Robust Out-of-Distribution Detection via Informative Outlier Mining (ATOM)

Presented by
Bastin Joseph, Yifei Ming

ICML Workshop on Uncertainty & Robustness in Deep Learning (ICML UDL), 2020
Authors: Jiefeng Chen, Yixuan Li, Xi Wu, Yingyu Liang, Somesh Jha
Agenda

- Quick recap: the problem of OOD detection and why
- Pre-trained model based detection methods
- Training with auxiliary dataset & modified network structure
- Robustify by adversarial training & outlier mining: ATOM
- Potential improvements of ATOM
- Discussions Q & A
Recap of last lecture

Deep neural networks can be over-confident to both in-distribution (Guo et al.) and out-of-distribution samples (Nauyen et al.)
Pre-trained model based OOD detection

Training examples: traffic signs

$$p_i(x) = \frac{\exp(f_i(x))}{\sum_{j=1}^{N} \exp(f_j(x))}$$

$$p_i(x; T) = \frac{\exp(f_i(x)/T)}{\sum_{j=1}^{N} \exp(f_j(x)/T)}$$

High confidence in classifying traffic signs.

MSP: directly based on confidence score (max softmax output)  
Hendrycks et al.

ODIN: input perturbation & temperature scaling for confidence calibration  
Liang et al.
Generative classifier: better characterization from the representation space?

Assumption: class-conditional Gaussian with tied covariance \( P(y = i \mid x) \) is softmax

\[
P(f(x) \mid y = i) = \mathcal{N}(f(x) \mid \mu_i, \Sigma)
\]

Training examples: traffic signs

Metric \( M(x) = \max_i - (f(x) - \hat{\mu}_i)^T \hat{\Sigma}^{-1} (f(x) - \hat{\mu}_i) \)

Measures the log of probability density of the test sample

Visualizaition by t-SNE
Beyond pre-trained models

Can we do better?

- Train with auxiliary OOD datasets?
- Introduce additional classes representing “OOD”?
- Modify the network structure to incorporate the detector?
- Robust against adversarial & corrupted OOD samples?
- Utilize the abundance of OOD data?
Improve the robustness of detectors with auxiliary OOD dataset for training

Outlier Exposure (OE): allow models to learn OOD features

\(D_{in}\) The distribution of the training data

\(D_{out}\) Any unknown distribution disjoint from \(D_{in}\)

\(D_{oe}\) OOD samples used to train the classifier

\(D_{test}\) OOD samples unseen by the classifier, used for test

Street View House Number (SVHN)

80 Million Tiny Image

MNIST

Hendrycks et al
OE for multi-class classification

$$
\mathbb{E}_{(x,y) \sim \mathcal{D}_{\text{in}}} \left[ - \log f_y(x) \right] + \lambda \mathbb{E}_{x \sim \mathcal{D}_{\text{out}}} \left[ H(\mathcal{U}; f(x)) \right]
$$

- Negative log probability of $f(x)$ belonging to real class $y$
- Cross entropy between a uniform distribution and posterior distribution of $f(.)$ for OOD examples

Hendrycks et al
Beyond pre-trained models

Can we do better?

- Train with auxiliary OOD datasets?
- Introduce additional classes representing “OOD”?
- Modify the network structure to incorporate the detector?
- Robust against adversarial & corrupted OOD samples?
- Utilize the abundance of OOD data?
Add auxiliary head with self-supervised learning

SOFL: add extra nodes in the last layer to train for outlier features

Mohseni et. al
Two step training procedure

Step 1: supervised in-distribution training with cross entropy loss

Step 2: self-supervised ood training with combined loss:

\[
E(x, y_{in}) \sim D_{in} \left[ - \log f_{y_{in}}(x) \right] + \lambda E_{x \sim D_{out}} \left[ - \log f_{y_{target}}(x) \right]
\]

\[
y_{in} \in \{1, 2, \cdots, K\}, \quad y_{target} \xrightarrow{\text{random sample}} y_{out} \in \{1, 2, \cdots, M\}
\]

Mohseni et. al
Beyond pre-trained models

Can we do better?

- Train with auxiliary OOD datasets?
- Introduce additional classes representing “OOD”?
- Modify the network structure to incorporate the detector?
- Robust against adversarial & corrupted OOD samples?
- Utilize the abundance of OOD data?
Adversarial Confidence Enhancing Tuning (ACET)

Decision Boundaries in NN are intersections of half spaces

Suppose $x \in \mathbb{R}^2$

$y \in \{1, 2, 3\}$

Each color represents the decision region for one class
Adversarial Confidence Enhancing Training (ACET)

- CEDA - Confidence Enhancing Data Augmentation
  - Force lower confidence for OOD samples

\[
\frac{1}{N} \sum_{i=1}^{N} \left( -\log(f_{y_i}(x_i)) \right) + \lambda \mathbb{E}[L_{pout}(f, Z)]
\]

- ACET
  - Enhance Adversarial Robustness by training on Adversarial OOD

\[
\frac{1}{N} \sum_{i=1}^{N} \left( -\log(f_{y_i}(x_i)) \right) + \lambda \mathbb{E}\left[ \max_{||u-Z||_p \leq \epsilon} L_{pout}(f, Z) \right]
\]

\[
L_{pout}(f, z) = \max_{l=1,2,\ldots,K} \log\left( \frac{e^{f_l(z)}}{\sum_{k=1}^{K} e^{f_k(z)}} \right)
\]

Hein et al
Beyond pre-trained models

Can we do better?

- Train with auxiliary OOD datasets?
- Introduce additional classes representing “OOD”?
- Modify the network structure to incorporate the detector?
- Robust against adversarial & corrupted OOD samples?
- Utilize the abundance of OOD data?
Robust OOD evaluation tasks on “harder” OOD samples

Adversarially perturbed OOD
e.g. for ODIN, MSP, OE (PGD attack):

$$x' = \arg\max_{x' \in \Omega_{\infty, \epsilon}(x)} - \frac{1}{K} \sum_{i=1}^{K} \log f_i(x')$$

$$\Omega_{\infty, \epsilon}(x) = \{ \delta \in \mathbb{R}^d \mid \|\delta\|_{\infty} \leq \epsilon, x + \delta \text{ valid} \}$$

Corruption attacked OOD

Compositionally attacked OOD
Beyond pre-trained models

Can we do better?

- Train with auxiliary OOD datasets?
- Introduce additional classes representing “OOD”?
- Modify the network structure to incorporate the detector?
- Robust against adversarial & corrupted OOD samples?
- Utilize the abundance of OOD data?
ATOM: better selection of informative OOD samples

Adversarial Training with informative Outlier Mining

$y_{in} \in \{1, 2, \cdots, K\} \quad y_{out} \in \{K + 1\}$

Chen et al
Procedure of OOD detection for ATOM

- OOD score
  \[ f(x)_{K+1} \]

- OOD detector
  \[ G(x) = \begin{cases} 
  -1 & \text{if } f(x)_{K+1} \geq \gamma \\
  +1 & \text{if } f(x)_{K+1} < \gamma 
\end{cases} \]

- classifier
  \[ f(x) = \arg\max_{y \in \{1,2,3,\ldots,K\}} f(x)_y \]
Adversarial training for ATOM

- Softmax output on $x : f(x)$

- Training objective

$$
\mathbb{E}_{(x,y) \sim D_{in}} \left[ - \log f_{y_{in}}(x) \right] + \lambda \mathbb{E}_{x \sim D_{out}^{\text{train}}} \max_{x' \in \Omega_{\infty, \epsilon(x)}} \left[ - \log f_{y_{K+1}}(x') \right]
$$

Chen et al
Adversarial training for ATOM

Minimize training set loss

\[ \mathbb{E}_{(x,y) \sim \mathcal{D}_{in}} [-\log f_{y_{in}}(x)] \]

Labelled in distribution data

Minimize loss of OOD data prediction

\[ \mathbb{E}_{x \sim \mathcal{D}^{train}} \max_{x' \in \Omega_{\infty,v(x)}} [-\log f_{y_{K+1}}(x')] \]

unlabelled OOD samples

Chen et al
How to select OOD data for training?

Abundance of Unlabelled, OOD data

Random sampling?

Converge quickly after few epochs

Chen et al.
Harder OOD & easy OOD

- Mine outliers

In-distribution data
e.g. two in-distribution classes:
- the lung Xrays with disease 1
- the normal lung Xrays

Chen et al. 23
Hardness based mining

- How do we mine?

Sort samples based on OOD scores

![Diagram showing histogram of OOD scores with three regions labeled as easy, medium, and hard.]

\[ D_{\text{auxilliary}} \rightarrow \text{Random Sampling} \rightarrow \text{N OOD samples} \rightarrow \text{Forward Pass} \rightarrow \text{OOD scores} \rightarrow \text{Select n samples with medium OOD scores} \rightarrow D_{\text{train}} \rightarrow D_{\text{out}} \]

Chen et al
Put it together: the training procedure

In-distribution samples

$D_{\text{train}}^{\text{in}}$

Natural OOD

$D_{\text{train}}^{\text{out}}$

Adversarial OOD

$qN$

$n$

$qN + n$

random sampling + sorted by OOD score

$D_{\text{auxiliary}}^{\text{out}}$

Chen et al
The ATOM algorithm

\begin{align*}
\text{input} & \quad \mathcal{D}_{\text{train}}^\text{in}, \mathcal{D}_{\text{out}}^\text{auxiliary}, \hat{F}_\theta, m, N, n, q \\
\text{output} & \quad \hat{F}, \hat{G} \\
\text{for} & \quad t = 1, 2, \cdots, m \text{ do} \\
& \quad \text{Randomly sample } N \text{ data points from } \mathcal{D}_{\text{out}}^\text{auxiliary} \text{ to get a candidate set } \mathcal{S}. \\
& \quad \text{Compute OOD scores on } \mathcal{S} \text{ using current model } \hat{F}_\theta \text{ to get set } \mathcal{V} = \{\hat{F}(x)_{K+1} \mid x \in \mathcal{S}\}. \\
& \quad \text{Sort scores in } \mathcal{V} \text{ from the lowest to the highest.} \\
& \quad \mathcal{D}_{\text{out}}^\text{train} \leftarrow \mathcal{V}[qN : qN + n] \quad \triangleright \quad \{q \in [0, 1 - n/N]\} \\
& \quad \text{Train } \hat{F}_\theta \text{ for one epoch using the training objective of (2).} \\
\text{end for}
\end{align*}
Ablation Study - Informative Outlier Mining

- How does the sampling parameter affect performance?

<table>
<thead>
<tr>
<th>$D_{\text{test}}$</th>
<th>Method</th>
<th>FPR (5% FNR)</th>
<th>AUROC</th>
<th>FPR (5% FNR)</th>
<th>AUROC</th>
<th>FPR (5% FNR)</th>
<th>AUROC</th>
<th>FPR (5% FNR)</th>
<th>AUROC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Natural OOD</td>
<td></td>
<td>Corruption OOD</td>
<td></td>
<td>$L_{\infty}$ OOD</td>
<td></td>
<td>Comp. OOD</td>
<td></td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>ATOM (q=0.0)</td>
<td>5.39</td>
<td>98.35</td>
<td>39.65</td>
<td>92.47</td>
<td>35.24</td>
<td>91.06</td>
<td>60.44</td>
<td>80.65</td>
</tr>
<tr>
<td></td>
<td>ATOM (q=0.125)</td>
<td>5.15</td>
<td>98.30</td>
<td>30.32</td>
<td>93.85</td>
<td>5.19</td>
<td>98.26</td>
<td>31.38</td>
<td>93.81</td>
</tr>
<tr>
<td></td>
<td>ATOM (q=0.25)</td>
<td>6.02</td>
<td>98.06</td>
<td>33.79</td>
<td>92.55</td>
<td>22.56</td>
<td>95.12</td>
<td>43.66</td>
<td>91.04</td>
</tr>
<tr>
<td></td>
<td>ATOM (q=0.5)</td>
<td>9.55</td>
<td>97.48</td>
<td>39.54</td>
<td>91.58</td>
<td>18.95</td>
<td>95.73</td>
<td>51.01</td>
<td>89.88</td>
</tr>
<tr>
<td></td>
<td>ATOM (q=0.75)</td>
<td>13.98</td>
<td>96.61</td>
<td>56.88</td>
<td>87.00</td>
<td>14.10</td>
<td>96.61</td>
<td>57.02</td>
<td>87.09</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>ATOM (q=0.0)</td>
<td>45.25</td>
<td>91.53</td>
<td>98.84</td>
<td>58.54</td>
<td>43.14</td>
<td>90.22</td>
<td>94.68</td>
<td>55.53</td>
</tr>
<tr>
<td></td>
<td>ATOM (q=0.125)</td>
<td>40.06</td>
<td>92.59</td>
<td>98.01</td>
<td>67.20</td>
<td>36.90</td>
<td>92.79</td>
<td>89.09</td>
<td>68.94</td>
</tr>
<tr>
<td></td>
<td>ATOM (q=0.25)</td>
<td>35.84</td>
<td>92.61</td>
<td>96.31</td>
<td>73.40</td>
<td>35.03</td>
<td>92.70</td>
<td>94.63</td>
<td>71.67</td>
</tr>
<tr>
<td></td>
<td>ATOM (q=0.5)</td>
<td>35.48</td>
<td>91.29</td>
<td>91.13</td>
<td>69.07</td>
<td>64.43</td>
<td>77.86</td>
<td>91.39</td>
<td>62.93</td>
</tr>
<tr>
<td></td>
<td>ATOM (q=0.75)</td>
<td>43.13</td>
<td>88.42</td>
<td>89.83</td>
<td>63.89</td>
<td>43.17</td>
<td>88.45</td>
<td>90.05</td>
<td>63.84</td>
</tr>
</tbody>
</table>
### Analysis

How does ATOM affect the in-distribution classification accuracy?

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>MSP</td>
<td>5.01</td>
<td>94.39</td>
<td>91.76</td>
</tr>
<tr>
<td></td>
<td>ODIN</td>
<td>5.01</td>
<td>94.39</td>
<td>91.02</td>
</tr>
<tr>
<td></td>
<td>Mahalanobis</td>
<td>5.01</td>
<td>94.39</td>
<td>89.71</td>
</tr>
<tr>
<td></td>
<td>SOFL</td>
<td>5.01</td>
<td>95.11</td>
<td>91.60</td>
</tr>
<tr>
<td></td>
<td>OE</td>
<td>5.01</td>
<td>94.79</td>
<td>91.86</td>
</tr>
<tr>
<td></td>
<td>ACET</td>
<td>5.01</td>
<td>91.70</td>
<td>88.64</td>
</tr>
<tr>
<td></td>
<td>CCU</td>
<td>5.01</td>
<td>94.89</td>
<td>91.88</td>
</tr>
<tr>
<td></td>
<td>ROWL</td>
<td>1.30</td>
<td>89.45</td>
<td>89.45</td>
</tr>
<tr>
<td></td>
<td>ATOM (ours)</td>
<td>5.01</td>
<td>94.98</td>
<td>91.14</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>MSP</td>
<td>5.01</td>
<td>75.05</td>
<td>73.87</td>
</tr>
<tr>
<td></td>
<td>ODIN</td>
<td>5.01</td>
<td>75.05</td>
<td>73.50</td>
</tr>
<tr>
<td></td>
<td>Mahalanobis</td>
<td>5.01</td>
<td>75.05</td>
<td>71.20</td>
</tr>
<tr>
<td></td>
<td>SOFL</td>
<td>5.01</td>
<td>74.37</td>
<td>72.62</td>
</tr>
<tr>
<td></td>
<td>OE</td>
<td>5.01</td>
<td>75.28</td>
<td>73.74</td>
</tr>
<tr>
<td></td>
<td>ACET</td>
<td>5.01</td>
<td>74.99</td>
<td>73.43</td>
</tr>
<tr>
<td></td>
<td>CCU</td>
<td>5.01</td>
<td>76.04</td>
<td>74.60</td>
</tr>
<tr>
<td></td>
<td>ROWL</td>
<td>0.40</td>
<td>67.53</td>
<td>67.53</td>
</tr>
<tr>
<td></td>
<td>ATOM (ours)</td>
<td>5.01</td>
<td>75.49</td>
<td>73.57</td>
</tr>
</tbody>
</table>

Chen et al
### Analysis

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>FPR (5% FNR)</th>
<th>AUROC</th>
<th>FPR (5% FNR)</th>
<th>AUROC</th>
<th>FPR (5% FNR)</th>
<th>AUROC</th>
<th>FPR (5% FNR)</th>
<th>AUROC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Natural OOD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>MSP</td>
<td>50.54</td>
<td>91.79</td>
<td>100.00</td>
<td>58.34</td>
<td>100.00</td>
<td>13.83</td>
<td>100.00</td>
<td>13.67</td>
</tr>
<tr>
<td></td>
<td>ODIN</td>
<td>21.72</td>
<td>94.72</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mahalanobis</td>
<td>28.50</td>
<td>89.60</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SOFL</td>
<td>2.78</td>
<td>99.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OE</td>
<td>3.66</td>
<td>98.82</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACET</td>
<td>13.13</td>
<td>97.61</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CCU</td>
<td>3.39</td>
<td>98.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ROWL</td>
<td>43.14</td>
<td>77.78</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATOM (ours)</td>
<td><strong>2.07</strong></td>
<td><strong>99.11</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>MSP</td>
<td>78.05</td>
<td>76.11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ODIN</td>
<td>53.03</td>
<td>84.45</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mahalanobis</td>
<td>43.25</td>
<td>85.65</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SOFL</td>
<td>43.36</td>
<td>91.21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OE</td>
<td>49.21</td>
<td>88.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ACET</td>
<td>47.69</td>
<td>88.47</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CCU</td>
<td>43.04</td>
<td>90.95</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ROWL</td>
<td>95.82</td>
<td>51.90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATOM (ours)</td>
<td><strong>34.06</strong></td>
<td><strong>93.79</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Note: Bold values indicate the best performance.*

---

Chen et al
Potential improvements via learning better representation

Self-training with rotation prediction improves robustness:

- Provides strong regularization to correct bias and concentrate on global structures/shapes
- Texture alone not sufficient for determining whether the zebra is flipped
Q & A
References

- Robust Out-of-distribution Detection via Informative Outlier Mining
- Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem
- Self-Supervised Learning for Generalizable Out-of-Distribution Detection
- A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks
- Enhancing the reliability of out-of-distribution image detection in neural networks
- A baseline for detecting misclassified and out-of-distribution examples in neural networks
- Deep anomaly detection with outlier exposure
- Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty
- Overview on Trustworthy ML