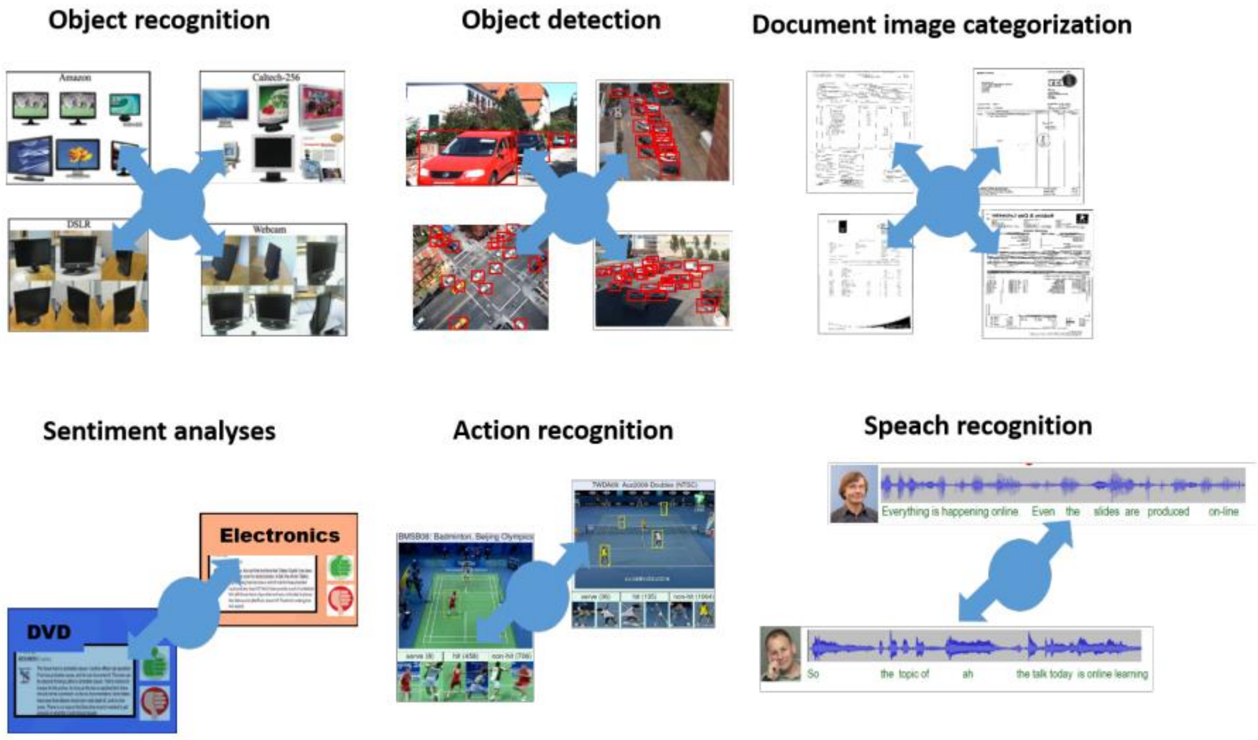
Domain Adaptation for Visual Applications
: A Comprehensive Survey

Domain Adaptation

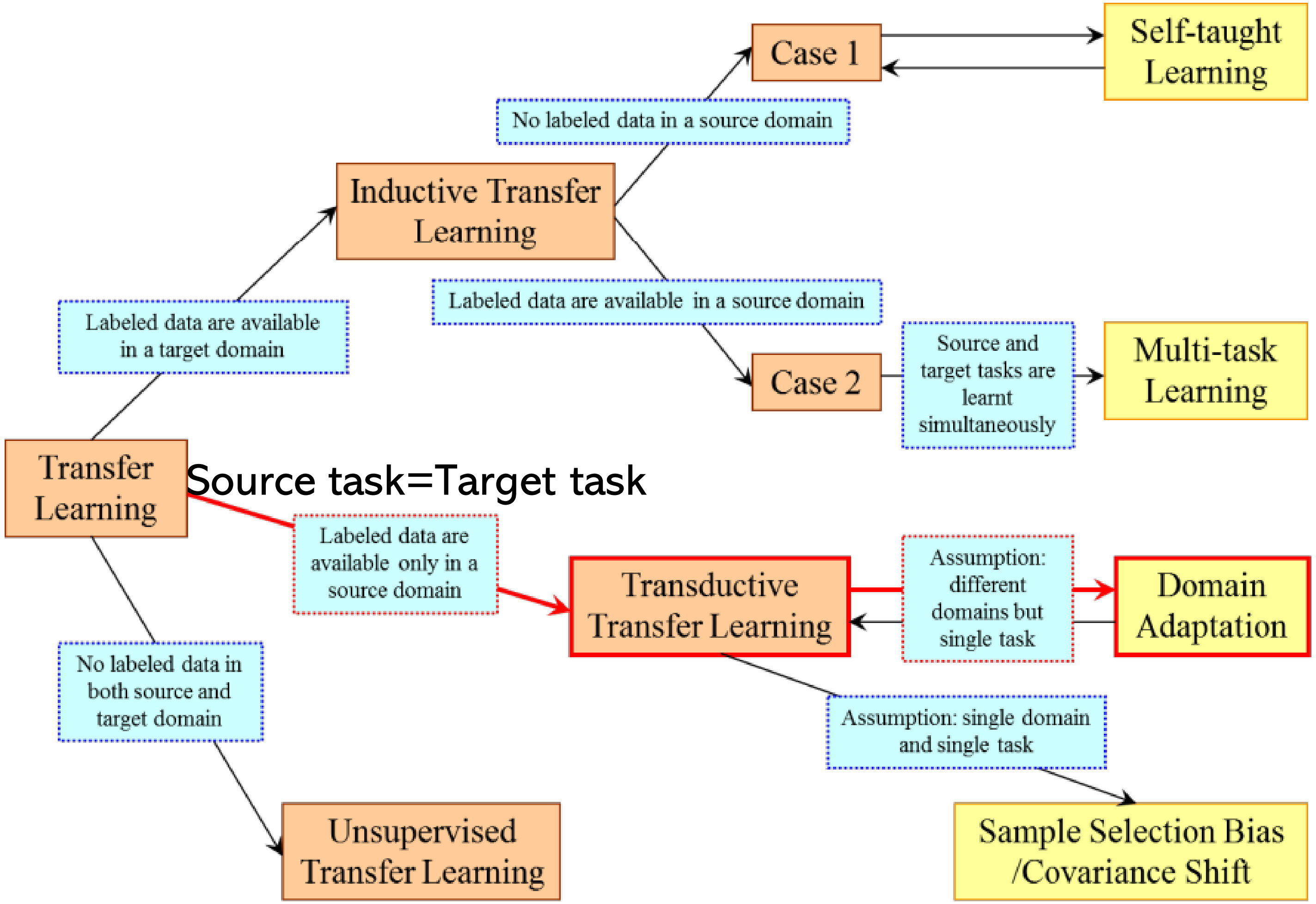
Domain Adaptation

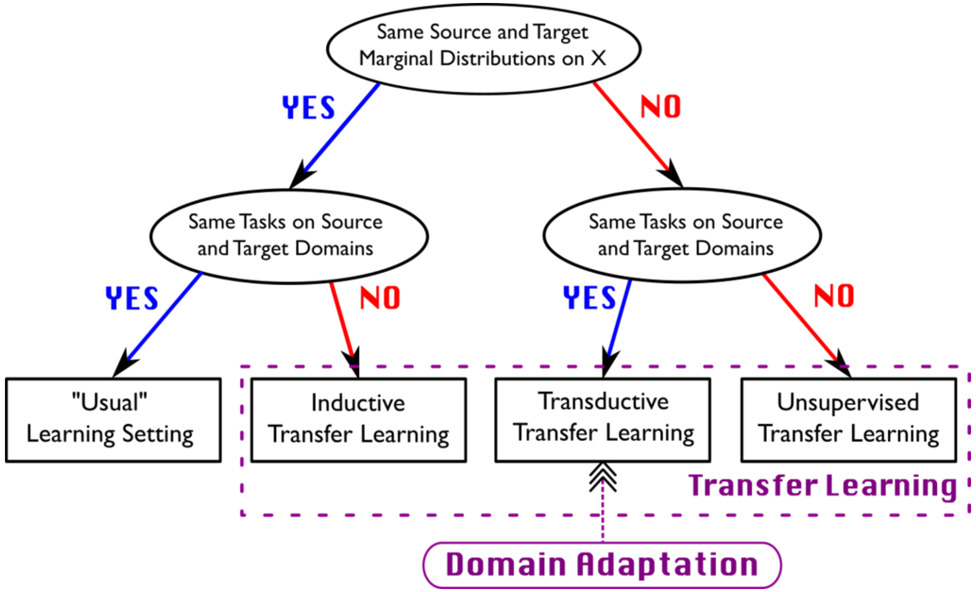
: a particular case of transfer learning (TL) that leverages labeled data in one or more related source domains, to learn a classifier for unseen or unlabeled data in a target domain



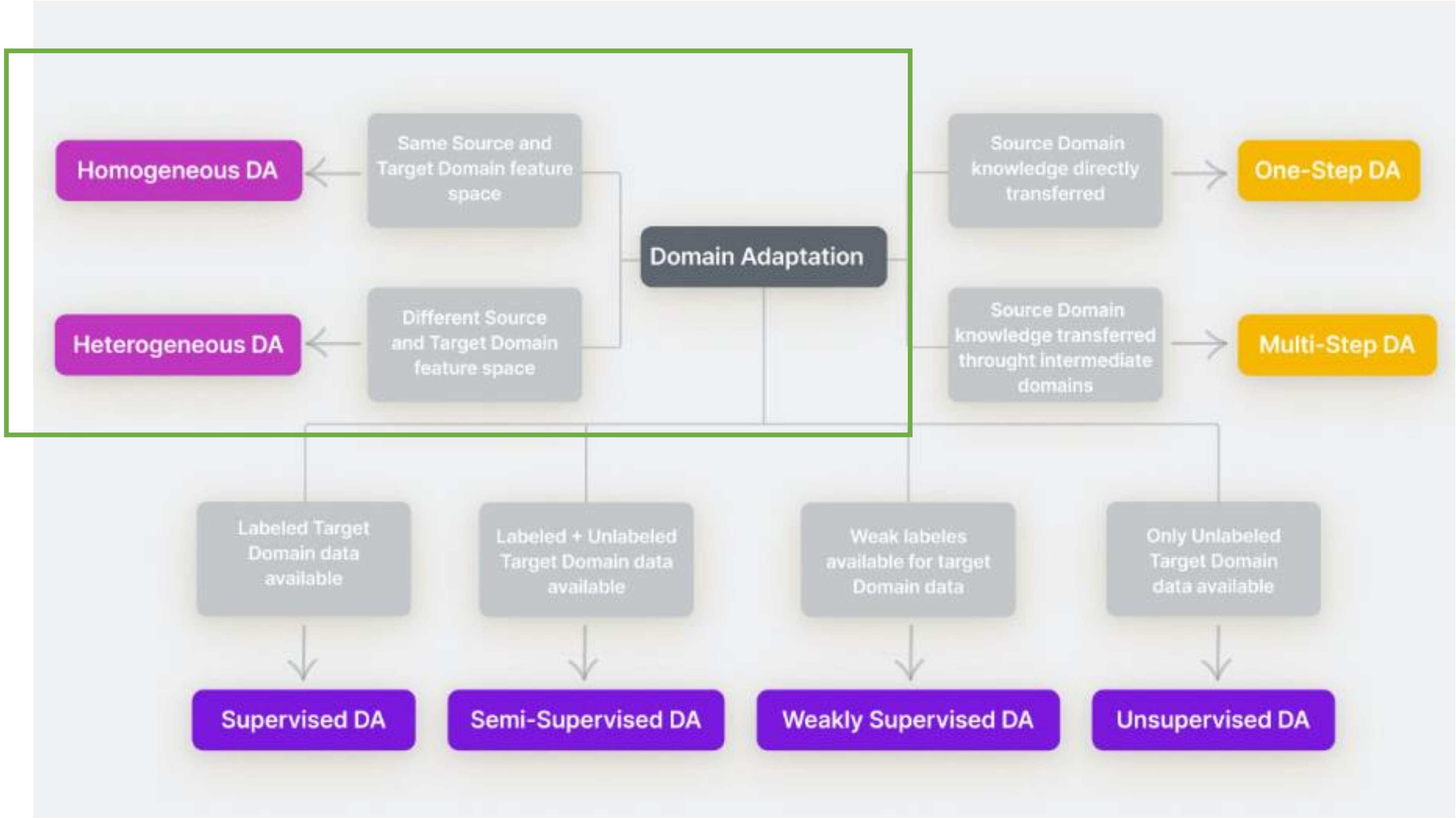
The source domains are assumed to be related to the target domain, but not identical

Transfer learning and Domain adaptation





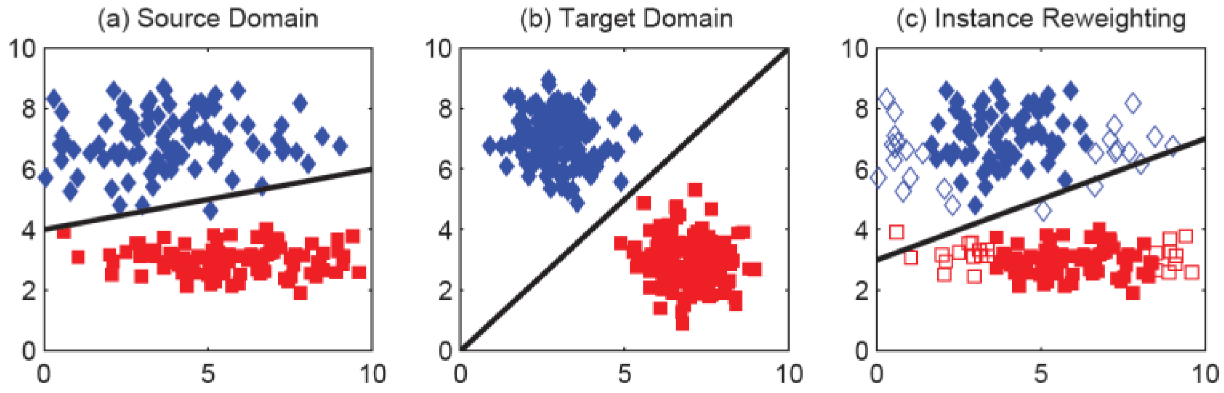
Transfer learning and Domain adaptation

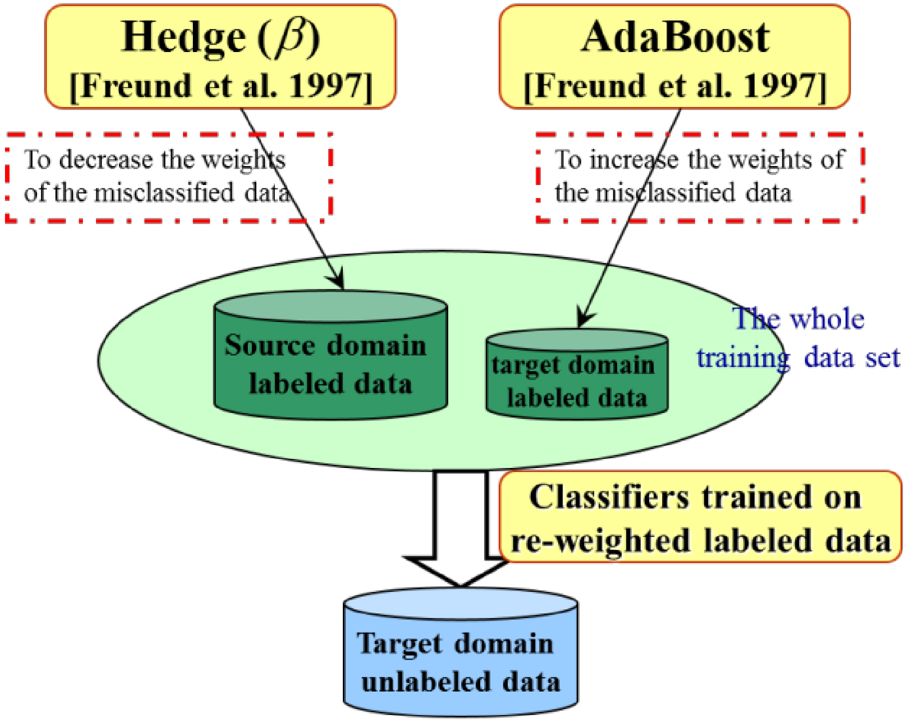


Homogeneous Domain Adaptation

1) Instance re-weighting methods

Source domain과 Target domain의 Data distribution이 다를 때

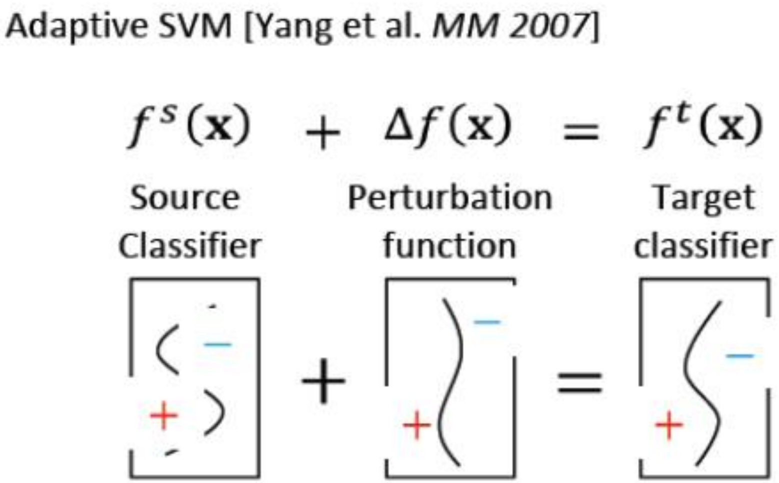


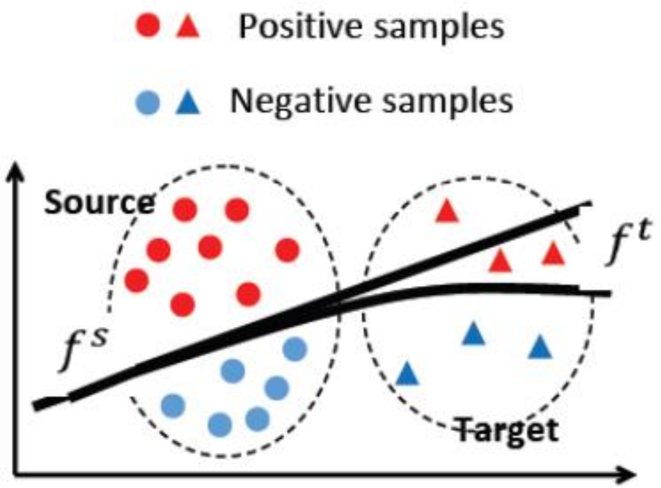


Homogeneous Domain Adaptation

2) Parameter adaptation methods

Adaptive SVM (A-SVM) progressively adjusts the decision boundaries of the source classifiers with the help of a set of perturbation functions built by exploiting predictions on the available labeled target examples





Homogeneous Domain Adaptation

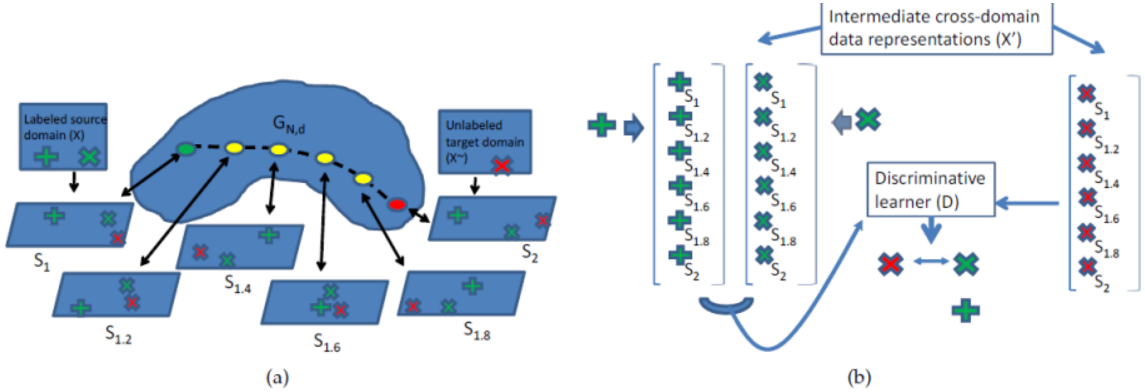
3) Feature augmentation

The original representation x is augmented with itself and a vector of the same size filled with zeros

4) Feature space alignment

Aligning the source features with the target ones

ex) Subspace Alignment: an alignment between the source subspace obtained by PCA and the target PCA subspace



Domains are embedded in d-dimensional linear subspaces

Algorithm 1: Subspace Alignment (SA) [19]

Input: Source data $\mathbf{X}^s$, target data $\mathbf{X}^t$, subspace dimension d

1: $\mathbf{P}_s \leftarrow PCA(\mathbf{X}^s, d)$, $\mathbf{P}_t \leftarrow PCA(\mathbf{X}^t, d)$;

2: $\mathbf{X}_a^s = \mathbf{X}^s \mathbf{P}_s \mathbf{P}_s^\top \mathbf{P}_t$, $\mathbf{X}_a^t = \mathbf{X}^t \mathbf{P}_t$;

Output: Aligned source, $\mathbf{X}_a^s$ and target, $\mathbf{X}_a^t$ data.

Algorithm 2: Correlation Alignment (CORAL) [21]

Input: Source data $\mathbf{X}^s$, target data $\mathbf{X}^t$

1: $\mathbf{C}_s = cov(\mathbf{X}^s) + eye(size(\mathbf{X}^s, 2))$, $\mathbf{C}_t = cov(\mathbf{X}^t) + eye(size(\mathbf{X}^t, 2))$

2: $\mathbf{X}_w^s = \mathbf{X}^s * \mathbf{C}_s^{-1/2}$ (whitening), $\mathbf{X}_a^s = \mathbf{X}_w^s * \mathbf{C}_t^{-1/2}$ (re-coloring)

Output: Source data $\mathbf{X}_a^s$ adjusted to the target.

Homogeneous Domain Adaptation

5) Unsupervised feature transformation

Transfer Component Analysis(TCA) : To discover common latent features having the same marginal distribution across the source and target domains

➔ transformation without using any class label
:After projecting the data in the new space, any classifier trained on the source set can be used to predict labels for the target data

6) Supervised feature transformation

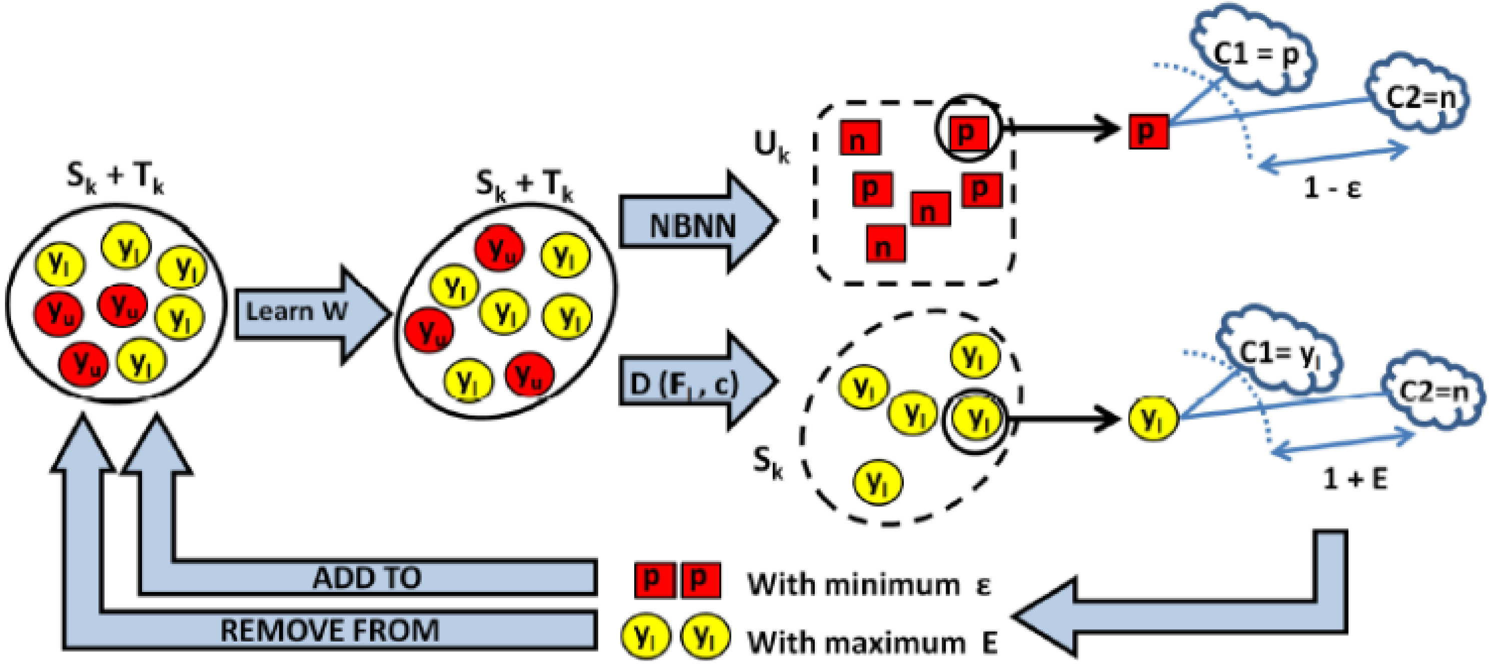
: to capitalize on class labels to learn a better transformation
- maximizing the alignment of the projections with the source labels and, when available, target labels

7) Metric learning based feature transformation
to bridge the relatedness between the source and target domains

Homogeneous Domain Adaptation

7) Metric learning based feature transformation

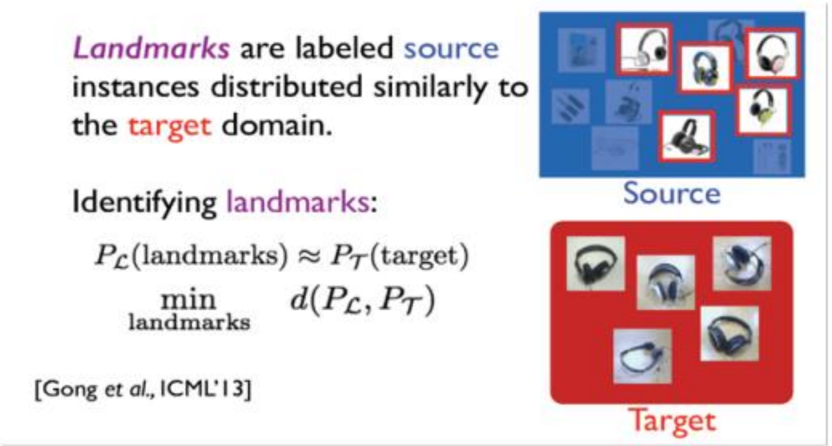
: to bridge the relatedness between the source and target domains



Homogeneous Domain Adaptation

8) Landmark Selection

In order to improve the feature learning process, selecting the most relevant instances from the source



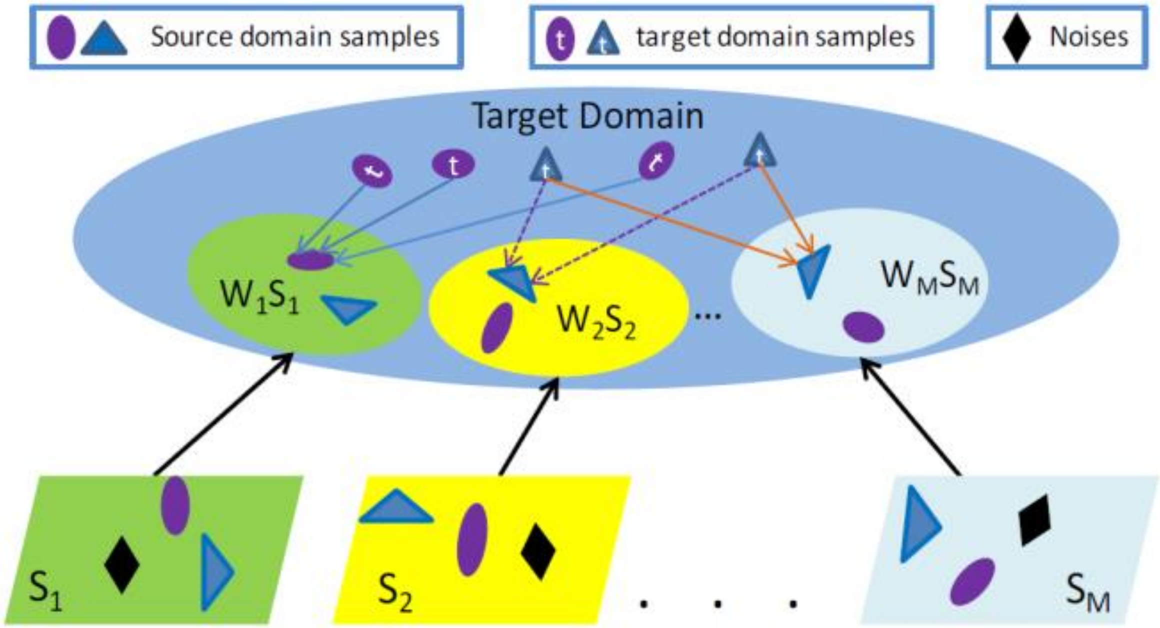
(a)

(b)

Multi-source Domain Adaptation

Multi-source DA models are able to exploit the specificity of each source domain

Source domain weighting
to select those domains that provide the best information transfer and to remove the ones that have more likely negatively impact on the final model



a compact source sample set is formed with a distribution close to the target domain

Heterogeneous Domain Adaptation

Heterogeneous Domain Adaptation
:representation spaces are different for the source and target domains and the tasks are assumed to be the same

strongly related to multi-view learning

: the presence of multiple information sources gives an opportunity to learn better representations (features) by analyzing the views simultaneously (audio and video, image and text)

Heterogeneous Domain Adaptation

1) Methods relying on auxiliary domains
: To exploit feature co-occurrences (e.g. between words and visual features) in the multi-view auxiliary domain

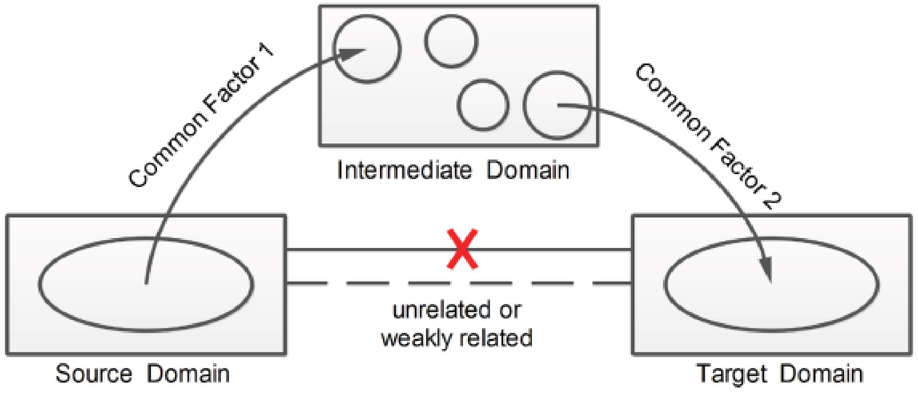
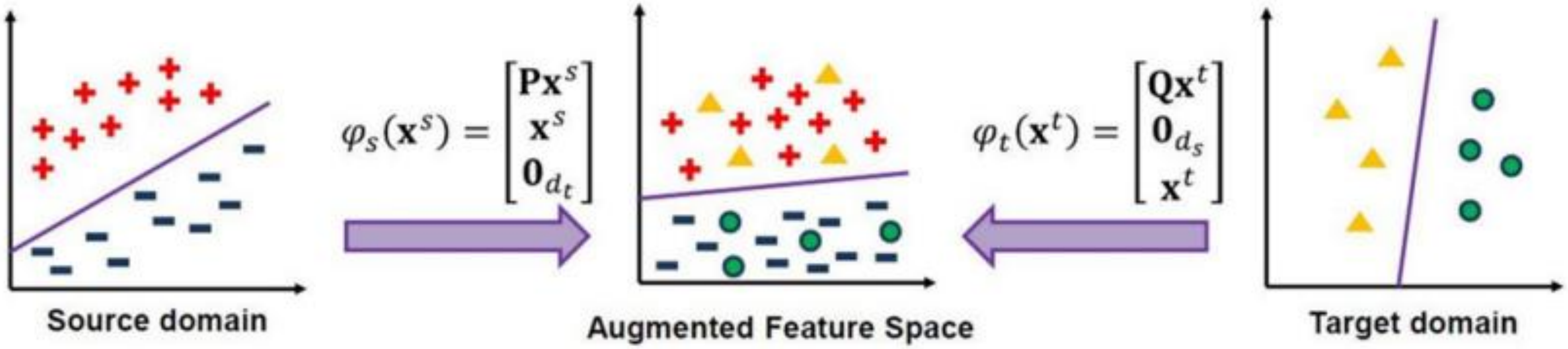


Fig. 12 Heterogeneous DA through an intermediate domain allowing to bridge the gap between features representing the two domains. For example, when the source domain contains text and the target images, the intermediate domain can be built from a set of crawled Web pages containing both text and images. (Image courtesy B. Tan [101]).

Heterogeneous Domain Adaptation

2) Symmetric feature transformation

:learn projections for both the source and target spaces into a common latent (embedding) feature space better suited to learn the task for the target



Heterogeneous Domain Adaptation

3) Asymmetric feature transformation
: aim to learn a projection of the source features into the target space such that the distribution mismatch within each class is minimized

Multiple Outlook MAPping algorithm
: Aims to find the transformation matrix by singular value decomposition process that encourage the marginal distributions within the classes to be aligned while maintaining the structure of the data

Deep domain adaptation methods

Compared to conventional methods, which learn shared feature subspaces or reuse important source instances with shallow representations, deep domain adaptation methods leverage deep networks to learn more transferable representations by embedding domain adaptation in the pipeline of deep learning

Deep DA is a method that utilizes a deep network to enhance the performance of DA

more difficulties to handle the domain differences

Deep domain adaptation methods

1) Shallow methods with deep features

to consider the deep network as feature extractor, where the activations of a layer or several layers of the deep architecture is considered as representation for the input image.

2) Fine-tuning deep CNN architectures

to fine-tune the deep network model on the new type of data and for the new task

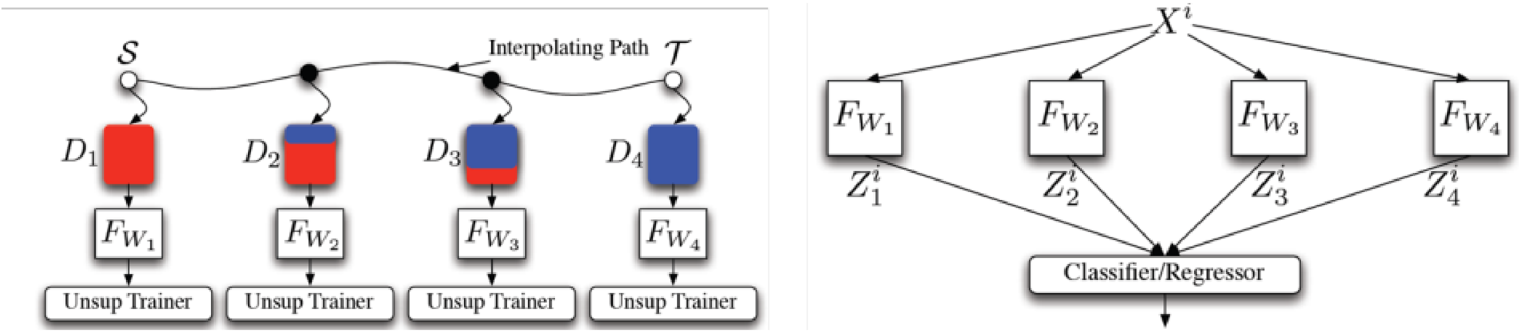
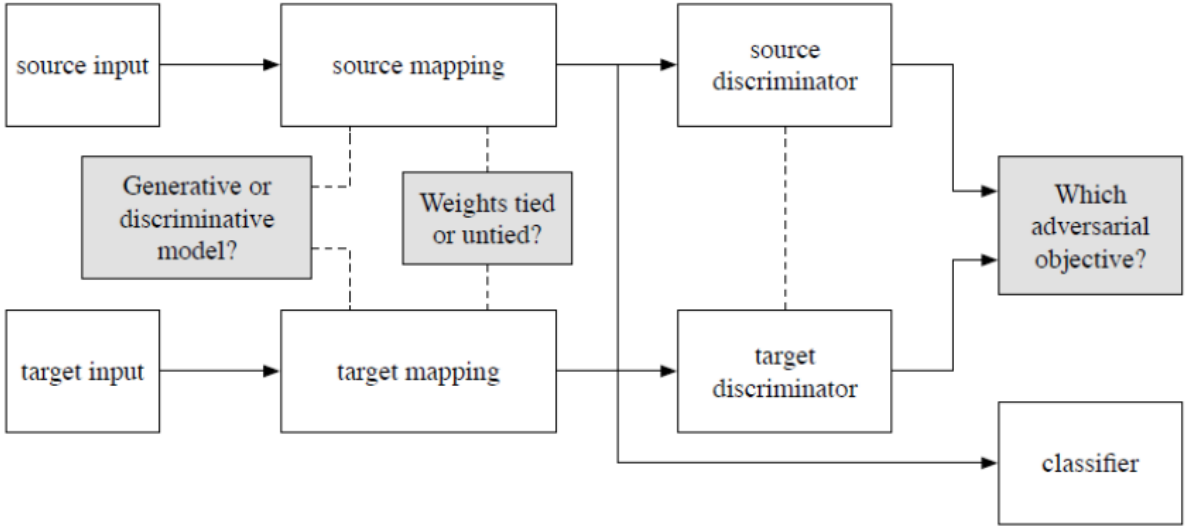


Fig. 17 The DLID model aims in interpolating between domains based on the amount of source and target data used to train each model. (Image courtesy S. Chopra [128]).

Deep domain adaptation methods

DeepDA architectures

- Most DeedDA methods follow a ‘Siamese architectures’ with two streams, representing the source and target models and are trained with a combination of a classification loss and a discrepancy loss or an adversarial loss
→The classification loss depends on the labeled source data
→ The discrepancy loss aims to diminish the shift between the two domains
→ The adversarial loss tries to encourage a common feature space through an adversarial objective with respect toa domain discriminator

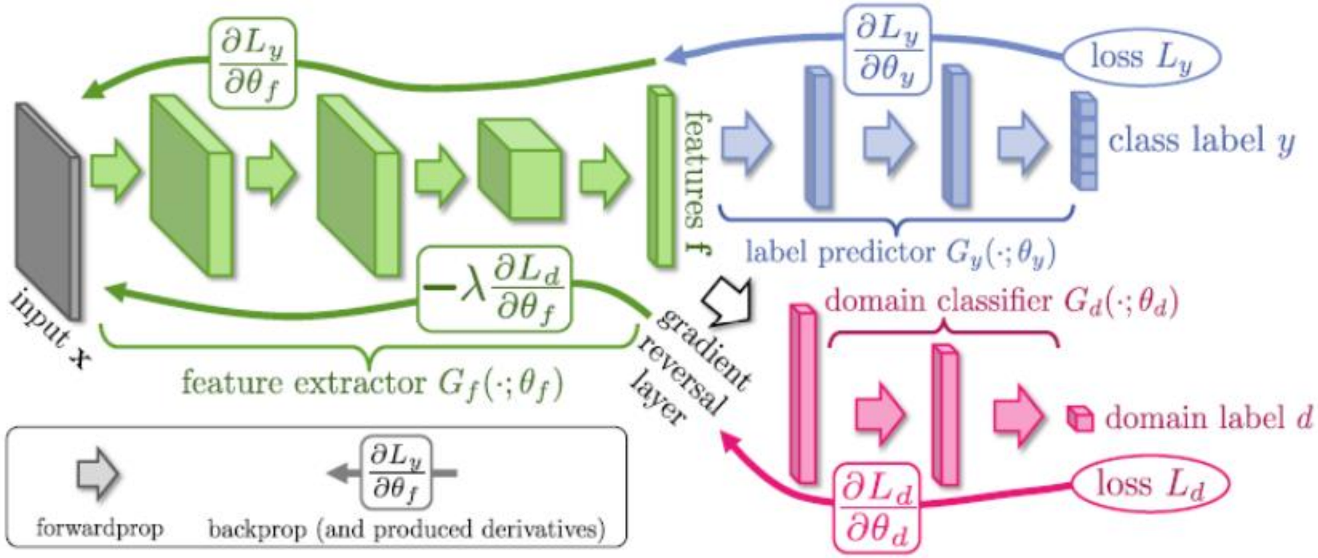


Deep domain adaptation methods

Discrepancy-based methods

- inspired by the shallow feature space transformation
-a discrepancy based on the sum of marginal distributions defined between corresponding activation layers of the two streams of the Siamese architecture

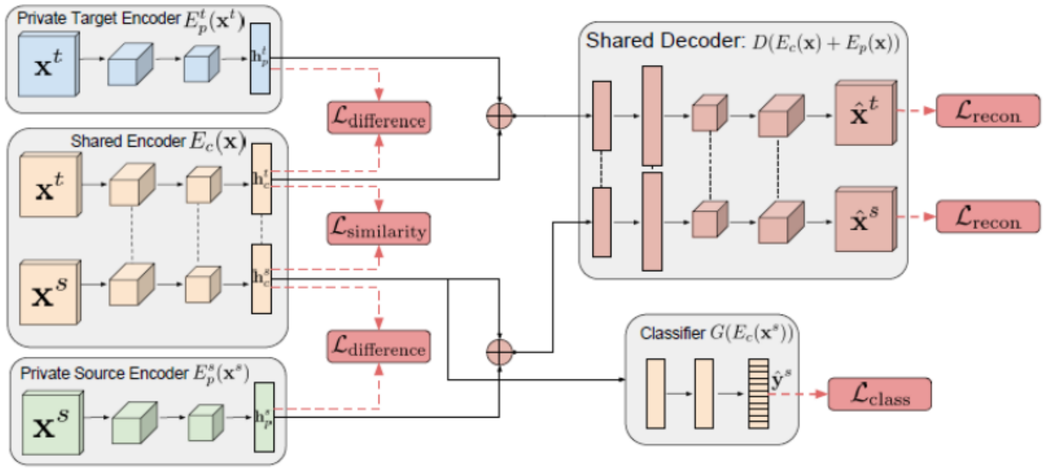
Adversarial discriminative models
-to encourage domain confusion through an adversarial objective with respect to a domain discriminator



Deep domain adaptation methods

Adversarial generative models

- These models combine the discriminative model with a generative component in general based on GANs
- Coupled Generative Adversarial Networks learns a joint distribution of multi-domain
images and enforces a weight sharing constraint to limit the network capacity

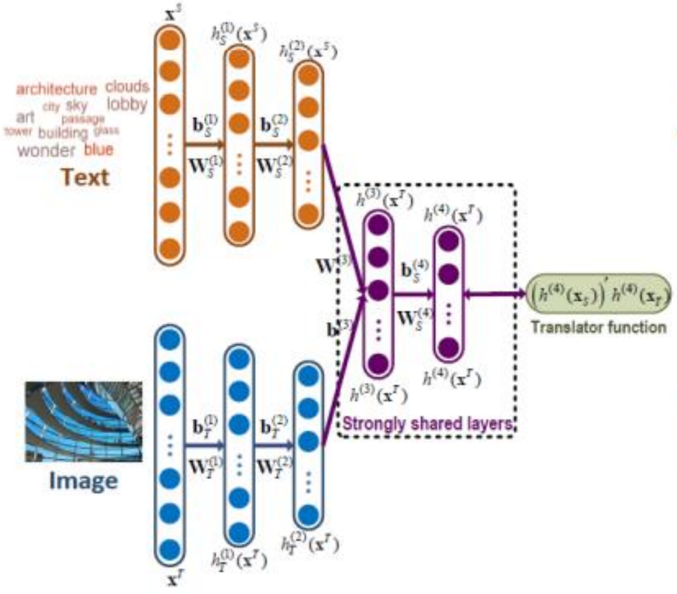


Data reconstruction (encoder-decoder) based methods
Domain Separation Networks (DSN) : a private subspace for each domain captures domain specific properties

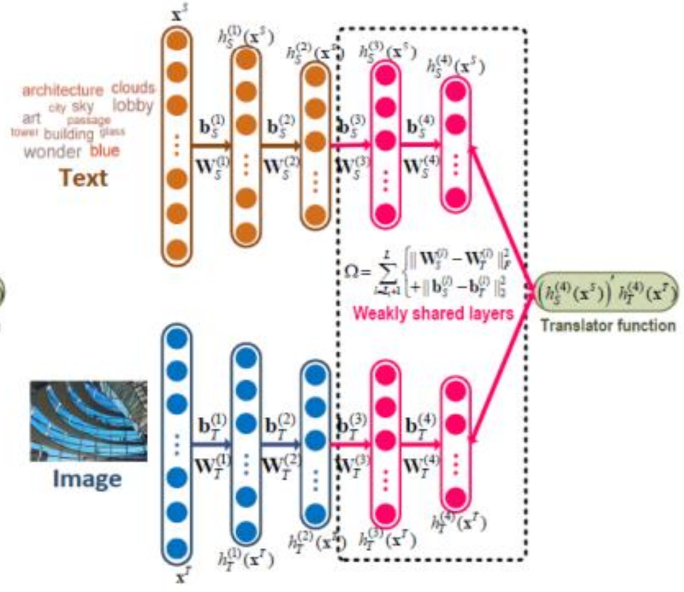
Deep domain adaptation methods

Heterogeneous deepDA

- one stream for each modality, where the weights in the latter stages of the network are shared
- As the prediction layer, a Transfer Neural Decision Forest (Transfer-NDF) is used that performs jointly adaptation and classification



Strongly parameter-shared DTNs



Weakly parameter-shared DTNs