

Adversarial Discriminative Domain Adaptation

Eric Tzeng

University of California, Berkeley
etzeng@eecs.berkeley.edu

Judy Hoffman

Stanford University
jhoffman@cs.stanford.edu

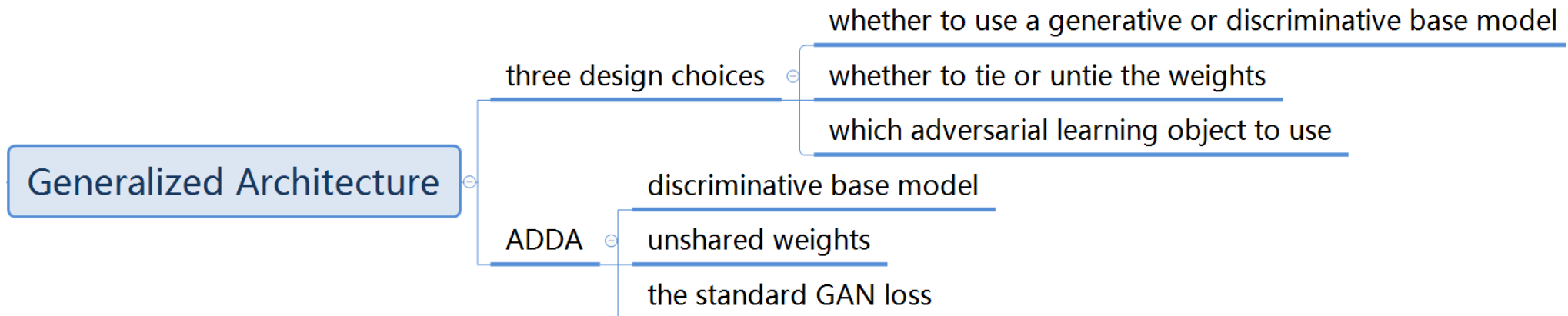
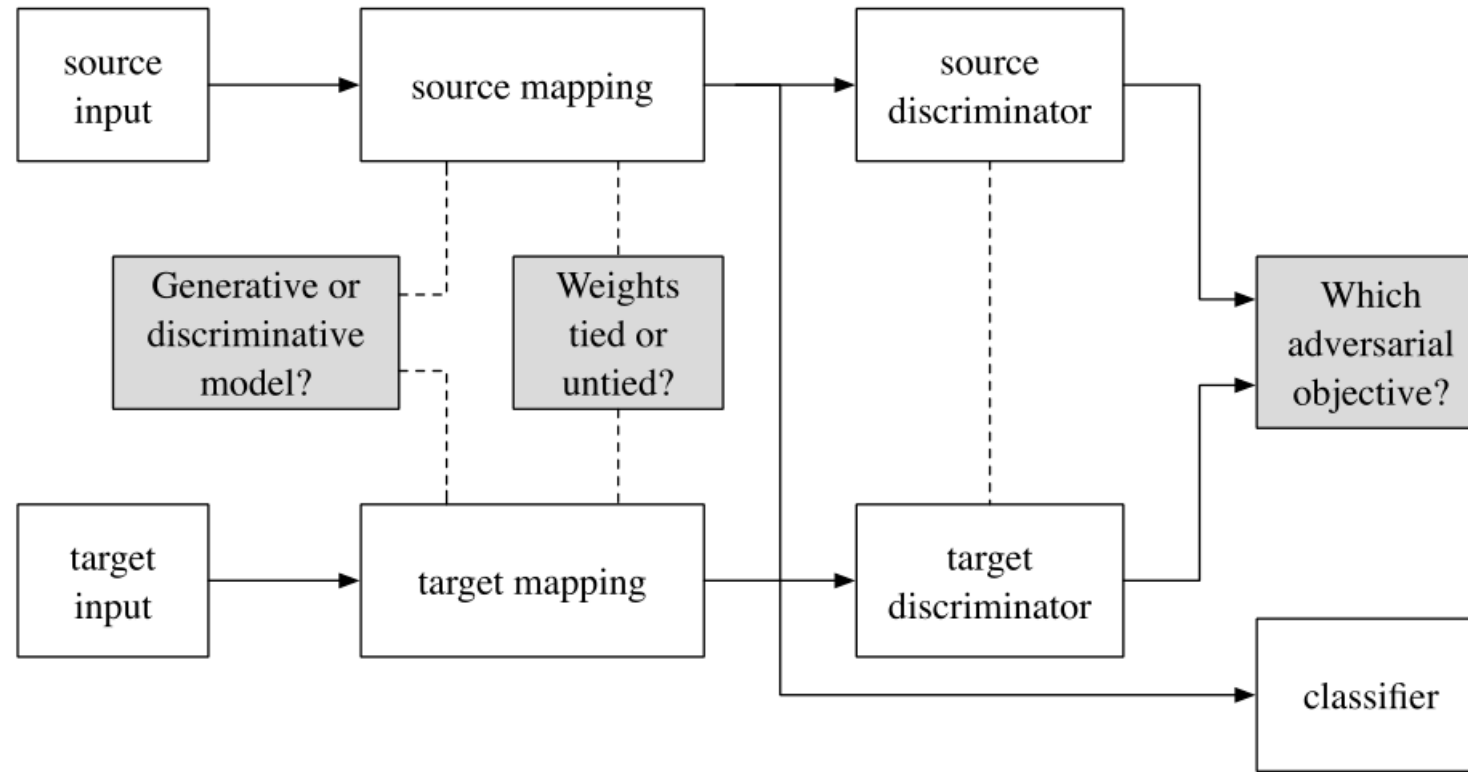
Kate Saenko

Boston University
saenko@bu.edu

Trevor Darrell

University of California, Berkeley
trevor@eecs.berkeley.edu

Generalized architecture for adversarial domain adaptation



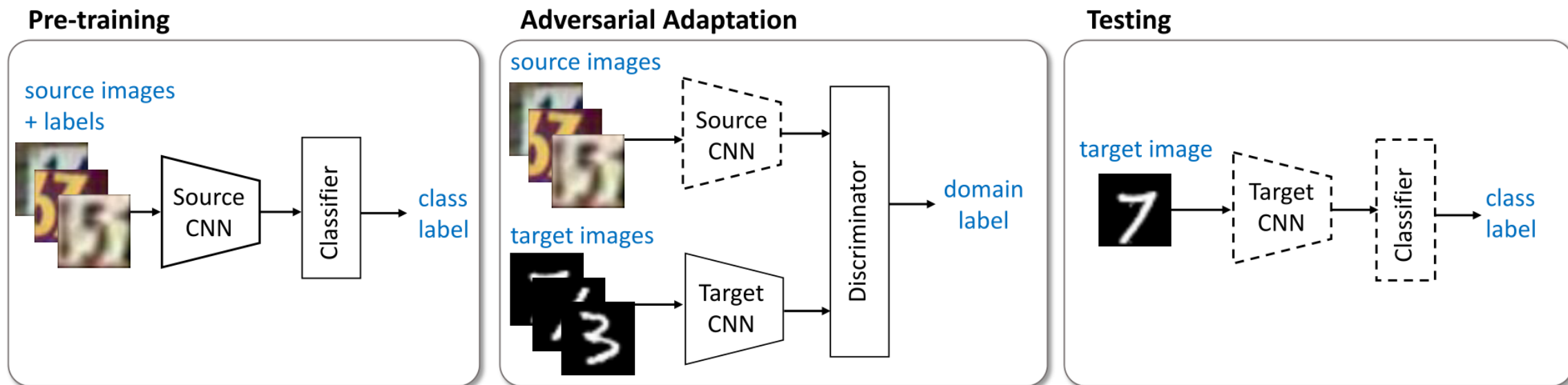


Figure 3: An overview of our proposed Adversarial Discriminative Domain Adaptation (ADDA) approach. We first pre-train a source encoder CNN using labeled source image examples. Next, we perform adversarial adaptation by learning a target encoder CNN such that a discriminator that sees encoded source and target examples cannot reliably predict their domain label. During testing, target images are mapped with the target encoder to the shared feature space and classified by the source classifier. Dashed lines indicate fixed network parameters.

1.源域的分类误差项

$$\min_{M_s, C} \mathcal{L}_{\text{cls}}(\mathbf{X}_s, Y_t) = \mathbb{E}_{(\mathbf{x}_s, y_s) \sim (\mathbf{X}_s, Y_t)} - \sum_{k=1}^K \mathbb{1}_{[k=y_s]} \log C(M_s(\mathbf{x}_s)) \quad (1)$$

X_s : source domain images

Y_s : labels of source images

X_t : target domain images

M_s : source representation

C_s : source classifier

M_t : target representation

C_t : classifier that can classify target images into one of K categories

3.生成器在目标域的误差项

$$\mathcal{L}_{\text{adv}_M}(\mathbf{X}_s, \mathbf{X}_t, D) = -\mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}_t} [\log D(M_t(\mathbf{x}_t))]. \quad (7)$$

2.域分类器的分类误差项

$$\mathcal{L}_{\text{adv}_D}(\mathbf{X}_s, \mathbf{X}_t, M_s, M_t) = -\mathbb{E}_{\mathbf{x}_s \sim \mathbf{X}_s} [\log D(M_s(\mathbf{x}_s))] - \mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}_t} [\log(1 - D(M_t(\mathbf{x}_t)))] \quad (2)$$

$$\min_{M_s, C} \mathcal{L}_{\text{cls}}(\mathbf{X}_s, Y_s) = -\mathbb{E}_{(\mathbf{x}_s, y_s) \sim (\mathbf{X}_s, Y_s)} \sum_{k=1}^K \mathbb{1}_{[k=y_s]} \log C(M_s(\mathbf{x}_s))$$

$$\min_D \mathcal{L}_{\text{adv}_D}(\mathbf{X}_s, \mathbf{X}_t, M_s, M_t) = -\mathbb{E}_{\mathbf{x}_s \sim \mathbf{X}_s} [\log D(M_s(\mathbf{x}_s))] - \mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}_t} [\log(1 - D(M_t(\mathbf{x}_t)))]$$

$$\min_{M_s, M_t} \mathcal{L}_{\text{adv}_M}(\mathbf{X}_s, \mathbf{X}_t, D) = -\mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}_t} [\log D(M_t(\mathbf{x}_t))]. \quad (9)$$

Base model

生成式模型还是判别式模型

生成式模型用随机噪声作为输入，在图像空间产生样本。一般会使用判别器的中间层特征来训练一个任务相关的分类器。

判别模型则会直接将图片映射到特征空间，然后输入到分类器中进行训练。

Adversarial loss

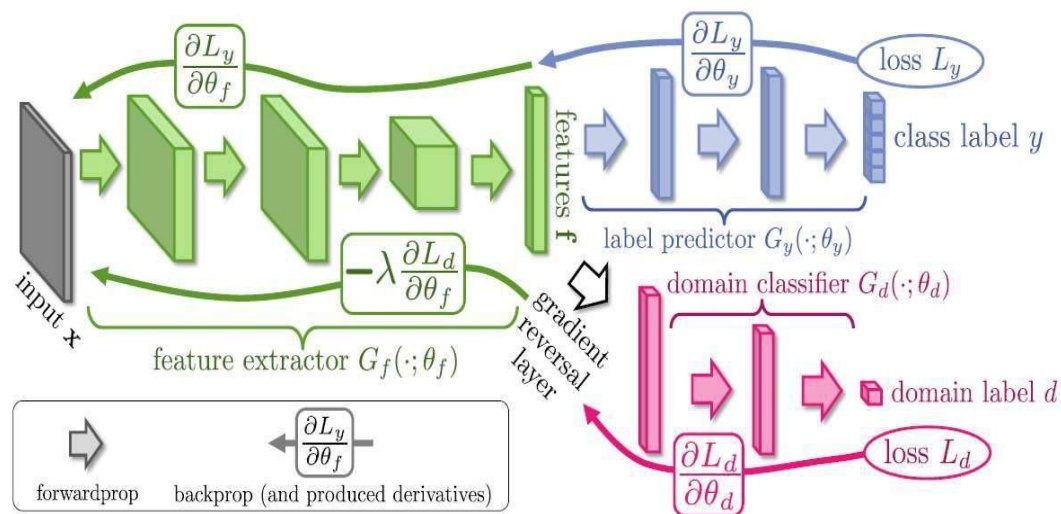
Adversarial loss

minmax loss $\ominus \mathcal{L}_{\text{adv}_M} = -\mathcal{L}_{\text{adv}_D}$

GAN loss $\ominus \mathcal{L}_{\text{adv}_M}(\mathbf{X}_s, \mathbf{X}_t, D) = -\mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}_t} [\log D(M_t(\mathbf{x}_t))]$

$$\mathcal{L}_{\text{adv}_M}(\mathbf{X}_s, \mathbf{X}_t, D) = - \sum_{d \in \{s, t\}} \mathbb{E}_{\mathbf{x}_d \sim \mathbf{X}_d} \left[\frac{1}{2} \log D(M_d(\mathbf{x}_d)) + \frac{1}{2} \log(1 - D(M_d(\mathbf{x}_d))) \right]$$

domain confusion loss \ominus



判别器损失函数：

$$-\mathbb{E}_{x \sim P_r} [\log D(x)] - \mathbb{E}_{x \sim P_g} [\log(1 - D(x))] \quad (\text{公式1})$$

生成器损失函数：

$$\mathbb{E}_{x \sim P_g} [\log(1 - D(x))] \quad (\text{公式2})$$

$$\mathbb{E}_{x \sim P_g} [-\log D(x)] \quad (\text{公式3})$$

GAN-1

GAN 的目标是实现生成器 (Generator) 和判别器 (Discriminator) 之间的平衡

记生成器为 G , 判别器为 D , 真实 (real) 数据分布为 P_r , 高斯随机噪声分布为 P_z , 生成器伪造的 (faked) 数据分布为 $P_f = G(P_z)$ 。

$$KL(P_r || P_f) = \mathbb{E}_{x \sim P_r} \left[\log \frac{P_r(x)}{P_f(x)} \right] = \int P_r(x) \log \frac{P_r(x)}{P_f(x)} dx \quad JS(P_r || P_f) = \frac{1}{2} KL \left(P_r || \frac{P_r + P_f}{2} \right) + \frac{1}{2} KL \left(P_f || \frac{P_r + P_f}{2} \right)$$

判别器D, 是一个二分类问题, 交叉熵损失函数作为优化器的优化目标, 令D(x)为图片x的真实概率, 那么损失函数为:

生成器G目标是与判别器D唱反调, 既然D的目标是最小化LD, 那么G的目标就是最小化-LD:

$$L_D = -E_{x \sim P_r} [\log(D(x))] - E_{z \sim P_z} [\log(1 - D(G(z)))] \quad L_G = E_{x \sim P_r} [\log(D(x))] + E_{z \sim P_z} [\log(1 - D(G(z)))]$$

$$\min_G \max_D V(G, D) = E_{x \sim P_r} [\log(D(x))] + E_{z \sim P_z} [\log(1 - D(G(z)))]$$

source and target mappings

Consider a layered representations where each layer parameters are denoted as, M_s^ℓ or M_t^ℓ , for a given set of equivalent layers, $\{\ell_1, \dots, \ell_n\}$. Then the space of constraints explored in the literature can be described through layerwise equality constraints as follows:

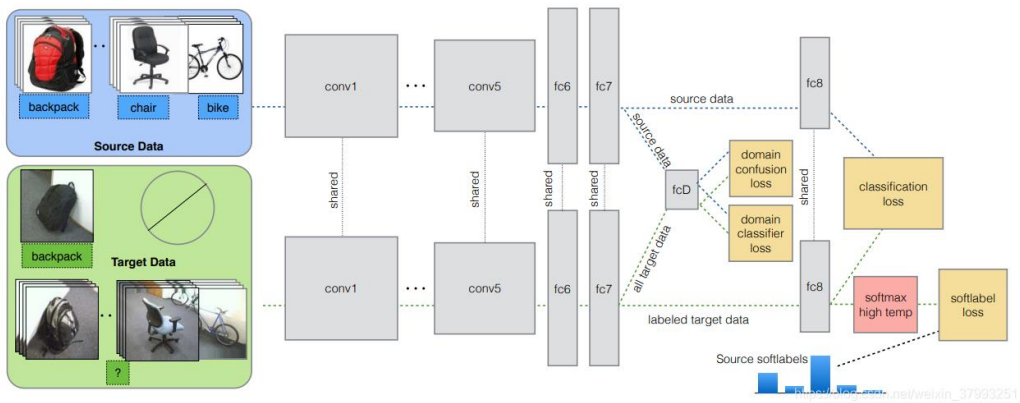
$$\psi(M_s, M_t) \triangleq \{\psi_{\ell_i}(M_s^{\ell_i}, M_t^{\ell_i})\}_{i \in \{1 \dots n\}} \quad (4)$$

dently. A very common form of constraint is source and target layerwise equality:

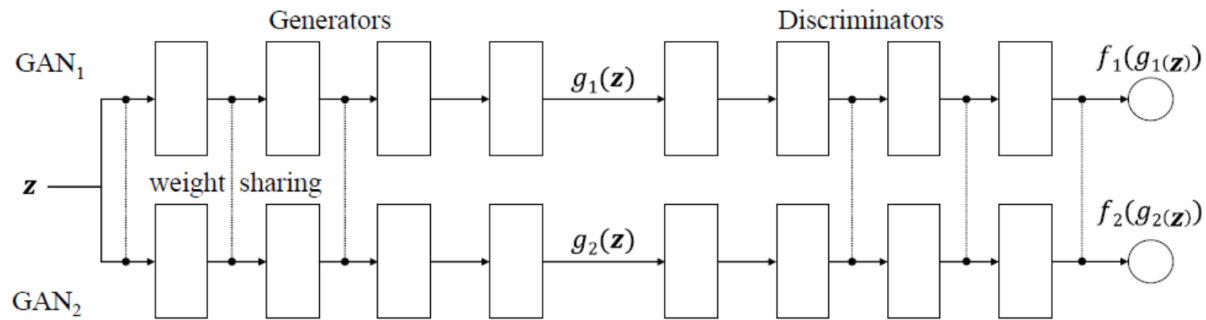
$$\psi_{\ell_i}(M_s^{\ell_i}, M_t^{\ell_i}) = (M_s^{\ell_i} = M_t^{\ell_i}). \quad (5)$$

Method	Base model	Weight sharing	Adversarial loss
Gradient reversal [16]	discriminative	shared	minimax
Domain confusion [12]	discriminative	shared	confusion
CoGAN [13]	generative	unshared	GAN
ADDA (Ours)	discriminative	unshared	GAN

Table 1: Overview of adversarial domain adaption methods and their various properties. Viewing methods under a unified framework enables us to easily propose a new adaptation method, adversarial discriminative domain adaptation (ADDA).



domain confusion



CoGAN

Experiments

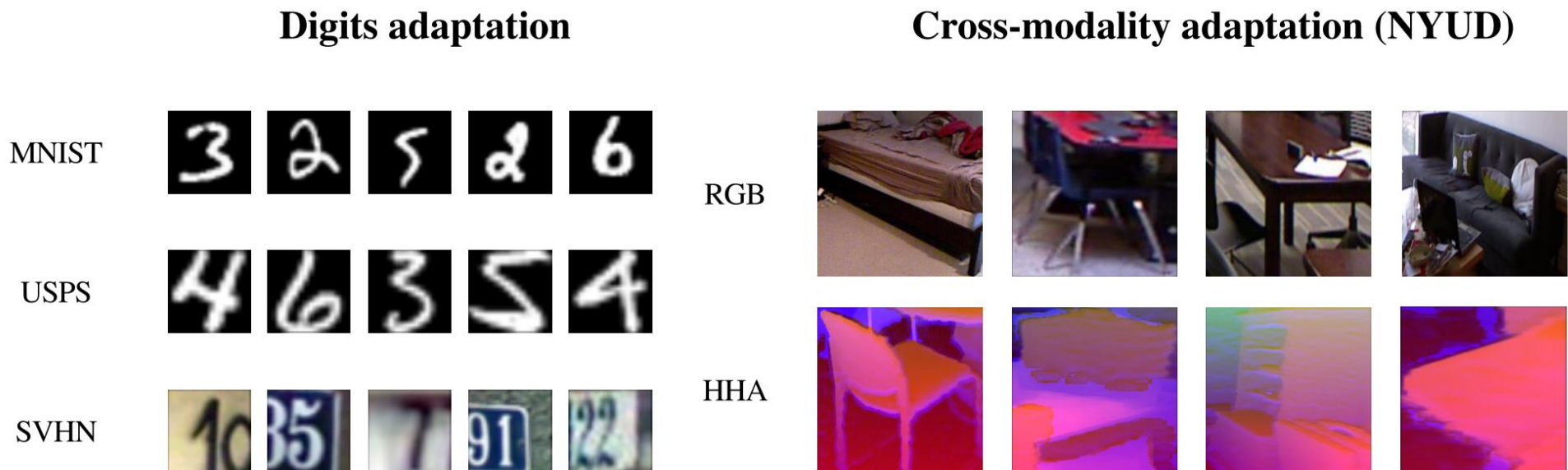




















Figure 4: We evaluate ADDA on unsupervised adaptation across four domain shifts in two different settings. The first setting is adaptation between the MNIST, USPS, and SVHN datasets (left). The second setting is a challenging cross-modality adaptation task between RGB and depth modalities from the NYU depth dataset (right).

Method	MNIST \rightarrow USPS	USPS \rightarrow MNIST	SVHN \rightarrow MNIST
	   \rightarrow   	   \rightarrow   	   \rightarrow   
Source only	0.752 ± 0.016	0.571 ± 0.017	0.601 ± 0.011
Gradient reversal	0.771 ± 0.018	0.730 ± 0.020	0.739 [16]
Domain confusion	0.791 ± 0.005	0.665 ± 0.033	0.681 ± 0.003
CoGAN	0.912 ± 0.008	0.891 ± 0.008	did not converge
ADDA (Ours)	0.894 ± 0.002	0.901 ± 0.008	0.760 ± 0.018

	bathtub	bed	bookshelf	box	chair	counter	desk	door	dresser	garbage bin	lamp	monitor	night stand	pillow	sink	sofa	table	television	toilet	overall
# of instances	19	96	87	210	611	103	122	129	25	55	144	37	51	276	47	129	210	33	17	2401
Source only	0.000	0.010	0.011	0.124	0.188	0.029	0.041	0.047	0.000	0.000	0.069	0.000	0.039	0.587	0.000	0.008	0.010	0.000	0.000	0.139
ADDA (Ours)	0.000	0.146	0.046	0.229	0.344	0.447	0.025	0.023	0.000	0.018	0.292	0.081	0.020	0.297	0.021	0.116	0.143	0.091	0.000	0.211
Train on target	0.105	0.531	0.494	0.295	0.619	0.573	0.057	0.636	0.120	0.291	0.576	0.189	0.235	0.630	0.362	0.248	0.357	0.303	0.647	0.468

Table 3: Adaptation results on the NYUD [20] dataset, using RGB images from the train set as source and depth images from the val set as target domains. We report here per class accuracy due to the large class imbalance in our target set (indicated in # instances). Overall our method improves average per category accuracy from 13.9% to 21.1%.

首先是预训练阶段（Pre-training Stage），源域上利用有标记数据训练，采用交叉熵损失：

推广到隐私保护

$$\min_{M_s, C} \mathcal{L}_{ce}(\mathcal{X}_s, \mathcal{Y}_s) = -E_{(x_s, y_s) \sim (\mathcal{X}_s, \mathcal{Y}_s)} \sum_{k=1}^K \mathcal{I}\{y_s = k\} \log C(M_s(x_s))$$

其中 M_s 为源域的特征提取器， C 为源域的分类器。

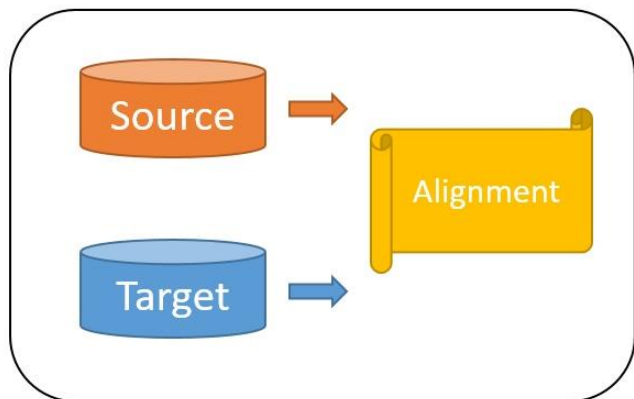
第一步训练域判别器（Discriminator）将源域的特征和目标域的特征进行区分开， D 代表域判别器：

$$\min_D \mathcal{L}_{adv_D}(\mathcal{X}_s, \mathcal{Y}_s, M_s, M_t) = -E_{x_s \sim \mathcal{X}_s} [\log D(M_s(x_s))] - E_{x_t \sim \mathcal{X}_t} [\log(1 - D(M_t(x_t)))]$$

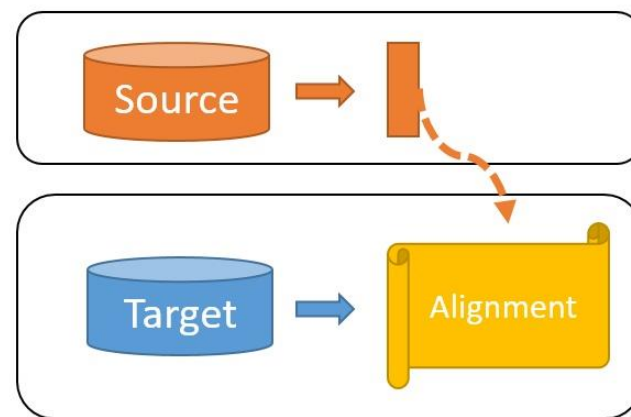
第二步，训练 M_t ，使得 $M_t(x_t)$ 让判别器尽可能分不开：

$$\min_{M_t} \mathcal{L}_{adv_M}(\mathcal{X}_s, \mathcal{X}_t, D) = -E_{x_t \sim \mathcal{X}_t} [\log D(M_t(x_t))]$$

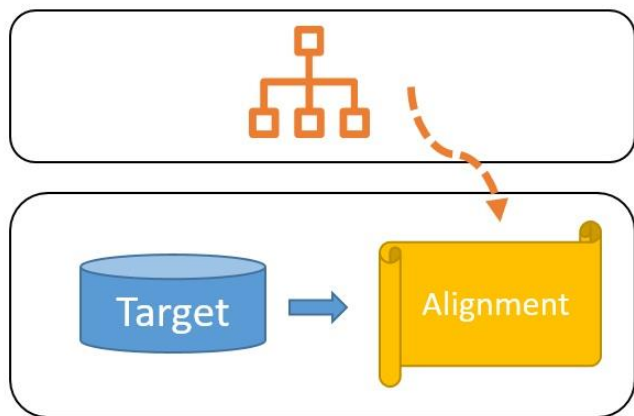
重复以上两步，直到收敛。



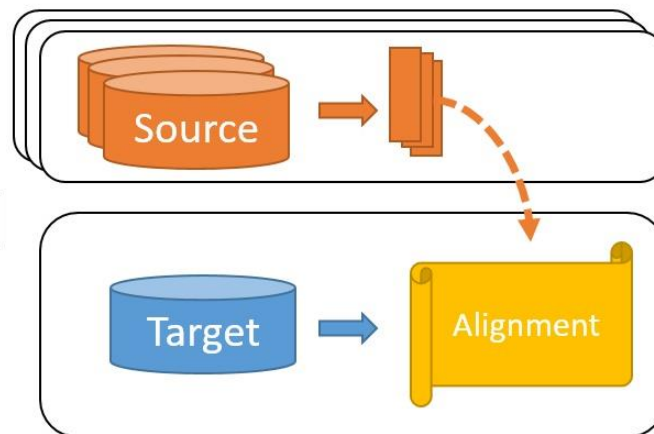
传统UDDA: 源域数据存在、
源域目标域可以混合对齐



ADDA: 单源域数据存在、只
需传输源域特征



SHOT: 源域数据不存在、只
有源域模型



FADA: 多源域数据存在但不
许外传, 特征可以传输