Adversarial Domain Adaptation and Adversarial Robustness

Judy Hoffman
Big data + Deep learning = success
Millions of Images

Challenge to recognize 1000 categories
Dataset Bias
Dataset Bias

Test Image

Deep Model

Dog is not recognized
Dataset Bias
Dataset Bias

Low resolution

Motion Blur
Dataset Bias

- Low resolution
- Motion Blur
- Pose Variety
Why not collect new annotations?
Why not collect new annotations?

Figure 1: Unsupervised domain adaptation for pixel-level semantic segmentation.

- **Car**
- **Road**
- **Sidewalk**
- **Person**
- **Sky**
- **Vegetation**
- **Street Sign**
- **Building**
Why not collect new annotations?

Expensive ($10-12 per image)

- Car
- Road
- Sidewalk
- Person
- Sky
- Vegetation
- Street Sign
- Building
Why not collect new annotations?

Large Potential for Change
Different: Weather, City, Car

Expensive ($10-12 per image)
Why not collect new annotations?

Proprietary

Private
Domain Adaptation: Train on Source Test on Target

Source Domain $\sim P_S(X_S, Y_S)$
- lots of labeled data

Target Domain $\sim P_T(X_T, Y_T)$
- unlabeled or limited labels
Adversarial Domain Adaptation

Adversarial Domain Adaptation

$y_s$  bottle

Source Data

Source CNN

Source feature vector $x_s$

Classifier

Target Data

Target CNN

Target feature vector $x_t$

Adversarial Domain Adaptation

Source Data

Target Data

$y_s$ bottle

Source CNN

Source feature vector $x_s$

Classifier

Minimize Discrepancy

Target CNN

Target feature vector $x_t$

Adversarial Domain Adaptation

Source Data

Source CNN

Source feature vector $x_s$

Classifier

Target CNN

Target feature vector $x_t$

Minimize Discrepancy

Adversarial Domain Adaptation

Source Data

Target Data

Source CNN

Target CNN

Source feature vector $x_s$

Target feature vector $x_t$

Classifier

Domain Classifier

Adversarial Loss

Minimize Discrepancy

Adversarial Domain Adaptation

\( y_s \) bottle

Source Data

Target Data

Minimize Discrepancy

CyCADA: Cycle Consistent Adversarial DA

Source Data

Semantically Consistent

Source to Target

Target to Source

Domain Adversarial

Reconstructed Source Data

Target Data

Hoffman et.al. ICML 2018
Synthetic to Real Pixel Adaptation

Train

GTA (synthetic)

Test

CityScapes (Germany)

Hoffman et.al. ICML 2018
Synthetic to Real Pixel Adaptation

Hoffman et.al. ICML 2018
Synthetic to Real Pixel Adaptation

Synthetic to Real Pixel Adaptation

CyCADA Results: CityScapes Evaluation

CityScapes Image

Before Adaptation

After Adaptation

Ground Truth

Hoffman et.al. ICML 2018
CyCADA Results: CityScapes Evaluation

CityScapes Image

Ground Truth

Before Adaptation

After Adaptation

Hoffman et.al. ICML 2018
CyCADA Results: CityScapes Evaluation

CityScapes Image

Ground Truth

Before Adaptation

After Adaptation

Hoffman et.al. ICML 2018
So Far: Adapting to Natural Shifts
So Far: Adapting to Natural Shifts
What about adversarial shifts?
Adversarial Examples

$\mathbf{x}$  
“panda”  
57.7% confidence

$\mathbf{x} + 0.007 \times \text{sign}(\nabla \mathbf{x} J(\mathbf{\theta}, \mathbf{x}, y))$  
“nematode”  
8.2% confidence

$\mathbf{x} + \varepsilon \text{sign}(\nabla \mathbf{x} J(\mathbf{\theta}, \mathbf{x}, y))$  
“gibbon”  
99.3% confidence

Goodfellow et al. ICLR 2015.
Visualize Perturbation Space
Visualize Perturbation Space

Training point
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).

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Domain confusion</th>
<th>Gradient reversal</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDA (Ours)</td>
<td>0.002 ± 0.008</td>
<td>0.005 ± 0.016</td>
</tr>
<tr>
<td>Source only</td>
<td>0.016 ± 0.017</td>
<td></td>
</tr>
</tbody>
</table>

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).
We find that our method, ADDA, greatly improves classification accuracy. For this experiment, our base architecture is the VGG-16 network, initializing from weights pretrained on ImageNet. 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 between RGB and HHA encoded depth images (right). For additional insight on what effect ADDA has on classification accuracy, we consider the task of adaptation between these RGB and HHA modalities. 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 between RGB and HHA encoded depth images (right). For additional insight on what effect ADDA has on classification accuracy, we consider the task of adaptation between these RGB and HHA modalities.

Visualize Perturbation Space
Visualize Perturbation Space

- Training point
- Vectorize
- Sweep over a grid of perturbations

Project onto random 2D orthonormal basis
Visualize Perturbation Space

Training point → Vectorize → Sweep over a grid of perturbations → Project onto random 2D orthonormal basis → Perturbed Image

28 × 28 → 784 → Perturbed Image
When training with ADDA, the adversarial discriminator output. With the exception of the output, these iterations, again with a batch size of 128.

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

<table>
<thead>
<tr>
<th>Method</th>
<th>CoGAN</th>
<th>Gradient reversal</th>
<th>ADDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>MNIST accuracy</td>
<td>0.95</td>
<td>0.94</td>
<td>0.96</td>
</tr>
<tr>
<td>USPS accuracy</td>
<td>0.89</td>
<td>0.88</td>
<td>0.90</td>
</tr>
<tr>
<td>SVHN accuracy</td>
<td>0.87</td>
<td>0.86</td>
<td>0.88</td>
</tr>
</tbody>
</table>

In contrast, the classifier trained using ADDA predicts a significantly higher accuracy for the target domain. This is especially true for classes with few labeled examples in the target domain. The confusion matrix for the target domain after adaptation shows a much wider variety of classes. This leads to decreased accuracy for the source only baseline. The table shows that ADDA achieves significantly higher accuracy for the target domain, particularly for classes with few labeled examples.

The visualization of the perturbation space shows the effect of the perturbations on the model scores. The model is trained on a grid of perturbations and the scores are projected onto a 2D orthonormal basis. The perturbed images are then classified by the model.
MNIST LeNet Decisions Around Training Point
MNIST LeNet Decisions Around Training Point

Training Data Point
MNIST LeNet Decisions Around Training Point

Training Data Point
MNIST LeNet Decisions Around Training Point

Non-smooth Decision Boundary

Training Data Point
MNIST LeNet Decisions Around Training Point

Non-smooth Decision Boundary

Small perturbations lead to new outputs
MNIST LeNet with L2 Regularization

Smooth Decision Boundary

Small perturbations lead to new outputs
MNIST LeNet with L2 Regularization

Smooth Decision Boundary

Small perturbations lead to new outputs
Jacobian Regularization

$y_s$  bottle

$\mathcal{x}_s$  

score vector $\mathcal{z}_s$

Classifier

Hoffman, Roberts, Yaida, In submission, 2019.
Jacobian Regularization

$$J_{c,i} = \frac{\partial z_c}{\partial x_i}$$

Hoffman, Roberts, Yaida, In submission, 2019.
Jacobian Regularization

**Input-output Jacobian matrix**

\[ J_{c,i} = \frac{\partial z_c}{\partial x_i} \]

**Minimize Frobenius Norm**

\[ \| J \|_F^2 \]

Hoffman, Roberts, Yaida, In submission, 2019.
MNIST LeNet with Jacobian Regularization

Mostly Smooth
Decision Boundary

Larger perturbations needed to lead to new outputs
MNIST LeNet with Jacobian Regularization

 Mostly Smooth Decision Boundary

Larger perturbations needed to lead to new outputs
Decision Boundary Comparison

No Regularization

L2 Regularization

Jacobian Regularization

Hoffman, Roberts, Yaida, In submission, 2019.
Robustness to Random Perturbations

Table 2: Generalization on clean test data from an unseen domain. Accuracy on data from the novel input domain of USPS test set using the LeNet' model learned with all MNIST training data. Here, each regularizer, including Jacobian, increases accuracy over an unregularized model. In addition, the regularizers may be combined for the strongest generalization effects. Averages and 95% confidence intervals are estimated over 5 distinct runs.

<table>
<thead>
<tr>
<th>Regularizer</th>
<th>No Regularization</th>
<th>$L^2$ Regularization</th>
<th>Dropout Regularization</th>
<th>Jacobian Regularization</th>
<th>Combine Three</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>80$\pm$4</td>
<td>83$\pm$7</td>
<td>81$\pm$8</td>
<td>88$\pm$1</td>
<td>81$\pm$4</td>
</tr>
</tbody>
</table>

(a) White noise

Figure 3: Robustness against random and adversarial input perturbations. This key result illustrates that Jacobian regularization significantly increases the robustness of a learned model against random and adversarial input perturbations. Accuracy under corruption of input test data for LeNet' models trained on the MNIST dataset. (a) Considering robustness under white noise perturbations, Jacobian minimization is the most effective regularizer. (b,c) Jacobian regularization alone outperforms an adversarial defense (base models all include $L^2$ and dropout regularization).

Error bars indicate 95% confidence intervals over 5 distinct runs.

3.2 Evaluating under Data Corruption

This section showcases the main robustness results of the Jacobian regularizer, highlighted in the case of both random and adversarial input perturbations.

Random Noise Corruption: The real world can differ from idealized experimental setups and input data can become corrupted by various natural causes such as random noise and occlusion. Robust models should minimize the impact of such corruption. As one evaluation of stability to natural corruption, we perturb each test input image $x$ to $e_x = d_x + \xi$ where each component of the perturbation vector is drawn from the normal distribution with variance $\xi_i \sim N(0, \sigma^2_{\text{noise}})$, and the perturbed image is then clipped to fit into the range $[0, 1]$ before preprocessing. As in the domain-adaptation experiment above, we take the model parameters with the best test accuracy, and then test them on corrupted data. Results in Figure 3a shows that models trained with the Jacobian regularization is more robust to white noise than others. This is in line with – and indeed quantitatively validates – the embiggening of decision cells as shown in Figure 1.

Adversarial Perturbations: The world is not only imperfect but also filled with evil agents that can deliberately attack models. Such adversaries seek a small perturbation to each input example that changes the model predictions while also being imperceptible to humans. Obtaining the actual smallest perturbation is likely computationally intractable, but there exist many tractable approximations. The simplest attack is the white-box untargeted fast gradient sign method (FGSM) [Goodfellow et al., 2014], which distorts the image as $e_x = d_x + \xi$ with $\xi_i = \text{"FGSM} \cdot \text{sign} \left( \frac{\partial L}{\partial z} \right)$;
Robustness to Adversarial Perturbations

Table 2: Generalization on clean test data from an unseen domain. Accuracy on data from the novel input domain of USPS test set using the LeNet' model learned with all MNIST training data. Here, each regularizer, including Jacobian, increases accuracy over an unregularized model. In addition, the regularizers may be combined for the strongest generalization effects. Averages and 95% confidence intervals are estimated over 5 distinct runs.

<table>
<thead>
<tr>
<th>No regularization</th>
<th>$L^2$ Dropout</th>
<th>Jacobian</th>
<th>All Combined</th>
</tr>
</thead>
<tbody>
<tr>
<td>80 $\pm$ 4</td>
<td>83 $\pm$ 7</td>
<td>81 $\pm$ 8</td>
<td>85 $\pm$ 9</td>
</tr>
</tbody>
</table>

Figure 3: Robustness against random and adversarial input perturbations. This key result illustrates that Jacobian regularization significantly increases the robustness of a learned model against random and adversarial input perturbations. Accuracy under corruption of input test data for LeNet' models trained on the MNIST dataset. (a) Considering robustness under white noise perturbations, Jacobian minimization is the most effective regularizer. (b,c) Jacobian regularization alone outperforms an adversarial defense (base models all include $L^2$ and dropout regularization). Error bars indicate 95% confidence intervals over 5 distinct runs.

3.2 Evaluating under Data Corruption

This section showcases the main robustness results of the Jacobian regularizer, highlighted in the case of both random and adversarial input perturbations.

Random Noise Corruption: The real world can differ from idealized experimental setups and input data can become corrupted by various natural causes such as random noise and occlusion. Robust models should minimize the impact of such corruption. As one evaluation of stability to natural corruption, we perturb each test input image $x$ to $e_x = d_x + \epsilon_{crop}$ where each component of the perturbation vector is drawn from the normal distribution with variance $\text{noise}$ as

$$\epsilon_i \sim N(0, \text{noise})$$

and the perturbed image is then clipped to fit into the range $[0, 1]$ before preprocessing. As in the domain-adaptation experiment above, we take the model parameters with the best test accuracy, and then test them on corrupted data. Results in Figure 3a shows that models trained with the Jacobian regularization is more robust to white noise than others. This is in line with – and indeed quantitatively validates – the embiggening of decision cells as shown in Figure 1.

Adversarial Perturbations: The world is not only imperfect but also filled with evil agents that can deliberately attack models. Such adversaries seek a small perturbation to each input example that changes the model predictions while also being imperceptible to humans. Obtaining the actual small-est perturbation is likely computationally intractable, but there exist many tractable approximations. The simplest attack is the white-box untargeted fast gradient sign method (FGSM) [Goodfellow et al., 2014], which distorts the image as $e_x = d_x + \epsilon_{crop}$ with

$$\epsilon_i = \text{FGSM} \cdot \text{sign} X c_{\partial L_{\text{bare}} z c} J c_{\partial z} ! \Delta$$

Hoffman, Roberts, Yaida, In submission, 2019.
Next Steps

Jacobian regularizer as unsupervised adaptive loss?

Adaptation to an adversarial domain?
Thank you

Taesung Park
UC Berkeley

Eric Tzeng
UC Berkeley

Jun-Yan Zhu
MIT

Dan Roberts
Diffeo

Phil Isola
MIT

Kate Saenko
Boston University

Trevor Darrell
UC Berkeley

Alyosha Efros
UC Berkeley

Sho Yaida
FAIR
Judy Hoffman
judyhoffman.io