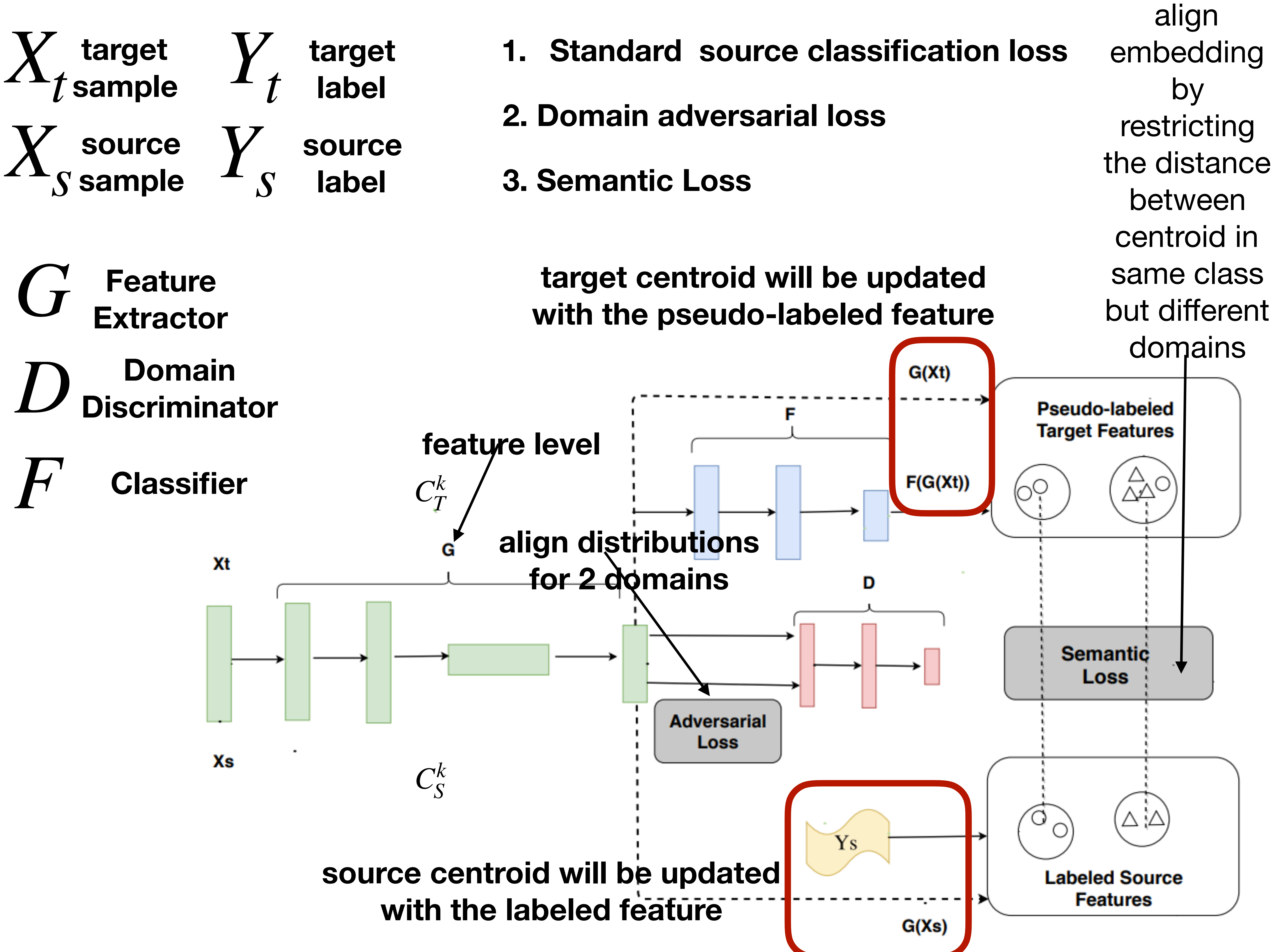
# Learning Semantic Representations for Unsupervised Domain Adaptation

Moving Semantic Transfer Network (MSTN)

$$f : X_T \rightarrow Y_T$$

“Ultimate goal is to develop a deep neural network that is able to  predict labels for the samples from target domain”



$X_t$ target sample
$Y_t$ target label
$X_S$ source sample
$Y_S$ source label
1. Standard source classification loss
2. Domain adversarial loss
3. Semantic Loss
align embedding by restricting the distance between centroid in same class but different domains
$G$ Feature Extractor
$D$ Domain Discriminator
$F$ Classifier
target centroid will be updated with the pseudo-labeled feature
G(Xt)
F
F(G(Xt))
Pseudo-labeled Target Features
feature level
$C_T^k$
G
Xt
align distributions for 2 domains
D
Semantic Loss
Adversarial Loss
Xs
$C_S^k$
Ys
Labeled Source Features
source centroid will be updated with the labeled feature
G(Xs)

$$f = F \circ G$$

**a visual classifier is trained by minimizing the source classification error and the discrepancy between source domain and target domain**

domain adversarial similarity loss
(discrepancy between source/target domain)

cross entropy loss
(source classification error)

$$\mathcal{L}(\mathcal{X}_S, \mathcal{Y}_S, \mathcal{X}_T) = \boxed{\mathcal{L}_C(\mathcal{X}_S, \mathcal{Y}_S)} + \boxed{\lambda \mathcal{L}_{DC}(\mathcal{X}_S, \mathcal{X}_T)}$$
$$\boxed{+ \gamma \mathcal{L}_{SM}(\mathcal{X}_S, \mathcal{Y}_S, \mathcal{X}_T),}$$

making alignment semantic
(centroid alignment)

$$\mathcal{L} = \underbrace{\mathbb{E}_{(x,y)\sim D_S}[J(f(x),y)]}_{\mathcal{L}_C(\mathcal{X}_S,\mathcal{Y}_S)} + \lambda \underbrace{d(\mathcal{X}_S,\mathcal{X}_T)}_{\mathcal{L}_{DC}(\mathcal{X}_S,\mathcal{X}_T)} \quad (1)$$

**1. Standard source classification loss (cross entropy loss) + divergence between two domains (domain adversarial similarity loss)**

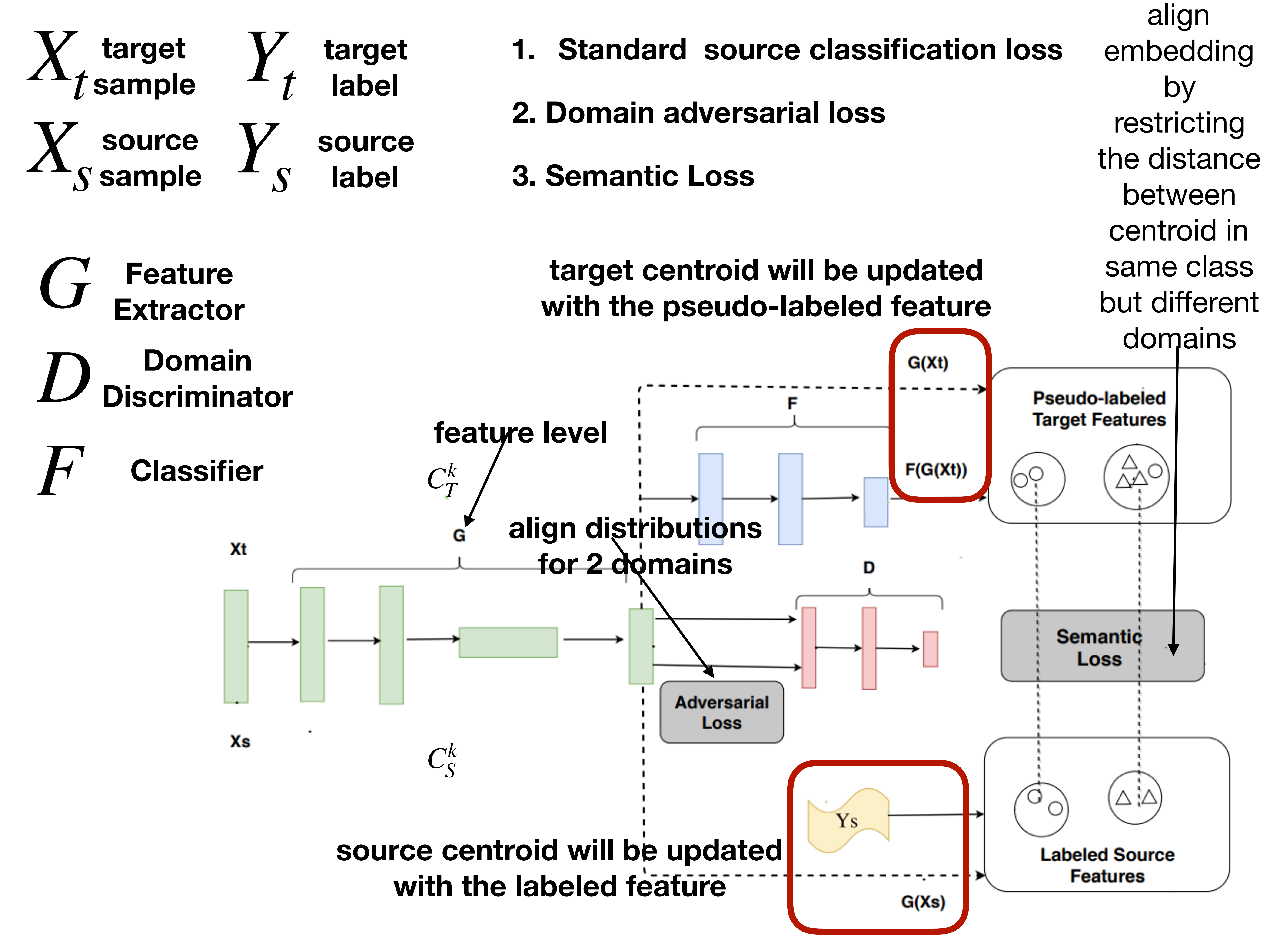
$$d(\mathcal{X}_S,\mathcal{X}_T) = \mathbb{E}_{x\sim D_S}[\log(1 - \mathrm{D} \circ \mathrm{G}(\mathrm{x}))] \\ \mathbb{E}_{x\sim D_T}[\log(\mathrm{D} \circ \mathrm{G}(\mathrm{x}))] \quad (2)$$

**domain adversarial similarity loss (Domain classifier D to tell whether features from G arise from source or target domain)**

**discriminability**

$$\mathcal{L}_{SM}^{UDA}(\mathcal{X}_S,\mathcal{Y}_S,\mathcal{X}_T) = \underbrace{\sum_{k=1}^{K} \Phi(C_S^k, C_T^k)}_{\mathcal{L}_{SM}(\mathcal{X}_S,\mathcal{Y}_S,\mathcal{X}_T)}, \quad (4)$$

**3. making alignment semantic with pseudo labeled target domain (centroid alignment)**


$X_t$ target sample
$Y_t$ target label
$X_S$ source sample
$Y_S$ source label
1. Standard source classification loss
2. Domain adversarial loss
3. Semantic Loss
align embedding by restricting the distance between centroid in same class but different domains
$G$ Feature Extractor
$D$ Domain Discriminator
$F$ Classifier
target centroid will be updated with the pseudo-labeled feature
G(Xt)
F
Pseudo-labeled Target Features
feature level
$C_T^k$
F(G(Xt))
G
align distributions for 2 domains
Xt
D
Semantic Loss
Adversarial Loss
Xs
$C_S^k$
Ys
source centroid will be updated with the labeled feature
Labeled Source Features
G(Xs)

**Algorithm 1** Moving semantic transfer loss computation in iteration $t$ in our model. $K$ is the number of classes.

**Input:** Labeled set S, Unlabeled set T, $N$ is the batch size, Training classifier $f$, Global centroids for two domains: $\{C_S^k\}_{k=1}^K$ and $\{C_T^k\}_{k=1}^K$

$S_t = \textsc{RandomSample}(S, N)$

$T_t = \textsc{RandomSample}(T, N)$

$\widehat{\boldsymbol{T_t}}$ =Labeling(G,$f$,$T_t$)

$\mathcal{L}_{SM} = 0$

**for** $k = 1$ to $K$ **do**

feature extraction

$\quad C_{S_{(t)}}^k \leftarrow \frac{1}{|S_t^k|} \sum_{(x_i,y_i)\in S_t^k} G(x_i)$ (From Scratch)

$\quad C_{T_{(t)}}^k \leftarrow \frac{1}{|\widehat{\boldsymbol{T_t^k}}|} \sum_{(x_i,y_i)\in \widehat{\boldsymbol{T_t^k}}} G(x_i)$ (From Scratch)

$\quad \boldsymbol{C_S^k \leftarrow \theta C_S^k + (1-\theta) C_{S_{(t)}}^k}$ (Moving Average) align distributions

$\quad \boldsymbol{C_T^k \leftarrow \theta C_T^k + (1-\theta) C_{T_{(t)}}^k}$ (Moving Average)

$\quad \mathcal{L}_{SM} \leftarrow \mathcal{L}_{SM} + \Phi(C_S^k, C_T^k)$ centroid alignment

**end for**

**return** $\mathcal{L}_{SM}$