Unsupervised Domain Adaptation

SOTA review and analysis

Michele Yin, Roberto Mazzaro, Andrea Bonora, Filippo Daniotti, Giovanni Ambrosi

Unsupervised Domain Adaptation

What is it:

- Train and test on dataset
- In real world we have a slightly different dataset
- Model doesn’t work!

https://arxiv.org/pdf/1804.09578.pdf

Unsupervised Domain Adaptation

What is it:

$$\{X_s, Y_s\}$$

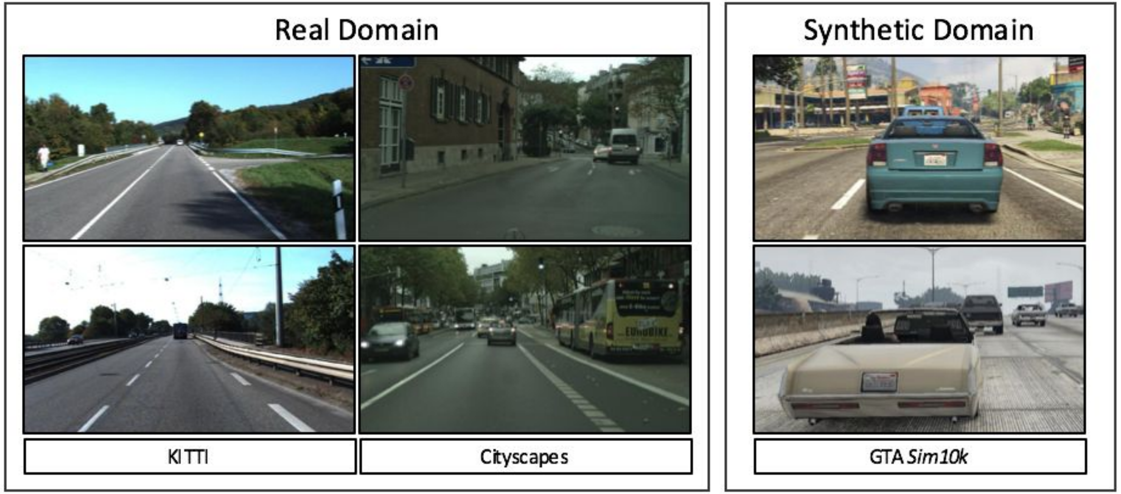
$$\{X_t\}$$

https://paperswithcode.com/paper/adaptiope-a-modern-benchmark-for-unsupervised

Unsupervised Domain Adaptation

Why:

- Reuse datasets
  - Labels are very expensive
- Improve real world performances



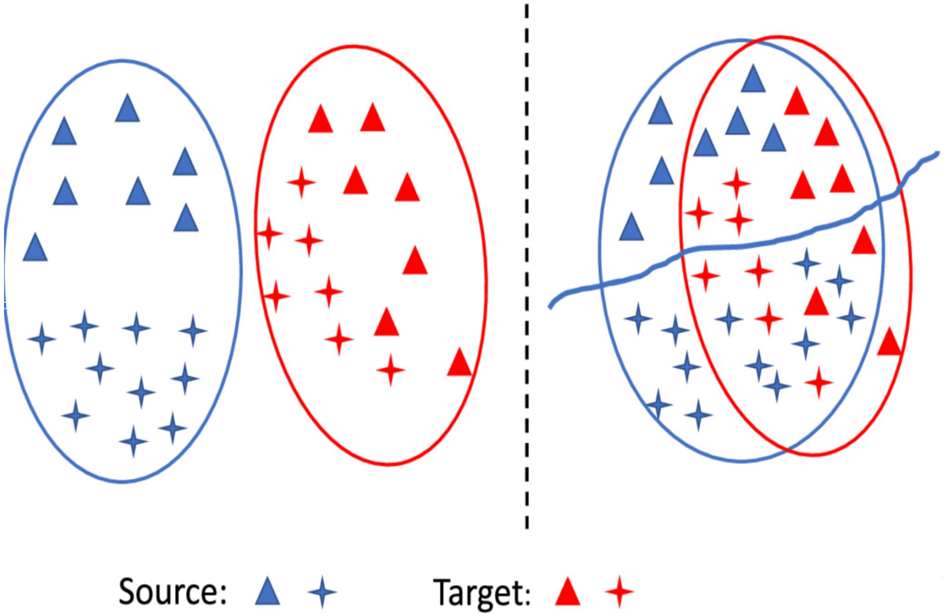
https://arxiv.org/abs/1810.05852

Unsupervised Domain Adaptation

Outline:

- Discrepancy based methods
- Adversarial based methods
- Other methods:
  - Teacher Student methods
  - Optimal Transport methods
  - Reconstruction-based methods
- Our experiments

Discrepancy Based



Key Idea:

- Align source and target feature distributions
- Hundreds or more techniques available and explored

Discrepancy Based

How:

- Measure a distance between source and target distributions
- Minimize this distance

$$\mathrm{MMD}(X_S, X_T) = \left\| \frac{1}{|X_S|} \sum_{x_s \in X_S} \phi(x_s) - \frac{1}{|X_T|} \sum_{x_t \in X_T} \phi(x_t) \right\|$$

- In general

$$L = L_{cls}(Y_s, \hat{Y}_s) + \lambda L_{align}(X_S, X_t)$$

https://arxiv.org/pdf/1412.3474.pdf

Discrepancy Based

Disadvantages:

- MMD is a measure of **first order** statistics. Distributions may have same mean but different variance.
- CORAL aligns the **second order** moments
  - Deep CORAL: Correlation Alignment for Deep Domain Adaptation [2016]
- Many variations to consider **higher order moments**
  - HoMM: Higher-order Moment Matching for Unsupervised Domain Adaptation [2019]
- Some ideas are to use a **kernel function** to map feature space into a Hilbert space

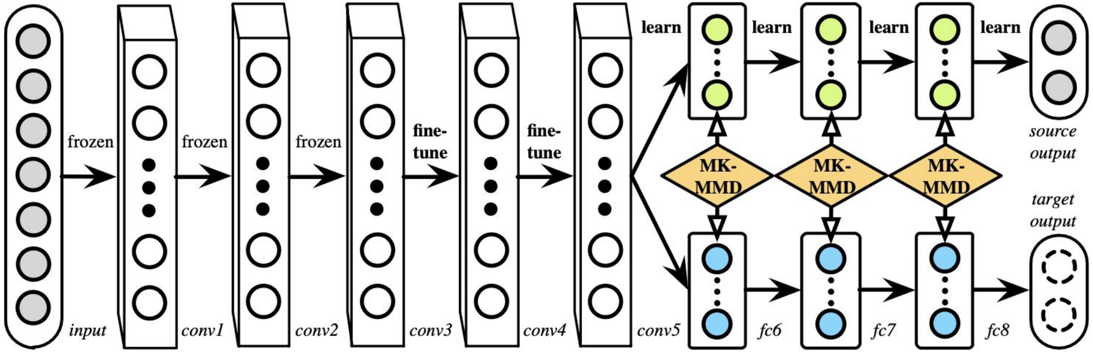
Many more

https://arxiv.org/abs/1607.01719 https://arxiv.org/abs/1912.11976

Discrepancy Based

Domain adaptation layers:

- Introduce domain adaptation layers
  - Learning Transferable Features with Deep Adaptation Networks [2015]



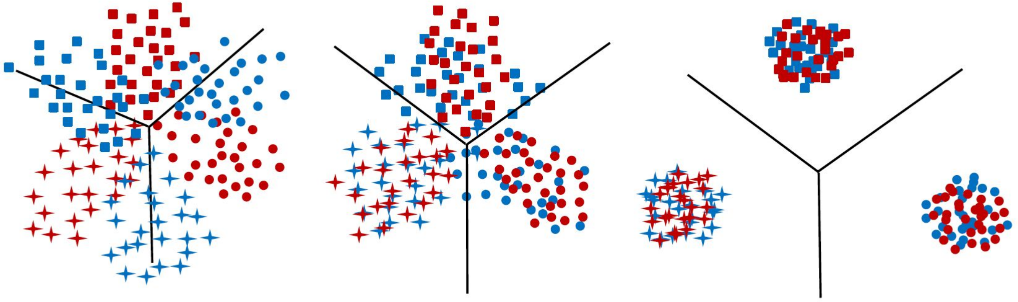
- **More layers** to learn a domain adaptation

https://arxiv.org/pdf/1502.02791.pdf

Discrepancy Based

Clustering or Entropy minimization:

- When target classes reside on a decision boundary we get bad domain adaptation results
- Use a **clustering** algorithm to push classes further from the decision boundary
  - Joint Domain Alignment and Discriminative Feature Learning for Unsupervised Deep Domain Adaptation [2018]



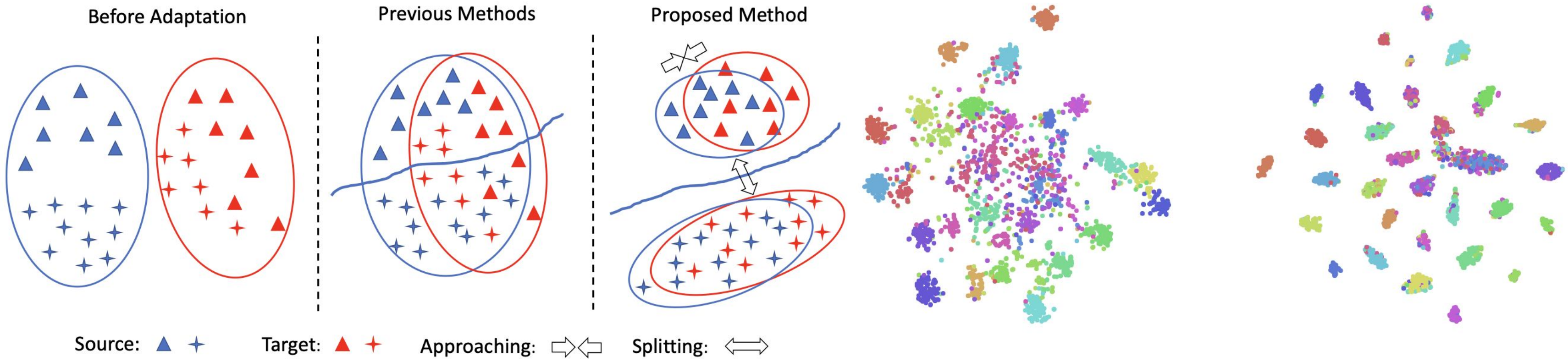
(a) (b) (c)

https://arxiv.org/abs/1808.09347

Discrepancy Based

Clustering or Entropy minimization:

- Consider **label information** to perform clustering
    - Contrastive Adaptation Network for Unsupervised Domain Adaptation [2019]

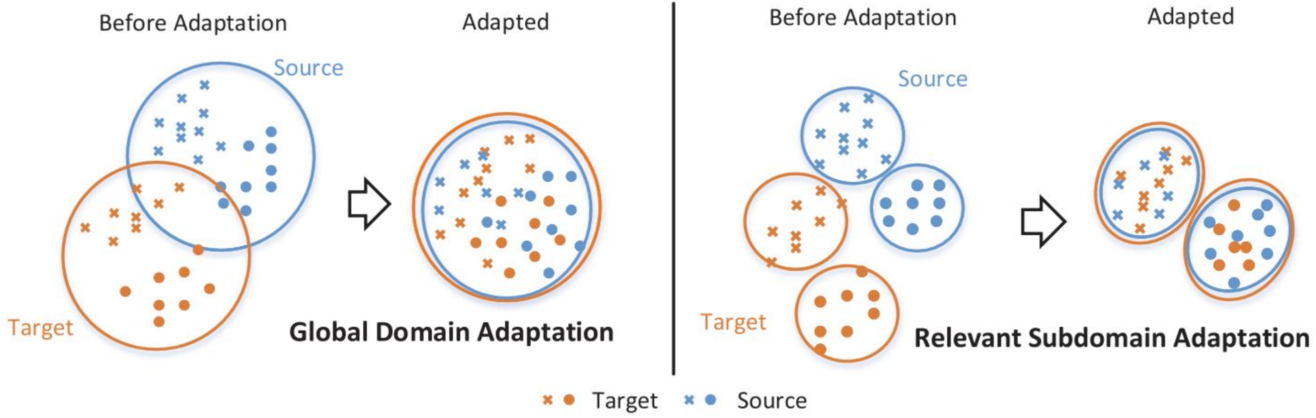


https://arxiv.org/abs/1901.00976

Discrepancy Based

Pseudo-labelling:

- Deep Subdomain Adaptation Network for Image Classification [2021]



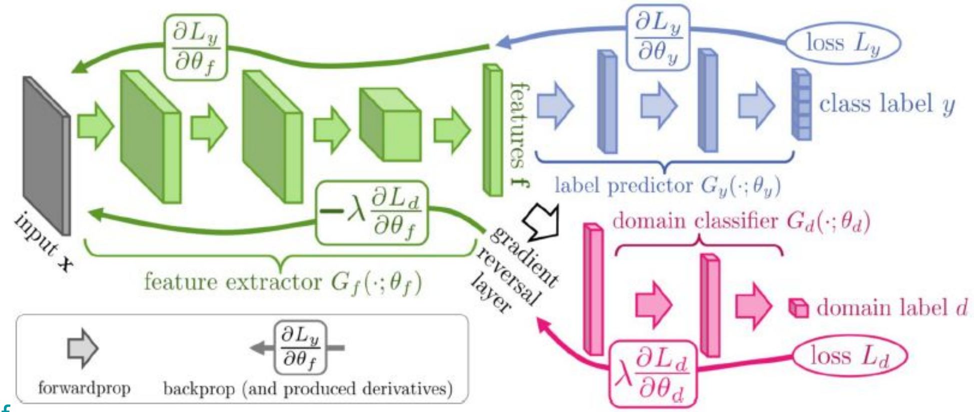
- Align label specific subdomain instead of global alignment

https://arxiv.org/pdf/2106.09388.pdf

Adversarial Based

Domain Adversarial Network [DANN 2015]:

- **Label Predictor** -> min. classification loss
- **Domain Predictor** -> min. domain loss
- **Feature Extractor** -> min. classification loss
  -> max. domain loss
- Important: **Gradient Reversal Layer (GRL)**

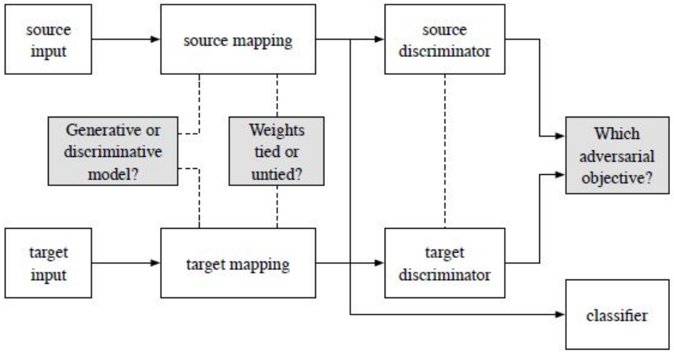


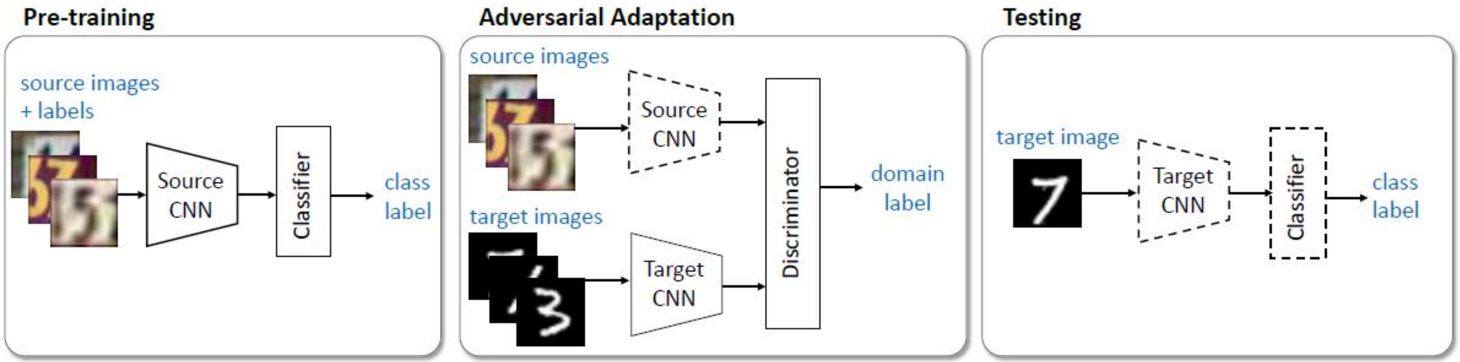
https://arxiv.org/pdf/1505.07818.pdf

Adversarial Based

Adversarial Discriminative DA [ADDA 2017]:

- Pre-train Source encoder
- Fool Discriminator -> learn **Target encoder**
- Use Target encoder with Source Classifier
- Important: Weight sharing, GAN loss



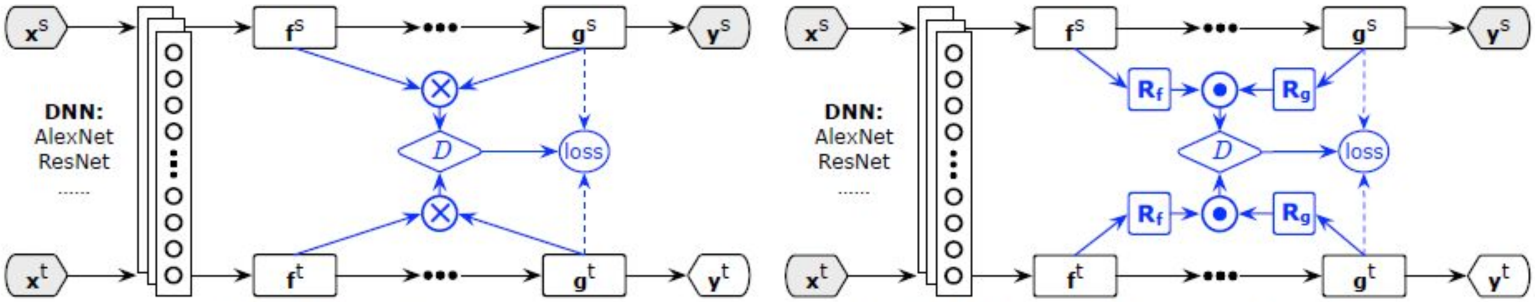


https://arxiv.org/pdf/1702.05464.pdf

Adversarial Based

Conditional Domain Adversarial Network [CDAN 2017]:

- Exploit discriminative information conveyed in the **classifier prediction**
- Exploit domain specific **features representation**
- Condition the discriminator using **Multilinear mapping** (small datasets) Randomized Multilinear mapping (bigger datasets) -> Entropy conditioning



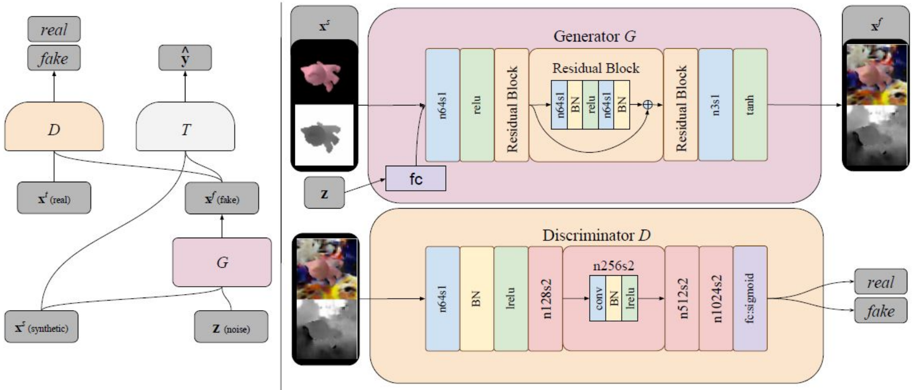
(a) Multilinear Conditioning (b) Randomized Multilinear Conditioning

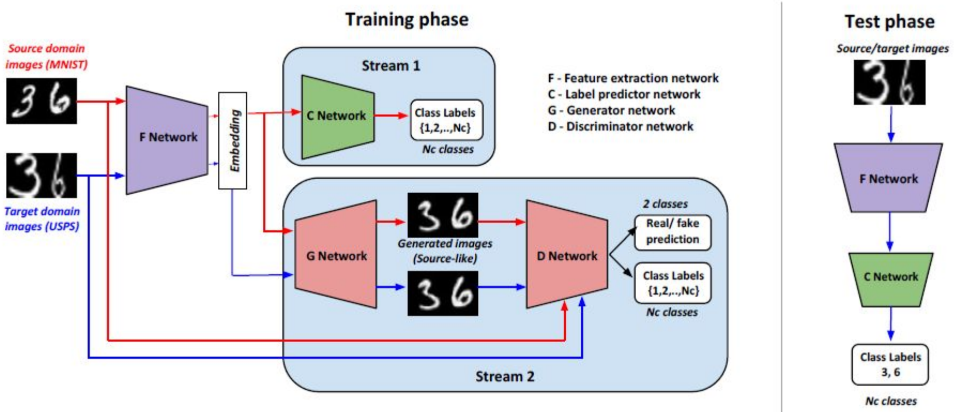
https://arxiv.org/pdf/1705.10667.pdf

Adversarial Based

GAN based methods:

- **PixelDA 2016:** generate **target images**, work at **pixel level,** train directly task specific **classifier**
- **GenerateToAdapt 2017:** generate target images, work at **feature level**, gen. images used only from the **discriminator**



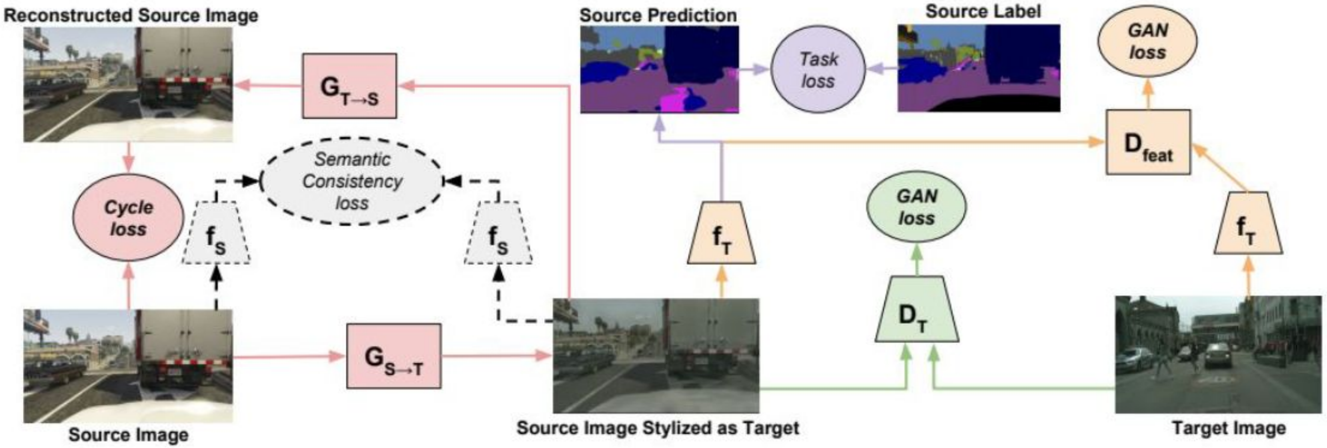


https://arxiv.org/pdf/1612.05424.pdf https://arxiv.org/pdf/1704.01705.pdf

Adversarial Based

Cycle Consistency [CyCADA 2018]:

- GAN based
- Introduce **cycle loss**
- Image level adaptation: pixel GAN loss, **cycle loss**, semantic consistency loss
- Feature level adaptation: feature GAN loss, source task loss

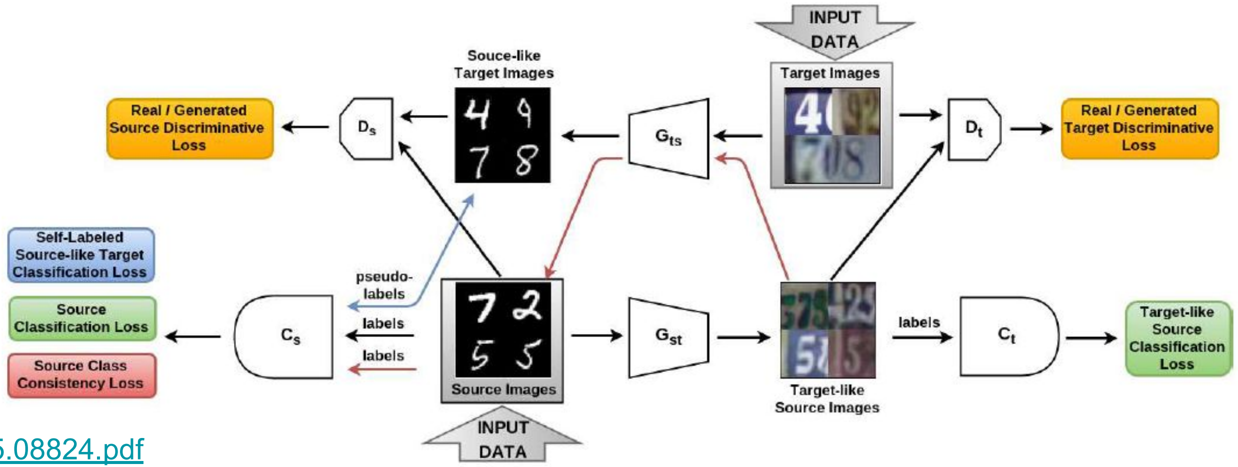


https://arxiv.org/pdf/1711.03213.pdf

Adversarial Based

Cycle Consistency [SBADA-GAN 2019]:

- GAN based
- **Cycle both for target and source**
- Source like **target images** are automatically **annotated with pseudo-labels** and are used by the classifier

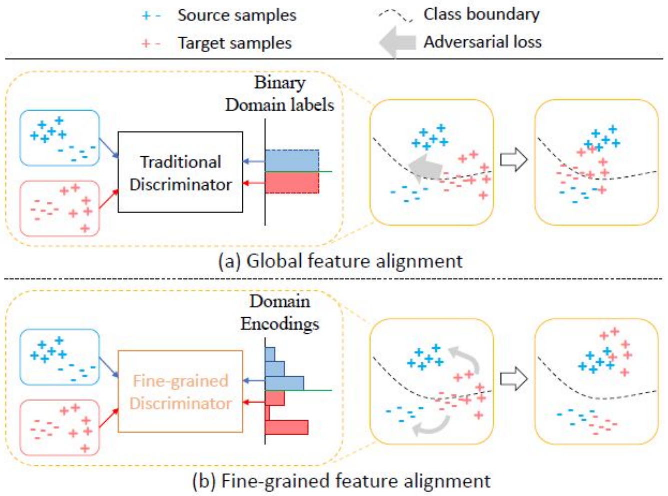


https://arxiv.org/pdf/1705.08824.pdf

Adversarial Based

Fine-grained ADA [FADA 2020]:

- Use **fine-grained discriminator**
- Include class information
- Allow class level alignment

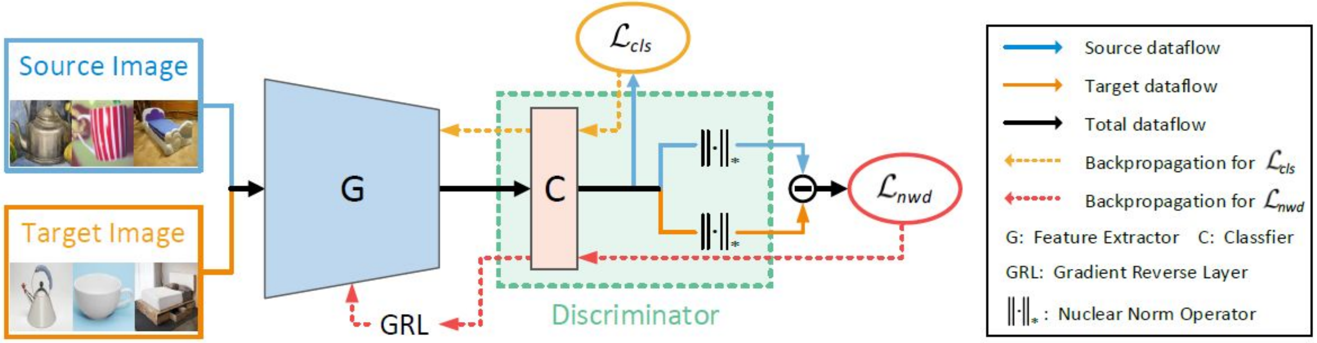


https://arxiv.org/pdf/2007.09222.pdf

Adversarial Based

Discriminator free ADA [DALN 2022]:

- **Category Classifier used as Discriminator**
- Introduce NWD (Nuclear-norm Wasserstein Discrepancy)
- **NWD + Classifier** used as **discriminator**
- High values on the diagonal of source self-correlation matrix (supervised training)
- High values also on off-diagonal element for target
- Encourage intra and inter class correlation between source and target
- Can be applied to other UDA methods

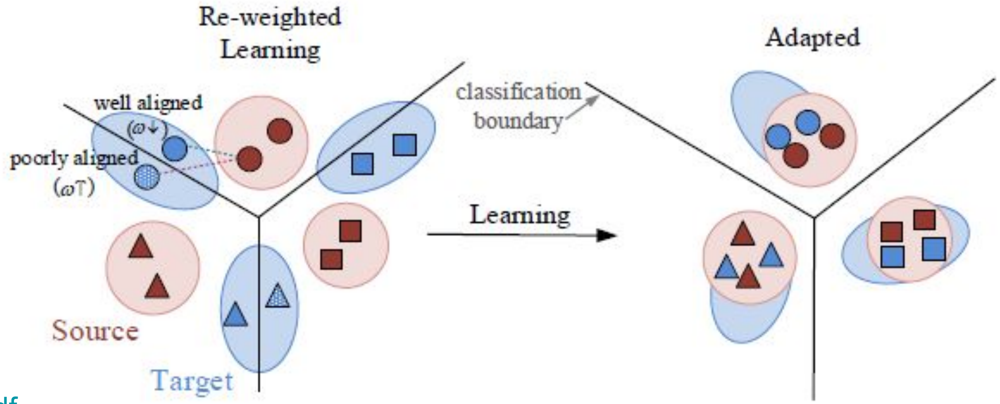


http://arxiv-export3.library.cornell.edu/pdf/2204.03838

Adversarial Based

Self-adaptive RE-weighted ADA [2020]:

- Use **conditional entropy** (obtained from conditional distribution) to reweight samples
- If **conditional entropy high** -> **poorly aligned** -> **increase weight** of adversarial loss
- Pseudo label for target
- Use **triplet loss** to obtain between source and pseudo labels to train feature extractor
- Allow good inter class separation and intra class compactenes

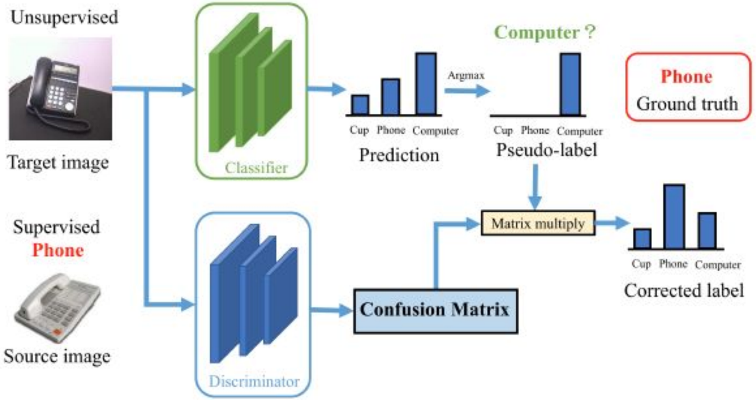


https://arxiv.org/pdf/2006.00223.pdf

Adversarial Based

Adversarial-Learned Loss for DA [ALDA 2020]:

- **Adversarial learning** only **aligns feature** distribution but don't consider if target features are discriminative
- **Self-training learn discriminative target features**
- **Combine** the **two methods** to obtain better features alignment
- Discriminator generate confusion matrix
- Obtain **corrected pseudo-labels** multiplying pseudo-labels by CM

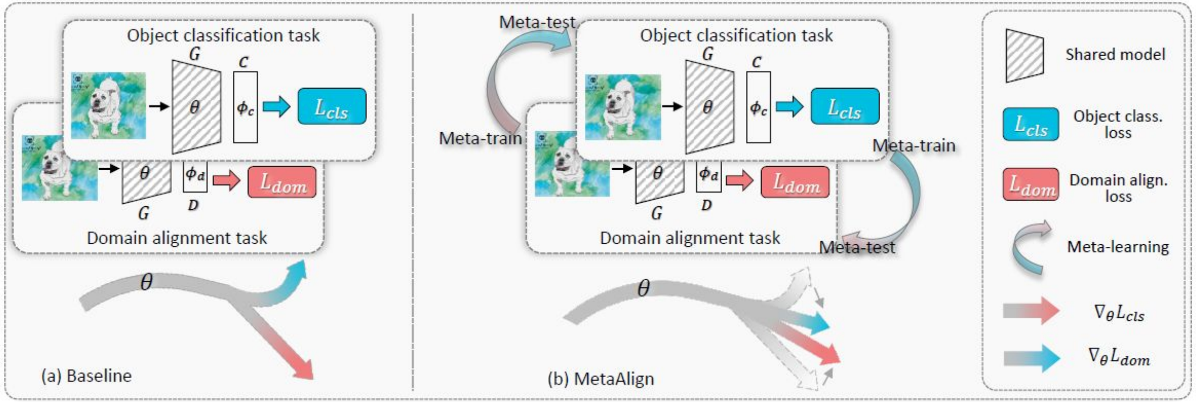


https://arxiv.org/pdf/2001.01046.pdf

Adversarial Based

MetaAlign [2021]:

- **Meta learning**: one task is **meta train task, other task** use for **validation**
- This scheme opt. in a **coordinated way** both tasks (domain alignment and classification)
- Maximize inner product of the gradients of the two tasks
- Can be applied to other UDA methods

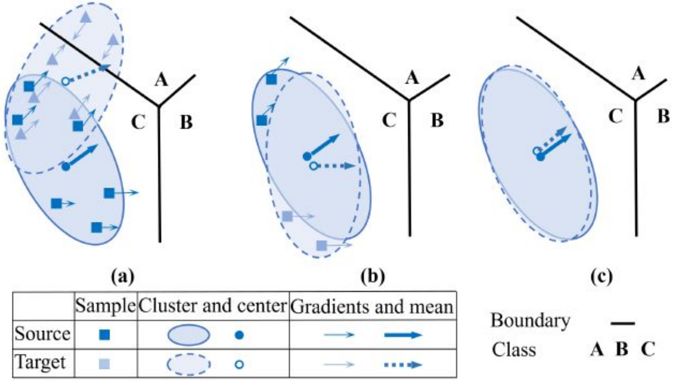


https://arxiv.org/pdf/2103.13575.pdf

Adversarial Based

Gradient Distribution Alignement [FGDA 2021]:

- **Constrain feature gradients** of **two domains** to have **similar distributions**
- Apply Jacobian regularization to improve model generalization
- **Pseudo labels** used to compute target loss
- Self-supervised pseudo labeling, online during first steps, then offline
- Can be applied to other methods

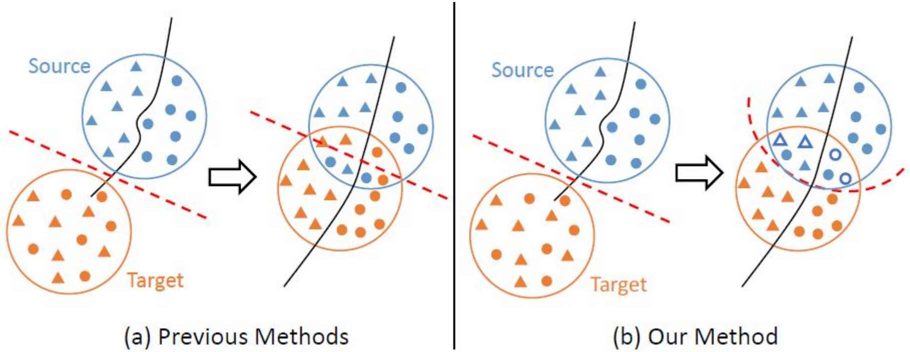


https://openaccess.thecvf.com/content/ICCV2021/papers/Gao_Gradient_Distribution_Alignment_Certificates_Better_Adversarial_Domain_Adaptation_ICCV_2021_paper.pdf

Adversarial Based

Re-energizing Domain Discrimination with Sample Relabeling [RADA 2021]:

- Use dynamic domain labels
- **Relabel well aligned target samples** as source domain
- Make less separable distributions more separable
- Compute average entropy of domain discrimination (if high poorer discrimination)
- Compute **MMD** (Maximum Mean Discrepancy) that indicate **how good is the alignment**
- Well aligned -> can't be well distinguish by domain discriminator (**entropy** higher than a **threshold**)
- Can be applied to other UDA methods

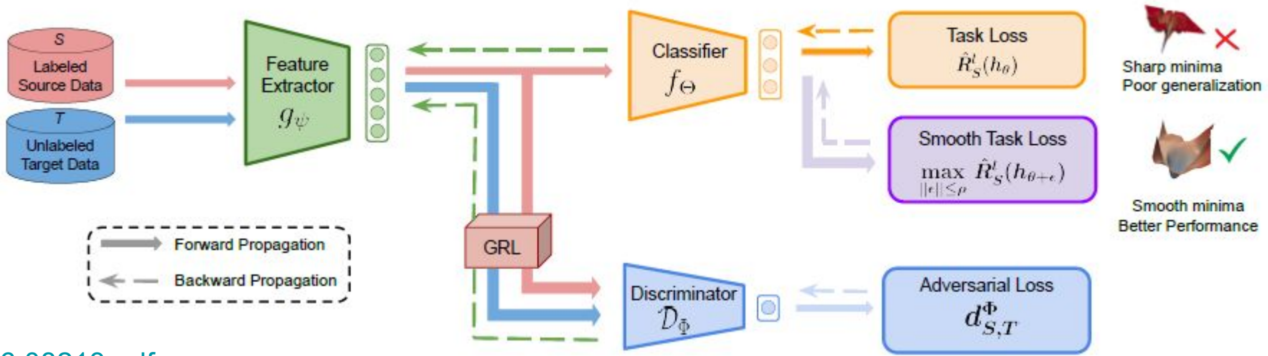


https://arxiv.org/pdf/2103.11661.pdf

Adversarial Based

Smooth Domain Adversarial Training [SDAT 2022]:

- Reach a **smoother minima** of **task loss** leads to better **generalization**
- Not the same for adversarial loss
- SDAT requires additional gradient computation step
- Compute **Hessian matrix** of classification task for source
- Compute Trace and the **maximum eigenvalue** -> indicative of high smoothness (if low better)
- Can be applied to other UDA methods



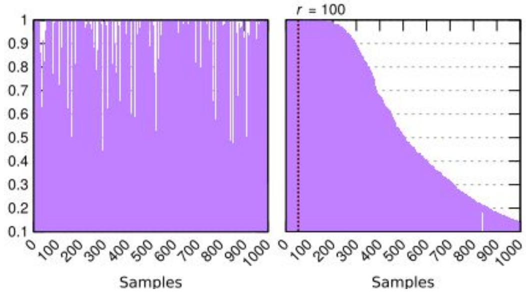
https://arxiv.org/pdf/2206.08213.pdf

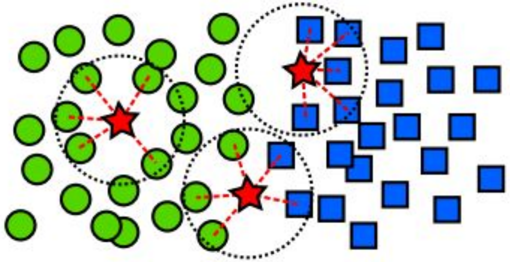
Incremental Methods

In incremental learning the input data are continuously used to extend the existing model's knowledge.

Incremental Unsupervised Domain-Adversarial Training of Neural Networks [iDANN 2020]:

- Builded upon the existing **DANN** approach
- Self-labeling
- Label-smoothing: $y_i' = (1 - \epsilon)y_i + \frac{\epsilon}{L}$
- Policies to select samples for the labeling phase:
  - *Confidence Policy*
  - *kNN Policy*





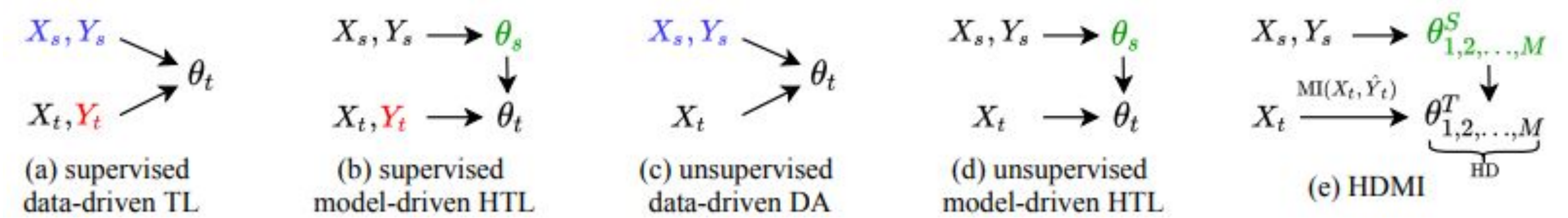
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9216604&isnumber=9591206

Information Based

Hypotheses Transfer Learning (**HTL**) + Unsupervised Domain Adaptation (**UDA**):
The knowledge from a source domain is transferred **solely through hypotheses** and adapted to the target domain in an unsupervised manner.

Hypothesis Disparity Regularized Mutual Information Maximization [HDMI 2020]:

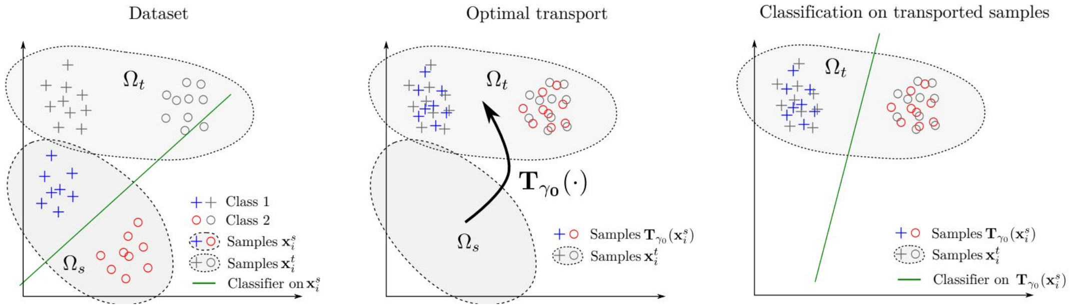
- Transfer knowledge from a set of source hypotheses to a corresponding target set of target hypotheses
- ***M*** hypothesis use a shared feature extractor and ***M*** independent classifier
- Adapt the source hypotheses into a set of corresponding target hypotheses by **maximizing the *MI*** between the empirical target input distribution and the predicted target label distribution induced by the target hypotheses.



https://arxiv.org/abs/2012.08072v1

Optimal Transport

Optimal transport is the general problem of moving **one distribution of mass to another** as efficiently as possible.



Different ways to compute the distance:

- Total variation
- Hellinger
- L_2
- X^2
- **Wasserstein Distance:** we allow the mass at ***x*** to split and move to more than one location

$$W_p(P,Q) = \left(inf_{J \in \Im(P,Q)} \int ||x - y||^p dJ(x,y) \right)^{1/p}$$

Optimal Transport

Teacher Imitation Domain Adaptation with Optimal Transport [TIDOT 2021]:

- Two cooperative agents: a **teacher** and a **student**
- P_S and P_T are the data distributions for the source and the target domain
- h_S: **well-qualified classifier** that gives accurate prediction for data instances on X_S sampled from P_S
- **Goal:** learn h_T
- Minimize the proposed objective function consisting in:
  - Loss of the teacher h_S
  - OT-based imitation learning term

$$\min_{h_S, h_T, G} \left\{ \mathcal{L}^S + \alpha \mathcal{R}^{WS} \right\},$$

$$\mathcal{L}^S = \frac{1}{N_S} \sum_{i=1}^{N_S} \ell \left(h_S \left(G \left(\mathbf{x}_i^S \right) \right), y_i^S \right)$$

$$\mathcal{R}^{WS} = \mathcal{W}_d \left(\mathbb{P}_{T,h_T}, \mathbb{P}_{S,h_S} \right)$$

https://www.ijcai.org/proceedings/2021/0394.pdf

Optimal Transport

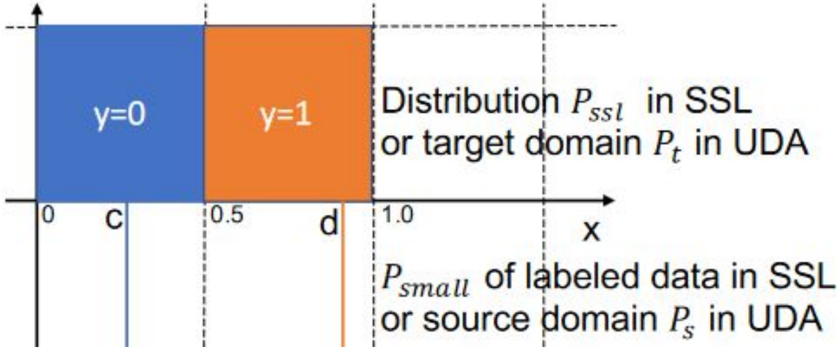
Other methods:

- MOST [2021]: multi source domain adaptation using teacher-student learning
- LAMDA [2021]: Wasserstein distance used to not only quantify the data shift but also to define the label shift directly
- MLOT [2020]: optimizes a Mahalanobis distance leading to a transportation plan that adapts better
- RWOT [2020]: inspired by prototypical networks. The idea is to shrink the subspace reliability to measure the sample-level domain discrepancy across domains by exploiting spatial prototypical information and intra-domain structure dynamically
- ETD [2020]: builds an attention-aware transport distance

SSL and UDA

Semi-supervised Models are Strong Unsupervised Domain Adaptation Learners [arXiv 2021]:

- Frame **SSL** as a **special case of UDA**



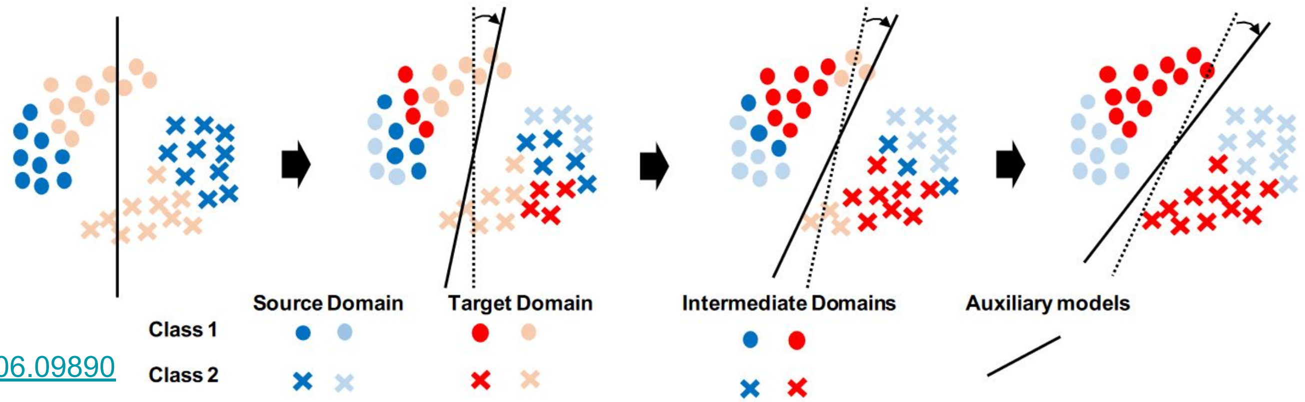
- Apply standard SSL methods on UDA tasks
    - consistent improvement over source only
    - **steady baseline**

https://arxiv.org/abs/2106.00417

Self-Training Based

Gradual Domain Adaptation via Self-Training of Auxiliary Models [arXiv 2021]:

- Models trained on source get worse as domain divergence increases
- Idea: **train auxiliary models on intermediate domains** through self-training
- Generate intermediate domains:
  - first: start with pure *source*
  - intermediate: gradually increase proportion of samples drawn from *target*
  - end with pure *target*



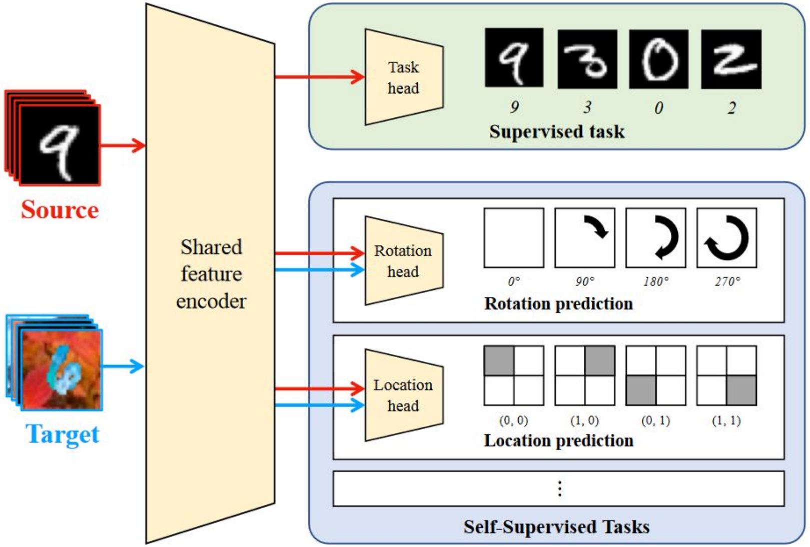
https://arxiv.org/pdf/2106.09890

Self-Supervised Based

Unsupervised Domain Adaptation through Self-Supervision [arXiv 2019]:

- Self-Supervision: train with auxiliary tasks on artificially altered versions of the available data
- In UDA:
  - shared feature extractor
  - supervised task head (source only)
  - SS task heads (source and target)
- Multi-task loss

$$\min_{\phi, h_k, k=1\ldots K} \quad \mathcal{L}_0(S;\phi,h_0) + \sum_{k=1}^{K} \mathcal{L}_k(S,T;\phi,h_k)$$



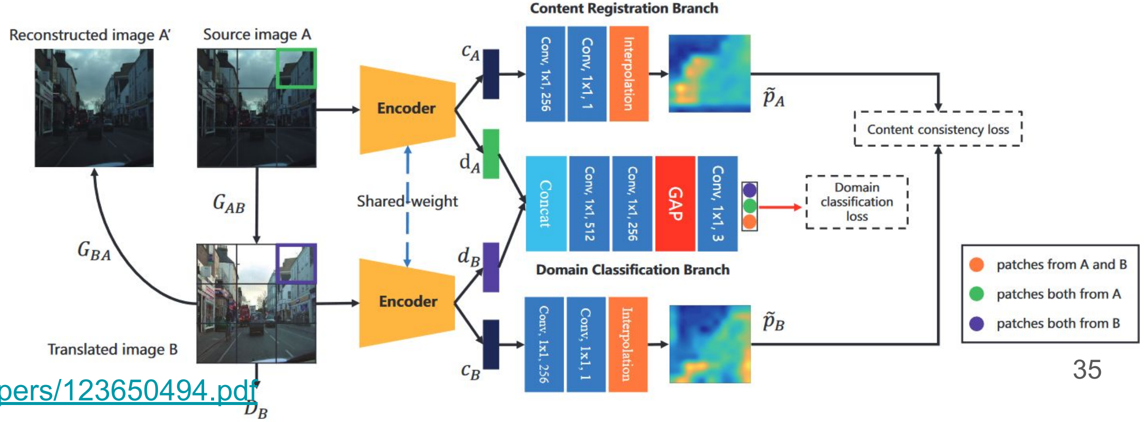
https://arxiv.org/pdf/1909.11825

Self-Supervised Based

Self-Supervised CycleGAN for Object-Preserving Image-to-Image Domain Adaptation [ECCV2020]:

- Pick CycleGAN, add Self-Supervised Siamese network *S*
- Divide image in patches, sample two
- Two SS tasks:
  - content registration: relative position
  - domain classification: patch classification
- content consistency loss

$$\mathcal{L}_{cc} = \frac{1}{M \times N}\sum_{x=1}^{M}\sum_{y=1}^{N}(\tilde{p}^{A}_{x,y} - \tilde{p}^{B}_{x,y})^2$$

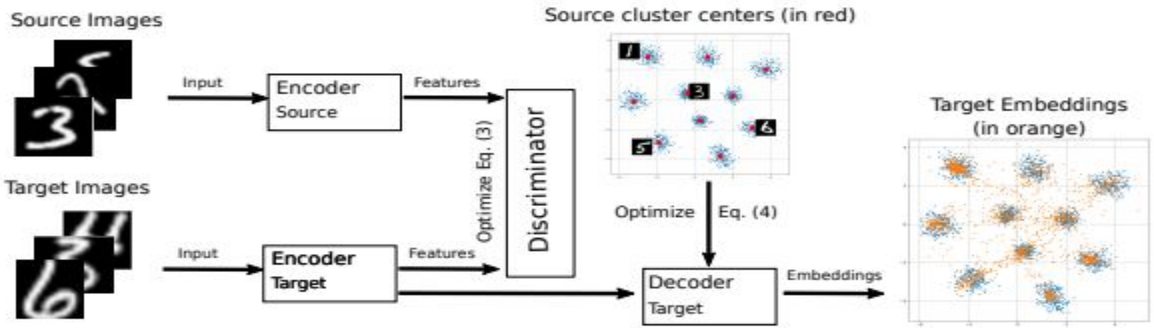


https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123650494.pdf

Deep metric learning

M(etric)-ADDA (2018):

- Learn a distance metric -> clustering
  - examples with same label close as possible
  - examples with different labels as far as possible;



https://arxiv.org/abs/1807.02552v1

Deep metric learning

Source model training:

- optimization of the **triplet loss**:

$$\mathcal{L}(\theta_S) = \sum_{(a_i, p_i, n_i)} \max(||f_{\theta_S}(a_i) - f_{\theta_S}(p_i)||^2 - \\ ||f_{\theta_S}(a_i) - f_{\theta_S}(n_i)||^2 + m, 0)$$

- a_i = anchor example (picked randomly)
- p_i = example with same label of ai
- n_i = example with different label wrt pi

Deep metric learning

Target model training:

- **Adapt loss** for target encoder:

$$\mathcal{L}_A(\theta_{T_E}, \theta_D) = \min_{\theta_D} \max_{\theta_{T_E}} - \sum_{i \in S} \log D_{\theta_D}(E_{\theta_S}(X_{S_i})) - \sum_{i \in T} \log(1 - D_{\theta_D}(E_{\theta_{T_E}}(X_{T_i})))$$

- **Magnet loss** for decoder:

$$\mathcal{L}_C(\theta_T) = \sum_{i \in T} \min_j ||f_{\theta_T}(x_i) - C_j||^2$$

Deep metric learning

- Final loss for target:

$$\mathcal{L}(\theta_T, \theta_D) = \underbrace{\mathcal{L}_A(\theta_{T_E}, \theta_D)}_{\text{Adapt}} + \underbrace{\mathcal{L}_C(\theta_T)}_{\text{C-Magnet}}$$

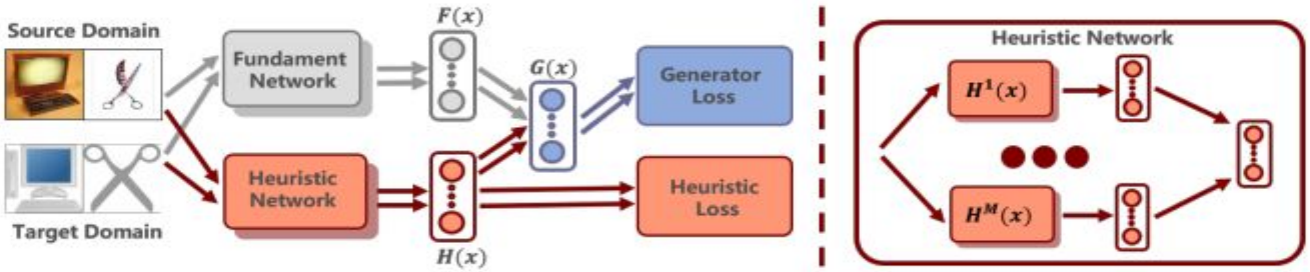
- M-ADDA works better than vanilla ADDA for the presence of the decoder
  - **improves the training of the target encoder** (unsupervised part)
  - **guarantees better alignment between the two domains**

Visual domain adaptation

Heuristic Domain Adaptation (2020):

- Based on A* search
- Heuristic function H(x) that guides a generator function G(x)

$$G(x) = F(x) - H(x)$$



https://arxiv.org/abs/2011.14540

Visual domain adaptation

- Fundament network computes:
  - adversarial discrepancy on classification responses;
  - **domain invariant representation**

- Heuristic network
  - learns local features
  - ensures that it **does not model any domain-specific representation**

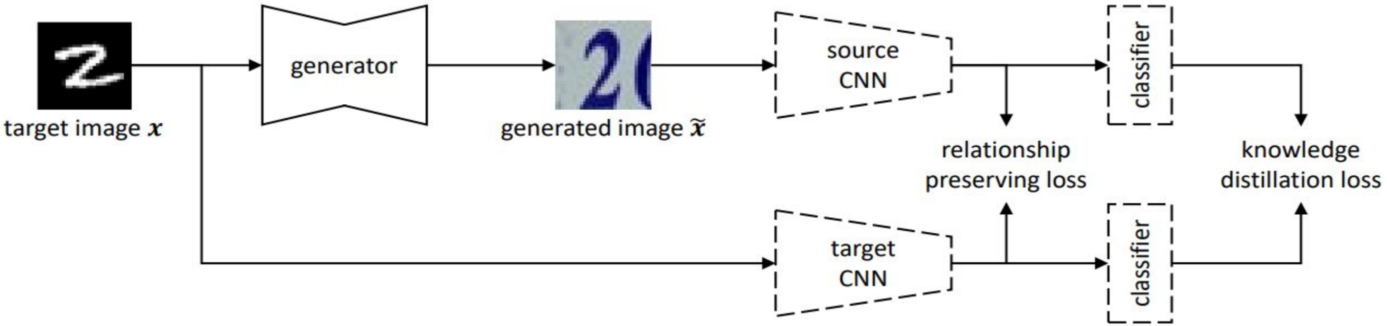
- Cosine similarity to look at the relationship between the representations of deep networks.

$$cos(\theta) = \frac{G(x) \cdot H(x)}{|G(x)|\,|H(x)|}$$

Visualizing adapted knowledge in DL

Visualizing adapted knowledge (2021):

- Translate a target image x from its domain to a new image x̌;
  - Feed source model with the generated image
  - The target model with the original one -> **source-free training**



https://arxiv.org/abs/2104.10602

Visualizing adapted knowledge in DL

- **Relationship preserving (loss)**: ensures similar distributions from the target and source CNNs after a successful knowledge distillation
  - MSE between Gram matrices;

- **Knowledge distillation (loss)**: learns semantic information and transfer it to the generator
  - Kullback-Leibler divergence;

Project status

Implementations:

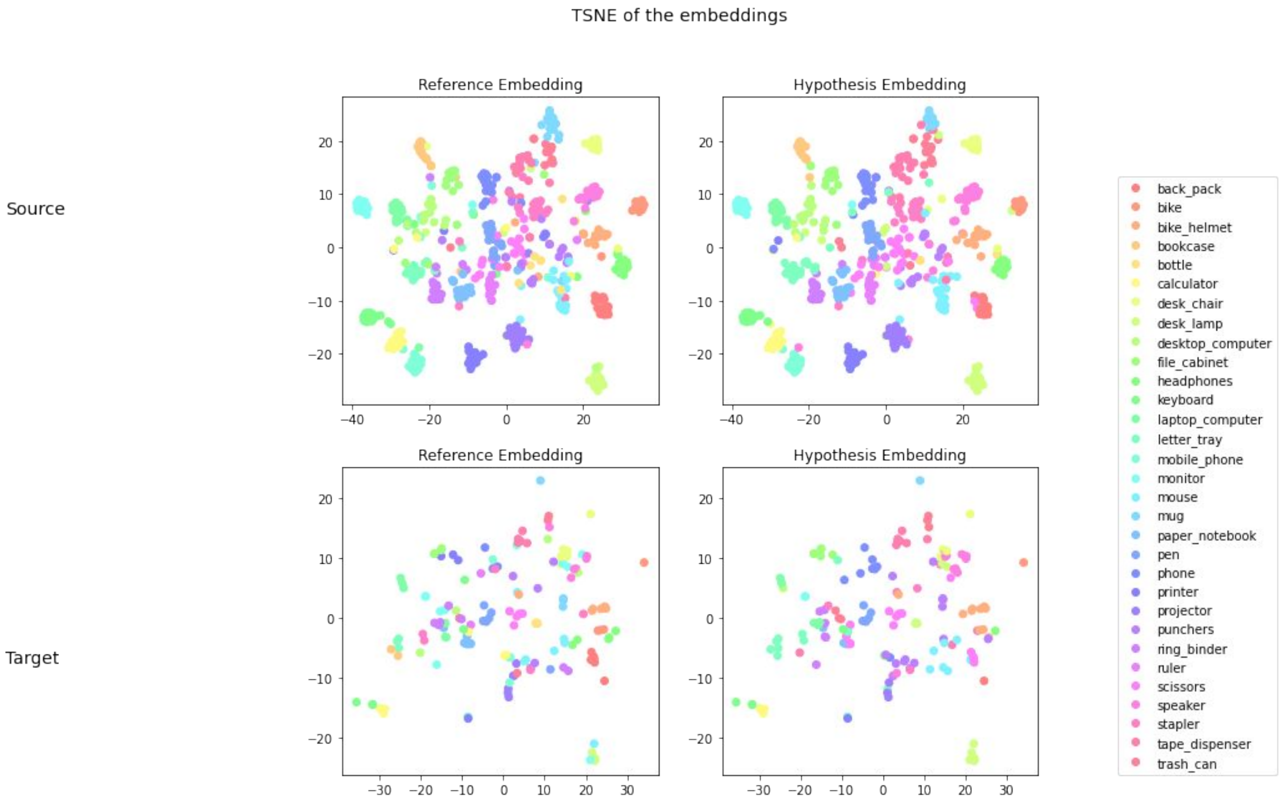
- universal t-SNE plotter for standardized testing

Tested methods:

- baseline model
- coral alignment
- dan
- cdan
- dann

Test environment: Google COLAB

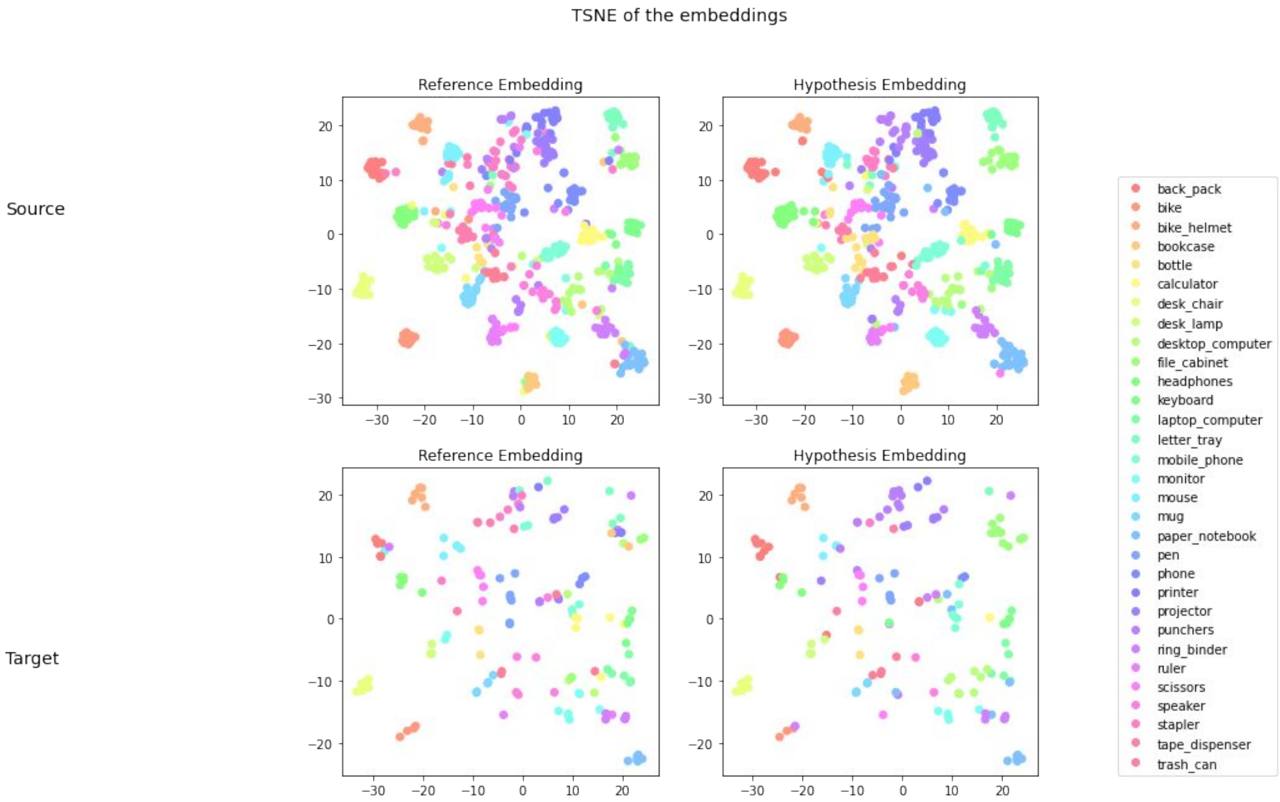
Source Only results



t-SNE Source Only:

- Backbone: ResNet-34
- Dataset: Office-31
  - Source: Amazon
  - Target: Webcam
- Drops accuracy from 83% to 48%

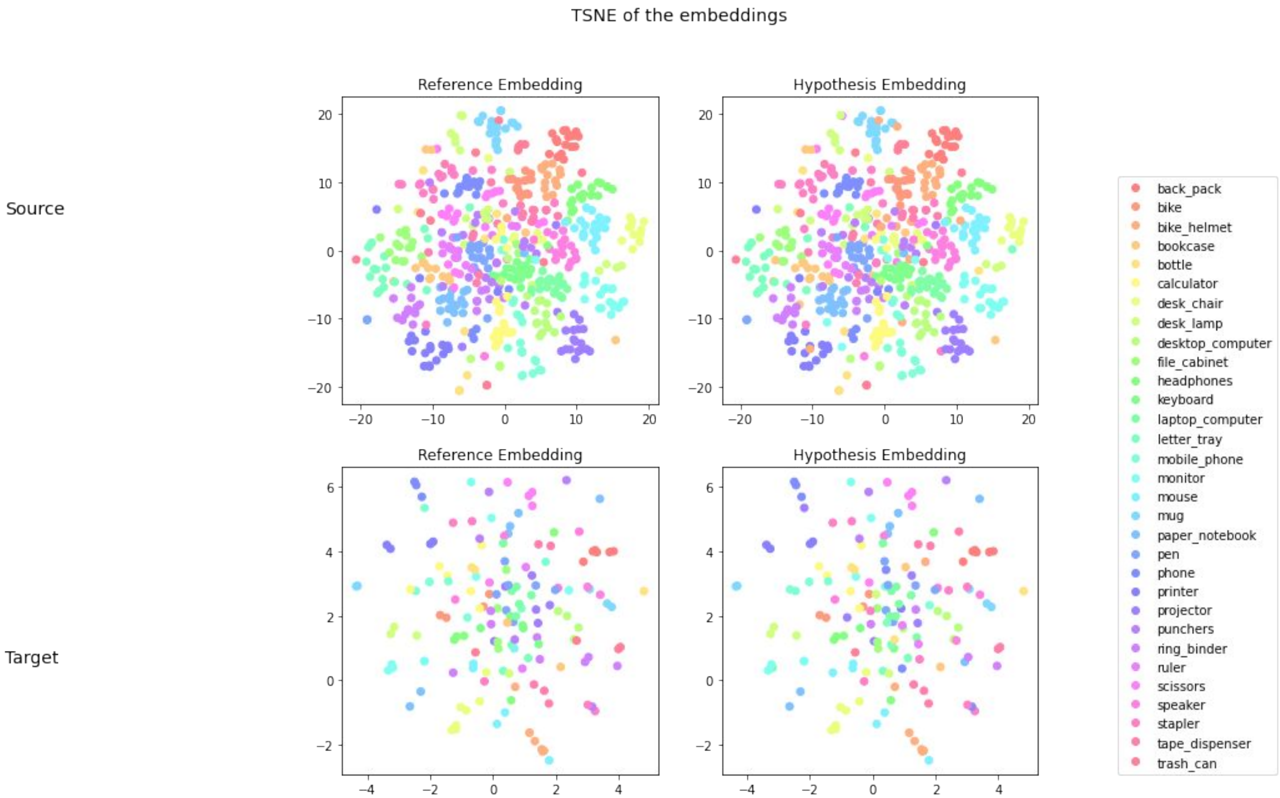
CORAL results



t-SNE CORAL:

- Backbone: ResNet-34
- Dataset: Office-31
  - Source: Amazon
  - Target: Webcam
- Improve from 48% accuracy to 61% on target

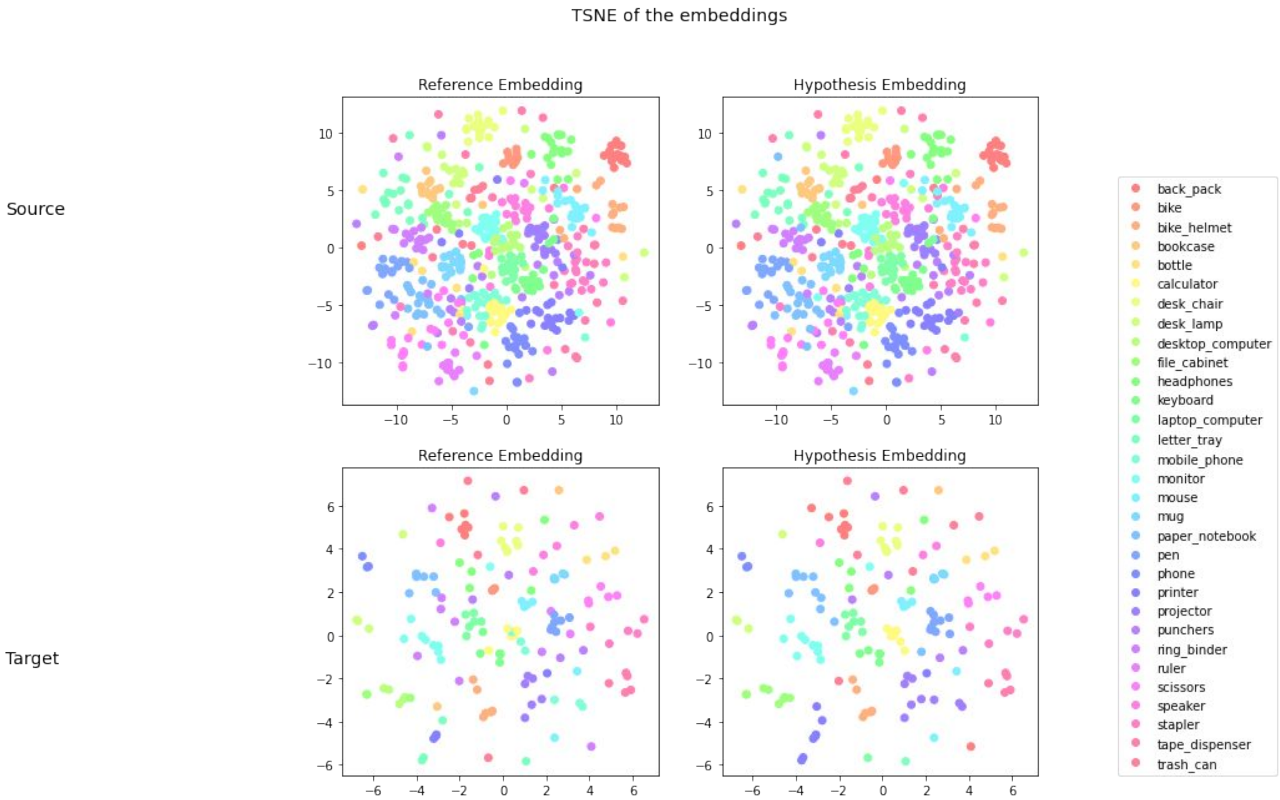
DAN results



t-SNE DAN:

- Backbone: ResNet-34
- Dataset: Office-31
  - Source: Amazon
  - Target: Webcam
- Improve from 48% accuracy to 81% on target

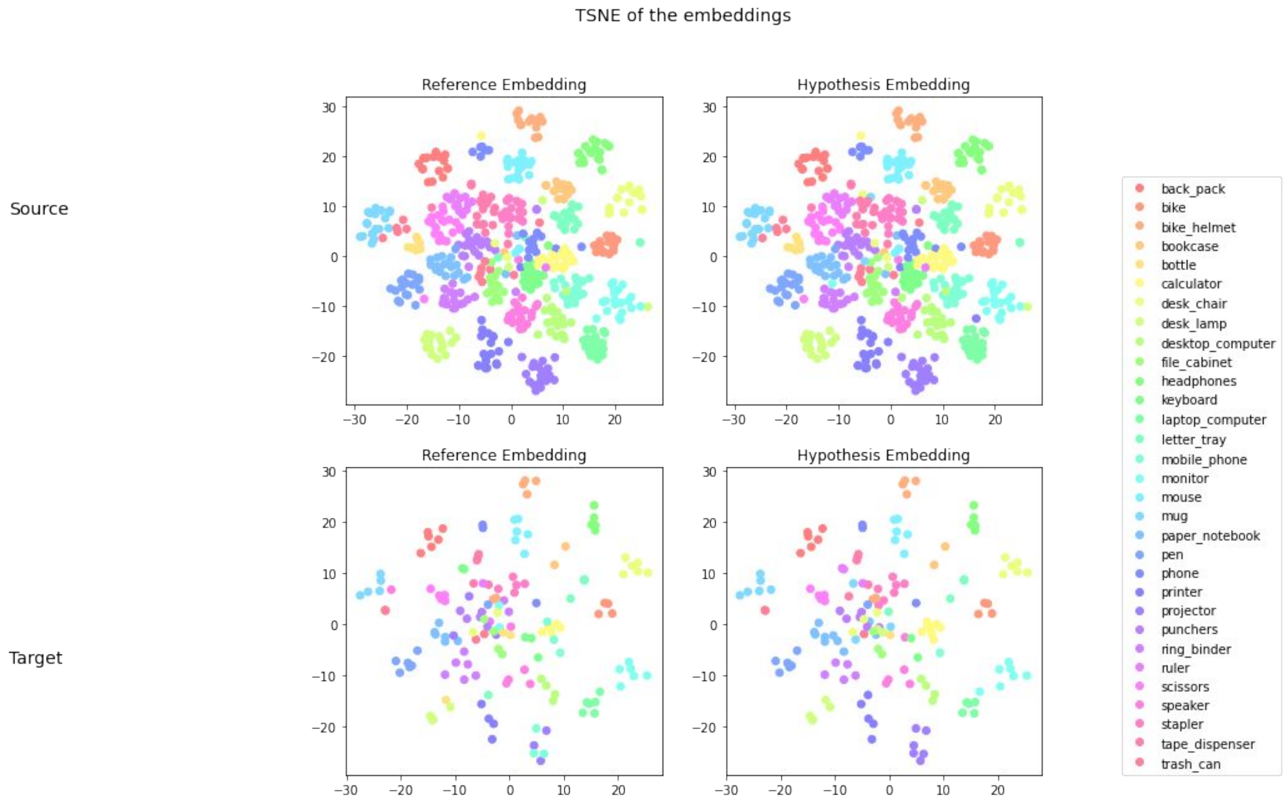
DANN results



t-SNE DANN:

- Backbone: ResNet-34
- Dataset: Office-31
  - Source: Amazon
  - Target: Webcam
- Improve from 48% accuracy to 81% on target

CDAN results



t-SNE CDAN:

- Backbone: ResNet-34
- Dataset: Office-31
  - Source: Amazon
  - Target: Webcam
- Improve from 48% accuracy to 86% on target

Future works

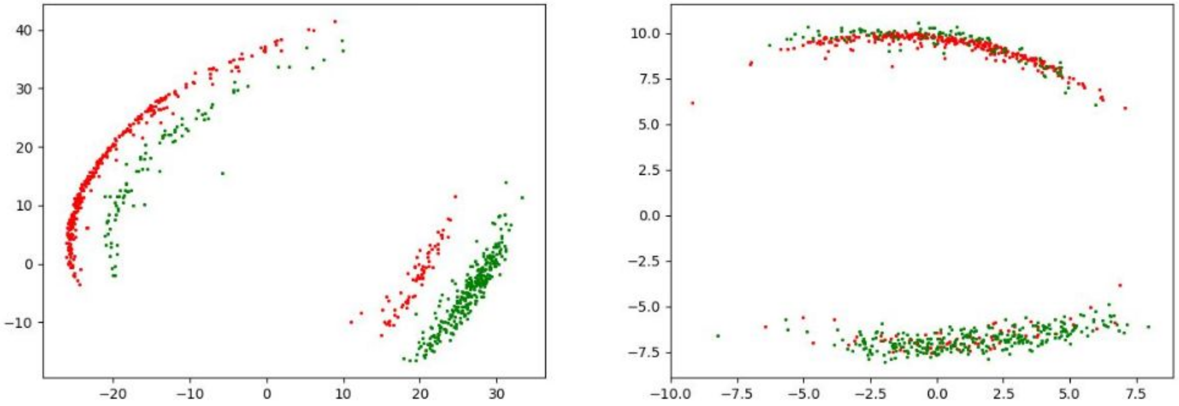
Ideas:

- Test more recent papers
- Merge different approaches
    - In particular add reconstruction and discrepancy to other methods

Discrepancy Based

Class imbalance:

- Normalized Wasserstein for Mixture Distributions with Applications in Adversarial Learning and Domain Adaptation

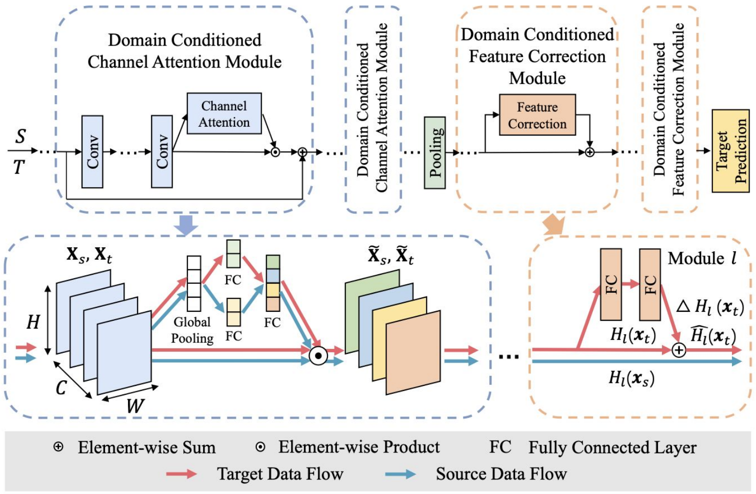


https://arxiv.org/abs/1902.00415

Discrepancy Based

One big problem:

- Assume that we can use the same backbone for both source and target
- It has been proven that in deep networks, eventually features do transition from general to domain specific.
- Attention module to select the domain
  - Domain Conditioned Adaptation Network

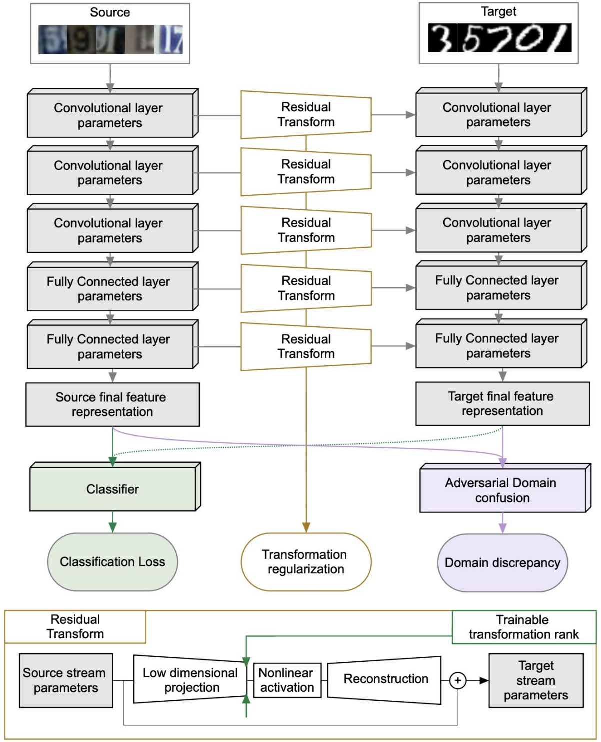


https://arxiv.org/abs/2005.06717

Discrepancy Based

One big problem:

- Adapt the backbone by learning transformation of source and target network
    - Residual Parameter Transfer for Deep Domain Adaptation



https://arxiv.org/abs/1711.07714

Adversarial Based

Summarizing:

- Domain Adversarial Network contain the core idea
- Next years several architectures have been developed
- **Equilibrium problem:** even though discriminator is fully confused, sufficient similarity between two distributions cannot be guaranteed because gradient for well aligned samples is low so we have few driving power for training
- Nowadays several strategies to solve this problem have been proposed
- Many of them are can be added to other UDA schemes

Adversarial Based

Dynamic Weighted Learning for Unsupervised Domain Adaptation(2020):

- Dynamically adjust weights in order to have balanced domain alignment and class discrimination;
  - measure degree of alignment each iteration
  - construct dynamic balance factor to control weights (**τ**)
- MMD and LDA or data alignment
- scatter matrix J(w) for class discrimination;
- training controlled by **τ**

Slide di Esempio

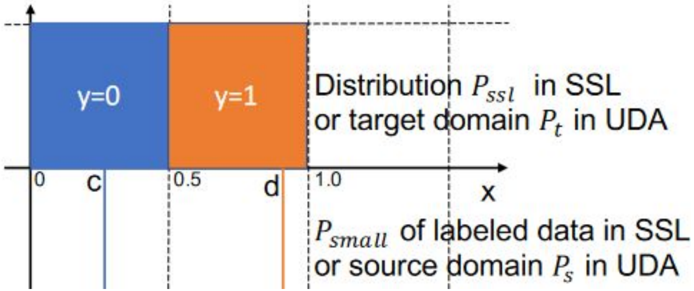
Advantages:

- monitoring degree of alignment real-time;
- avoid model bias during training;
- more universal and applicable to cross-domain data scenarios;
- more efficient with unbalanced number of sample or with different statistics/distributions;

SSL and UDA

Semi-supervised Models are Strong Unsupervised Domain Adaptation Learners [arXiv 2021]:

- Frame SSL as a special case of UDA
- Consider a SSL setting:
  - labelled data can only represent a subdomain of distribution P_{ssl}
  - pick the sub-domain with smallest possible support P_{small}
  - P_{small} and P_{ssl} are distributions of source and target domains respectively
- Apply standard SSL methods on UDA tasks
  - consistent improvement over baseline
- Combinations:
  - SSL regularizers on UDA techniques
  - UDA approaches to SSL tasks

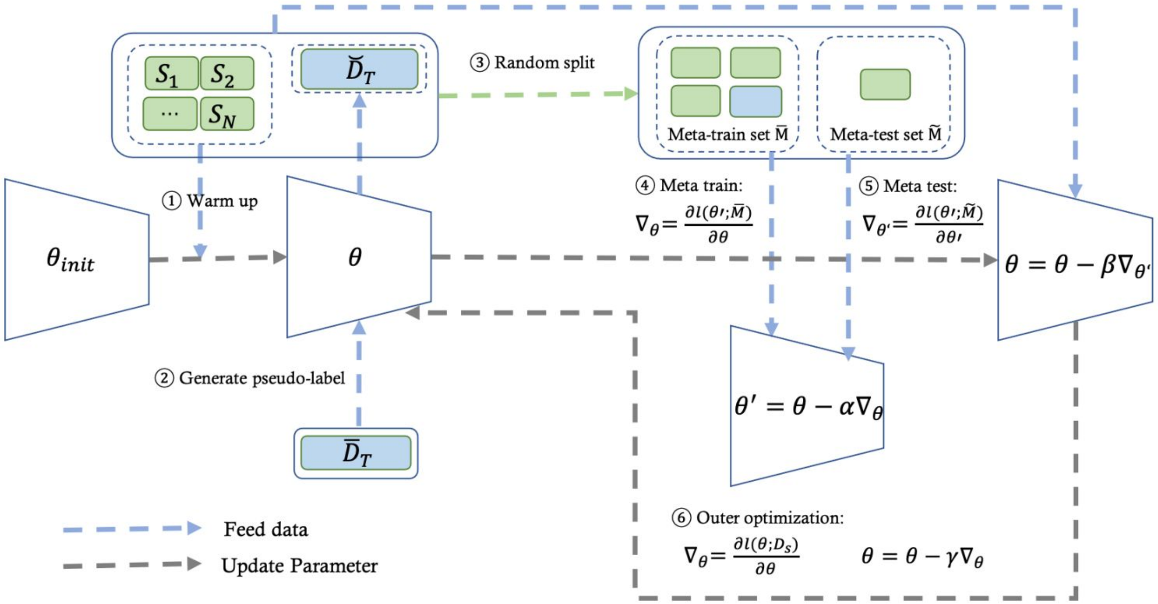


https://arxiv.org/abs/2106.00417

Self-Training Based

Meta Self-Learning for Multi-Source Domain Adaptation [ICCV Workshop 2021]:

- Combine self-learning with the idea of meta-learning
- Meta-learning: ‘training operations on model itself (“learn to learn”)
- MAML: find best initialization parameters
  - requires second-order derivative



https://arxiv.org/pdf/2108.10840

Optimal Transport

Joint Distribution Optimal Transportation for Domain Adaptation [JDOT 2017]:

- Assumption: there exists a non-linear transformation between the joint feature/label space distribution of the two domain (source and train) that can be estimated with optimal transport.
- Handle a change in both marginal and conditional distributions
- Transformation T will be expressed through a coupling between both joint distribution:

$$\gamma_0 = \underset{\gamma \in \Pi(\mathcal{P}_s, \mathcal{P}_t)}{\operatorname{argmin}} \int_{(\Omega \times \mathcal{C})^2} \mathcal{D}(\mathbf{x}_1, y_1; \mathbf{x}_2, y_2) d\gamma(\mathbf{x}_1, y_1; \mathbf{x}_2, y_2),$$

- Impossible to find the optimal coupling
- Replace y2 by a proxy f(x2): $\mathcal{P}_t^f = (\mathbf{x}, f(\mathbf{x}))_{\mathbf{x} \sim \mu_t}$ (f: Ω -> C)
- The goal is to estimate a prediction f on the target domain

$$\min_{f, \gamma \in \Delta} \sum_{ij} \mathcal{D}(\mathbf{x}_i^s, \mathbf{y}_i^s; \mathbf{x}_j^t, f(\mathbf{x}_j^t)) \gamma_{ij} \quad \equiv \quad \min_f W_1(\hat{\mathcal{P}}_s, \hat{\mathcal{P}}_t^f)$$

(W1 is the 1-Wasserstein distance for the loss)