

Adversarial Discriminative Domain Adaptation

Eric Tzeng¹, Judy Hoffman², Kate Saenko³, Trevor Darrell¹

¹University of California Berkeley

²Stanford University

³Boston University

Introduction

Domain bias or Domain shift

- A change in data distribution between the source and target domain

Typical solution:

- Fine-tune networks on task-specific datasets
 - Expensive to obtain labeled data to properly fine-tune parameters for a deep multilayer network

Domain adaptation methods:

- Learn deep neural transformations to map both domains into a common feature space
 - Minimize some measure of domain shift

Adversarial adaptation method

- Minimize an approximate domain discrepancy distance
- Utilize generative adversarial learning
 - Goal: Discriminator should not be able to distinguish between distributions of training and test domain examples

Related Work

Goal: Transferring representations from a labeled source domain to an unlabeled target domain

- Main strategy: minimize difference between the source and target feature distributions
 - Maximum Mean Discrepancy (MMD) loss is commonly used
 - MMD computes norm of difference between two domain means
 - Adversarial loss has been implemented to minimize domain shift
 - Add a domain classifier with separate loss
 - Gradient reversal (ReverseGrad)
 - Domain confusion loss
 - Adversarial learning for generative tasks using general adversarial network (GAN)
 - Train two GANs to generate source and target images respectively (CoGAN)
 - Achieves a domain invariant feature space
 - Downside: “relies on generators finding a mapping from the shared high-level layer feature space to full images in both domains”. This can be difficult for increasingly distinct domains

Motivation

“In this work, we propose a novel unified framework for adversarial domain adaptation, allowing us to effectively examine the different factors of variation between the existing approaches and clearly view the similarities they each share”.

“...we observe that generative modeling of input image distributions is not necessary, as the ultimate task is to learn a discriminative representation. On the other hand, asymmetric mappings can better model the difference in low level features than symmetric ones. We therefore propose a previously unexplored unsupervised adversarial adaptation method, Adversarial Discriminative Domain Adaptation (ADDA)”.

“In this paper, we observe that modeling the image distributions is not strictly necessary to achieve domain adaptation, as long as the latent feature space is domain invariant, and propose a discriminative approach”

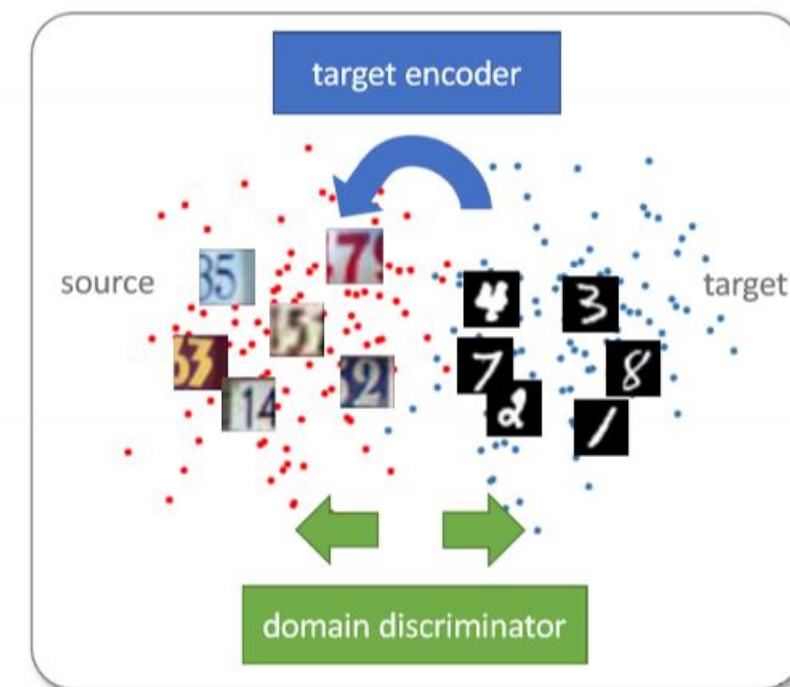


Figure 1: We propose an improved unsupervised domain adaptation method that combines adversarial learning with discriminative feature learning. Specifically, we learn a discriminative mapping of target images to the source feature space (target encoder) by fooling a domain discriminator that tries to distinguish the encoded target images from source examples.

Generalized Adversarial Adaptation

Source images \mathbf{X}_s and labels Y_s from source domain distribution $p_s(x, y)$

Target images \mathbf{X}_t from target domain distribution $p_t(x, y)$

Goal: Learn target representation M_t and classifier C_t to correctly classify target images into one of K categories despite the lack of in domain annotations.

Domain adaptation version: Learn M_s and C_s and adapt model to the target domain

Adversarial adaptive method: Learn M_s and M_t so as to minimize distance between empirical source and target mapping distributions $M_s(X_s)$ and $M_t(X_t)$.

“If this is the case then the source classification model, C_s , can be directly applied to the target representation, eliminating the need to learn a separate target classifier...”

$$C = C_s = C_t$$

Generalized Adversarial Adaptation Cont.

Source classification model is trained using standard supervised loss:

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

Define a discriminator, D , which classifies whether a data point is drawn from source or target domain. D is optimized according to standard supervised loss:

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

Source and target mappings are optimized according to constrained adversarial objective. Generic formulation for domain adversarial techniques:

$$\begin{aligned} & \min_D \mathcal{L}_{\text{adv}_D}(\mathbf{X}_s, \mathbf{X}_t, M_s, M_t) \\ & \min_{M_s, M_t} \mathcal{L}_{\text{adv}_M}(\mathbf{X}_s, \mathbf{X}_t, D) \\ & \text{s.t. } \psi(M_s, M_t) \end{aligned} \quad (3)$$

Source and Target Mappings

How to best minimize distance between source and target mappings when target domain is unlabeled?

Must choose how to parameterize these mappings, i.e. define a constraint between source and target mappings:

$$\psi(M_s, M_t)$$

Goal: “Make sure that the target mapping is set so as to minimize the distance between the source and target domains under their respective mappings, while crucially also maintaining a target mapping that is category discriminative”

Consider a layered representation with layer parameters denoted as M_s^ℓ or M_t^ℓ for a set of layers $\{\ell_1, \dots, \ell_n\}$

The space of constraints explored in literature can be described through layerwise equality constraints as follows:

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

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

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

Downside: optimization is poorly conditioned since the same network must handle images from two domains

Alternative: learn an asymmetric transformation, i.e. only constrain a subset of layers, enforcing partial alignment

Adversarial Losses

Choose an adversarial loss function to learn the mapping

Options:

- Gradient reversal

$$\mathcal{L}_{\text{adv}_M} = -\mathcal{L}_{\text{adv}_D}. \quad (6)$$

- Downside: quick convergence in early training causes gradient to vanish

- GAN loss function:

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

- Downside: this objective will lead to oscillation

Proposed solution:

- Cross-entropy loss function:

$$\begin{aligned} \mathcal{L}_{\text{adv}_M}(\mathbf{X}_s, \mathbf{X}_t, D) = & \\ & - \sum_{d \in \{s, t\}} \mathbb{E}_{\mathbf{x}_d \sim \mathbf{X}_d} \left[\frac{1}{2} \log D(M_d(\mathbf{x}_d)) \right. \\ & \left. + \frac{1}{2} \log(1 - D(M_d(\mathbf{x}_d))) \right]. \end{aligned} \quad (8)$$

- Advantage: adversarial discriminator views the two domains identically

Adversarial Discriminative Domain Adaptation

“...designing a new method has now been simplified to the space of making three design choices:

1. whether to use a generative or discriminative base model
2. whether to tie or untie the weights
3. which adversarial learning objective to use.”

The ADDA method:

1. Discriminative base model
2. Unshared weights
3. Standard GAN loss

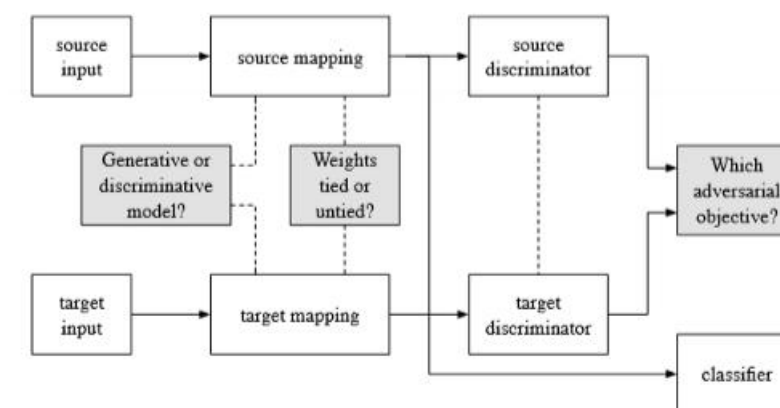


Figure 2: Our generalized architecture for adversarial domain adaptation. Existing adversarial adaptation methods can be viewed as instantiations of our framework with different choices regarding their properties.

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

Method

“First we choose a discriminative base model, as we hypothesize that much of the parameters required to generate convincing in-domain samples are irrelevant for discriminative adaptation tasks.

Next, we choose to allow independent source and target mappings by untying the weights. This is a more flexible learning paradigm as it allows more domain specific feature extraction to be learned.

However, note that the target domain has no label access, and thus without weight sharing a target model may quickly learn a degenerate solution...

Therefore we use the pre-trained source model as an initialization for the target representation space and fix the source model during adversarial training.

“In doing so we are effectively learning an asymmetric mapping, in which we modify the target model so as to match the source distribution [which] is most similar to the original generative adversarial learning setting...

Therefore, we choose the inverted label GAN loss...

Method

“Our proposed method, ADDA, thus corresponds to the following unconstrained optimization:”

$$\begin{aligned}
 \min_{M_s, C} \mathcal{L}_{\text{cls}}(\mathbf{X}_s, Y_s) = & \\
 & - \mathbb{E}_{(\mathbf{x}_s, y_s) \sim (\mathbf{X}_s, Y_s)} \sum_{k=1}^K \mathbb{1}_{[k=y_s]} \log C(M_s(\mathbf{x}_s)) \\
 \min_D \mathcal{L}_{\text{adv}_D}(\mathbf{X}_s, \mathbf{X}_t, M_s, M_t) = & \\
 & - \mathbb{E}_{\mathbf{x}_s \sim \mathbf{X}_s} [\log D(M_s(\mathbf{x}_s))] \\
 & - \mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}_t} [\log(1 - D(M_t(\mathbf{x}_t)))] \\
 \min_{M_s, M_t} \mathcal{L}_{\text{adv}_M}(\mathbf{X}_s, \mathbf{X}_t, D) = & \\
 & - \mathbb{E}_{\mathbf{x}_t \sim \mathbf{X}_t} [\log D(M_t(\mathbf{x}_t))].
 \end{aligned}
 \tag{9}$$

Optimization progression:

1. Optimize \mathcal{L}_{cls} over M_s and C by training using the source data, \mathbf{X}_s and Y_s
 - M_s is fixed while learning M_t
2. Optimize $\mathcal{L}_{\text{adv}_D}$ and $\mathcal{L}_{\text{adv}_M}$ without revisiting the first objective

Method

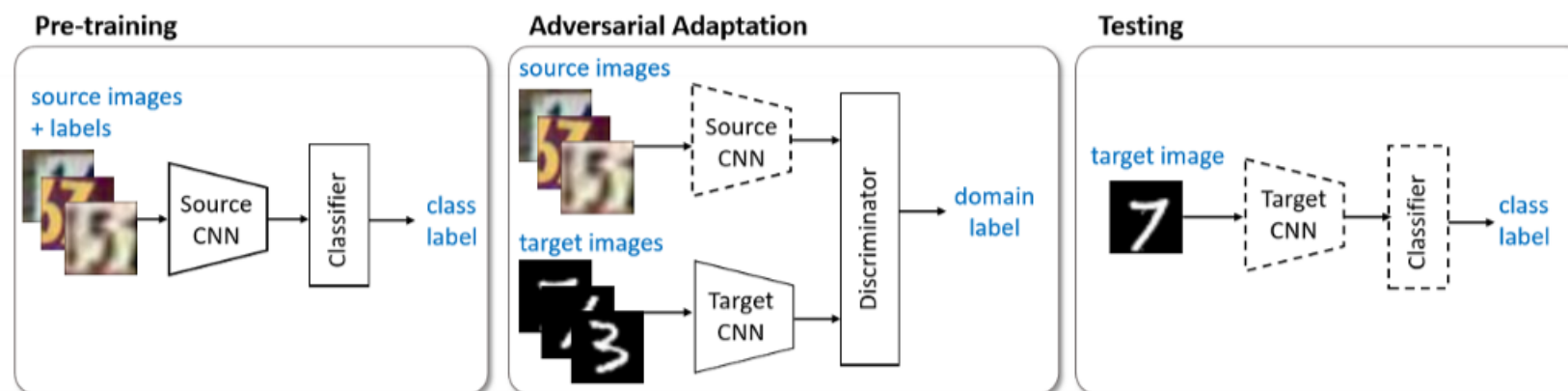


Figure 3: An overview of our proposed Adversarial Discriminative Domain Adaptation (ADDA) approach. We first pre-train a source encoder CNN using labeled source image examples. Next, we perform adversarial adaptation by learning a target encoder CNN such that a discriminator that sees encoded source 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.

Experiment – Digit Datasets

Three digit datasets of varying difficulty:

1. MNIST
2. USPS
3. SVHN

Considered adaptation in three directions:

1. MNIST \rightarrow USPS
2. USPS \rightarrow MNIST
3. SVHN \rightarrow MNIST

Architecture:

- LeNet

Discriminator:

- 3 fully connected layers:
 - 2 layers of 500 hidden units
 - Final discriminator output

Digits adaptation

MNIST



USPS



SVHN



Results
















Method	MNIST \rightarrow USPS   \rightarrow   	USPS \rightarrow MNIST    \rightarrow  	SVHN \rightarrow MNIST    \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

Table 2: Experimental results on unsupervised adaptation among MNIST, USPS, and SVHN.

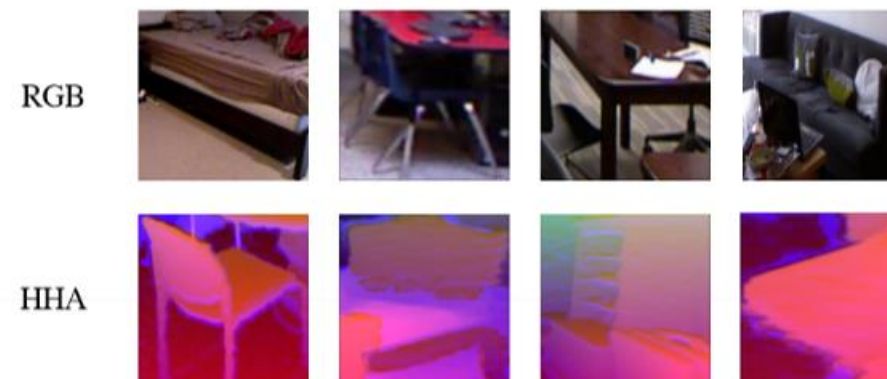
Experiment – Modality Adaptation

Cross-modality adaptation:

1. NYUD

- Source domain: RGB images
 - 2,186 labeled images
- Target domain: depth images
 - 2,401 unlabeled images

Cross-modality adaptation (NYUD)



Source Only

Architecture:

- VGG-16
 - Initializing from weights pretrained on ImageNet
 - Fine-tune

ADDA

Discriminator:

- 3 fully connected layers:
 - 1024 hidden units
 - 2048 hidden units
 - Final discriminator output

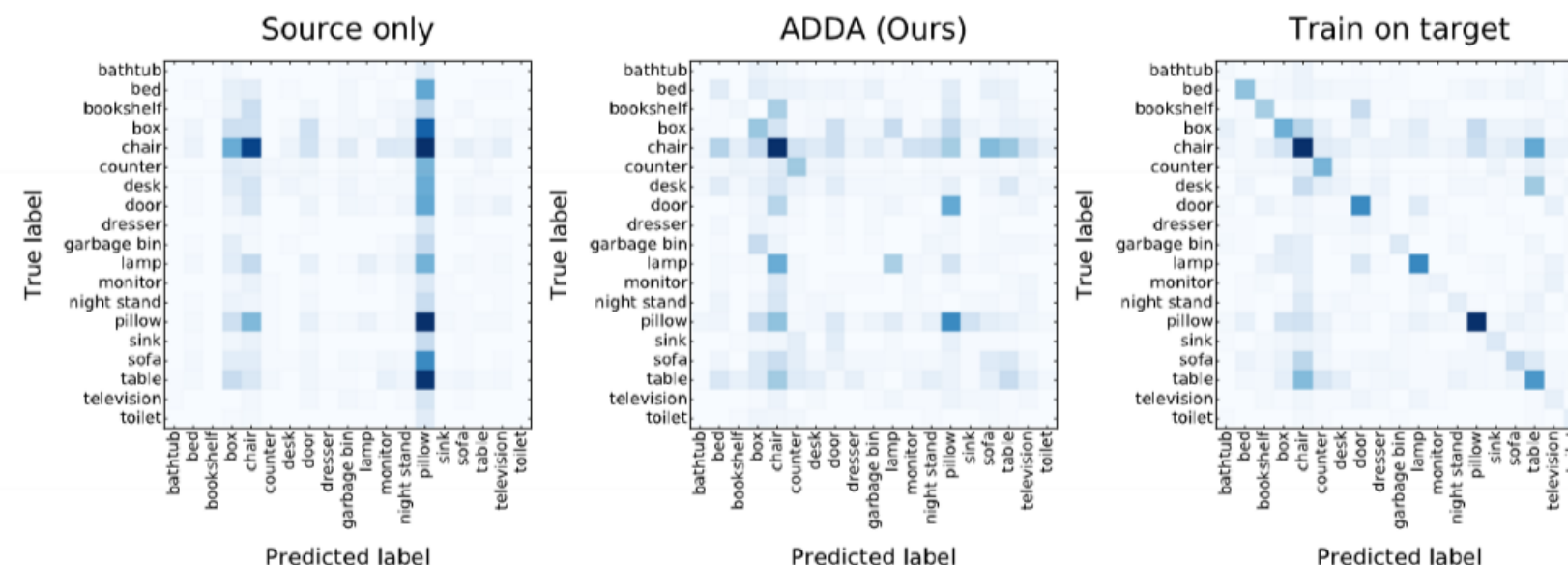
Results

Average accuracy was improved from 13.9% to 21.1%

- Large improvement on certain categories (*counter* accuracy improved from 2.9% to 44.7%)
- ADDA network made “reasonable” mistakes; for instance, confusing *chair* and *table* classes.

	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%.



Conclusion

- “Our framework provides a simplified and cohesive view by which we may understand and connect the similarities and differences between recently proposed adaptation methods”.

Benefits of ADDA method:

- Generalizes well across a variety of tasks
- Performs well on cross-modality adaptation tasks
- Effective at partially undoing effects of domain shift