS observations

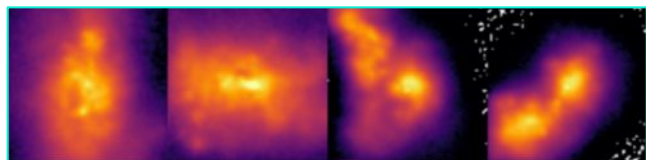


← This is how the network sees the data.
2D representation of network's latent space.

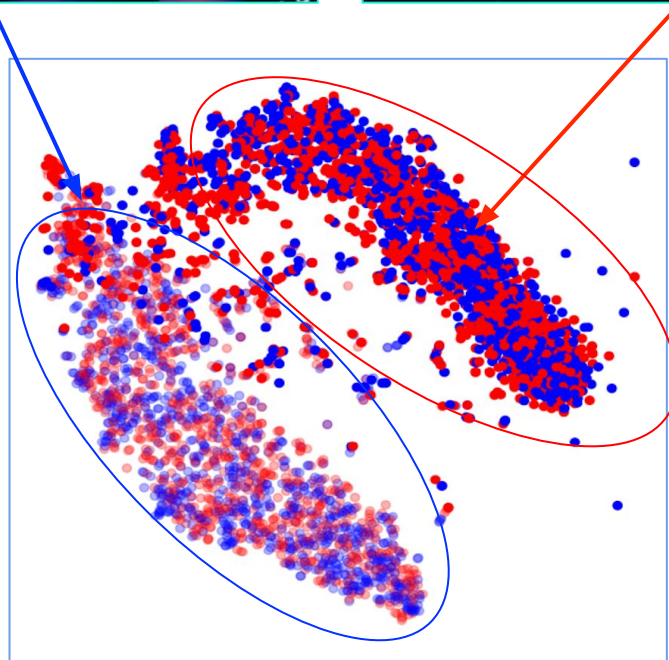
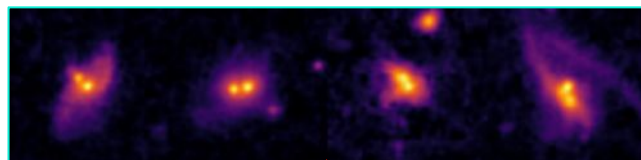
Ćiprijanović et al. 2020.
Ćiprijanović et al. 2021.

⚙️ Latent Space Alignment

Source - Illustris



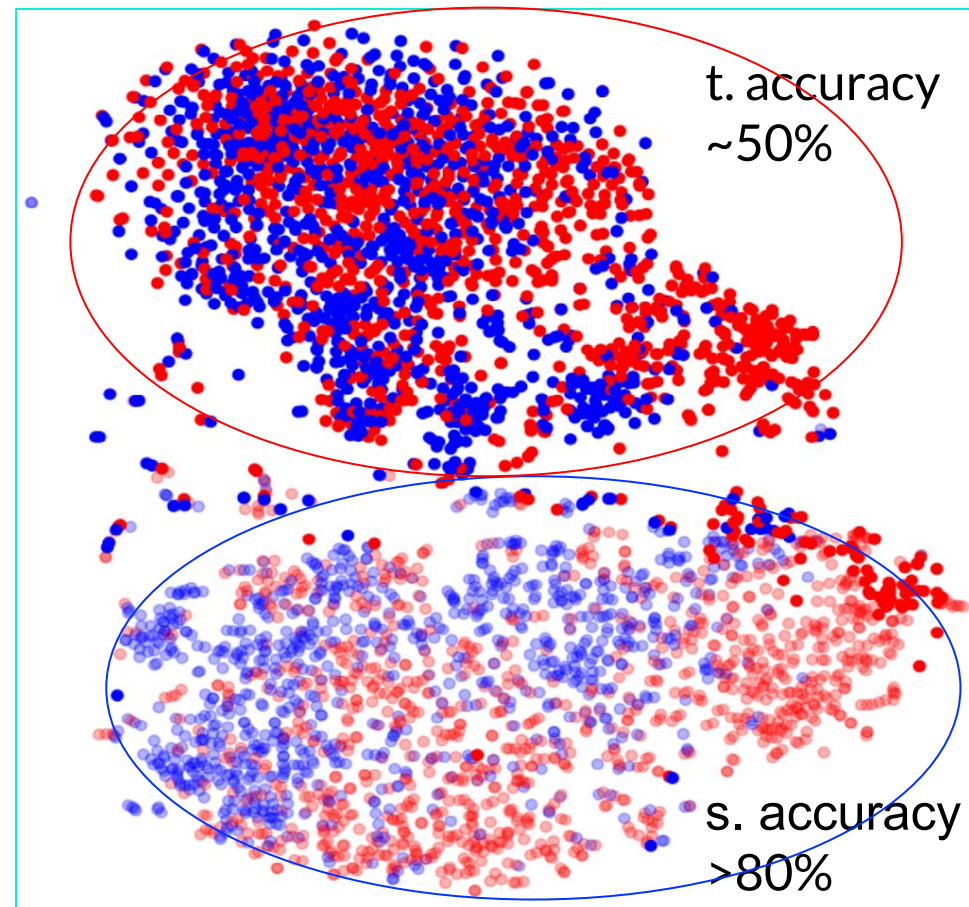
Target - SDSS observations



Ćiprijanović et al. 2020.
Ćiprijanović et al. 2021.



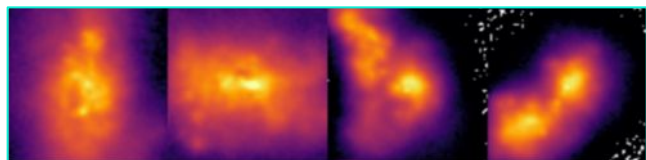
Regular Training



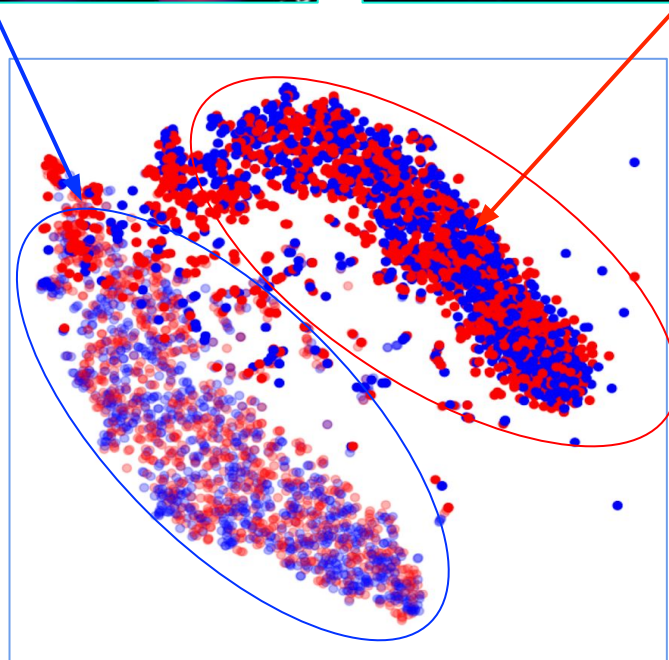
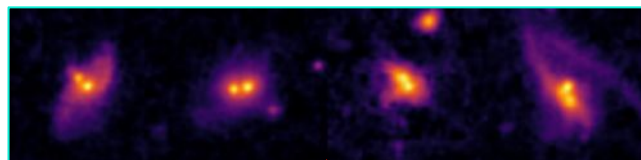


Latent Space Alignment

Source - Illustris



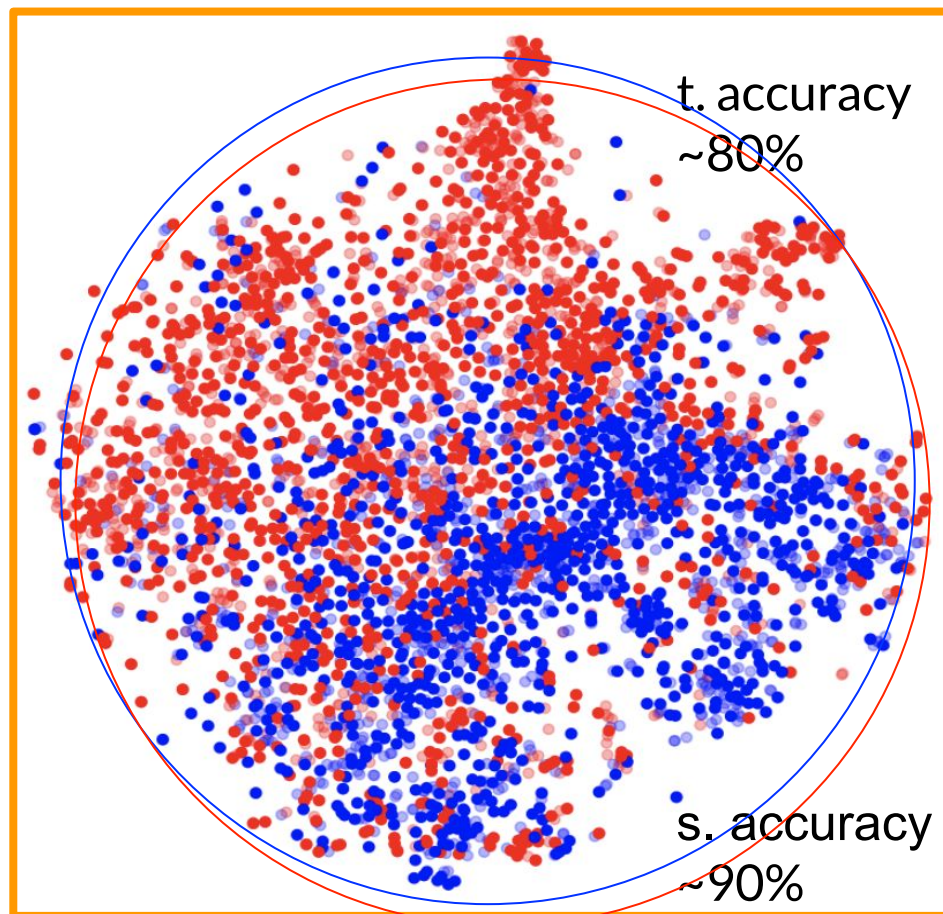
Target - SDSS observations



Ćiprijanović et al. 2020.
Ćiprijanović et al. 2021.

Domain Adaptation

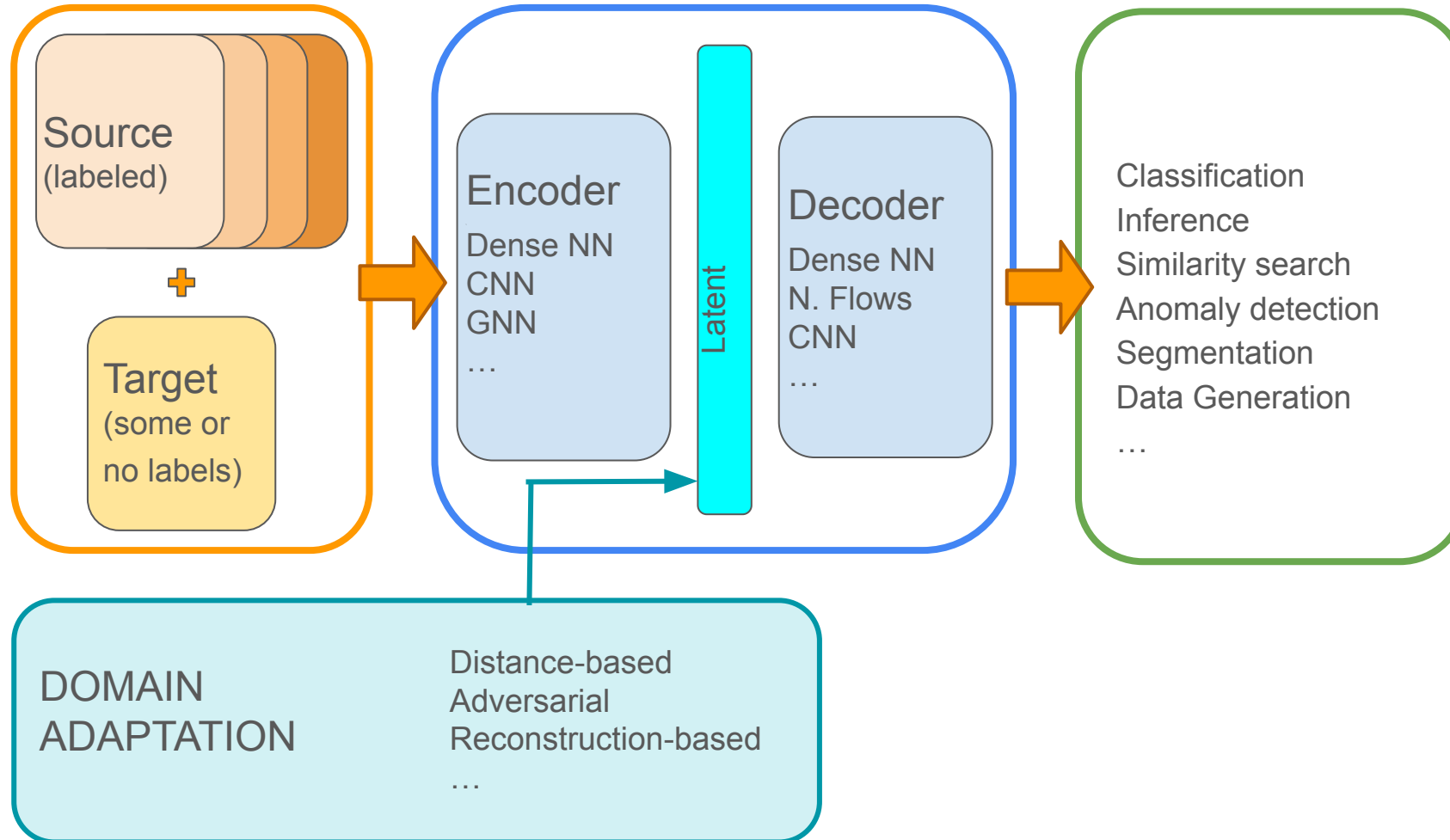
Up to 30% increase!





Where can we use it?

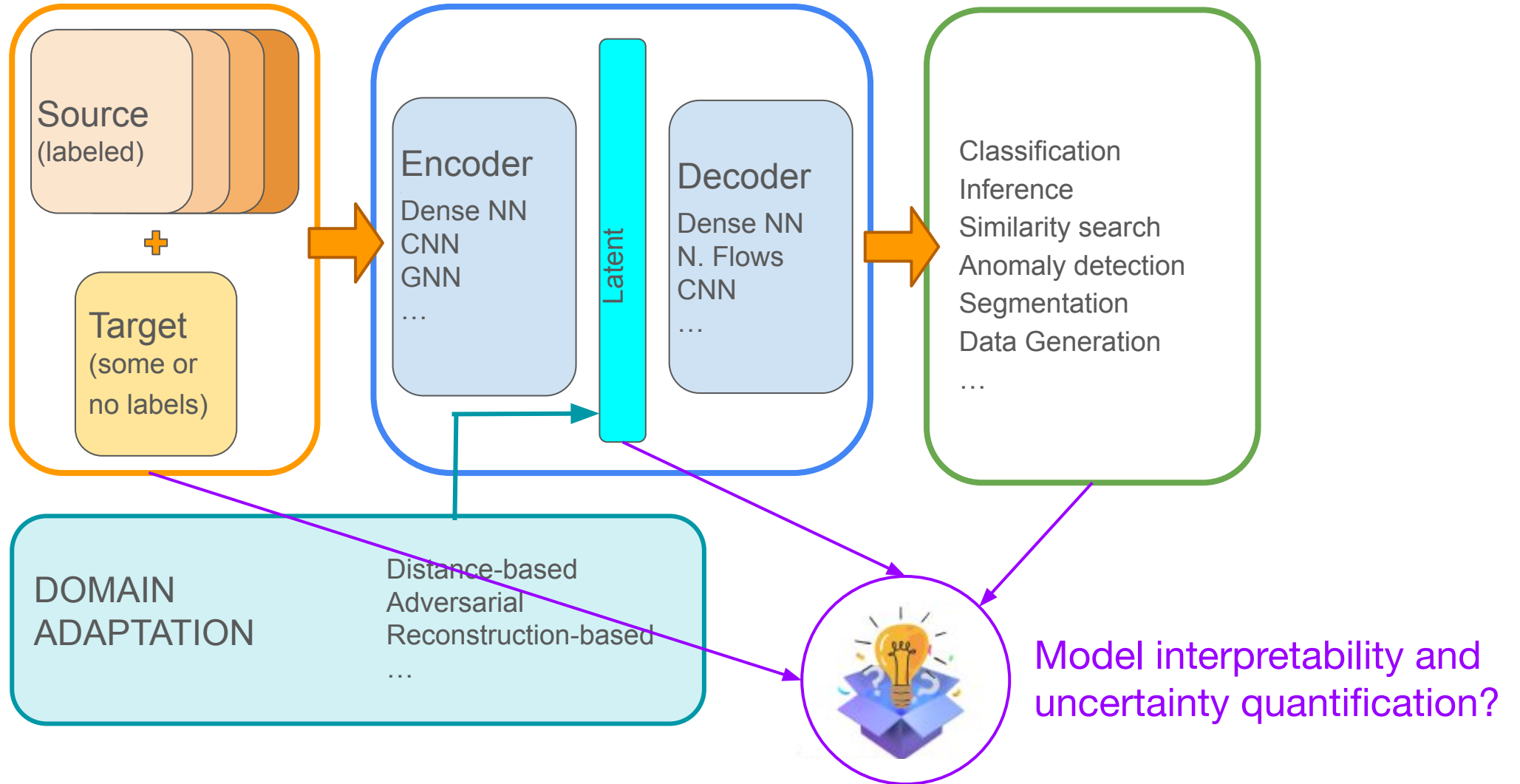
Any data, any task, any problem (within reason!)





Where can we use it?

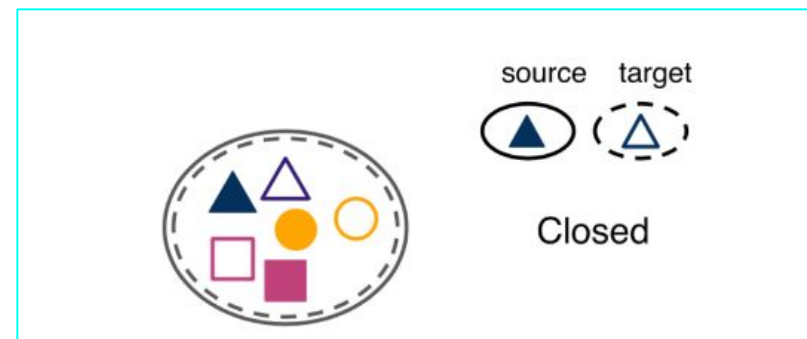
Any data, any task, any problem (within reason!)





Limitations and things to be careful about

Types of domain overlap

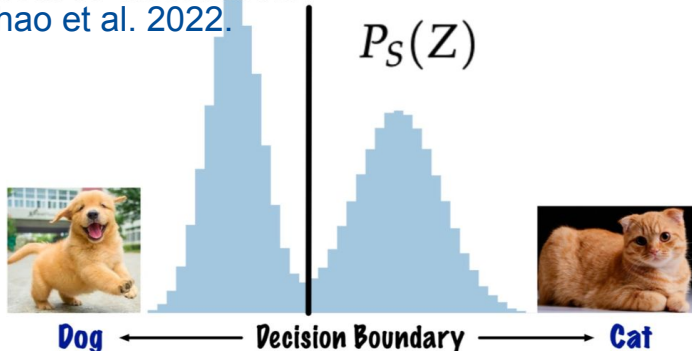


Limitations and things to be careful about

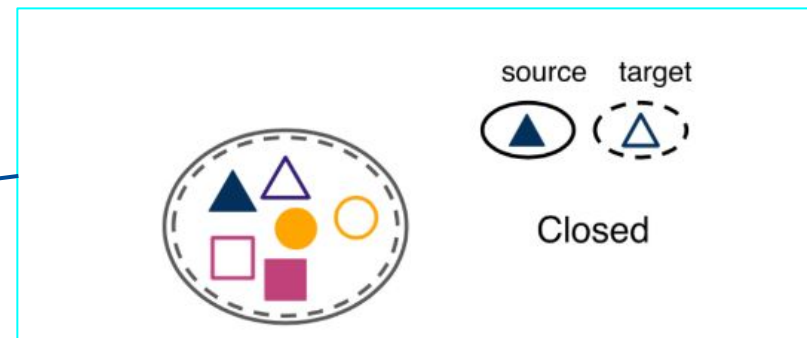
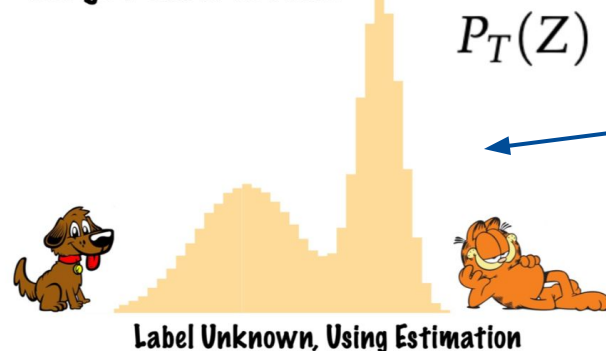
Types of domain overlap

Source distribution

Zhao et al. 2022.



Target distribution

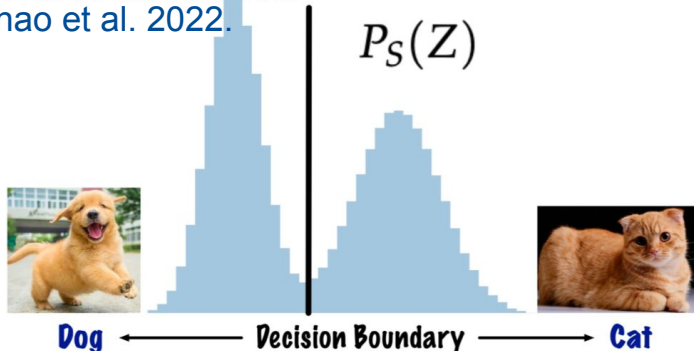


Limitations and things to be careful about

Types of domain overlap

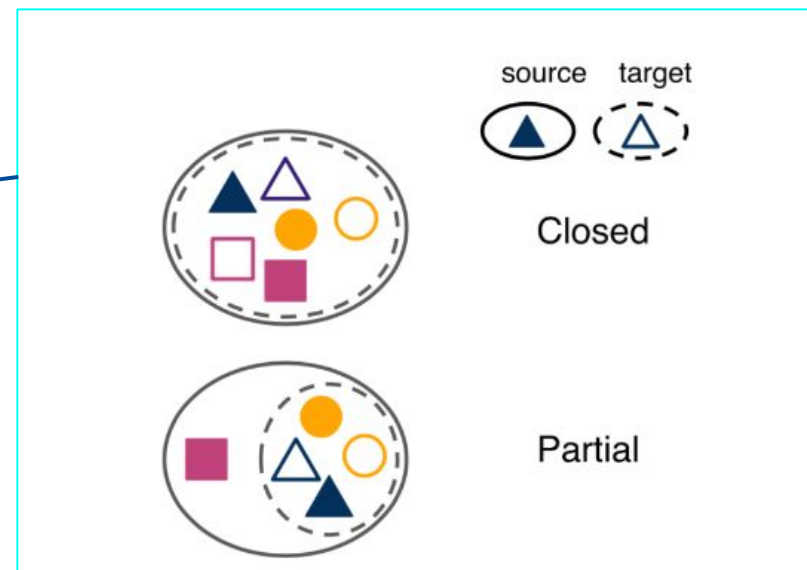
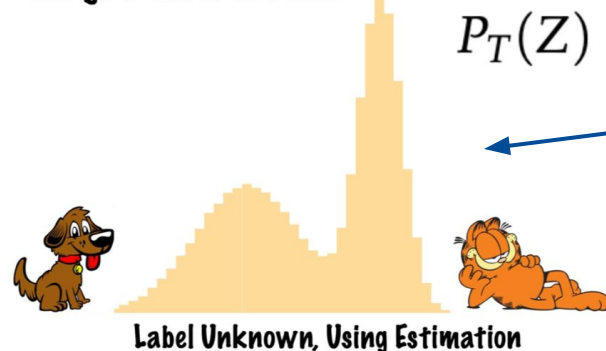
Source distribution

Zhao et al. 2022.



"Bird"

Target distribution

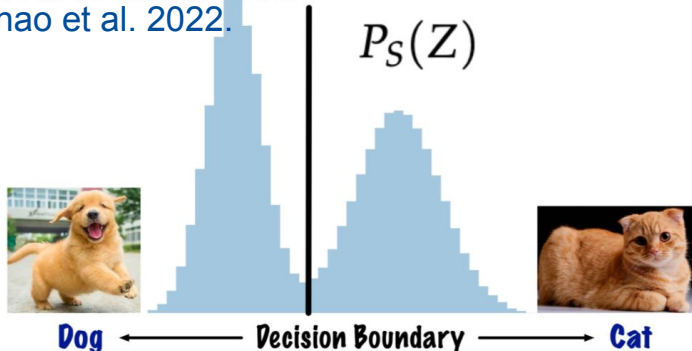


Limitations and things to be careful about

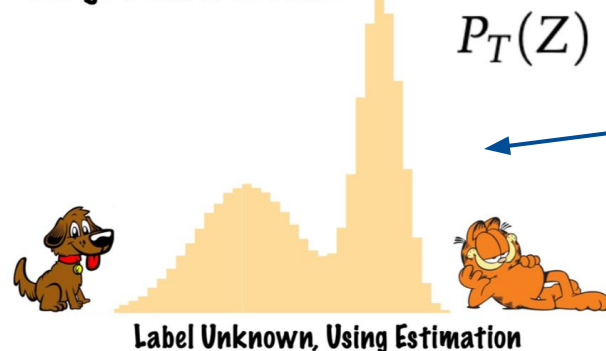
Types of domain overlap

Source distribution

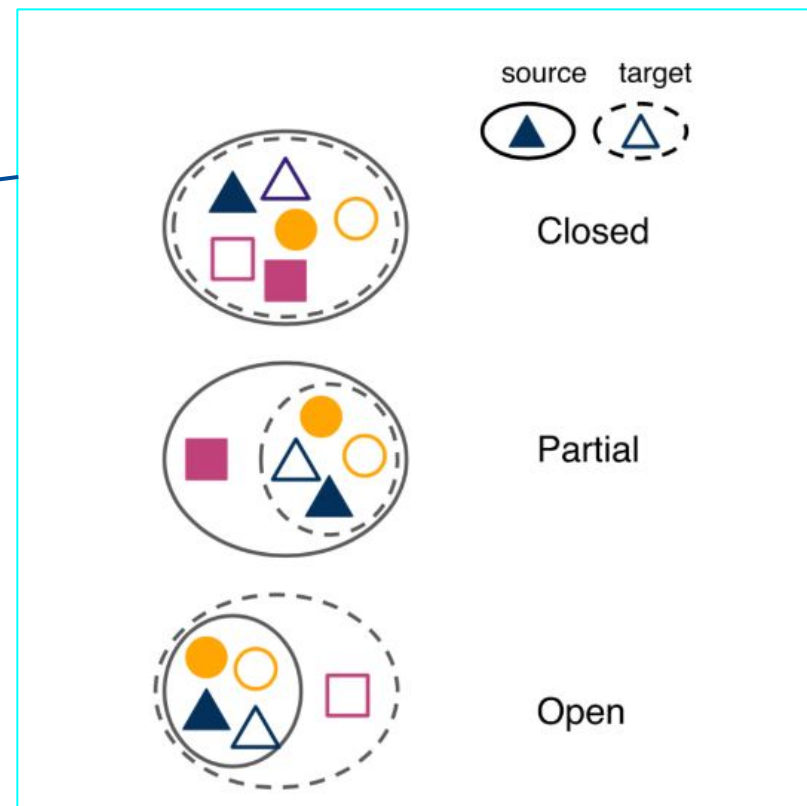
Zhao et al. 2022.



Target distribution



"Bird"

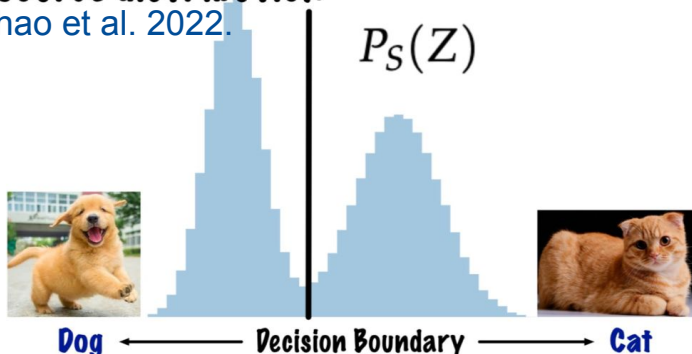


Limitations and things to be careful about

Types of domain overlap

Source distribution

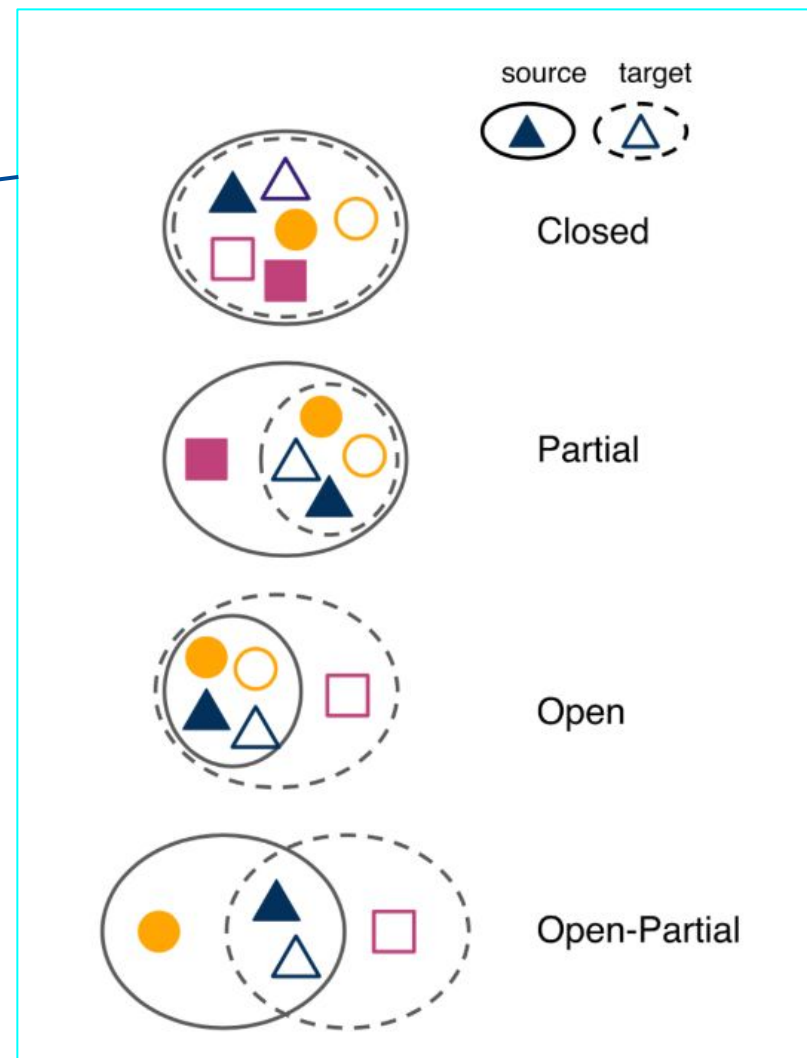
Zhao et al. 2022.



Target distribution



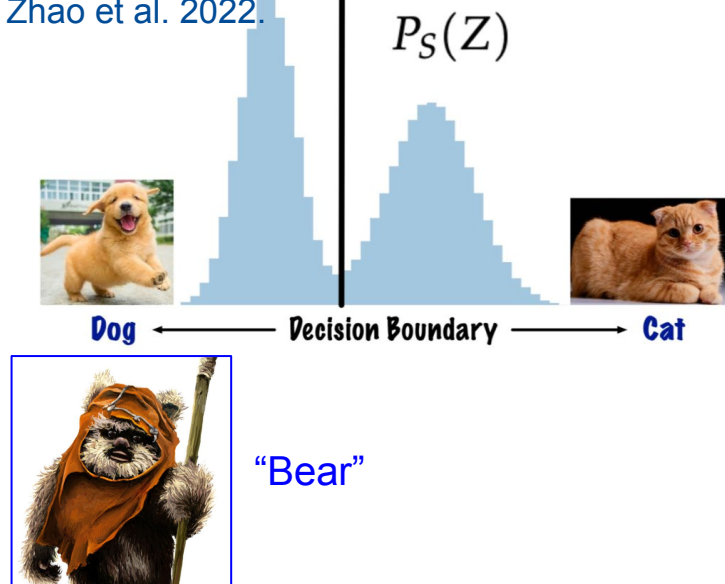
Label Unknown, Using Estimation



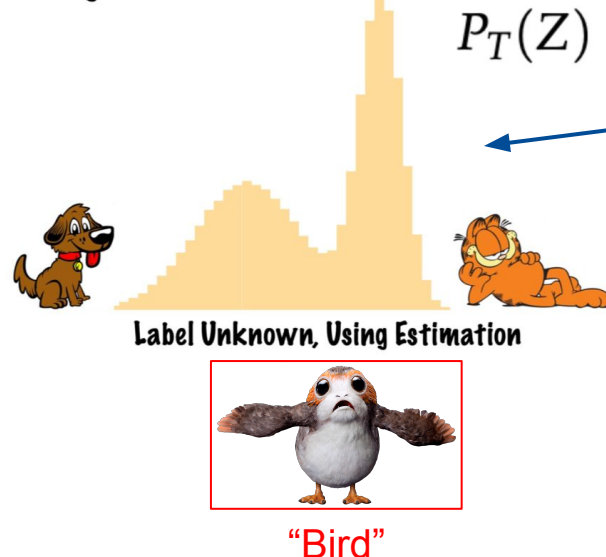
Limitations and things to be careful about

Types of domain overlap

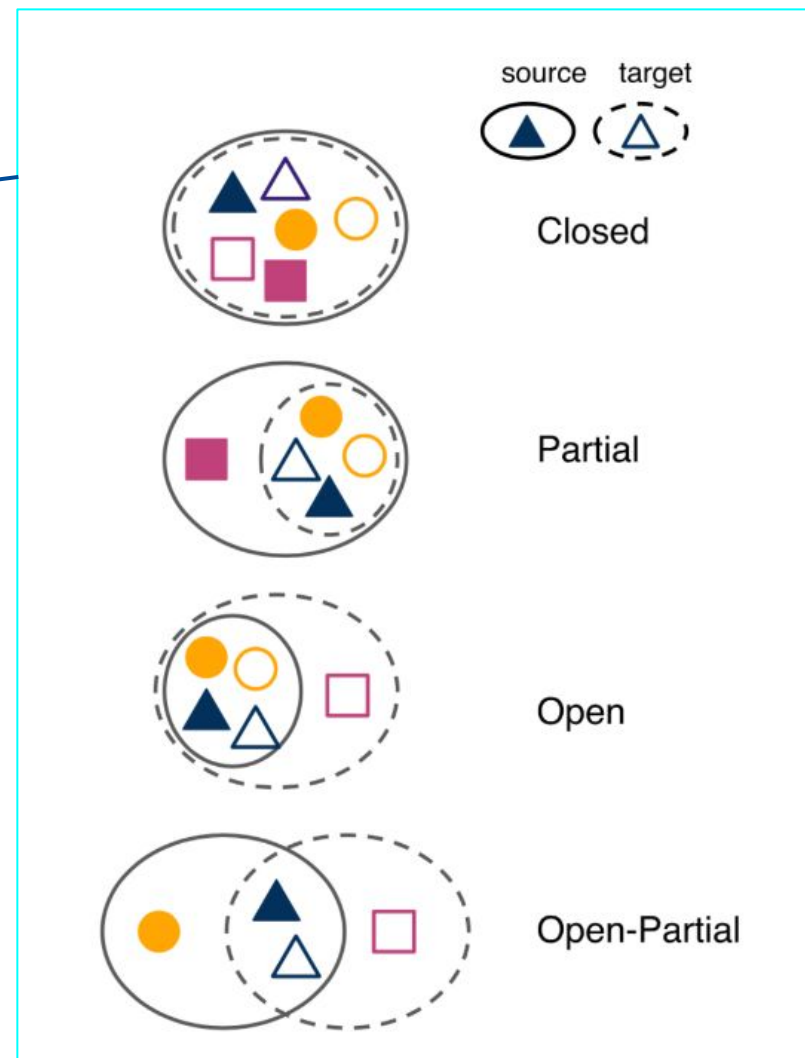
Source distribution
Zhao et al. 2022.



Target distribution



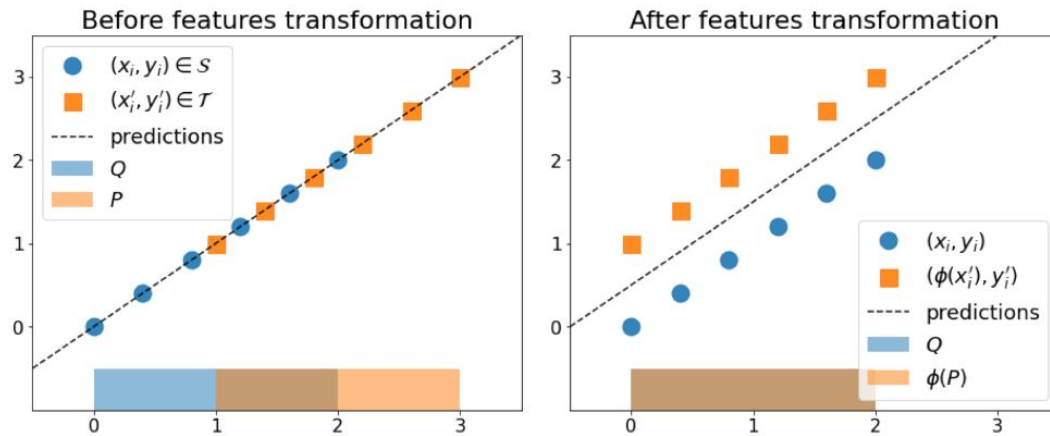
- We can have extra classes present in only one or both of the datasets!
- We might not even know about it!
 - Mutual classes should overlap.
 - Non-overlapping classes should not be aligned with anything - not all DA methods can do this!





Limitations for regression problems

Be careful if prior ranges of parameters being inferred do not match across domains!



de Mathelin et al. 2021.

Fig. 1. Negative transfer caused by feature-based methods. Matching Q and P is detrimental here, as the shift between the two distributions is informative with respect to the task.

Feature-based algorithms (distance or adversarial based) cannot be used if prior ranges don't match.

Only is differences between the domains is related to parameters not being inferred (noise, pixel scale etc).

Instance-based algorithms are necessary
- reweighting importance of labeled instances in the overlapping region.

⚙️ Domain Adaptation Problems Quiz !

Scenario 1:

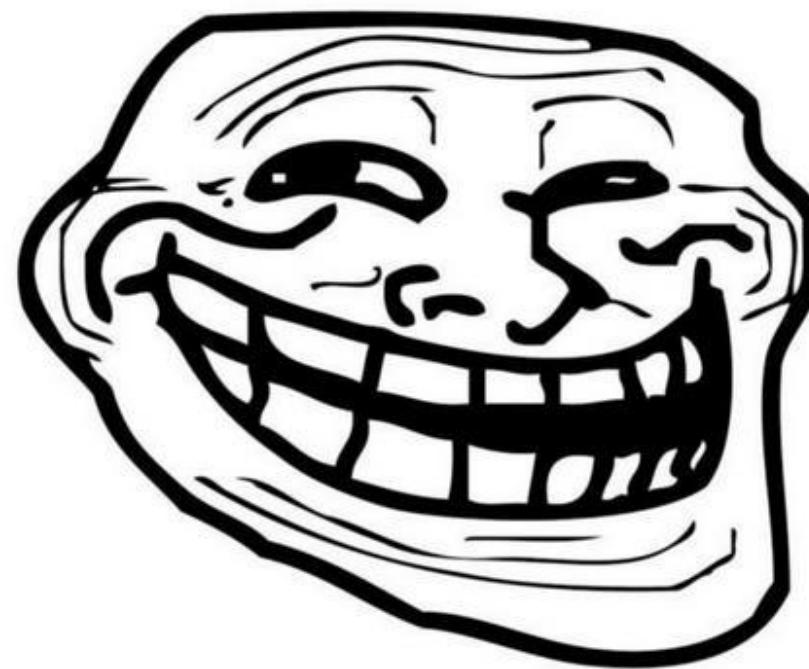
You apply a domain adaptation technique to align galaxy images from two telescopes. After adaptation, your model performs **worse** on the source data and loses its ability to distinguish spiral and elliptical galaxies. What likely went wrong?

Learning rate too small

Overfitting to source data

Too aggressive domain alignment

Answer:



problem?

⚙️ Domain Adaptation Problems Quiz !

Scenario 1:

You apply a domain adaptation technique to align galaxy images from two telescopes. After adaptation, your model performs **worse** on the source data and loses its ability to distinguish spiral and elliptical galaxies. What likely went wrong?

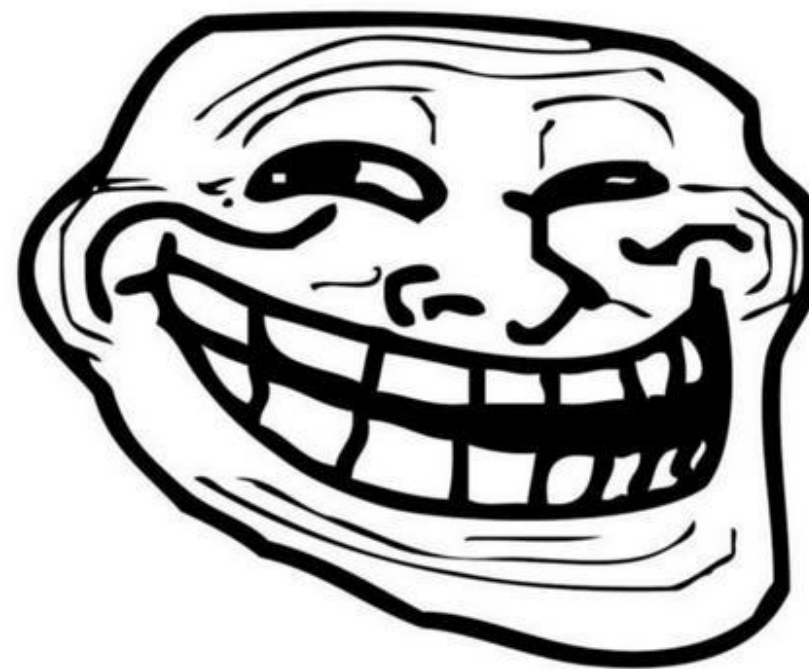
Learning rate too small

Overfitting to source data

Too aggressive domain alignment

Answer:

This is a classic **over-alignment** issue: when features are overly forced to match across domains, the model can discard **discriminative features** that are important for the task. This is especially risky in scientific domains where small differences matter, or domain overlap is small. Careful balance between alignment and task supervision is crucial.



problem?

⚙️ Domain Adaptation Problems Quiz !

Scenario 2:

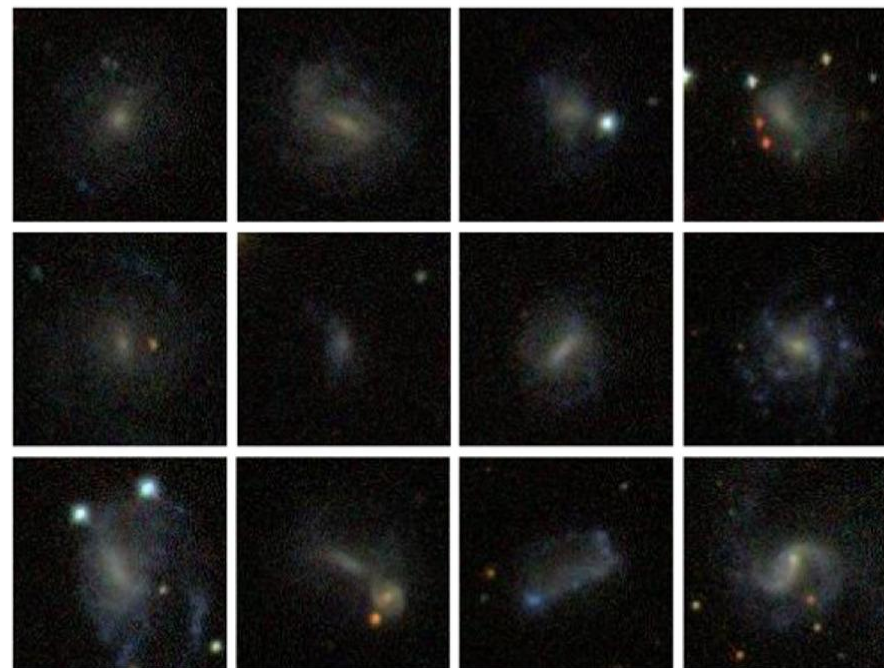
You trained a classifier on old survey data where low-surface brightness galaxies are rare, but in the new LSST survey, they dominate. Your adapted model performs poorly on the target domain, despite good feature alignment. What is the most likely cause?

We didn't account for label distribution shift

Feature extractor wasn't pre-trained

Training didn't converge

Answer:



Domain Adaptation Problems Quiz !

Scenario 2:

You trained a classifier on old survey data where low-surface brightness galaxies are rare, but in the new LSST survey, they dominate. Your adapted model performs poorly on the target domain, despite good feature alignment. What is the most likely cause?

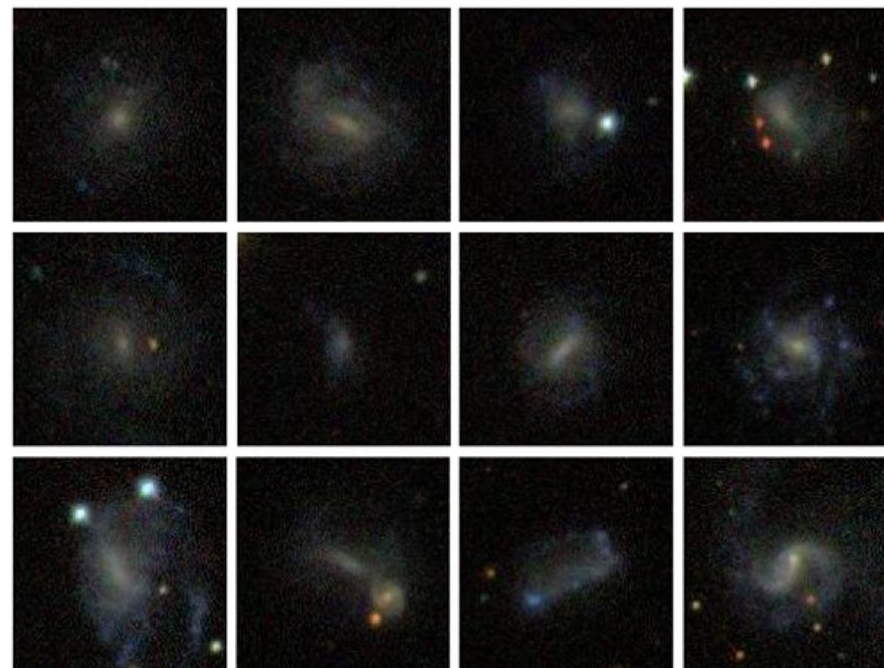
We didn't account for label distribution shift

Feature extractor wasn't pre-trained

Training didn't converge

Answer:

Most domain adaptation methods assume that $P(\mathbf{Y}|\mathbf{X})$ stays the same (covariate shift). If instead $P(\mathbf{Y})$ or $P(\mathbf{Y}|\mathbf{X})$ changes, models trained on one distribution may misclassify dominant classes in the new domain. Solutions like target-prior correction, self-supervision or instance reweighting are needed.



Domain Adaptation Problems Quiz !

Scenario 3:

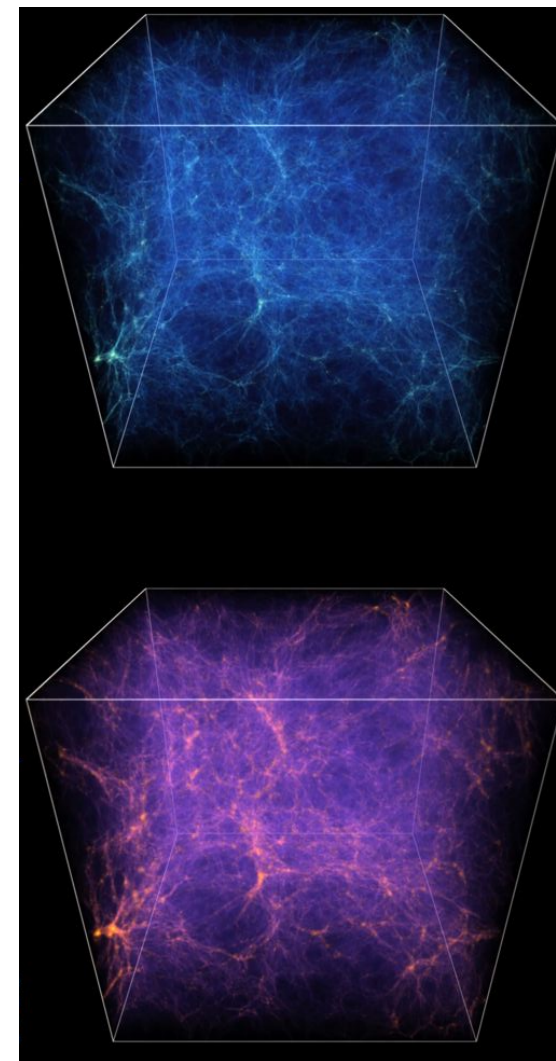
You apply a domain adaptation method assuming two datasets come from distinct domains. However, both were collected under nearly identical conditions, just with different random seeds. Model performance drops after adaptation. What likely went wrong?

You overestimated domain shift

Domain alignment was overdone

Overfitting

Answer:



Domain Adaptation Problems Quiz !

Scenario 3:

You apply a domain adaptation method assuming two datasets come from distinct domains. However, both were collected under nearly identical conditions, just with different random seeds. Model performance drops after adaptation. What likely went wrong?

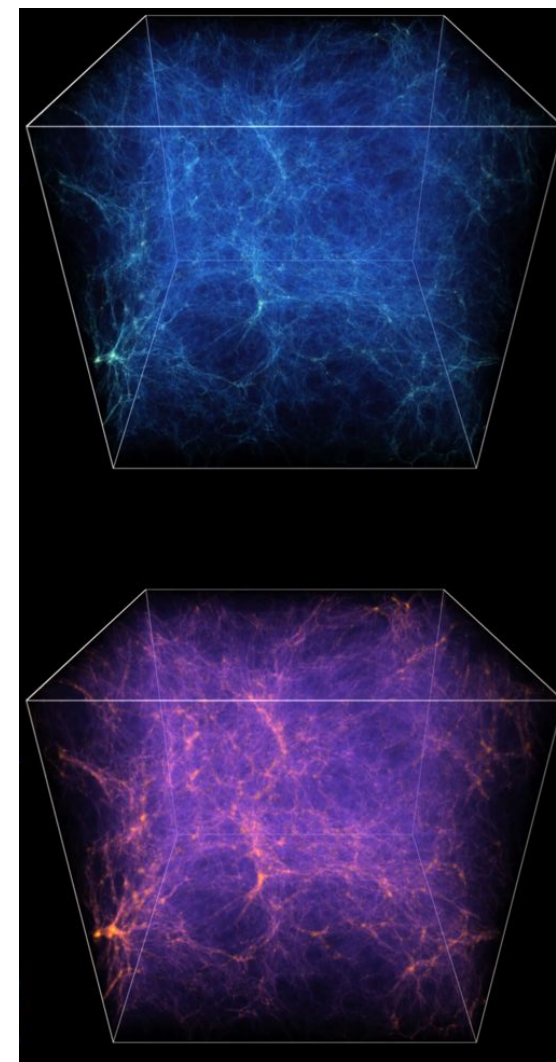
You overestimated domain shift

Domain alignment was overdone

Overfitting

Answer:

Applying domain adaptation when **no meaningful domain gap exists** can **inject noise** or unnecessary transformations into the model. **False domain shift** can degrade performance because of **spurious alignment pressure**. You end up learning representations to solve a non-existent problem, which may reduce task performance and hurt generalization.



Useful Resources:

- [“A Kernel Two-Sample Test” Gretton et al. 2012., Video 1, Video 2](#)
- [“Domain Adversarial Training of Neural Networks” Ganin et al. 2016.](#)
- [“Deep Reconstruction-Classification Networks for Unsupervised Domain Adaptation” Ghifary et al. 2016.](#)
- [“Interpolating between Optimal Transport and MMD using Sinkhorn Divergences” Feydy et al. 2018](#)

03

Diving Deeper Into Distance-Based Methods

Combining Datasets

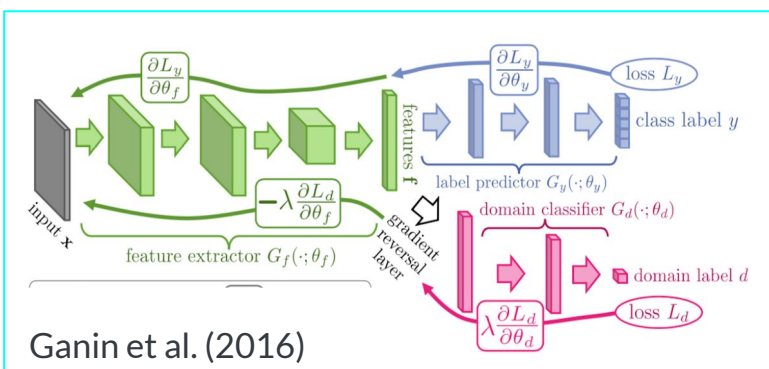
DOMAIN ADAPTATION

Align data distributions in the latent space of the network by forcing the network to **find more robust domain-invariant features**.

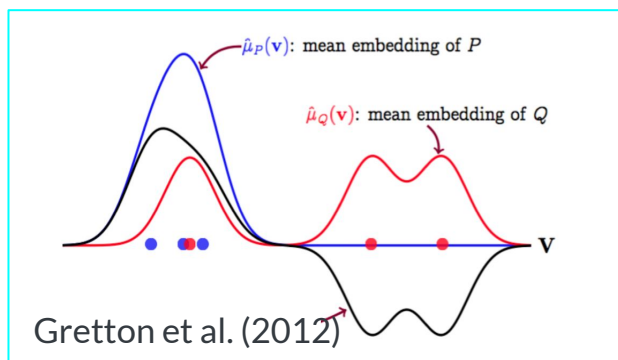
Adversarial

Distance-based
methods

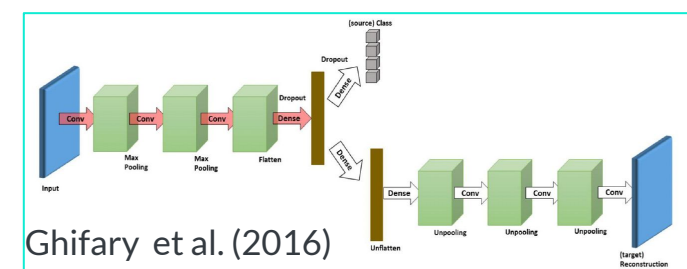
Reconstruction-based



Using domain discriminators to encourage domain confusion through an adversarial objective.



Minimize a distance metric between source and target data.



Data reconstruction as an auxiliary task to ensure feature invariance.



Common types of distance measures

φ -divergences

Csiszar 1963.

- Use convex function φ to measure the ratio of the two distributions.
- Likelihood based models and information theory.

$$D_{\varphi}(\alpha \parallel \beta) = \int \varphi \left(\frac{d\alpha}{d\beta} \right) d\beta$$

- Examples:

Kullback-Liebler (KL) divergence ($\varphi = x \log x$)
Jensen-Shannon (JS) distance
Total Variation (TV) distance
 χ^2 distance

Kernel distances

Gretton 2006. ("king of kernels")

- Kernel-based metric between distributions.
- Measures distance in Reproducing Kernel Hilbert Space (RKHS).
- Used in generative modeling and domain adaptation.

$$\text{MMD}^2(\alpha, \beta) = \mathbb{E}_{x, x' \sim \alpha} [k(x, x')] + \mathbb{E}_{y, y' \sim \beta} [k(y, y')] - 2\mathbb{E}_{x \sim \alpha, y \sim \beta} [k(x, y)]$$

Optimal transport

Monge 1781, Kantorovitch 1942

- Finding the transport plan π that minimized the total cost of moving all the mass (c - the cost of moving a unit mass) from α to β ?

$$\text{OT}(\alpha, \beta) = \inf_{\gamma \in \Pi(\alpha, \beta)} \int c(x, y) d\gamma(x, y)$$

$$c(x, y) = \|x - y\|^p$$

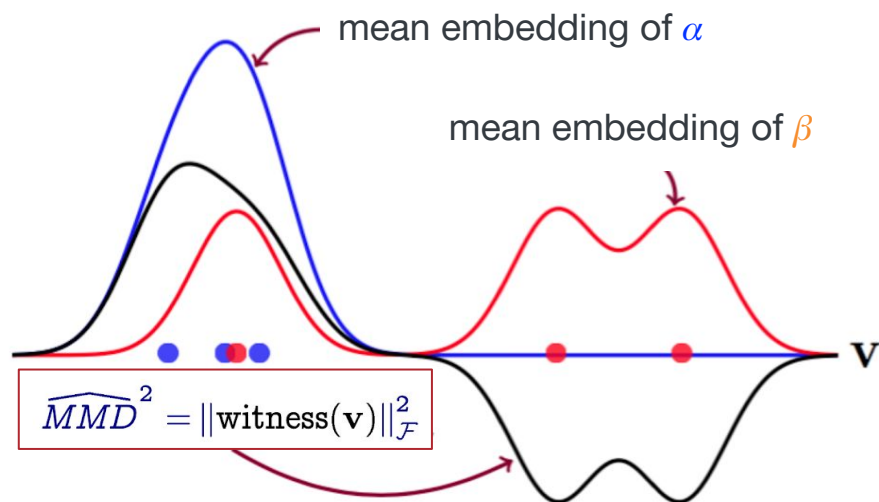
- Examples metrics:

Wasserstein distance ($p=2$)
Earthmovers distance ($p=1$)
Sinkhorn divergence (regularized OT)

Maximum Mean Discrepancy (MMD)

Gaussian (RBF) kernel

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$



$$\text{MMD}^2(\alpha, \beta) = \mathbb{E}_{x, x' \sim \alpha}[k(x, x')] + \mathbb{E}_{y, y' \sim \beta}[k(y, y')] - 2\mathbb{E}_{x \sim \alpha, y \sim \beta}[k(x, y)]$$

Maximum Mean Discrepancy (MMD)

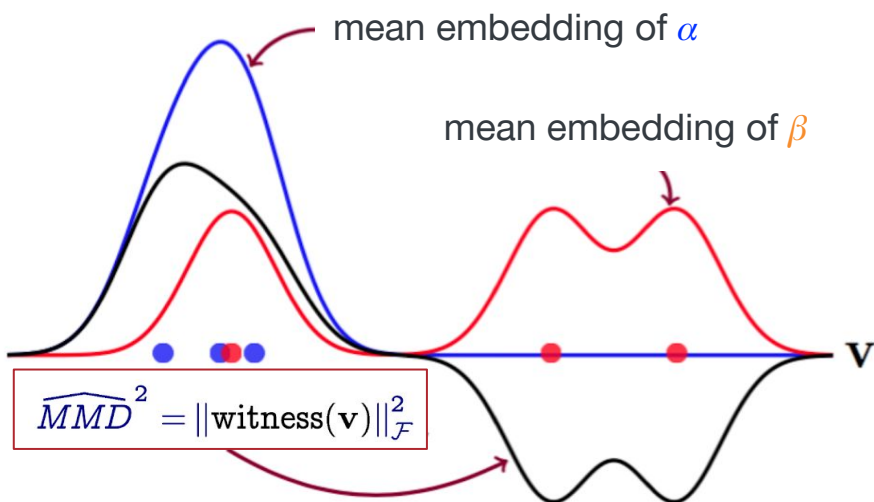
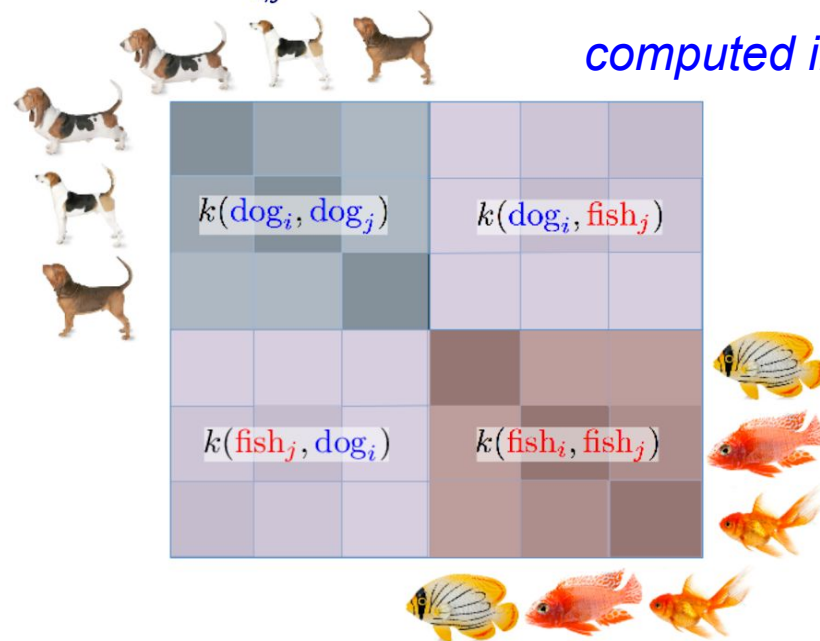
Smola et al. (2007)
Gretton et al. (2012)

Gaussian (RBF) kernel

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

$$\widehat{MMD}^2 = \frac{1}{n(n-1)} \sum_{i \neq j} k(\text{dog}_i, \text{dog}_j) + \frac{1}{n(n-1)} \sum_{i \neq j} k(\text{fish}_i, \text{fish}_j) - \frac{2}{n^2} \sum_{i,j} k(\text{dog}_i, \text{fish}_j)$$

computed in $O(n^2)$



$$\text{MMD}^2(\alpha, \beta) = \mathbb{E}_{x, x' \sim \alpha} [k(x, x')] + \mathbb{E}_{y, y' \sim \beta} [k(y, y')] - 2\mathbb{E}_{x \sim \alpha, y \sim \beta} [k(x, y)]$$

Maximum Mean Discrepancy (MMD)

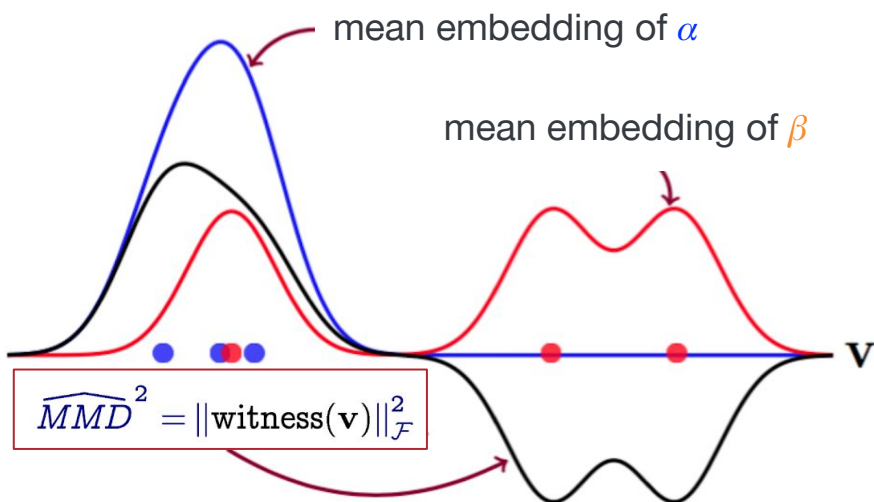
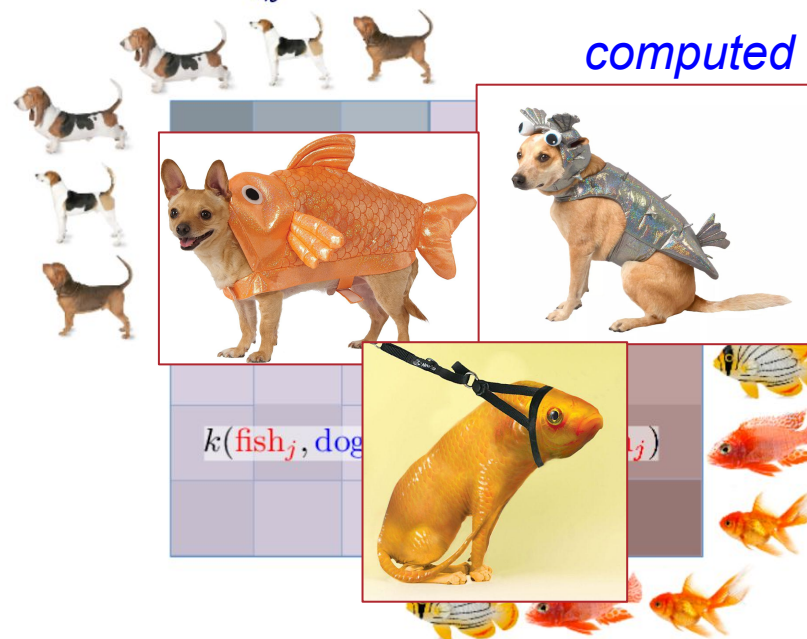
Smola et al. (2007)
Gretton et al. (2012)

Gaussian (RBF) kernel

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

$$\widehat{MMD}^2 = \frac{1}{n(n-1)} \sum_{i \neq j} k(\text{dog}_i, \text{dog}_j) + \frac{1}{n(n-1)} \sum_{i \neq j} k(\text{fish}_i, \text{fish}_j) - \frac{2}{n^2} \sum_{i, j} k(\text{dog}_i, \text{fish}_j)$$

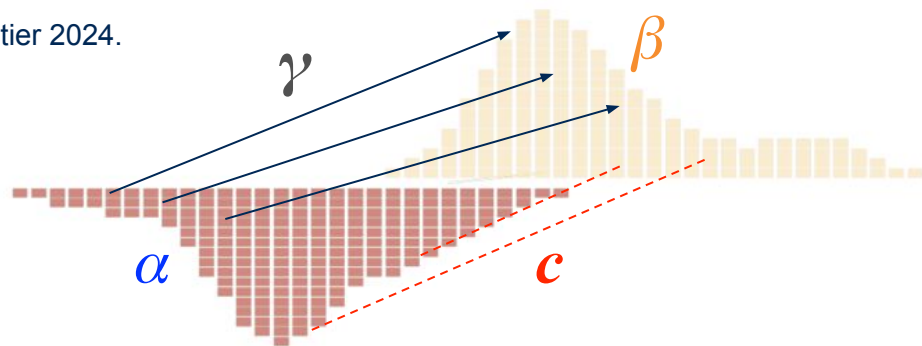
computed in $O(n^2)$



$$\text{MMD}^2(\alpha, \beta) = \mathbb{E}_{x, x' \sim \alpha} [k(x, x')] + \mathbb{E}_{y, y' \sim \beta} [k(y, y')] - 2\mathbb{E}_{x \sim \alpha, y \sim \beta} [k(x, y)]$$

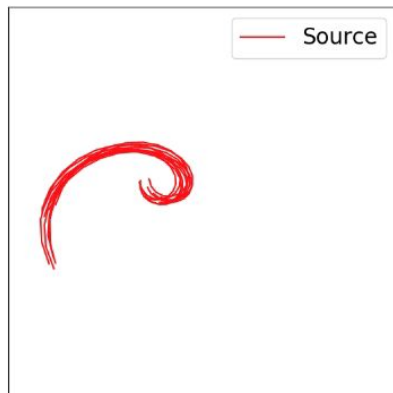
Optimal Transport

Source: Charpentier 2024.

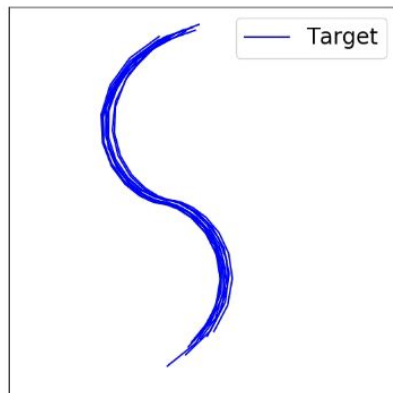


OPTIMAL TRANSPORT:

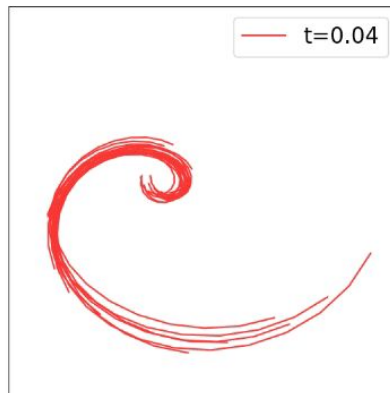
Finding the “most efficient” transportation plan γ that minimizes the total transportation cost c of moving the entire mass from probability α distribution to β .



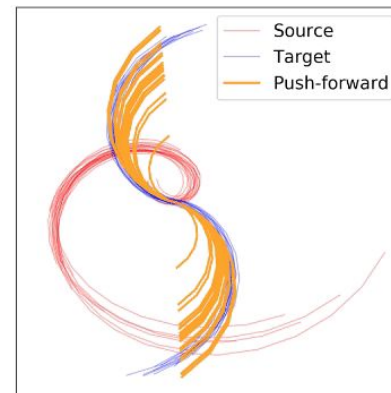
Source data:
Swiss-roll curves



Target data:
Wave curves



The **geodesic** of the
Mapping



The resulting
push-forward

Source: Zhu et al. 2023.

Optimal Transport

Finding the “most efficient” transportation plan γ that minimizes the total transportation cost c of moving the entire mass from probability α distribution to β .

Optimal Transport

$$\text{OT}(\alpha, \beta) = \inf_{\gamma \in \Pi(\alpha, \beta)} \int c(x, y) d\gamma(x, y)$$

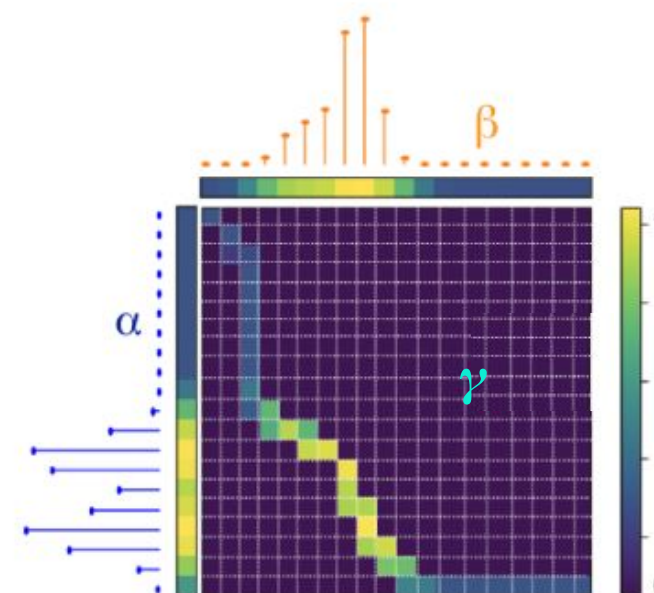
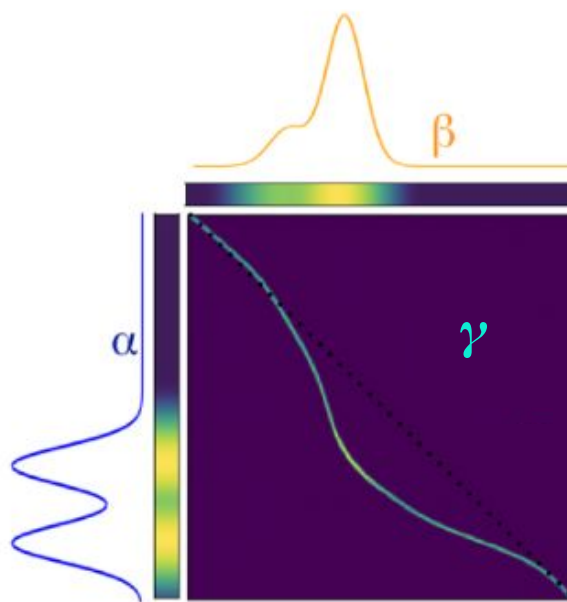
Solving OT exactly becomes impossible for large datasets as it scales as $O(n^3 \log n)$!

Wasserstein distance
p=2

$$W_p(\alpha, \beta) = \left(\inf_{\gamma \in \Pi(\alpha, \beta)} \int \|x - y\|^p d\gamma(x, y) \right)^{1/p}$$

$c(x, y)$ cost of moving a unit of mass from x to y

$\gamma(x, y)$ transport plan (coupling) - how much mass moves from x to y



Genevay 2019.

Optimal Transport

Finding the “most efficient” transportation plan γ that minimizes the total transportation cost c of moving the entire mass from probability α distribution to β .

Optimal Transport

$$\text{OT}(\alpha, \beta) = \inf_{\gamma \in \Pi(\alpha, \beta)} \int c(x, y) d\gamma(x, y)$$

Solving OT exactly becomes impossible for large datasets as it scales as $O(n^3 \log n)$!

Wasserstein distance
 $p=2$

$$W_p(\alpha, \beta) = \left(\inf_{\gamma \in \Pi(\alpha, \beta)} \int \|x - y\|^p d\gamma(x, y) \right)^{1/p}$$

Entropy Regularization

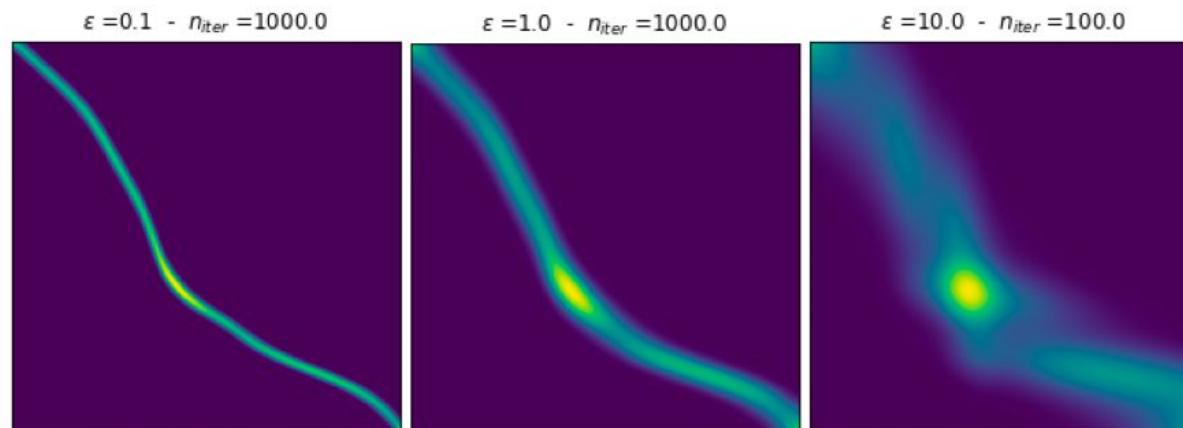
$$OT_{\varepsilon, p}(\alpha, \beta) = \inf_{\gamma \in \Pi(\alpha, \beta)} \int_{\mathcal{X} \times \mathcal{Y}} \|x - y\|^p d\gamma(x, y) - \varepsilon \int_{\mathcal{X} \times \mathcal{Y}} \log \left(\frac{d\gamma}{dx dy} \right) d\gamma(x, y)$$

The **entropy term** smooths the optimization and ensures faster convergence with better numerical properties. ε controls the **strength of the entropy regularization**.

Optimal Transport

Finding the “most efficient” transportation plan γ that minimizes the total transportation cost c of moving the entire mass from probability α distribution to β .

ε smooths the problem to avoid overfitting.



Genevay 2019.

Entropy Regularization

$$OT_{\varepsilon,p}(\alpha, \beta) = \inf_{\gamma \in \Pi(\alpha, \beta)} \int_{\mathcal{X} \times \mathcal{Y}} \|x - y\|^p d\gamma(x, y) - \varepsilon \int_{\mathcal{X} \times \mathcal{Y}} \log \left(\frac{d\gamma}{dx dy} \right) d\gamma(x, y)$$

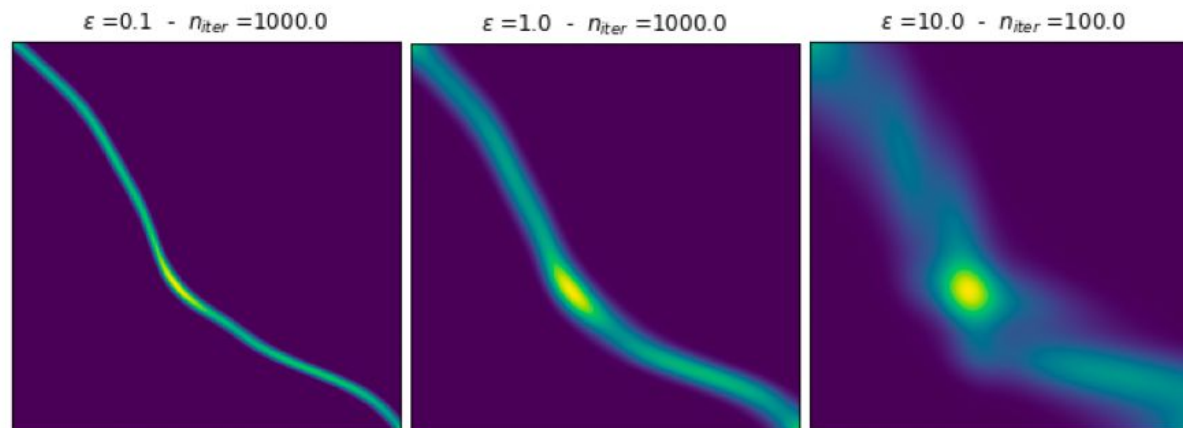
The **entropy term** smooths the optimization and ensures faster convergence with better numerical properties. ε controls the **strength of the entropy regularization**.

Optimal Transport

Finding the “most efficient” transportation plan γ that minimizes the total transportation cost c of moving the entire mass from probability α distribution to β .

ε smooths the problem to avoid overfitting.

But **Entropy Regularization** introduces a bias. $OT_{\varepsilon,p}(\alpha, \alpha) \neq 0$



Genevay 2019.

Entropy Regularization

$$OT_{\varepsilon,p}(\alpha, \beta) = \inf_{\gamma \in \Pi(\alpha, \beta)} \int_{\mathcal{X} \times \mathcal{Y}} \|x - y\|^p d\gamma(x, y) - \varepsilon \int_{\mathcal{X} \times \mathcal{Y}} \log \left(\frac{d\gamma}{dx dy} \right) d\gamma(x, y)$$

The **entropy term** smooths the optimization and ensures faster convergence with better numerical properties. ε controls the **strength of the entropy regularization**.



Sinkhorn Algorithm

Debiased and efficient OT

$$\text{Sink}_\varepsilon(\alpha, \beta) = OT_\varepsilon(\alpha, \beta) - \frac{1}{2} [OT_\varepsilon(\alpha, \alpha) + OT_\varepsilon(\beta, \beta)]$$

Entropic OT between source and target distributions.

Self-distances of source and target (should be zero in classical OT, but not in entropic OT).

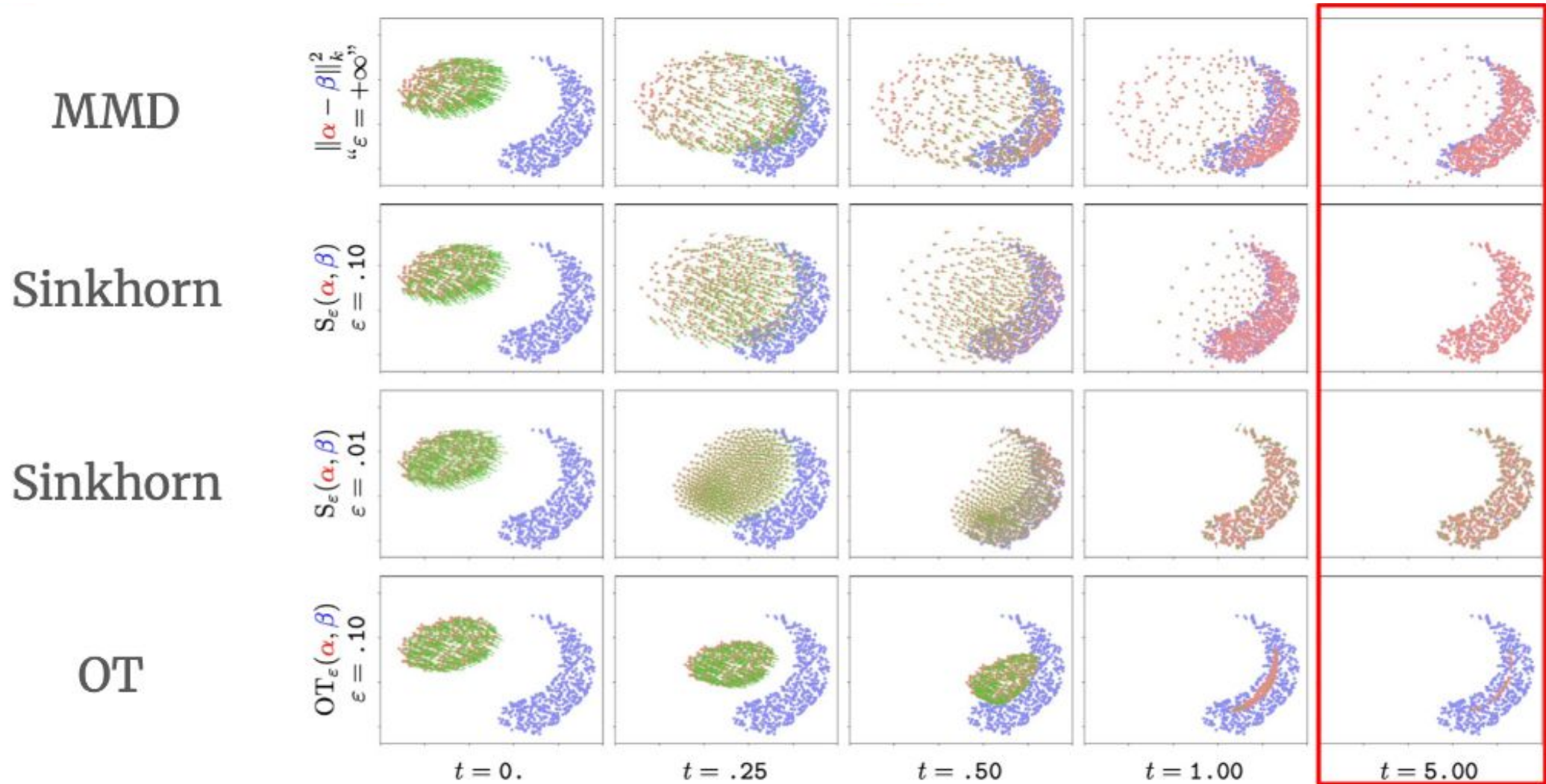
Asymptotic behaviours:

$$\text{Sink}_\varepsilon(\alpha, \beta) \rightarrow \begin{cases} \text{Wasserstein} & \text{as } \varepsilon \rightarrow 0 \\ \text{MMD-like mean-matching} & \text{as } \varepsilon \rightarrow \infty \end{cases}$$

Feydy et al. 2018.

Sinkhorn - best of both worlds

Interpolating between MMD and Optimal Transport (OT)



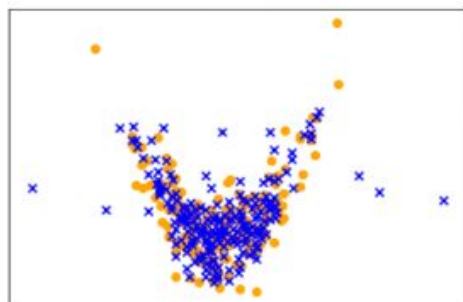
MMD and Optimal Transport (OT)

MMD PROS:

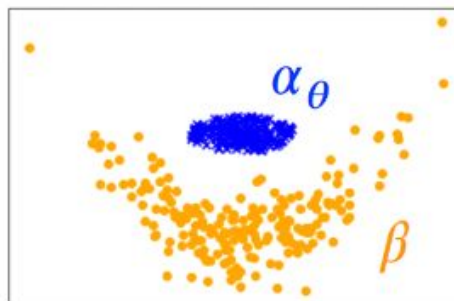
- **Simple and fast** to compute - $O(n^2)$.
- **Differentiable**
- Works well in **high dimensions** (if kernel is well-chosen).
- Captures subtle distributional differences.

CONS:

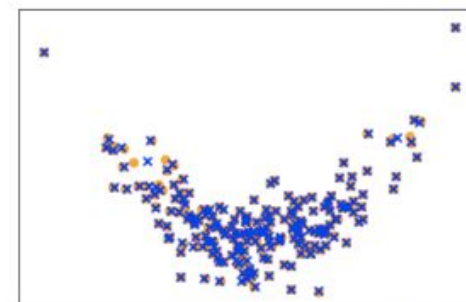
- **Sensitive to kernel choice.**
- Bad in low-density environments.
- Fails to capture **geometric structure**
- Less interpretable.



$$MMD_k - k = - \|\cdot\|_2^{1.5}$$



Initial Setting



$$W_c - c = \|\cdot\|_2^{1.5}$$

OT PROS:

- Captures **geometry of distributions**.
- More **interpretable**: measures "how much mass needs to move".
- Good for structured data (images, point clouds, time series).

CONS:

- **Computationally expensive** $O(n^3 \log(n))$.
- Curse of dimensionality.
- **Non-differentiable** (needs entropic regularization, but this requires tuning). **Sinkhorn to the rescue!**

MMD and Optimal Transport (OT)

MMD PROS:

- **Simple and fast** to compute - $O(n^2)$.
- **Differentiable**
- Works well in **high dimensions** (if kernel is well-chosen).
- Captures subtle distributional differences.

CONS:

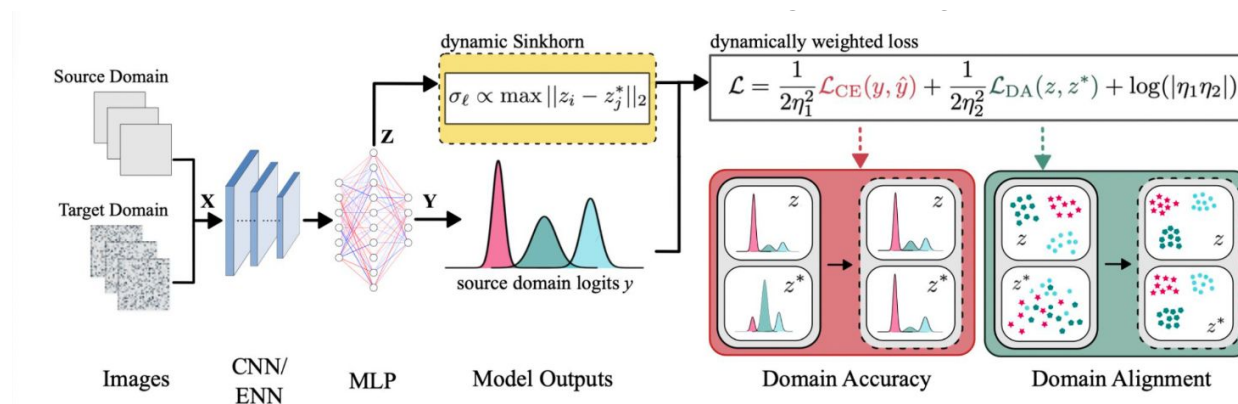
- **Sensitive to kernel choice.**
- Bad in low-density environments.
- Fails to capture **geometric structure**
- Less interpretable.

OT PROS:

- Captures **geometry of distributions**.
- More **interpretable**: measures "how much mass needs to move".
- Good for structured data (images, point clouds, time series).

CONS:

- **Computationally expensive** $O(n^3 \log(n))$.
- Curse of dimensionality.
- **Non-differentiable** (needs entropic regularization, but this requires tuning). **Sinkhorn to the rescue!**



Pandya 2025.

Useful Resources:

- [“A survey of uncertainty in deep neural networks” Gawlikowski et al. 2023.](#)
- [“On the Need to Align Intent and Implementation in Uncertainty Quantification for Machine Learning” Trivedi & Nord 2025.](#)

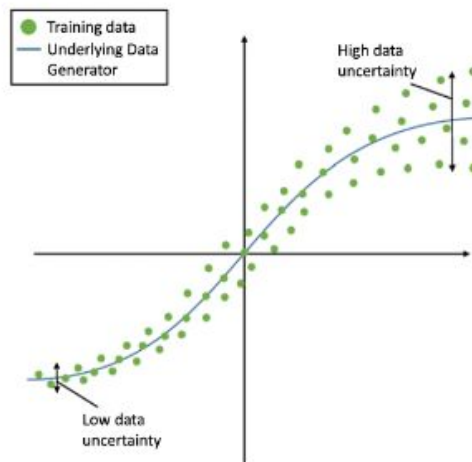
04

Uncertainty Quantification

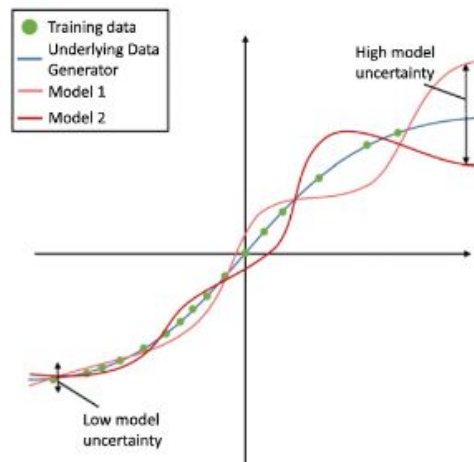


Sources of uncertainties

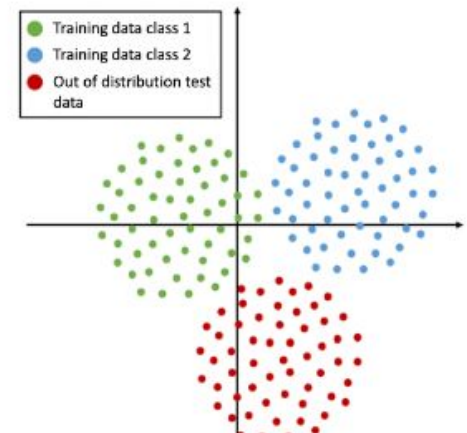
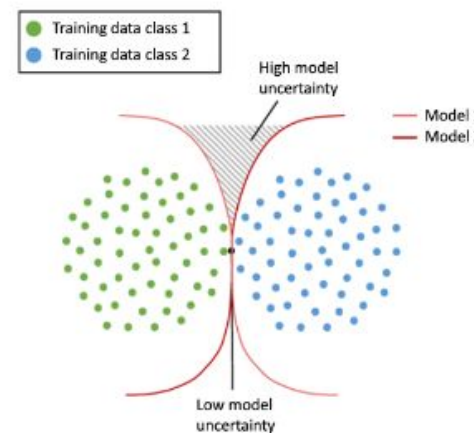
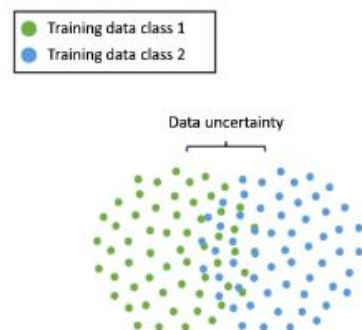
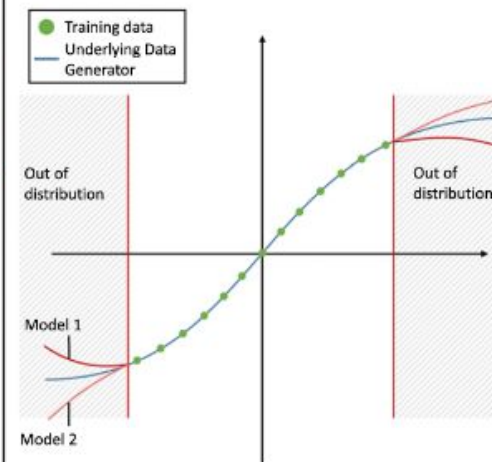
Data



Model



Distributional



Gawlikowski et al. 2023.

Sources of uncertainties

All of this will influence model parameters and model outputs!

Systematic

persistent bias from instruments, calibration, or imperfect theory

Statistical

variation due to limited data

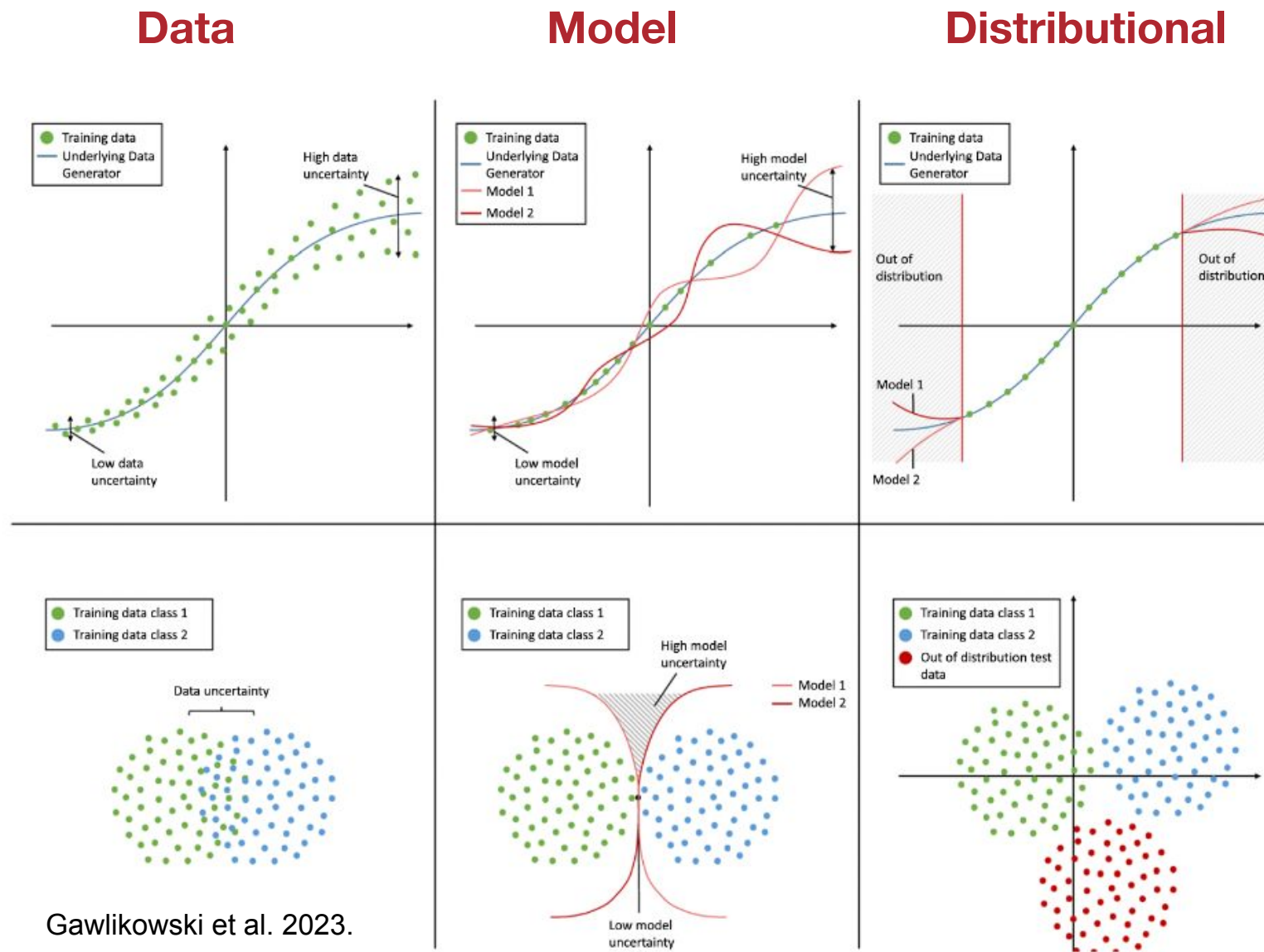
Epistemic

model or data-driven uncertainty

Aleatoric

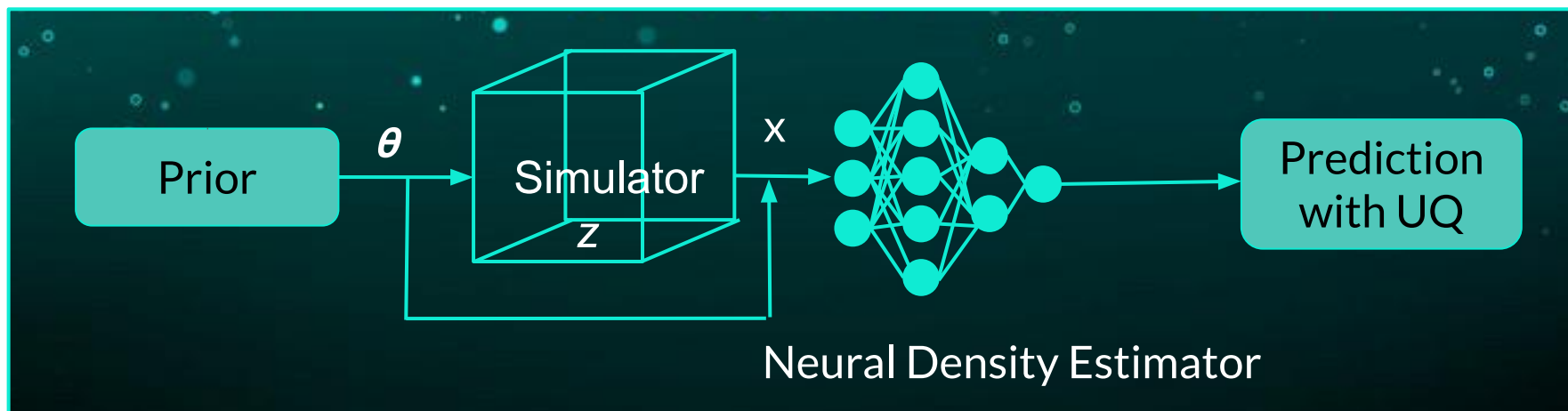
irreducible noise in the data

These vocabularies are not interchangeable!



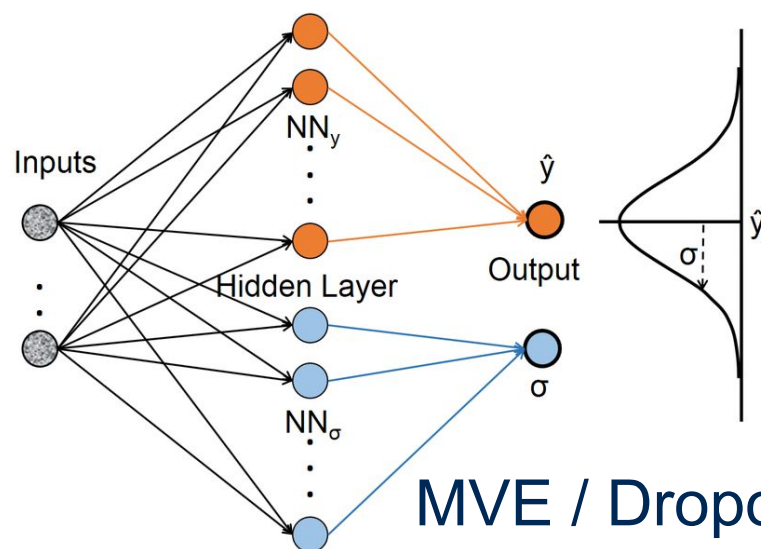
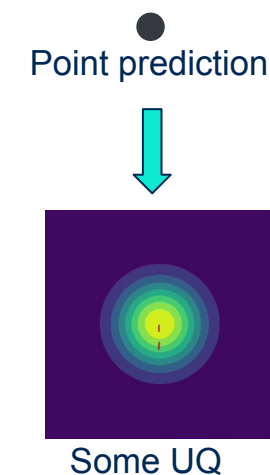
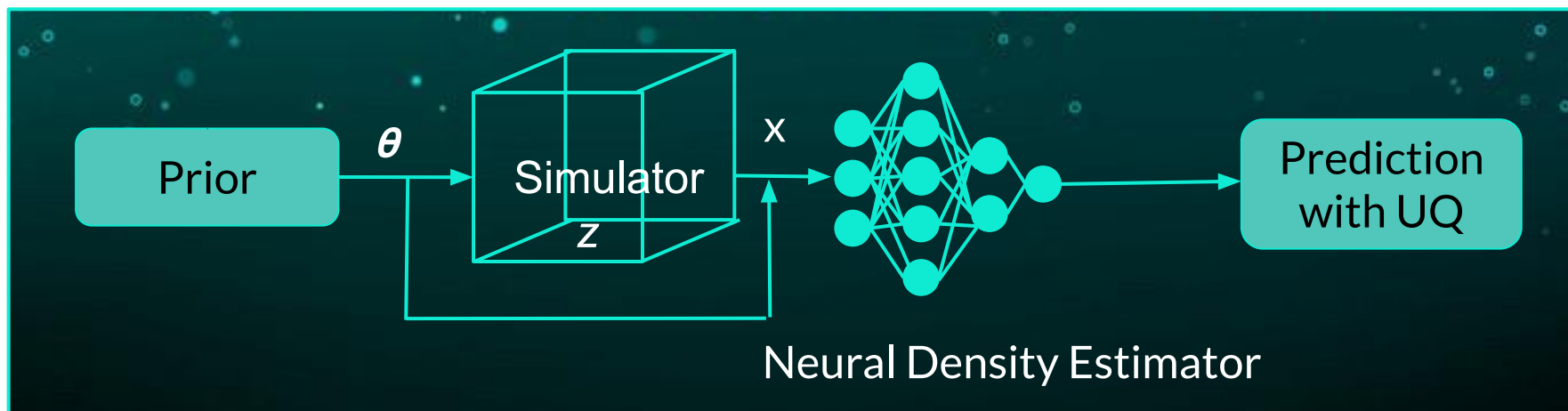
Gawlikowski et al. 2023.

Inference with UQ



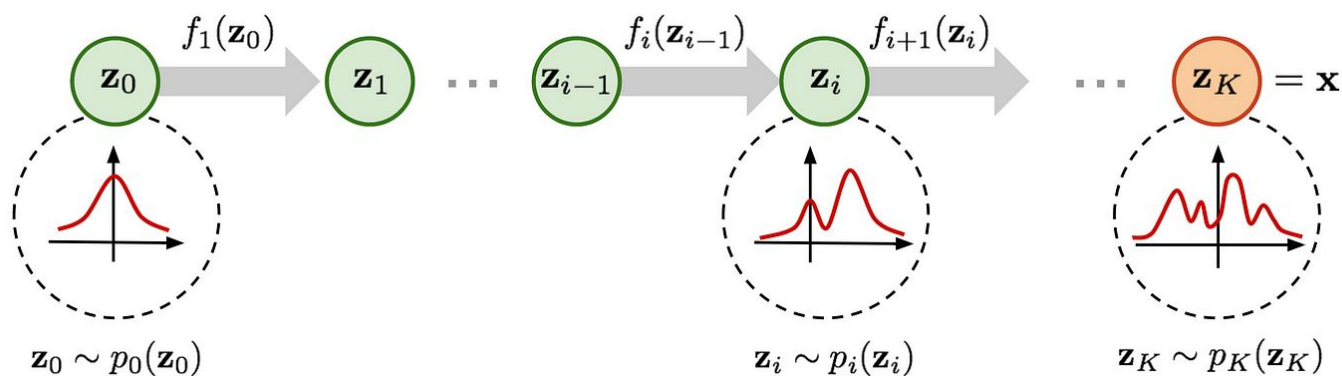
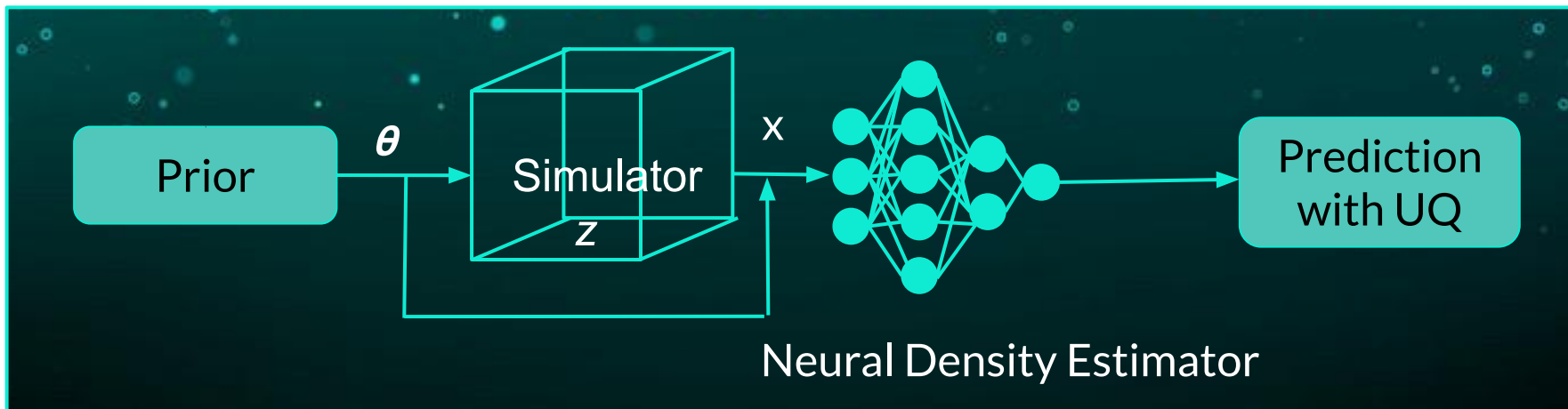
●
Point prediction

Inference with UQ

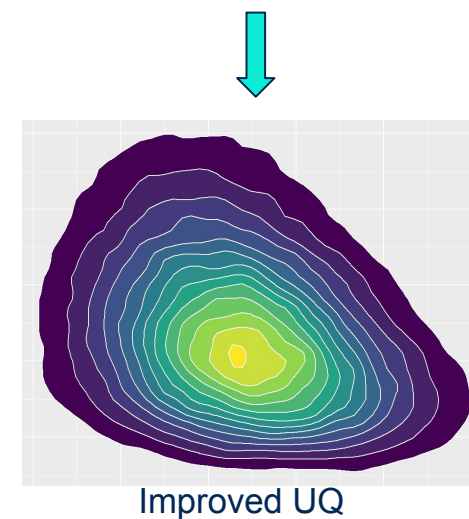
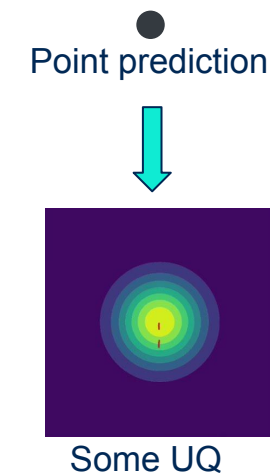


MVE / Dropout / BNN

Inference with UQ



Estimate the full posterior (Normalizing Flows)





UQ and Domain Shift

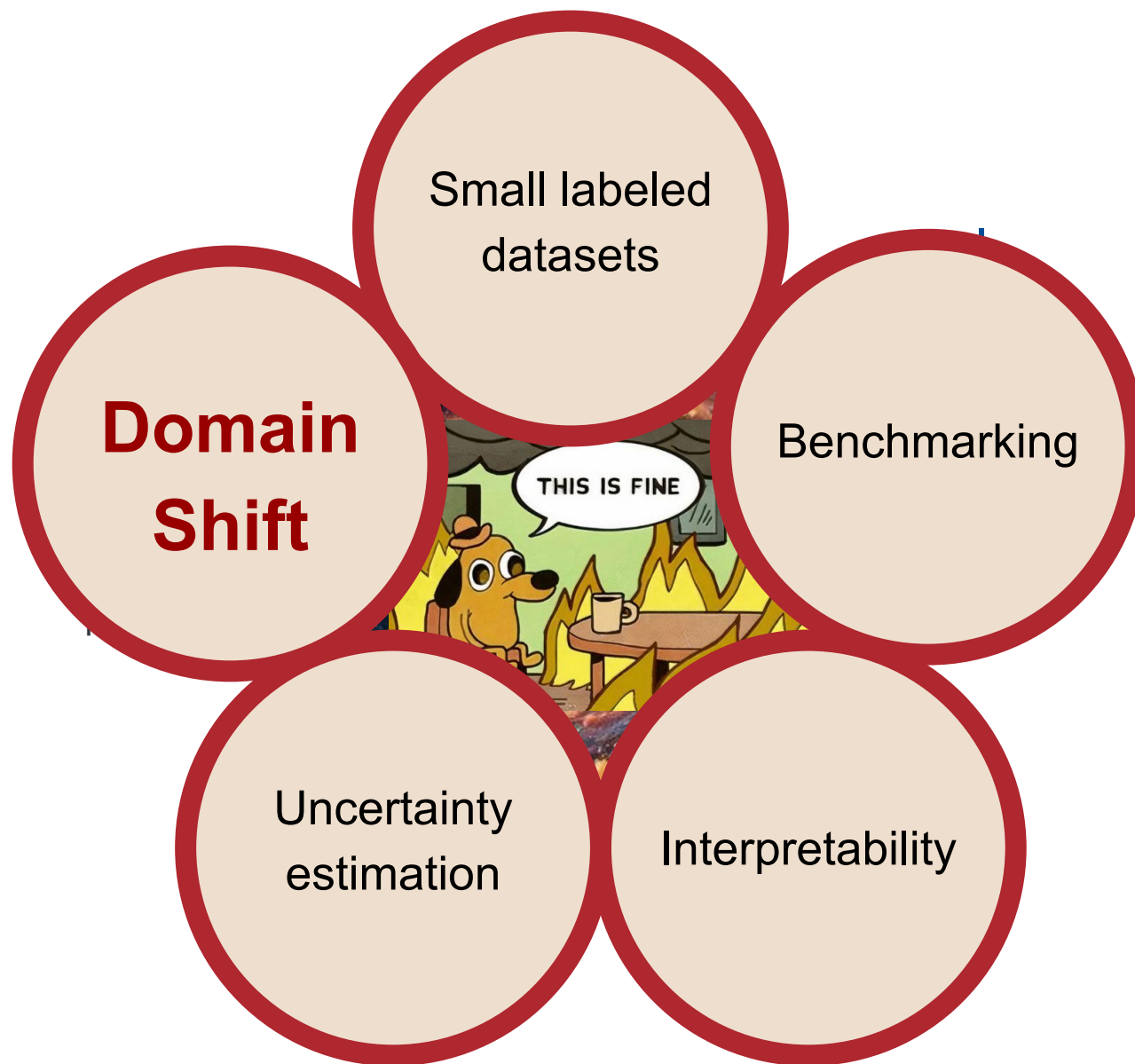
No UQ method is robust to domain shift by default.

- Standard UQ methods often assume **i.i.d. data** → can lead to **misleading uncertainty estimates** when this is violated.
- For scientific or safety-critical applications, **overconfident but wrong predictions** under domain shift are a serious risk.

- **Key challenges:**

- Overconfidence - highly certain (small errors) even on out of domain data.
- Shifted epistemic/aleatoric balance - epistemic error dominates.
- Miscalibrated models - confidence $\neq$ accuracy.
- Lack of target labels - we cannot evaluate UQ.





THANK YOU!

- Era of big scientific datasets - **large amount of data but not all of it labeled** (i.e. suitable for easy AI applications).
- Simulations and data are different which leads to **domain shift problem!**
- **Domain Adaptation** can help but it cannot solve all problems. So be careful how and when you use it!



Fermi**FORWARD**



U.S. DEPARTMENT
of ENERGY