Department of Computer Sciences
UNIVERSITY OF WISCONSIN–MADISON

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

Presented by
Bastin Joseph, Yifei Ming

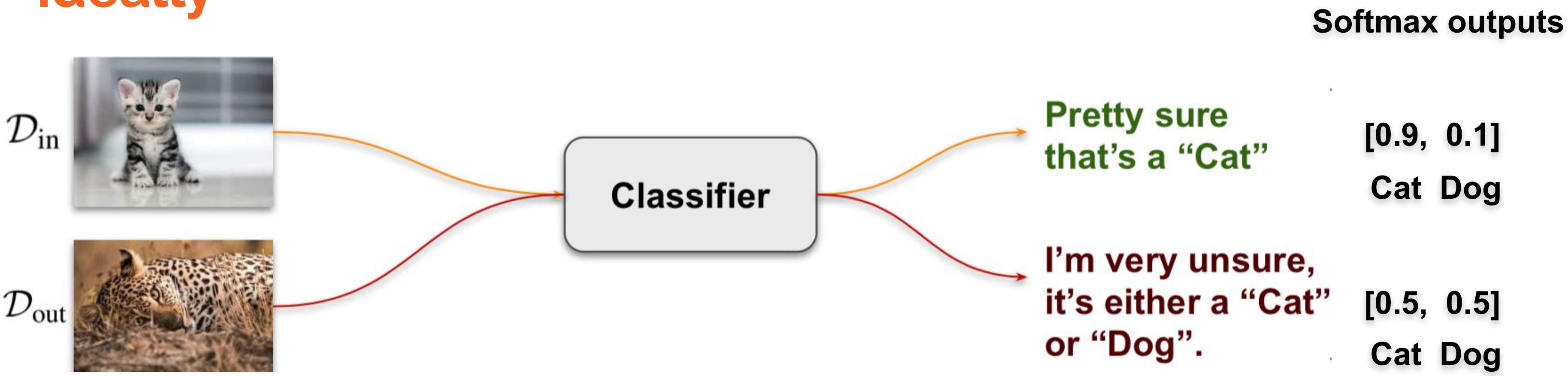
ICML Workshop on Uncertainty & Robustness in Deep Learning (ICML UDL), 2020
Authors : Jiefeng Chen, Yixuan Li, Xi Wu, Yingyu Liang, Somesh Jha

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

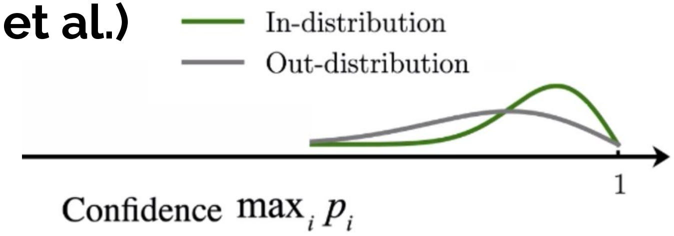
Recap of last lecture

Ideally



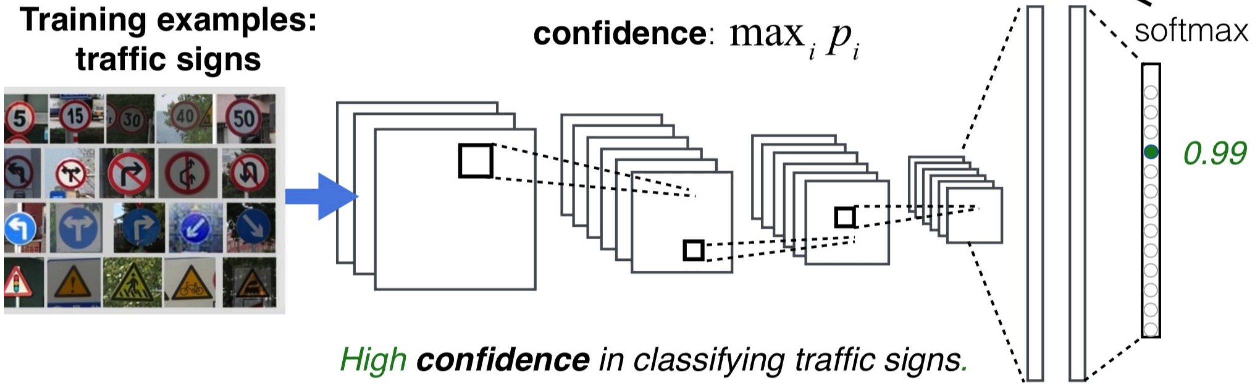
Reality

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

$$p_i(\boldsymbol{x}) = \frac{\exp\left(f_i(\boldsymbol{x})\right)}{\sum_{j=1}^{N} \exp\left(f_j(\boldsymbol{x})\right)} \xrightarrow{\text{ODIN}} p_i(\boldsymbol{x};T) = \frac{\exp(f_i(\boldsymbol{x})/T)}{\sum_{j=1}^{N} \exp(f_j(\boldsymbol{x})/T)}$$



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

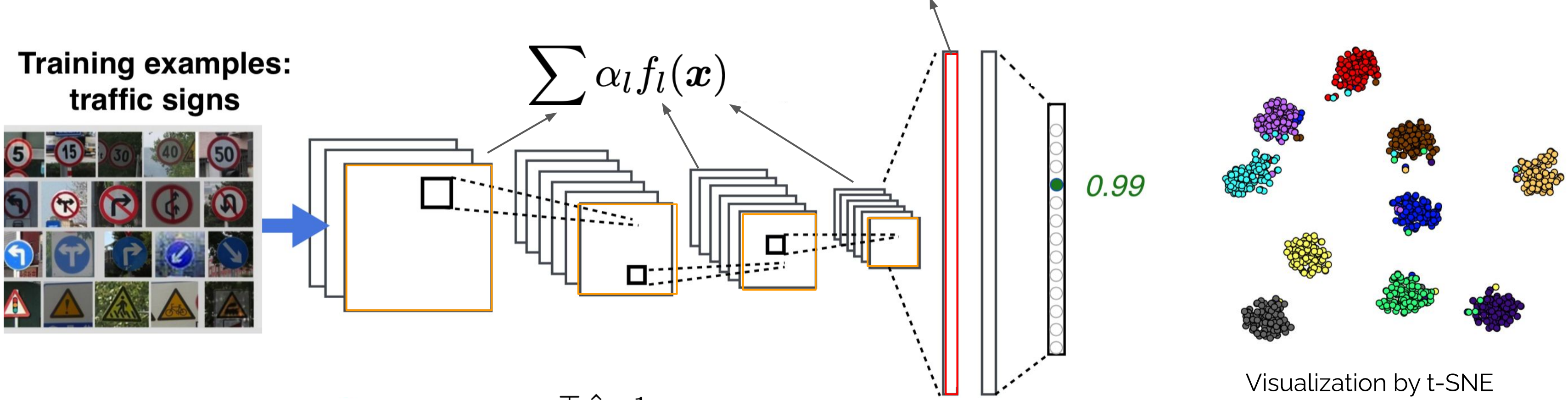
ODIN: input perturbation & temperature scaling for confidence calibration [Liang et.al](#)

Generative classifier: better characterization from the representation space?

Assumption: class-conditional Gaussian with tied covariance $\xrightarrow{\text{GDA}}$ posterior $P(y = i \mid \boldsymbol{x})$ is softmax

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

$f(\boldsymbol{x})$ from penultimate layer



Visualization by t-SNE

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

Measures the log of probability density of the test sample

Lee 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 ?

Improve the robustness of detectors with auxiliary OOD dataset for training

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

$\mathbb{D}_{in}$	The distribution of the training data	$\mathbb{D}_{out}^{oe}$	OOD samples used to train the classifier
$\mathbb{D}_{out}$	Any unknown distribution disjoint from $\mathbb{D}_{in}$	$\mathbb{D}_{out}^{test}$	OOD samples unseen by the classifier, used for test

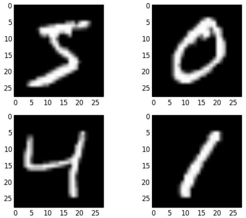
$\mathbb{D}_{in}$

Street View House Number (SVHN)

$\mathbb{D}_{out}^{oe}$

80 Million Tiny Image

$\mathbb{D}_{out}^{test}$

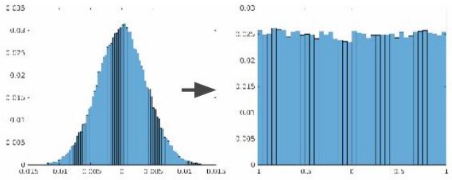


MNIST

Hendrycks et al

OE for multi-class classification

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{in}}}\underbrace{[-\log f_y(x)]}_{\text{Negative log probability of f(x) belonging to real class y}} + \lambda\mathbb{E}_{x\sim\mathcal{D}_{\text{out}}^{\text{OE}}}\underbrace{[H(\mathcal{U}; f(x))]}_{\text{Cross entropy between a uniform distribution and posterior distribution of f(.) for OOD examples}}$$



Hendrycks et al

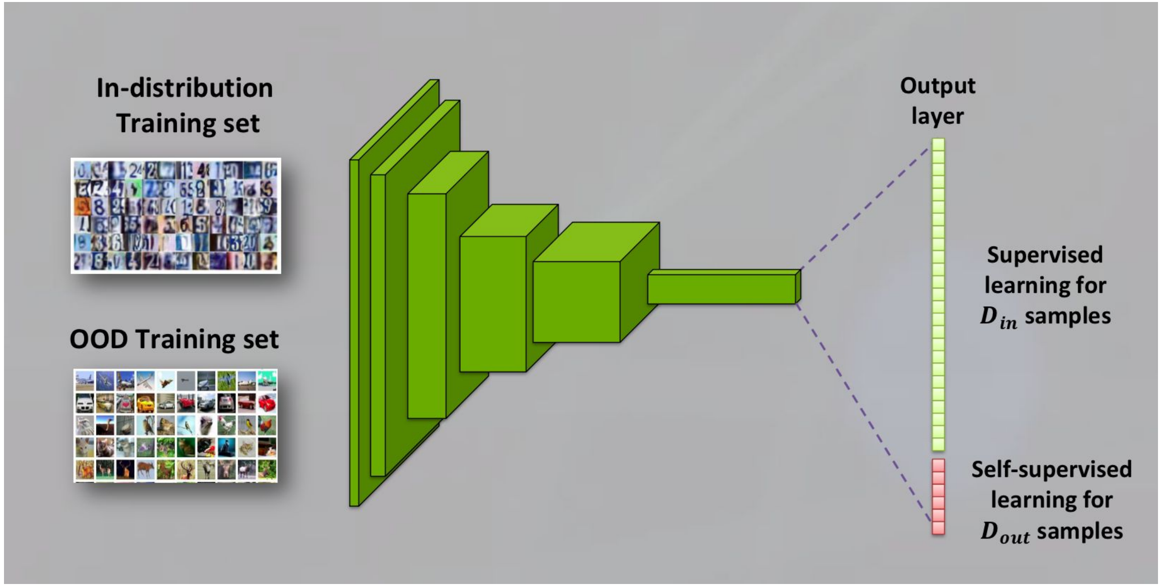
Beyond pre-trained models

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:

$$\mathbb{E}_{(x,y_{in})\sim\mathcal{D}_{in}}[-\log f_{y_{in}}(x)] + \lambda\mathbb{E}_{x\sim\mathcal{D}_{out}}[-\log f_{y_{target}}(x)]$$

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

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 ?

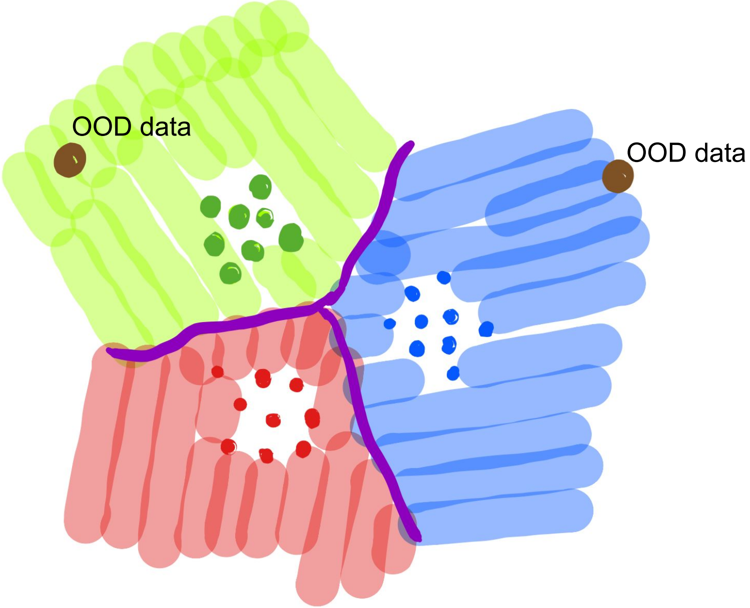
Adversarial Confidence Enhancing Tuning (ACET)

Decision Boundaries in NN are intersections of half spaces

Suppose $x \in R^2$

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

Each color represents the decision region for one class



Decomposition of R^2 for a two-hidden layers ReLU network

Hein et al

Adversarial Confidence Enhancing Training (ACET)

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

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

- ACET
  - Enhance Adversarial Robustness by training on Adversarial OOD

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

$$L_{p_{out}}(f, z) = max_{l=1,2,3,..,K}\, log(\frac{e^{f_l(z)}}{\sum_{k=1}^{K} e^{f_k(z)}})$$

[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

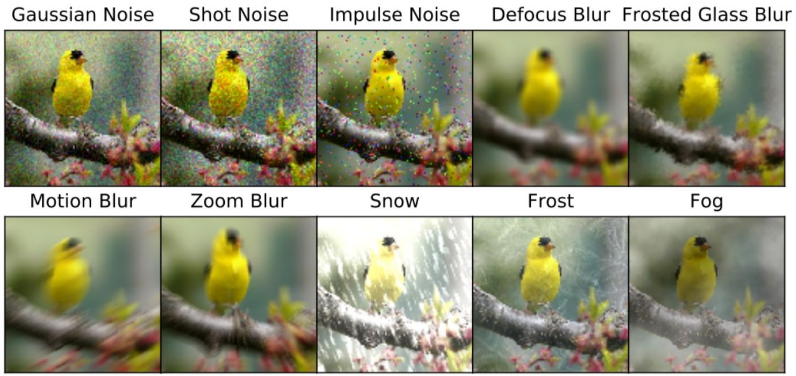
Adversarially perturbed OOD

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

$$\boldsymbol{x}' = \underset{\boldsymbol{x}' \in \Omega_{\infty,\epsilon}(\boldsymbol{x})}{\operatorname{argmax}} \; -\frac{1}{K}\sum_{i=1}^{K} \log f_i(\boldsymbol{x}')$$

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

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 ?

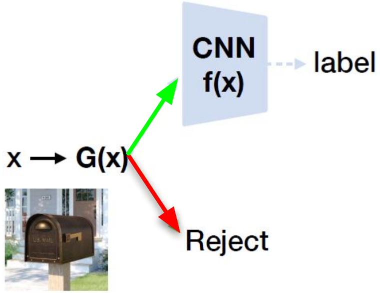
ATOM: better selection of informative OOD samples

Adversarial **T**raining with informative **O**utlier **M**ining

K+1 class classifier

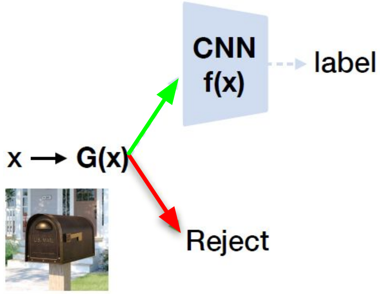
$$y_{in} \in \{1, 2, \cdots, K\}$$

$$y_{out} \in \{K+1\}$$



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

$$f(x) = argmax_{y \in \{1,2,3,..,K\}} f(x)_y$$

Chen et al

Adversarial training for ATOM

- Softmax output on $\boldsymbol{x}: f(\boldsymbol{x})$

- Training objective

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

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

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

unlabelled OOD samples

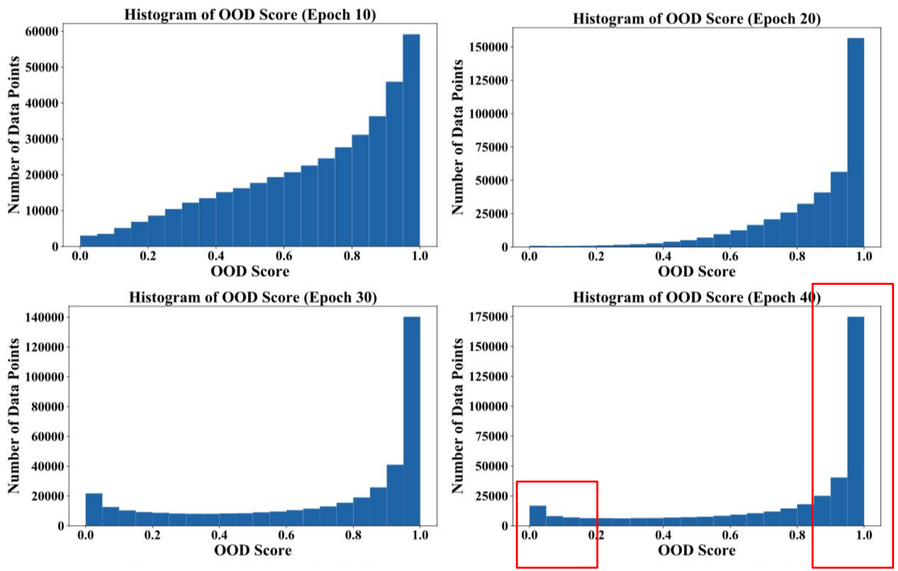
[Chen et al](#)

How to select OOD data for training?

Abundance of **Unlabelled** , **OOD** data

Random sampling ?

Converge quickly after few epochs

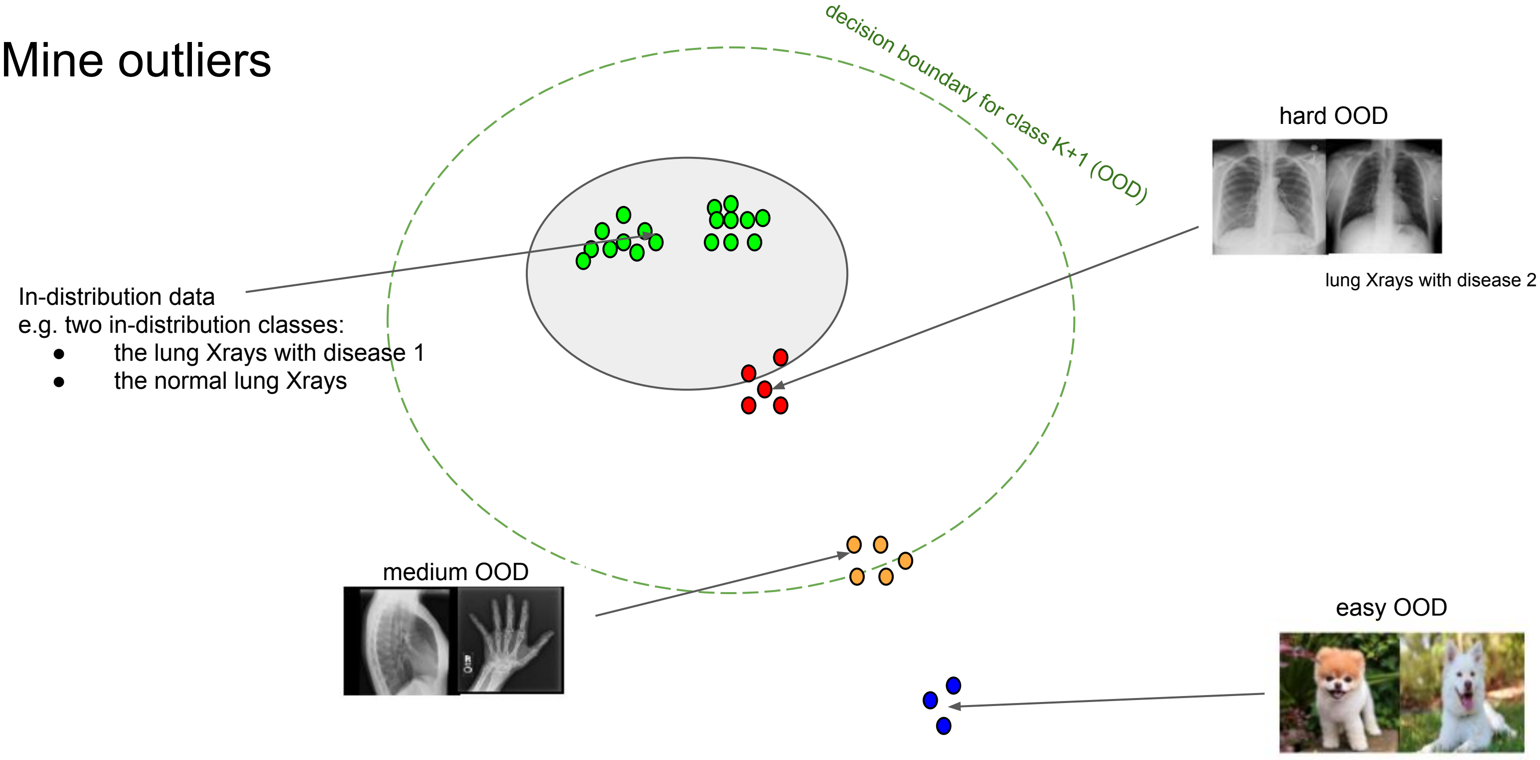


Chen et al

Department of Computer Sciences
UNIVERSITY OF WISCONSIN–MADISON

Harder OOD & easy OOD

- Mine outliers



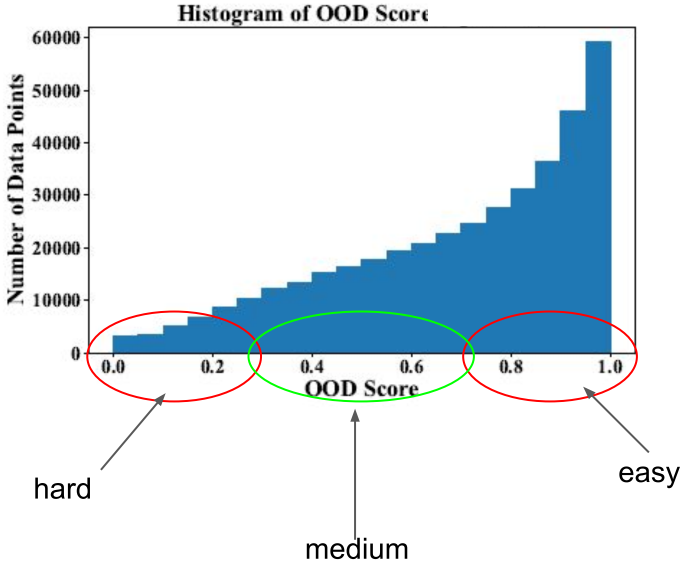
Chen et al

Department of Computer Sciences
UNIVERSITY OF WISCONSIN–MADISON

Hardness based mining

- How do we mine ?

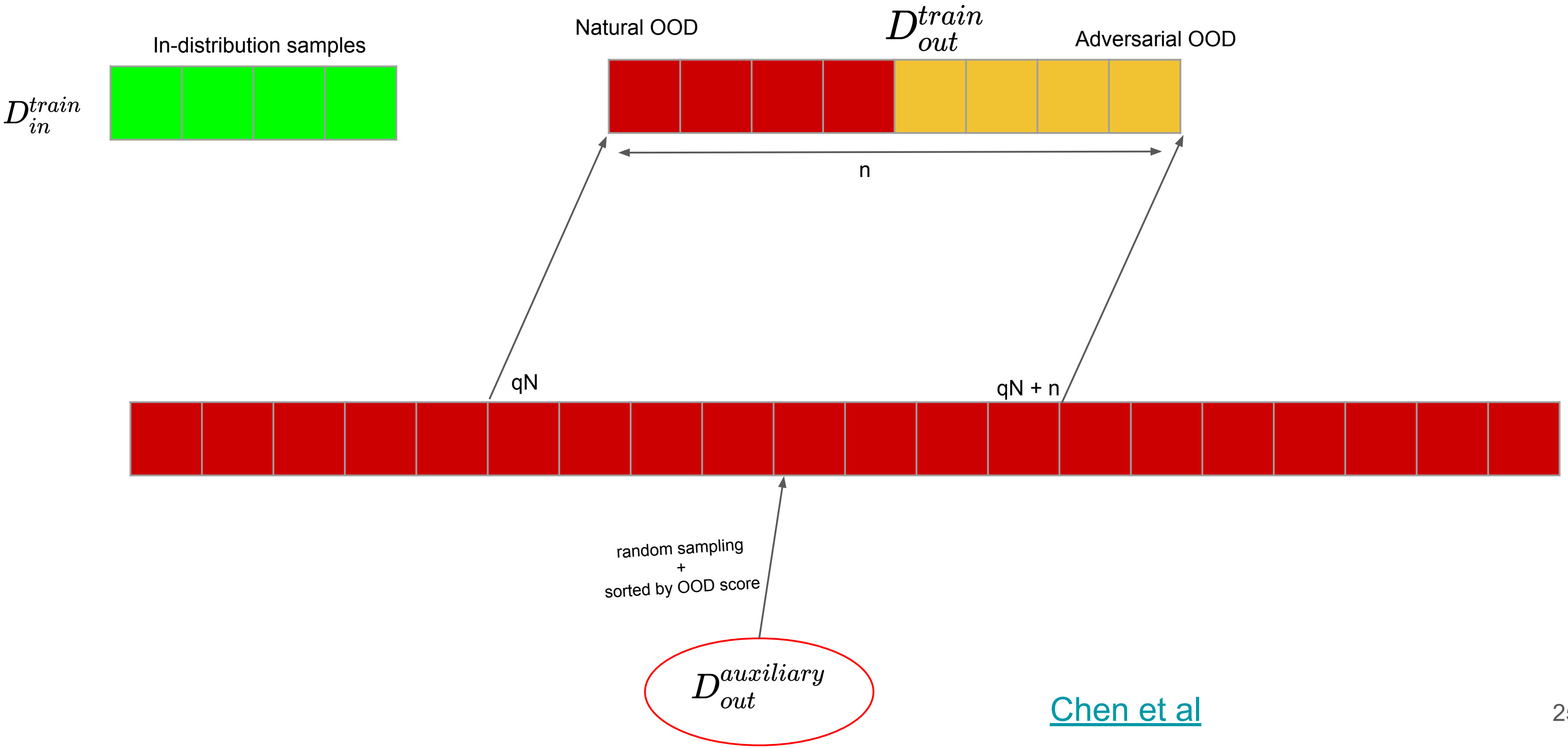
  Sort samples based on OOD scores



$$D_{out}^{auxilliary} \xrightarrow{\text{Random Sampling}} \boxed{\text{N OOD samples}} \xrightarrow{\text{Forward Pass}} \boxed{\text{OOD scores}} \xrightarrow{\text{Select n samples with medium OOD scores}} D_{out}^{train}$$

Chen et al

Put it together: the training procedure



Chen et al

The ATOM algorithm

Department of Computer Sciences
UNIVERSITY OF WISCONSIN–MADISON

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

Chen et al

Department of Computer Sciences
UNIVERSITY OF WISCONSIN–MADISON

Ablation Study - Informative Outlier Mining

- How does the sampling parameter affect performance?

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

Chen et al

Analysis

How does ATOM affect the in-distribution classification accuracy?

$\mathcal{D}_{\text{in}}^{\text{test}}$	Method	FNR	Pred. Acc.	End-to-end. Pred. Acc.
CIFAR-10	MSP	5.01	94.39	91.76
	ODIN	5.01	94.39	91.02
	Mahalanobis	5.01	94.39	89.71
	SOFL	5.01	95.11	91.60
	OE	5.01	94.79	91.86
	ACET	5.01	91.70	88.64
	CCU	5.01	94.89	91.88
	ROWL	1.30	89.45	89.45
	ATOM (ours)	5.01	94.98	91.14
CIFAR-100	MSP	5.01	75.05	73.87
	ODIN	5.01	75.05	73.50
	Mahalanobis	5.01	75.05	71.20
	SOFL	5.01	74.37	72.62
	OE	5.01	75.28	73.74
	ACET	5.01	74.99	73.43
	CCU	5.01	76.04	74.60
	ROWL	0.40	67.53	67.53
	ATOM (ours)	5.01	75.49	73.57

[Chen et al]()

Department of Computer Sciences
UNIVERSITY OF WISCONSIN–MADISON

Analysis

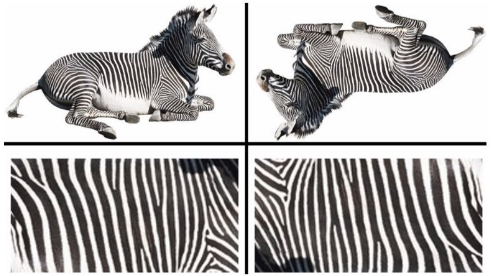
$\mathcal{D}_{\text{in}}^{\text{test}}$	Method	FPR (5% FNR) ↓	AUROC ↑	FPR (5% FNR) ↓	AUROC ↑	FPR (5% FNR) ↓	AUROC ↑	FPR (5% FNR) ↓	AUROC ↑
		Natural OOD		Corruption OOD		L_∞ OOD		Comp. OOD	
CIFAR-10	MSP	50.54	91.79	100.00	58.34	100.00	13.83	100.00	13.67
	ODIN	21.72	94.72	99.30	52.32	99.99	0.17	100.00	0.01
	Mahalanobis	28.50	89.60	94.58	37.76	97.67	3.90	99.93	0.32
	SOFL	2.78	99.04	61.82	88.72	99.98	1.08	100.00	0.77
	OE	3.66	98.82	56.44	90.66	99.95	0.35	99.99	0.16
	ACET	13.13	97.61	68.54	88.00	75.86	77.66	97.86	52.99
	CCU	3.39	98.92	56.50	89.34	99.90	0.36	99.98	0.21
	ROWL	43.14	77.78	94.19	52.26	93.40	52.65	97.86	50.42
	ATOM (ours)	**2.07**	**99.11**	**26.95**	**94.96**	**15.31**	**97.33**	**36.08**	**93.78**
CIFAR-100	MSP	78.05	76.11	100.00	30.08	100.00	2.35	100.00	2.13
	ODIN	53.03	84.45	100.00	36.36	100.00	0.40	100.00	0.01
	Mahalanobis	43.25	85.65	**96.62**	33.47	95.13	26.71	99.91	10.32
	SOFL	43.36	91.21	99.92	45.20	100.00	0.42	100.00	0.30
	OE	49.21	88.05	99.96	45.10	100.00	0.97	100.00	0.59
	ACET	47.69	88.47	99.86	43.38	79.33	50.59	98.60	24.96
	CCU	43.04	90.95	99.90	48.32	100.00	0.78	100.00	0.47
	ROWL	95.82	51.90	100.00	49.80	99.99	49.81	100.00	49.80
	ATOM (ours)	**34.06**	**93.79**	99.08	**72.27**	**52.89**	**82.61**	**96.83**	**68.93**

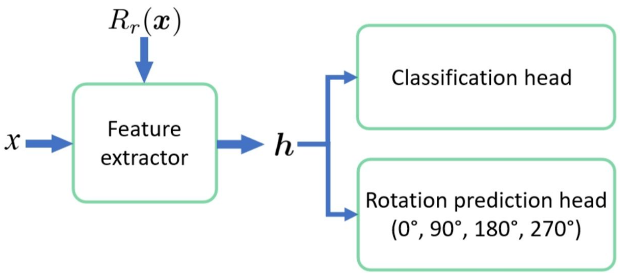
Chen et al

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





$\mathcal{D}_{in}$	$\mathcal{D}_{out}^{test}$	AUROC ↑ MSP	AUROC ↑ Rotation
CIFAR-10	Gaussian	96.3	99.0
	Rademacher	97.5	99.1
	Blobs	94.6	98.9
	Textures	87.9	97.4
	SVHN	91.9	98.9
	Places365	87.7	92.2
	LSUN	88.5	93.2
	CIFAR-100	87.2	90.9
	Mean	91.4	**96.2**

[Hendrycks et al](#)

Q & A

References

Department of Computer Sciences
UNIVERSITY OF WISCONSIN–MADISON

- 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