Robustness of Graph OOD

Jiaqing Xie

5.10.2024

EMORY

Overview

First Part

1. OOD Detectors
2. Attack on OOD Detectors
3. Defense on OOD Detectors

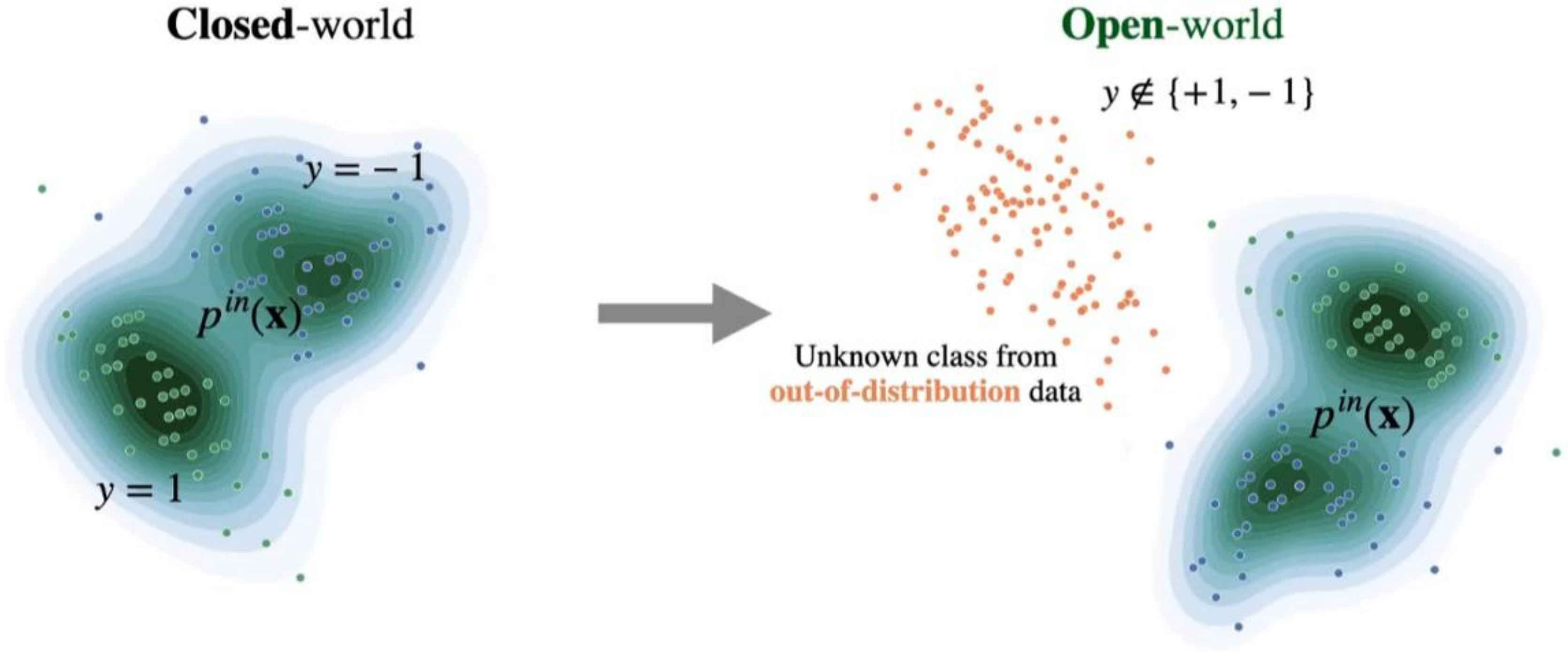
Second Part

1. Graph OOD Detectors
2. Attack and robustness on Graph OOD Detectors

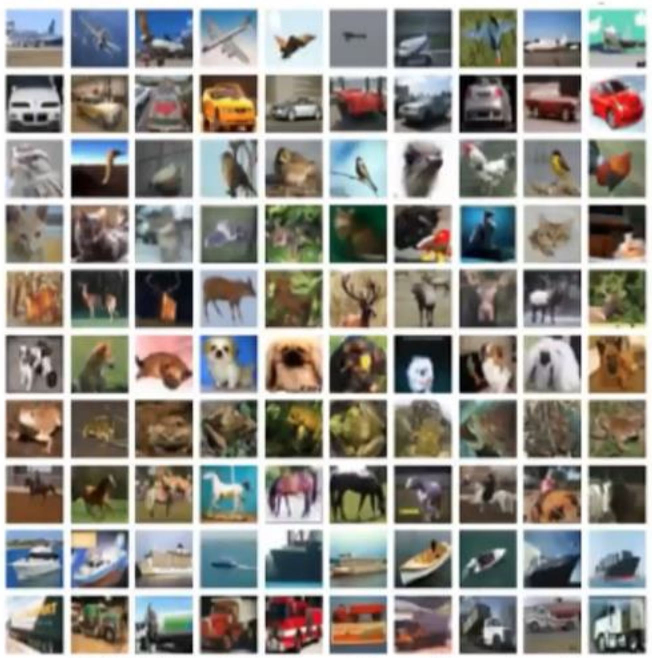
Part I

1. OOD Detectors

Out-of-distribution (OOD) Data



Out-of-distribution (OOD) Data

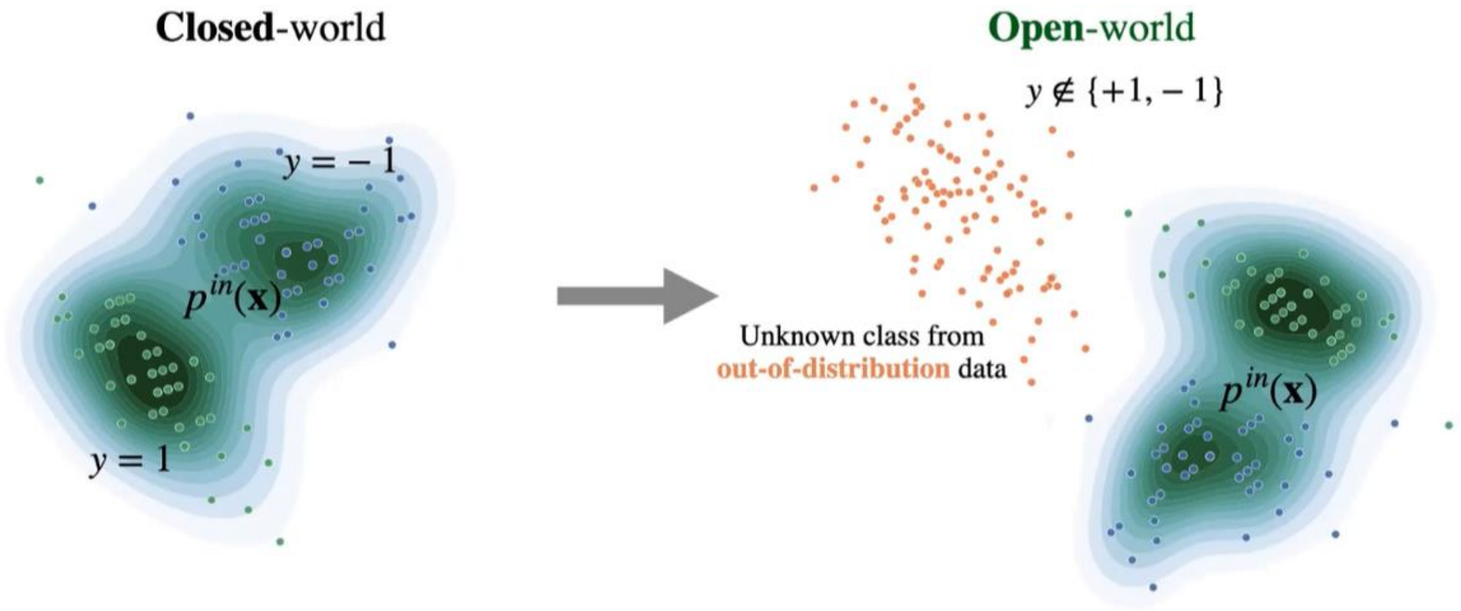


CIFAR-10 (in-distribution)

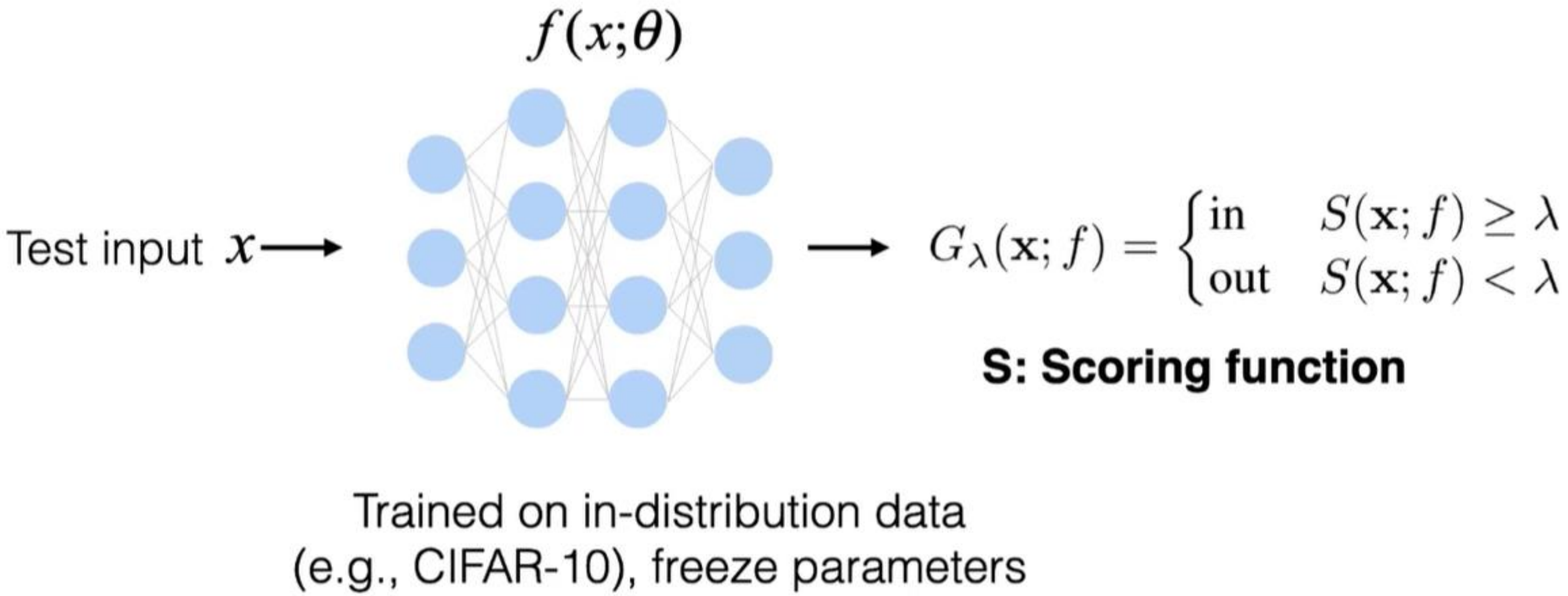
SVHN (OOD)

Challenges

- Supervise only in-distribution data
- OOD data are in high dimensional space



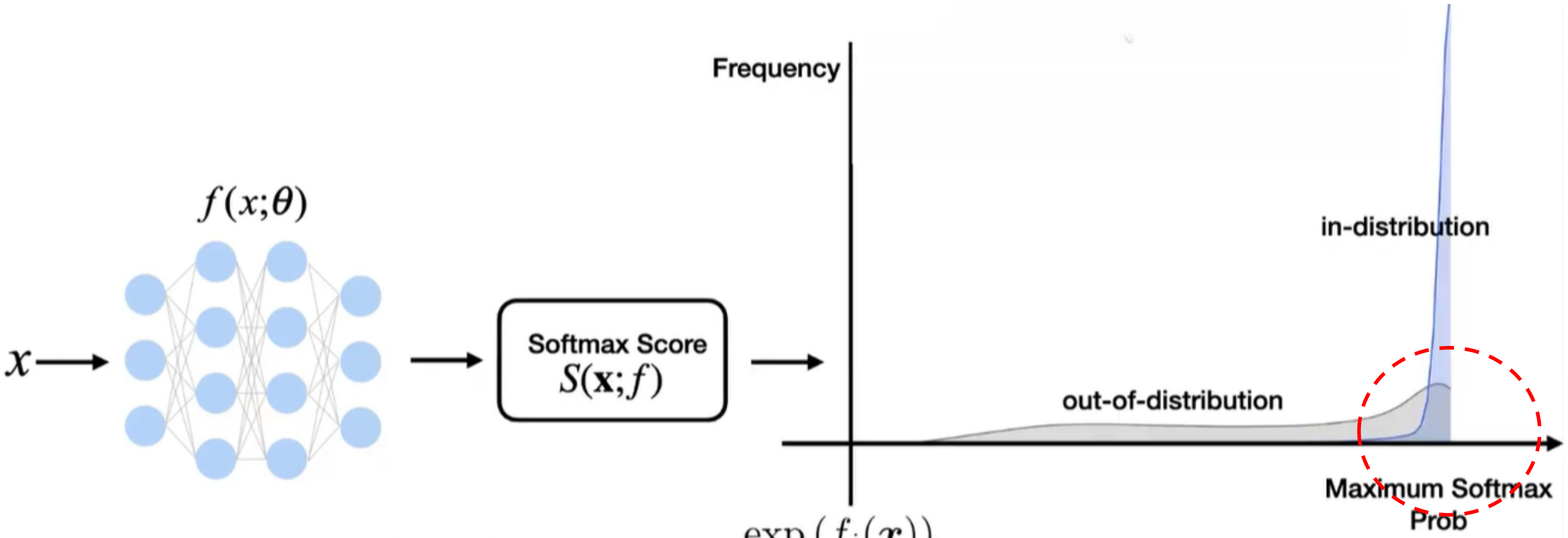
OOD Detection



Question: How to design the scoring function ?

Motivation: Output-Based

- Maximum Softmax Probability



$$S(\mathbf{x};f) = \max_i \frac{\exp\left(f_i(\boldsymbol{x})\right)}{\sum_{j=1}^{N} \exp\left(f_j(\boldsymbol{x})\right)},$$

Overlapped

A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks, ICLR 2017

Motivation: Output-Based

- ODIN (scaling)

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

- Input processing

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \varepsilon \mathrm{sign}(-\nabla_{\boldsymbol{x}} \log S_{\hat{y}}(\boldsymbol{x}; T)).$$

- OOD Detector

$$g(\boldsymbol{x}; \delta, T, \varepsilon) = \begin{cases} 1 & \text{if} \max_i p(\tilde{\boldsymbol{x}}; T) \le \delta, \\ 0 & \text{if} \max_i p(\tilde{\boldsymbol{x}}; T) > \delta. \end{cases}$$

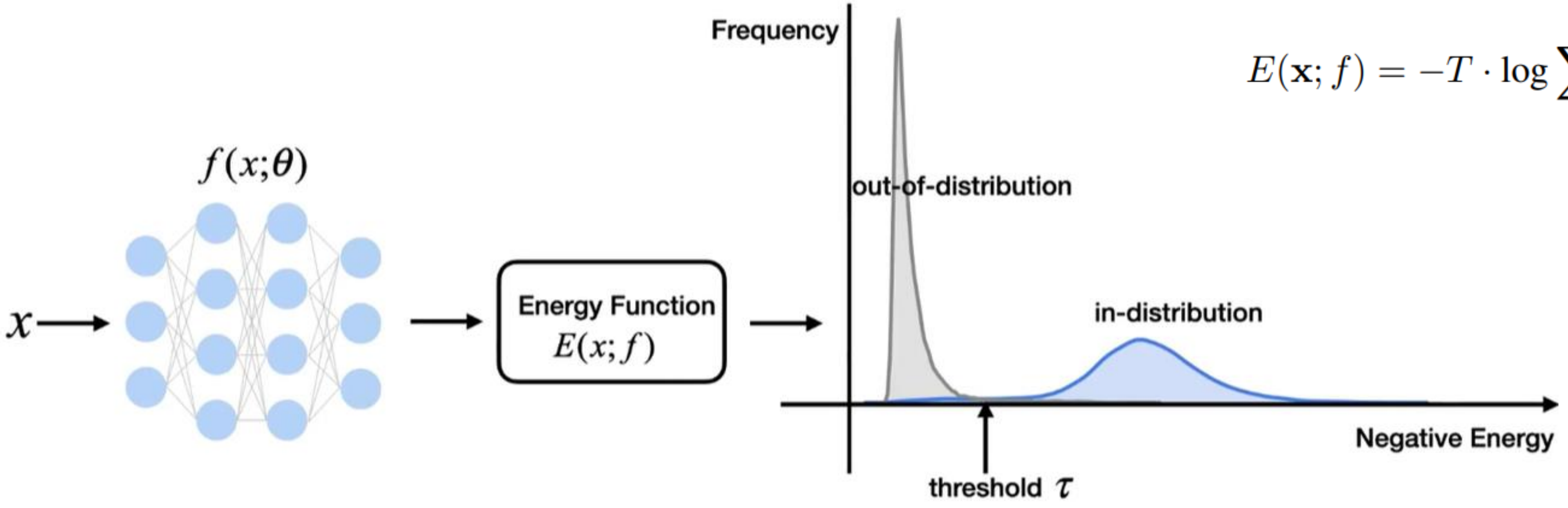
Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks, ICLR 2018

Motivation: Output-Based

- Energy

$$p(y \mid \mathbf{x}) = \frac{e^{f_y(\mathbf{x})/T}}{\sum_i^K e^{f_i(\mathbf{x})/T}}$$

$$E(\mathbf{x}; f) = -T \cdot \log \sum_i^K e^{f_i(\mathbf{x})/T}$$

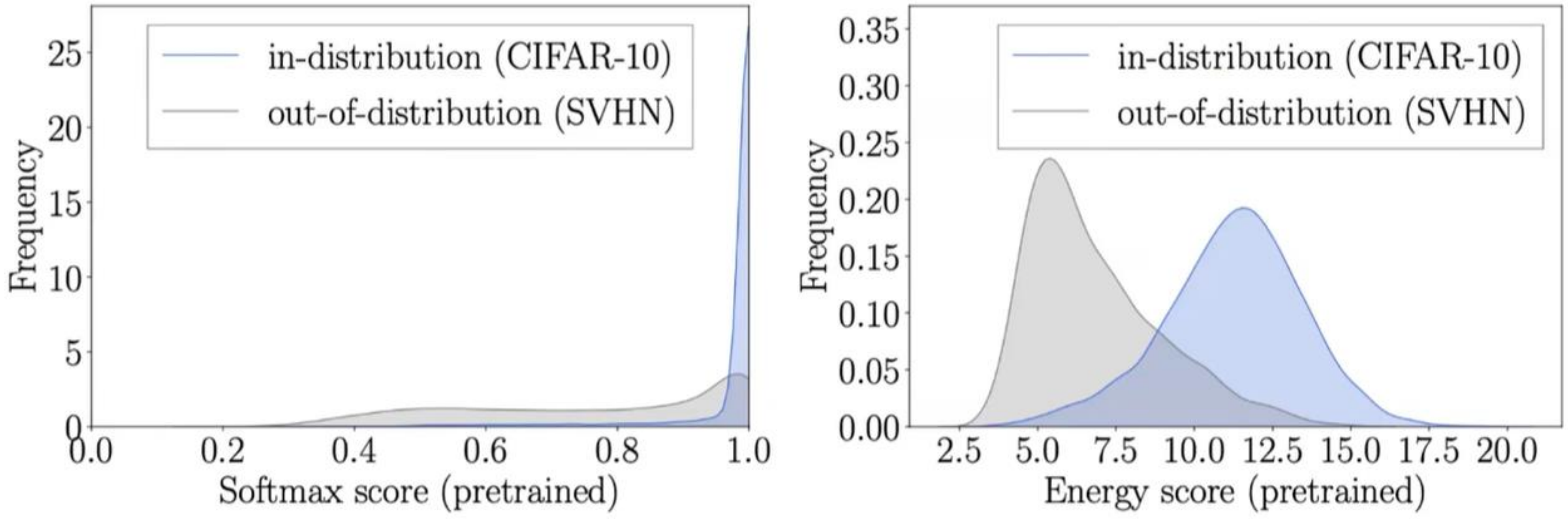


$$g(\mathbf{x}; \tau, f) = \begin{cases} 0 & \text{if } -E(\mathbf{x}; f) \leq \tau \\ 1 & \text{if } -E(\mathbf{x}; f) > \tau \end{cases}$$

Energy-based Out-of-distribution Detection, NeurIPS 2020

Comparison

- In distribution data: CIFAR-10
- OOD data: SVHN

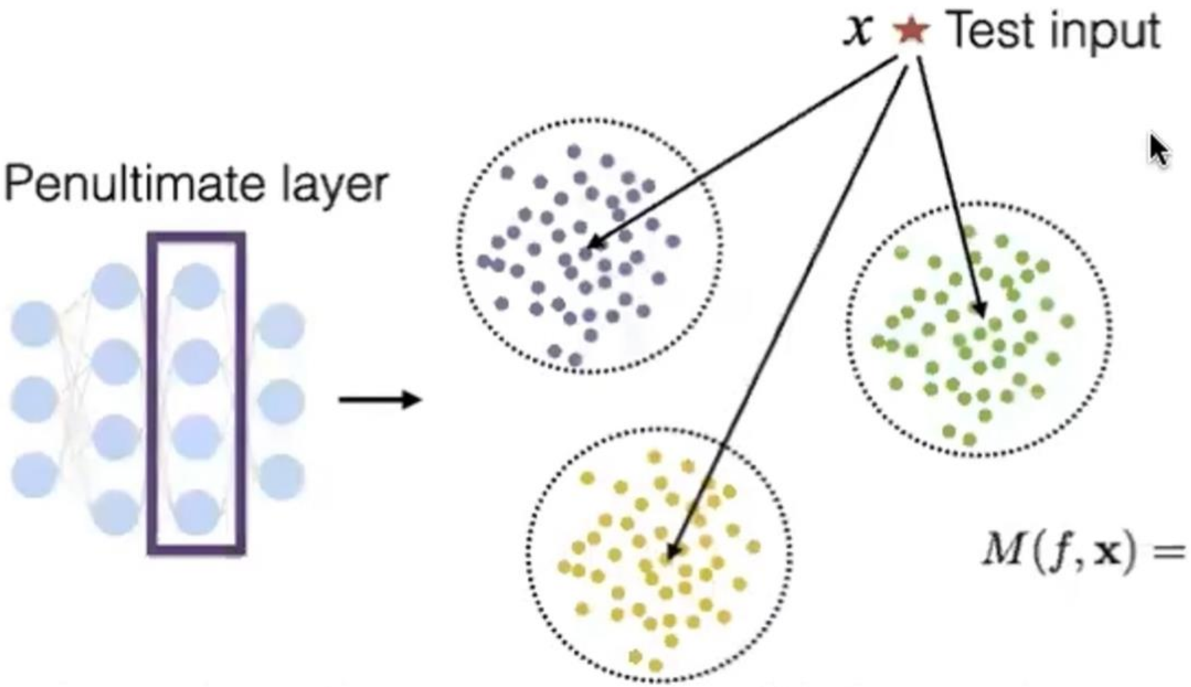


(a) FPR95: 48.87

(b) FPR95: 35.68

Motivation: Distance-Based

- Mahalanobis distance
- Idea: Model feature space as a mixture of multivariate Gaussian



$$\widehat{\mu}_c = \frac{1}{N_c}\sum_{i:y_i=c} f(\mathbf{x}_i),$$

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum\sum \left(f(\mathbf{x}_i)-\widehat{\mu}_c\right)\left(f(\mathbf{x}_i)-\widehat{\mu}_c\right)^\top$$

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

A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks, NeurIPS 2018

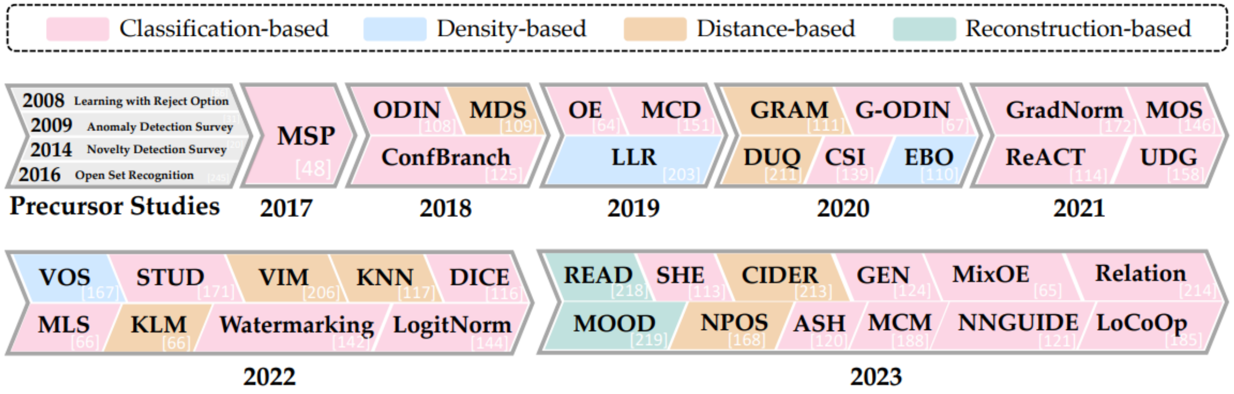
Motivation: Outlier Exposure

- Outliers D_{OE} as auxiliary training data
- During Inference: Detect whether a query is sampled from D_{in} or D_{OE}
- Training Objective

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{in}}}[\mathcal{L}(f(x),y) + \lambda\mathbb{E}_{x'\sim\mathcal{D}_{\text{out}}^{\text{OE}}}[\mathcal{L}_{\text{OE}}(f(x'),f(x),y)]]$$

Deep Anomaly Detection with Outlier Exposure, ICLR 2019

OOD Detection is mature



Sections			References
§ 3.1 Classification	§ 3.1.1 Output-based Methods	a: Training-free	[48, 108, 109, 110, 111, 112, 113, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124]
		b: Training-based	[67, 118, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150]
	§ 3.1.1 Outlier Exposure	a: Real Outliers	[57, 64, 65, 132, 132, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162]
		b: Data Generation	[163, 164, 165, 166, 167, 168, 169, 170, 171]
	§ 3.1.3: Gradient-based Methods		[108, 172, 173]
	§ 3.1.4: Bayesian Models		[174, 175, 176, 177, 178, 179, 180]
	§ 3.1.5: OOD for Foundation Models		[149, 181, 182, 183, 184, 185, 186, 187, 188, 189]
§ 3.2: Density-based Methods			[109, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206]
§ 3.3: Distance-based Methods			[109, 117, 207, 208, 209, 210, 211, 212, 213, 214]
§ 3.4: Reconstruction-based Methods			[215, 216, 217, 218, 219]
§ 3.5: Theoretical Analysis			[33, 56, 58, 59, 220, 221, 222, 223]

Generalized Out-of-Distribution Detection: A Survey 2023

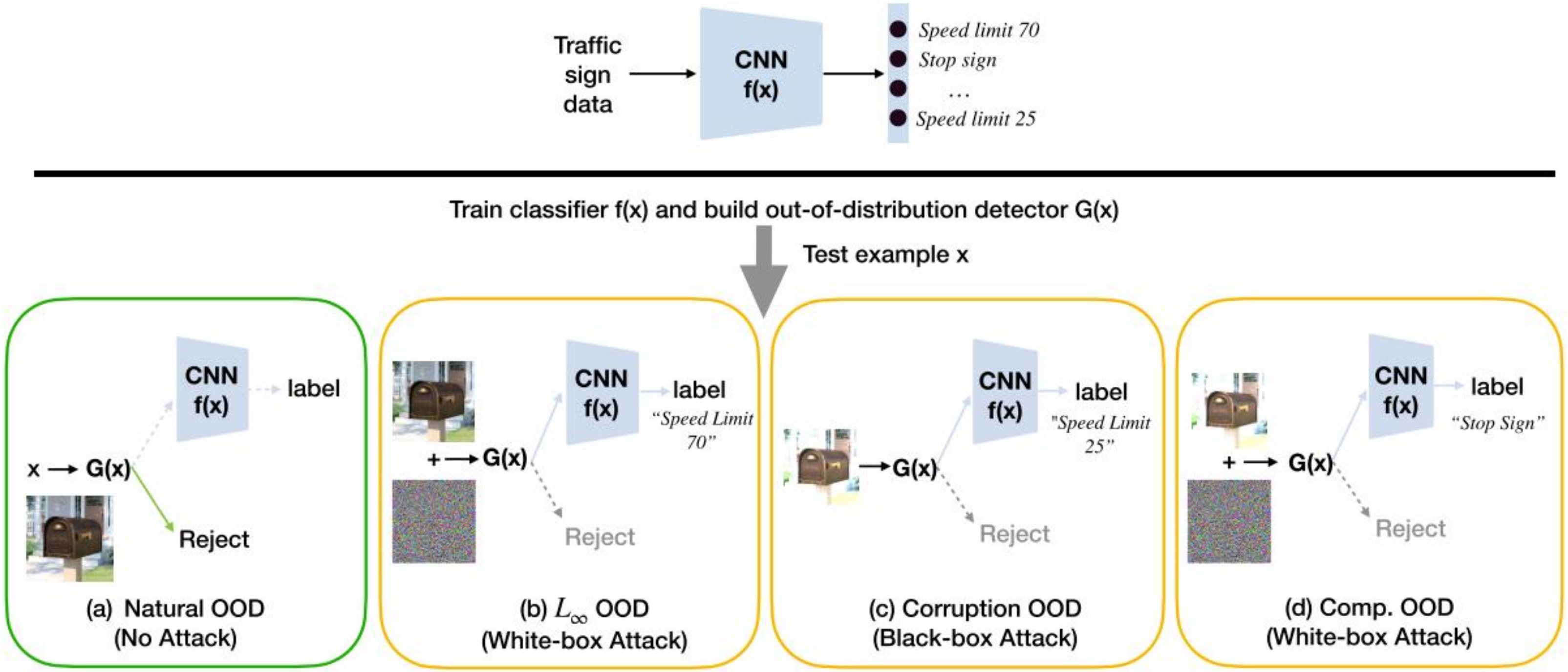
Related Software

- OpenOOD

OpenOOD: Benchmarking Generalized Out-of-Distribution Detection, NeurIPS 2022

2. Attack on OOD Detectors

Adversarial OOD Data



Aim: Fool the detector

ATOM: Robustifying Out-of-distribution Detection Using Outlier Mining, ECML 2021

White-box attack

- L_∞ Attack

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

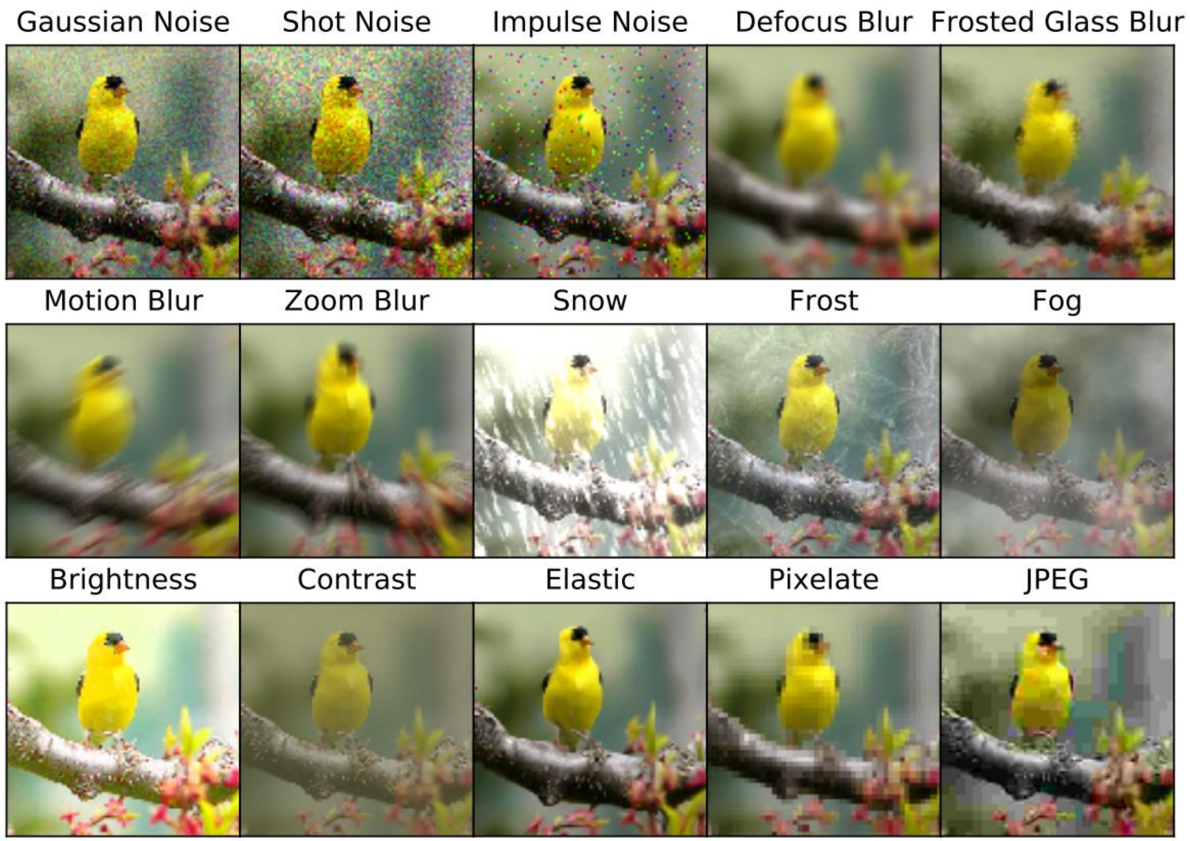
- Valid： within pixel value range (0, 255]
- For MSP, ODIN, OE:

$$\mathbf{x}' = \underset{\mathbf{x}' \in \Omega_{\infty,\epsilon}(\mathbf{x})}{\arg\max} -\frac{1}{K}\sum_{i=1}^{K} \log F(\mathbf{x}')_i$$

- For Mahalanobis:

$$\mathbf{x}' = \underset{\mathbf{x}' \in \Omega_{\infty,\epsilon}(\mathbf{x})}{\arg\max} -\log \frac{1}{1 + e^{-(\sum_\ell \alpha_\ell M_\ell(\mathbf{x}') + b)}}$$

Black-box attack



Select with the one with the lowest OOD score

Benchmarking neural network robustness to common corruptions and perturbations, ICLR 2019

Inlier Attack

- Previous Attacks are on OOD data
- Adversarial In-distribution Data
- In-distribution ===> OOD

Motivation: Inlier Attack

- For softmax confidence measurement such as MSP, ODIN, OE, we let **In-distribution data** close to uniform distribution, and maximize the likelihood for **OOD data**.

$\delta \leftarrow$ randomly choose a vector from $B(x, \epsilon)$
for $t = 1, 2, \cdots, m$ **do**
  $x' \leftarrow x + \delta$
  if x is in-distribution **then**
    $\ell(x') \leftarrow L_{\text{CE}}(F(x'), \mathcal{U}_K)$
  else
    $\ell(x') \leftarrow -\sum_{i=1}^{K} F_i(x') \log F_i(x')$
  end if
  $\delta' \leftarrow \delta - \xi \cdot \text{sign}(\nabla_x \ell(x'))$
  $\delta \leftarrow \prod_{B(x,\epsilon)} \delta'$ $\triangleright$ projecting δ' to $B(x, \epsilon)$

Robust Out-of-distribution Detection for Neural Networks, AAAI 2022

Motivation: Inlier Attack

- For Mahalanobis distance measurement, we want to make the logistic regressor predict wrongly.

$x' \leftarrow x + \delta$

$p(x') \leftarrow \frac{1}{1+e^{-(\sum_\ell \alpha_\ell M_\ell(x')+b)}}$

if x is in-distribution **then**

  $\ell(x') \leftarrow -\log p(x')$

else

  $\ell(x') \leftarrow -\log(1 - p(x'))$

end if

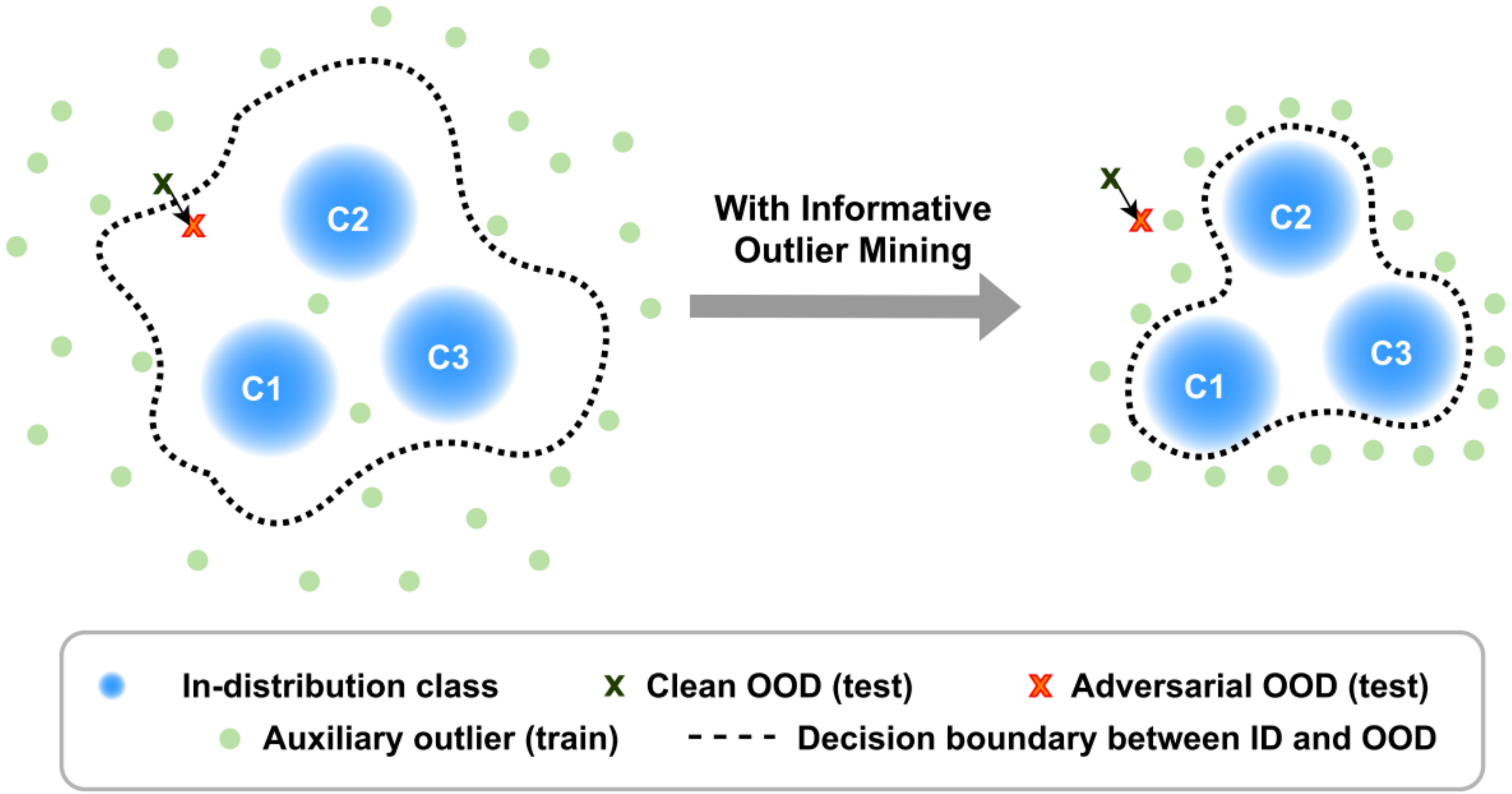
$\delta' \leftarrow \delta + \xi \cdot \text{sign}(\nabla_x \ell(x'))$

$\delta \leftarrow \prod_{B(x,\epsilon)} \delta'$ $\triangleright$ projecting δ' to $B(x, \epsilon)$

Robust Out-of-distribution Detection for Neural Networks, AAAI 2022

3. Defense on OOD Detectors

Motivation: Informative OOD Mining



ATOM: Robustifying Out-of-distribution Detection Using Outlier Mining, ECML 2021

Adversarial Training

- Learning Objective

$$\underset{\theta}{\text{minimize}} \quad \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_{\text{in}}^{\text{train}}}[\ell(\mathbf{x},y;F_\theta)] + \lambda \cdot \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_{\text{out}}^{\text{train}}} \max_{\mathbf{x}'\in\Omega_{\infty,\epsilon}(\mathbf{x})} [\ell(\mathbf{x}',K+1;F_\theta)] \quad (1)$$

- GAN Training Objective

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

- Algorithm

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 F_θ to get set $V = \{F(\mathbf{x})_{K+1} \mid \mathbf{x} \in \mathcal{S}\}$. Sort scores in V from the lowest to the highest;
  $\mathcal{D}_{\text{out}}^{\text{train}} \leftarrow V[qN : qN+n]$; /* $q \in [0, 1-n/N]$ */
  Train F_θ for one epoch using the training objective of (1);
end

ATOM: Robustifying Out-of-distribution Detection Using Outlier Mining, ECML 2021

Defense for Inlier Attack

- Recall Inlier Attack:
  - For Inlier Data, attack should bring down data log-likelihood

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{in}}^{\text{train}}} \max_{\delta\in B(x,\epsilon)} [-\log F_\theta(x+\delta)_y]$$

  - For OOD Data, attack should increase data log-likelihood

$$\mathbb{E}_{x\sim\mathcal{D}_{\text{out}}^{\text{OE}}} \max_{\delta\in B(x,\epsilon)} [L_{\text{CE}}(F_\theta(x+\delta),\mathcal{U}_K)]$$

- Follow Adversarial Training Settings

$$\begin{aligned}\underset{\theta}{\text{minimize}} \quad & \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{in}}^{\text{train}}} \max_{\delta\in B(x,\epsilon)} [-\log F_\theta(x+\delta)_y] \\ & +\lambda\cdot\mathbb{E}_{x\sim\mathcal{D}_{\text{out}}^{\text{OE}}} \max_{\delta\in B(x,\epsilon)} [L_{\text{CE}}(F_\theta(x+\delta),\mathcal{U}_K)]\end{aligned}$$

Robust Out-of-distribution Detection for Neural Networks, AAAI 2022

Results:

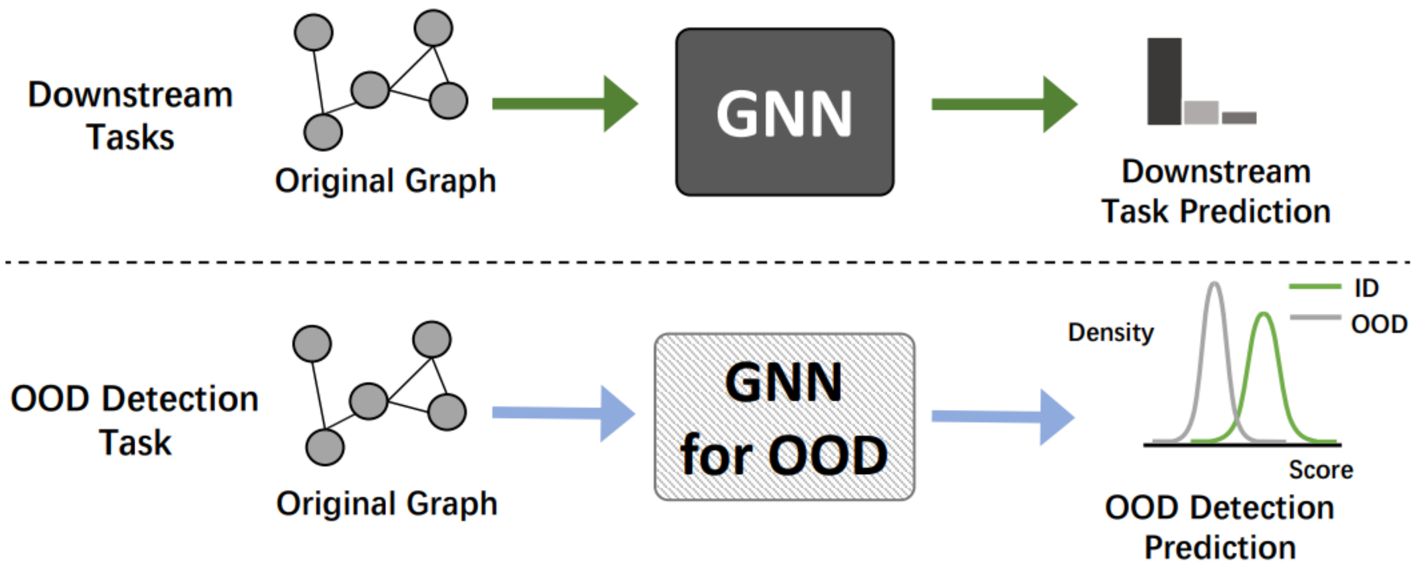
$\mathcal{D}_{\text{in}}^{\text{test}}$	**Method**	**FPR (95% TPR)** ↓	**Detection Error** ↓	**AUROC** ↑	**FPR (95% TPR)** ↓	**Detection Error** ↓	**AUROC** ↑
		without attack			**with attack ($\epsilon = 1/255$, $m = 10$)**		
GTSRB	MSP (Hendrycks and Gimpel 2016)	1.13	2.42	98.45	97.59	26.02	73.27
	ODIN (Liang, Li, and Srikant 2017)	1.42	2.10	98.81	75.94	24.87	75.41
	Mahalanobis (Lee et al. 2018)	1.31	2.87	98.29	100.00	29.80	70.45
	OE (Hendrycks, Mazeika, and Dietterich 2018)	0.02	**0.34**	**99.92**	25.85	5.90	96.09
	OE+ODIN	0.02	0.36	99.92	14.14	5.59	97.18
	ADV (Madry et al. 2017)	1.45	2.88	98.66	17.96	6.95	94.83
	AOE	0.00	0.62	99.86	1.49	2.55	98.35
	ALOE (ours)	**0.00**	0.44	99.76	**0.66**	1.80	98.95
	ALOE+ODIN (ours)	0.01	0.45	99.76	0.69	**1.80**	**98.98**
CIFAR-10	MSP (Hendrycks and Gimpel 2016)	51.67	14.06	91.61	99.98	50.00	10.34
	ODIN (Liang, Li, and Srikant 2017)	25.76	11.51	93.92	93.45	46.73	28.45
	Mahalanobis (Lee et al. 2018)	31.01	15.72	88.53	89.75	44.30	32.54
	OE (Hendrycks, Mazeika, and Dietterich 2018)	4.47	4.50	98.54	99.99	50.00	25.13
	OE+ODIN	**4.17**	**4.31**	**98.55**	99.02	47.84	34.29
	ADV (Madry et al. 2017)	66.99	19.22	87.23	98.44	31.72	66.73
	AOE	10.46	6.58	97.76	88.91	26.02	78.39
	ALOE (ours)	5.47	5.13	98.34	53.99	14.19	91.26
	ALOE+ODIN (ours)	4.48	4.66	98.55	**41.59**	**12.73**	**92.69**
CIFAR-100	MSP (Hendrycks and Gimpel 2016)	81.72	33.46	71.89	100.00	50.00	2.39
	ODIN (Liang, Li, and Srikant 2017)	58.84	22.94	83.63	98.87	49.87	21.02
	Mahalanobis (Lee et al. 2018)	53.75	27.63	70.85	95.79	47.53	17.92
	OE (Hendrycks, Mazeika, and Dietterich 2018)	56.49	19.38	87.73	100.00	50.00	2.94
	OE+ODIN	**47.59**	**17.39**	**90.14**	99.49	50.00	20.02
	ADV (Madry et al. 2017)	85.47	33.17	71.77	99.64	44.86	41.34
	AOE	60.00	23.03	84.57	95.79	43.07	53.80
	ALOE (ours)	61.99	23.56	83.72	92.01	40.09	61.20
	ALOE+ODIN (ours)	58.48	21.38	85.75	**88.50**	**36.20**	**66.61**

Robust Out-of-distribution Detection for Neural Networks, AAAI 2022

Part II

1.Graph OOD Detectors

GNN OOD Detection



GNN Baseline

- GCN

$$Z^{(l)} = \sigma\left(D^{-1/2}\tilde{A}D^{-1/2}Z^{(l-1)}W^{(l)}\right), \quad Z^{(l-1)} = [\mathbf{z}_i^{(l-1)}]_{i\in\mathcal{I}}, \quad Z^{(0)} = X$$

$$h_\theta(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i}) = \mathbf{z}_i^{(L)}.$$

- Predictor

$$p(y \mid \mathbf{x}, \mathcal{G}_{\mathbf{x}}) = \frac{e^{h_\theta(\mathbf{x},\mathcal{G}_{\mathbf{x}})_{[y]}}}{\sum_{c=1}^{C} e^{h_\theta(\mathbf{x},\mathcal{G}_{\mathbf{x}})_{[c]}}}.$$

GNNSafe (1) Data dependence

- Energy

$$E(\mathbf{x}, \mathcal{G}_{\mathbf{x}}, y; h_\theta) = -h_\theta(\mathbf{x}, \mathcal{G}_{\mathbf{x}})_{[y]}$$

- Free energy function

$$E(\mathbf{x}, \mathcal{G}_{\mathbf{x}}; h_\theta) = -\log \sum_{c=1}^{C} e^{h_\theta(\mathbf{x}, \mathcal{G}_{\mathbf{x}})_{[c]}}$$

- Loss Objective

$$\mathcal{L}_{sup} = \mathbb{E}_{(\mathbf{x}, \mathcal{G}_{\mathbf{x}}, y) \sim \mathcal{D}_{in}} \left(-\log p(y \mid \mathbf{x}, \mathcal{G}_{\mathbf{x}})\right)$$

$$= \sum_{i \in \mathcal{I}_s} \left(-h_\theta(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i})_{[y_i]} + \log \sum_{c=1}^{C} e^{h_\theta(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i})_{[c]}} \right)$$

Predictor

$$p(y \mid \mathbf{x}, \mathcal{G}_{\mathbf{x}}) = \frac{e^{h_\theta(\mathbf{x}, \mathcal{G}_{\mathbf{x}})_{[y]}}}{\sum_{c=1}^{C} e^{h_\theta(\mathbf{x}, \mathcal{G}_{\mathbf{x}})_{[c]}}}$$

Motivation 1: Label Propagation

- Problem: not all graph data are labeled

- Solution: Label Propagation, a non-parametric semi-supervised learning algorithm

GNNSafe (2) Label Propagation

- Initialize Energy

$$\mathbf{E}^{(0)} = [E(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i}; h_\theta)]_{i \in \mathcal{I}}$$

- Belief Propagation

$$\mathbf{E}^{(k)} = \alpha \mathbf{E}^{(k-1)} + (1-\alpha) D^{-1} A \mathbf{E}^{(k-1)}, \quad \mathbf{E}^{(k)} = [E_i^{(k)}]_{i \in \mathcal{I}}$$

- Learning Objective

$$\tilde{E}(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i}; h_\theta) = E_i^{(K)}$$

$$G(\mathbf{x}, \mathcal{G}_{\mathbf{x}}; h_\theta) = \begin{cases} 1, & \text{if} \quad \tilde{E}(\mathbf{x}, \mathcal{G}_{\mathbf{x}}; h_\theta) \leq \tau \\ 0, & \text{if} \quad \tilde{E}(\mathbf{x}, \mathcal{G}_{\mathbf{x}}; h_\theta) > \tau \end{cases}$$

GNNSafe (3) Regularization

- Loss Objective

$$\mathcal{L}_{reg} = \frac{1}{|\mathcal{I}_s|} \underbrace{\sum_{i \in \mathcal{I}_s} \left(\mathrm{ReLU} \left(\tilde{E} \left(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i}; h_\theta \right) - t_{in} \right) \right)^2}_{\text{In distribution}} + \frac{1}{|\mathcal{I}_o|} \underbrace{\sum_{j \in \mathcal{I}_o} \left(\mathrm{ReLU} \left(t_{out} - \tilde{E} \left(\mathbf{x}_j, \mathcal{G}_{\mathbf{x}_j}; h_\theta \right) \right) \right)^2}_{\text{Out-of distribution}}$$

Motivation 2: Uncertainty Estimation

- **Vacuity**

$$vac(\omega) \equiv u = K/S \qquad S = \sum_{k=1}^{K} \alpha_k$$

α_k refers to the Dirichlet strength

- **Dissonance**

$$diss(\omega) = \sum_{i=1}^{K} \Big(\frac{b_i \sum_{j \neq i} b_j \mathrm{Bal}(b_j, b_i)}{\sum_{j \neq i} b_j} \Big)$$

$$\mathrm{Bal}(b_j, b_i) = 1 - |b_j - b_i| / (b_j + b_i)$$

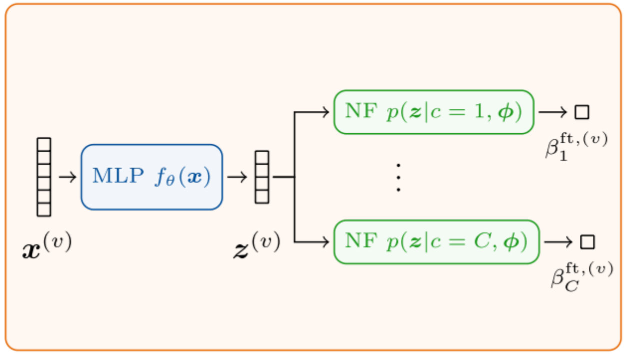
- **Epistemic**, **Aleatoric** and **Entropy**

$$P(y|x) = \int P(y|x; \boldsymbol{\theta}) P(\boldsymbol{\theta}|\mathcal{G}) d\boldsymbol{\theta}$$

$$\underbrace{I(y, \boldsymbol{\theta}|x, \mathcal{G})}_{\textit{Epistemic}} = \underbrace{\mathcal{H}\big[\mathbb{E}_{P(\boldsymbol{\theta}|\mathcal{G})}[P(y|x; \boldsymbol{\theta})]\big]}_{\textit{Entropy}} - \underbrace{\mathbb{E}_{P(\boldsymbol{\theta}|\mathcal{G})}\big[\mathcal{H}[P(y|x; \boldsymbol{\theta})]\big]}_{\textit{Aleatoric}}$$

Uncertainty Aware Semi-Supervised Learning on Graph Data, NeurIPS 2020

Motivation 3: Posterior



- Low-dimensional Space Mapping

$$\boldsymbol{z}^{(v)} = f_\theta(\boldsymbol{x}^{(v)}) \in \mathbb{R}^H$$

- Density measurement (pseudo-counts)

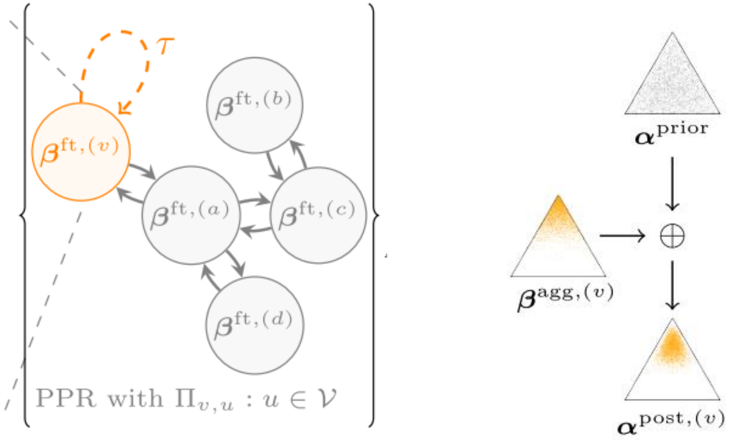
$$\beta_c^{\mathrm{ft},(v)} \propto \mathbb{P}(\boldsymbol{z}^{(v)} \mid c; \boldsymbol{\phi})$$

- Input-dependent param update

$$\beta_c^{\mathrm{agg},(v)} = \sum_{u \in \mathcal{V}} \Pi_{v,u}^{ppr} \beta_c^{\mathrm{ft},(u)}$$

$$\boldsymbol{\alpha}^{\mathrm{post},(v)} = \boldsymbol{\alpha}^{\mathrm{prior}} + \boldsymbol{\beta}^{(v)}$$

$$\boldsymbol{p}^{(v)} \sim \mathrm{Dir}(\boldsymbol{\alpha}^{\mathrm{post},(v)})$$



Graph Posterior Network: Bayesian Predictive Uncertainty for Node Classification, NeurIPS 2021

Motivation 4 **Attention** + Regularizer

- Attention Computation

$$e_{ij} = 1 - |w(i) - w(j)|$$

$$\alpha_{ij} = softmax_j(e_{ij}) = \frac{exp(e_{ij})}{\sum_{k \in \mathcal{N}(i) \cup \{v_i\}} exp(e_{ik})}$$

$$\mathbf{z}_i = softmax\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j \in \mathcal{N}(i) \cup \{v_i\}} \alpha_{ij}^{k}\mathbf{W}^{k}\mathbf{h}_j\right)$$

- Learning Objective: Negative Loss-Likelihood

Motivation 4 Attention + **Regularizer**

- Consistency Regularizer

- W: OOD Score predicted by classifier

$$\mathbf{w} = [w_1, w_2, \cdots, w_{|\mathcal{V}|}]^\top$$
$$w_i = \sigma(\mathbf{a}^\top \mathbf{W}\mathbf{h}_i)$$

- E: OOD Score given by entropy

$$\mathbf{e} = [\sigma(\tilde{e_1}), \sigma(\tilde{e_2}), \cdots, \sigma(\tilde{e_{|\mathcal{V}|}})]^\top$$
$$\tilde{e_i} = \frac{e_i - \mu_e}{\sigma_e}$$
$$e_i = H(\mathbf{z}_i) = -\sum_{j=1}^{|\mathcal{Y}_l|} z_{ij} log(z_{ij})$$

- Loss:

$$\mathcal{L}_{con} = -\cos(\mathbf{w}, \mathbf{e})$$

Learning on Graphs with Out-of-Distribution Nodes, KDD 2022

Motivation 4 Attention + **Regularizer**

- Entropy Regularizer

- Loss:

$$\mathcal{L}_{ent} = \frac{\sum_{i=1}^{|\mathcal{V}|} CE\left(\mathbf{u}, \mathbf{z}_i\right) \delta\big(w(i) > \epsilon\big)}{\sum_{i=1}^{|\mathcal{V}|} \delta\big(w(i) > \epsilon\big)}$$

Motivation 4 Attention + **Regularizer**

- Discrepancy Regularizer

- Two-layer GCN Loss:

$$\mathcal{L}_{dis} = -\cos\left(\mathbf{w}^1, \mathbf{w}^2\right)$$

- Total loss:

$$\mathcal{L}_{OODGAT} = -\frac{1}{|\mathcal{V}_l|}\sum_{i=1}^{|\mathcal{V}_l|} log(z_{iy_i}) \ + a^{b\times t}(\beta\mathcal{L}_{con} + \gamma\mathcal{L}_{ent} + \zeta\mathcal{L}_{dis})$$

Learning on Graphs with Out-of-Distribution Nodes, KDD 2022

Results

	GAT (base)[29]	ODIN [16]	Mahalanobis -Distance[14]	CaGCN [31]	OODGAT -ENT	OODGAT -ATT
			AUROC ↑ / FPR@95 ↓			
Cora	90.7/36.8	90.7/37.2	87.3/50.3	89.9/45.7	93.4/29.6	**94.1**/**25.0**
AmazonCS	84.1/51.9	84.4/51.2	81.8/78.8	83.6/56.2	91.3/**47.2**	**92.3**/52.0
AmazonPhoto	94.3/21.7	94.3/26.5	77.1/59.6	94.4/24.1	98.3/7.3	**98.4**/**4.2**
CoauthorCS	96.2/19.6	96.1/19.8	94.0/25.3	95.8/22.1	99.1/2.4	**99.6**/**1.4**
LastFMAsia	78.5/60.7	81.1/52.9	83.4/51.0	89.6/30.4	**91.4**/**25.4**	90.5/26.8
Wiki-CS	80.4/62.5	80.4/62.5	74.0/74.4	82.7/54.7	**88.7**/50.0	88.6/**49.0**

Learning on Graphs with Out-of-Distribution Nodes, KDD 2022

2. Attack and robustness on Graph OOD Detectors

Review Graph Attack

Table 2: Categorization of representative attack methods

Attack Methods	Attack Knowledge	Targeted or Non-targeted	Evasion or Poisoning	Perturbation Type	Application	Victim Model
PGD, Min-max [76]	White-box	Untargeted	Both	Add/Delete edges	Node Classification	GNN
IG-FGSM [72] IG-JSMA [72]	White-box	Both	Evasion	Add/Delete edges Modify features	Node Classification	GNN
Wang et al. [64]	White-box Gray-box	Targeted	Poisoning	Add/Delete edges	Node Classification	GNN
Nettack [89]	Gray-box	Targeted	Both	Add/Delete edges Modify features	Node Classification	GNN
Metattack [91]	Gray-box	Untargeted	Poisoning	Add/Delete edges	Node Classification	GNN
NIPA [57]	Gray-box	Untargeted	Poisoning	Inject nodes	Node Classification	GNN
RL-S2V [17]	Black-box	Targeted	Evasion	Add/Delete edges	Graph Classification Node Classification	GNN
ReWatt [46]	Black-box	Untargeted	Evasion	Add/Delete edges	Graph Classification	GNN
Liu et al. [43]	White-box Gray-box	Untargted	Poisoning	Flip label Modify features	Classification Regression	G-SSL
FGA [13]	White-box	Targeted	Both	Add/Delete edges	Node Classification Community Detection	Network Emebdding
GF-Attack [9]	Black-box	Targeted	Evasion	Add/Delete edges	Node Classification	Network Emebdding
Bojchevski et al. [5]	Black-box	Both	Poisoning	Add/Delete edges	Node Classification Community Detection	Network Emebdding
Zhang et al. [81]	White-box	Targeted	Poisoning	Add/Delete facts	Plausibility Prediction	Knowledge Graph Embedding
CD-Attack [38]	Black-box	Targeted	Poisoning	Add/Delete edges	Community Detection	Community Detection Algorithm

Adversarial Attacks and Defenses on Graphs: A Review, A Tool and Empirical Studies, SIGKDD Explorations

Traditional Graph Attack

Bad

- To fool the classifier but not OOD detector

Good

- Can create OOD data from In-distribution data.

My Idea

1. Use graph attack to generate some OOD nodes from the original graph.

2. Use Inlier / Outlier attack from ALOE to built adversarial samples

3. Test the robustness of OOD detectors such as GNNSafe.

4. Adversarial Training on graph OOD.

Any Question ?

Thanks !