PEARL: Perceptual Adaptive Representation Learning in the Wild

Adversarial Domain Adaptation

Kate Saenko
Trevor Darrell
Eric Tzeng
Judy Hoffman
Has deep learning solved AI?

pedestrian detection FAIL

https://www.youtube.com/watch?v=w2pxv8rFkU
“What you saw is not what you get”

What your net is trained on

What it’s asked to label

“Dataset Bias”
“Domain Shift”
“Domain Adaptation”
“Domain Transfer”
Example shift: scene segmentation

Train on Cityscapes, Test on Cityscapes

Example shift: scene segmentation

Train on Cityscapes, Test on San Francisco
Today: solving the domain shift problem

- From dataset to dataset
- From RGB to depth
- From simulated to real control
- From CAD models to real images
Background: Domain Adaptation from source to target distribution

Source Domain $\sim P_S(X, Y)$
- lots of labeled data

$$D_S = \{(x_i, y_i), \forall i \in \{1, \ldots, N\}\}$$

Target Domain $\sim P_T(Z, H)$
- unlabeled or limited labels

$$D_T = \{(z_j, ?), \forall j \in \{1, \ldots, M\}\}$$
How to adapt a deep network?
How to adapt a deep network?

- Applying source classifier to target domain can yield inferior performance…
How to adapt a deep network?

IDEA: align feature distributions

• Fine tune?
  …..Zero or few labels in target domain
Adversarial networks
**Adversarial networks**

**Encoder**
Generates features such that their distribution P matches reference distribution Q

**Adversary**
Tries to discriminate between samples from P and samples from Q
Adversarial networks

**Encoder**
Generates features such that their distribution $P$ matches reference distribution $Q$

*fools adversary*

**Adversary**
Tries to discriminate between samples from $P$ and samples from $Q$

*tries harder*
Adversarial domain adaptation

Source Data + Labels

Unlabeled Target Data

Encoder

Classifier

classification loss

Encoder

can be shared
Adversarial domain adaptation

Source Data + Labels

Unlabeled Target Data

Encoder

Encoder

Classifier

Discriminator

classification loss

Adversarial loss

can be shared
Adversarial domain adaptation

Source Data + Labels

Unlabeled Target Data

Encoder

Classifier

Discriminator

Adversarial loss

can be shared

classification loss

Encoder
Design choices in adversarial adaptation

Generative or discriminative?

Shared or not?

“confusion” Which loss?

<table>
<thead>
<tr>
<th>Method</th>
<th>Base model</th>
<th>Weight sharing</th>
<th>Adversarial loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient reversal [16]</td>
<td>discriminative</td>
<td>shared</td>
<td>minimax confusion</td>
</tr>
<tr>
<td>Domain confusion [12]</td>
<td>discriminative</td>
<td>shared</td>
<td>GAN</td>
</tr>
<tr>
<td>CoGAN [13]</td>
<td>generative</td>
<td>unshared</td>
<td></td>
</tr>
</tbody>
</table>

Deep domain confusion

Train a network to minimize classification loss AND confuse two domains

\[
\min_{\theta_D} \mathcal{L}_D(x_S, x_T, \theta_{\text{repr}}; \theta_D)
\]

\[
\min_{\theta_{\text{repr}}} \mathcal{L}_{\text{conf}}(x_S, x_T; \theta_D; \theta_{\text{repr}}).
\]
Deep domain confusion

Train a network to minimize classification loss AND confuse two domains

\[ \mathcal{L}_D(x_S, x_T, \theta_{repr}; \theta_D) = - \sum_d \mathbb{1}[y_D = d] \log q_d \]

\[ q = \text{softmax}(\theta_D^T f(x; \theta_{repr})) = p(y_D = 1|x) \]

\[ \mathcal{L}_{\text{conf}}(x_S, x_T, \theta_D; \theta_{repr}) = - \sum_d \frac{1}{D} \log q_d \] (cross-entropy with uniform distribution)

iterate
What is a good adversarial loss function?

**Confusion loss** [Tzeng 2015]

\[
\max_D \mathbb{E}_{x \sim p_S(x)} \left[ \log D(M_S(x)) \right] + \mathbb{E}_{x \sim p_T(x)} \left[ \log(1 - D(M_T(x))) \right]
\]

\[
\max_{M_S, M_T} \sum_{d \in \{S, T\}} \mathbb{E}_{x \sim p_d(x)} \left[ \frac{1}{2} \log D(M_d(x)) + \frac{1}{2} \log(1 - D(M_d(x))) \right]
\]

**Minimax loss** [Ganin 2015]

\[
\min_{M_S, M_T} \max_D V(D, M_S, M_T) = \mathbb{E}_{x \sim p_S(x)} \left[ \log D(M_S(x)) \right] + \mathbb{E}_{x \sim p_T(x)} \left[ \log(1 - D(M_T(x))) \right]
\]

**GAN loss** [Goodfellow 2014]

\[
\max_D \mathbb{E}_{x \sim p_S(x)} \left[ \log D(M_S(x)) \right] + \mathbb{E}_{x \sim p_T(x)} \left[ \log(1 - D(M_T(x))) \right]
\]

\[
\max_{M_T} \mathbb{E}_{x \sim p_T(x)} \left[ \log D(M_T(x)) \right].
\]

“stronger gradients”
Adversarial Discriminative Domain Adaptation (ADDA) [Tzeng CVPR17]

<table>
<thead>
<tr>
<th>Method</th>
<th>Base model</th>
<th>Weight sharing</th>
<th>Adversarial loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient reversal [16]</td>
<td>discriminative</td>
<td>shared</td>
<td>minimax</td>
</tr>
<tr>
<td>Domain confusion [12]</td>
<td>discriminative</td>
<td>shared</td>
<td>confusion</td>
</tr>
<tr>
<td>CoGAN [13]</td>
<td>generative</td>
<td>unshared</td>
<td>GAN</td>
</tr>
<tr>
<td>ADDA</td>
<td>discriminative</td>
<td>unshared</td>
<td>GAN</td>
</tr>
</tbody>
</table>

Generative or discriminative?  
Shared or not?  
Which loss?  
GAN

Kate Saenko, Trevor Darrell, PEARL: Perceptual Adaptive Representation Learning in the Wild
Applications to different types of domain shift

- From dataset to dataset
- From RGB to depth
- From simulated to real control
- From CAD models to real images
Fully Convolutional Network with Domain Confusion Loss [Hoffman 2016]
Results on Cityscapes to SF adaptation [Hoffman 2016]

Before domain confusion

After domain confusion

### ADDA: Adaptation on digits

**MNIST**

![Digits](image)

**USPS**

![Digits](image)

**SVHN**

![Digits](image)

<table>
<thead>
<tr>
<th>Method</th>
<th>MNIST $\rightarrow$ USPS</th>
<th>USPS $\rightarrow$ MNIST</th>
<th>SVHN $\rightarrow$ MNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source only</td>
<td>$0.752 \pm 0.016$</td>
<td>$0.571 \pm 0.017$</td>
<td>$0.601 \pm 0.011$</td>
</tr>
<tr>
<td>Gradient reversal</td>
<td>$0.771 \pm 0.018$</td>
<td>$0.730 \pm 0.020$</td>
<td>$0.739 \ [16]$</td>
</tr>
<tr>
<td>Domain confusion</td>
<td>$0.791 \pm 0.005$</td>
<td>$0.665 \pm 0.033$</td>
<td>$0.681 \pm 0.003$</td>
</tr>
<tr>
<td>CoGAN</td>
<td>$0.912 \pm 0.008$</td>
<td>$0.891 \pm 0.008$</td>
<td>did not converge</td>
</tr>
<tr>
<td>ADDA (Ours)</td>
<td>$0.894 \pm 0.002$</td>
<td>$0.901 \pm 0.008$</td>
<td>$0.760 \pm 0.018$</td>
</tr>
</tbody>
</table>
Office dataset: historical progress

Unsupervised adaptation in 2016/2017

<table>
<thead>
<tr>
<th>Method</th>
<th>$A \rightarrow W$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDC (Tzeng et al., 2014)</td>
<td>0.618</td>
</tr>
<tr>
<td>DAN (Long &amp; Wang, 2015)</td>
<td>0.685</td>
</tr>
<tr>
<td>DRCN (Ghifary et al., 2016)</td>
<td>0.687</td>
</tr>
<tr>
<td>DANN (Ganin et al., 2016)</td>
<td>0.730</td>
</tr>
<tr>
<td>ADDA (Ours)</td>
<td><strong>0.751</strong></td>
</tr>
</tbody>
</table>
Applications to different types of domain shift

- From dataset to dataset
- From RGB to depth
- From simulated to real control
- From CAD models to real images
ADDA: Adaptation on RGB-D

Train on RGB

Test on depth

<table>
<thead>
<tr>
<th></th>
<th>bathtub</th>
<th>bed</th>
<th>bookshelf</th>
<th>box</th>
<th>chair</th>
<th>counter</th>
<th>desk</th>
<th>door</th>
<th>dresser</th>
<th>garbage bin</th>
<th>lamp</th>
<th>monitor</th>
<th>night stand</th>
<th>pillow</th>
<th>sink</th>
<th>sofa</th>
<th>table</th>
<th>television</th>
<th>toilet</th>
<th>overall</th>
</tr>
</thead>
<tbody>
<tr>
<td># of instances</td>
<td>19</td>
<td>96</td>
<td>87</td>
<td>210</td>
<td>611</td>
<td>103</td>
<td>122</td>
<td>129</td>
<td>25</td>
<td>55</td>
<td>144</td>
<td>37</td>
<td>51</td>
<td>276</td>
<td>47</td>
<td>129</td>
<td>210</td>
<td>33</td>
<td>17</td>
<td>2401</td>
</tr>
<tr>
<td>Source only</td>
<td>0.000</td>
<td>0.010</td>
<td>0.011</td>
<td>0.124</td>
<td>0.188</td>
<td>0.029</td>
<td>0.041</td>
<td>0.047</td>
<td>0.000</td>
<td>0.000</td>
<td>0.069</td>
<td>0.000</td>
<td>0.039</td>
<td>0.587</td>
<td>0.008</td>
<td>0.010</td>
<td>0.000</td>
<td>0.139</td>
<td>0.019</td>
<td>0.211</td>
</tr>
<tr>
<td>ADDA (Ours)</td>
<td>0.000</td>
<td>0.146</td>
<td>0.046</td>
<td>0.229</td>
<td>0.344</td>
<td>0.447</td>
<td>0.025</td>
<td>0.023</td>
<td>0.000</td>
<td>0.018</td>
<td>0.292</td>
<td>0.081</td>
<td>0.020</td>
<td>0.297</td>
<td>0.021</td>
<td>0.116</td>
<td>0.143</td>
<td>0.091</td>
<td>0.000</td>
<td>0.211</td>
</tr>
<tr>
<td>Train on target</td>
<td>0.105</td>
<td>0.531</td>
<td>0.494</td>
<td>0.295</td>
<td>0.619</td>
<td>0.573</td>
<td>0.057</td>
<td>0.636</td>
<td>0.120</td>
<td>0.291</td>
<td>0.576</td>
<td>0.189</td>
<td>0.235</td>
<td>0.630</td>
<td>0.362</td>
<td>0.248</td>
<td>0.357</td>
<td>0.303</td>
<td>0.647</td>
<td>0.468</td>
</tr>
</tbody>
</table>
Not covered today: simulation-to-real shifts

From dataset to dataset

From RGB to depth

From simulated to real control

From CAD models to real images
Thank you

References

• Eric Tzeng, Judy Hoffman, Trevor Darrell, Kate Saenko, Simultaneous Deep Transfer Across Domains and Tasks, ICCV 2015
• Eric Tzeng, Coline Devin, Judy Hoffman, Chelsea Finn, Pieter Abbeel, Sergey Levine, Kate Saenko, Trevor Darrell. Adapting Deep Visuomotor Representations with Weak Pairwise Constraints, WAFR 2016
• Baochen Sun, Jiashi Feng, Kate Saenko, Return of Frustratingly Easy Domain Adaptation, AAAI 2016
• Baochen Sun, Kate Saenko, Deep CORAL: Correlation Alignment for Deep Domain Adaptation, TASK-CV Workshop at ICCV 2016
• Eric Tzeng, Judy Hoffman, Trevor Darrell, Kate Saenko, Adversarial Discriminative Domain Adaptation, accepted to CVPR 2017
• Synthetic to Real Adaptation with Deep Generative Correlation Alignment Networks, arxiv.org