POEM: Out-of-Distribution Detection with Posterior Sampling

Yifei Ming*

Ying Fan*

Yixuan Li

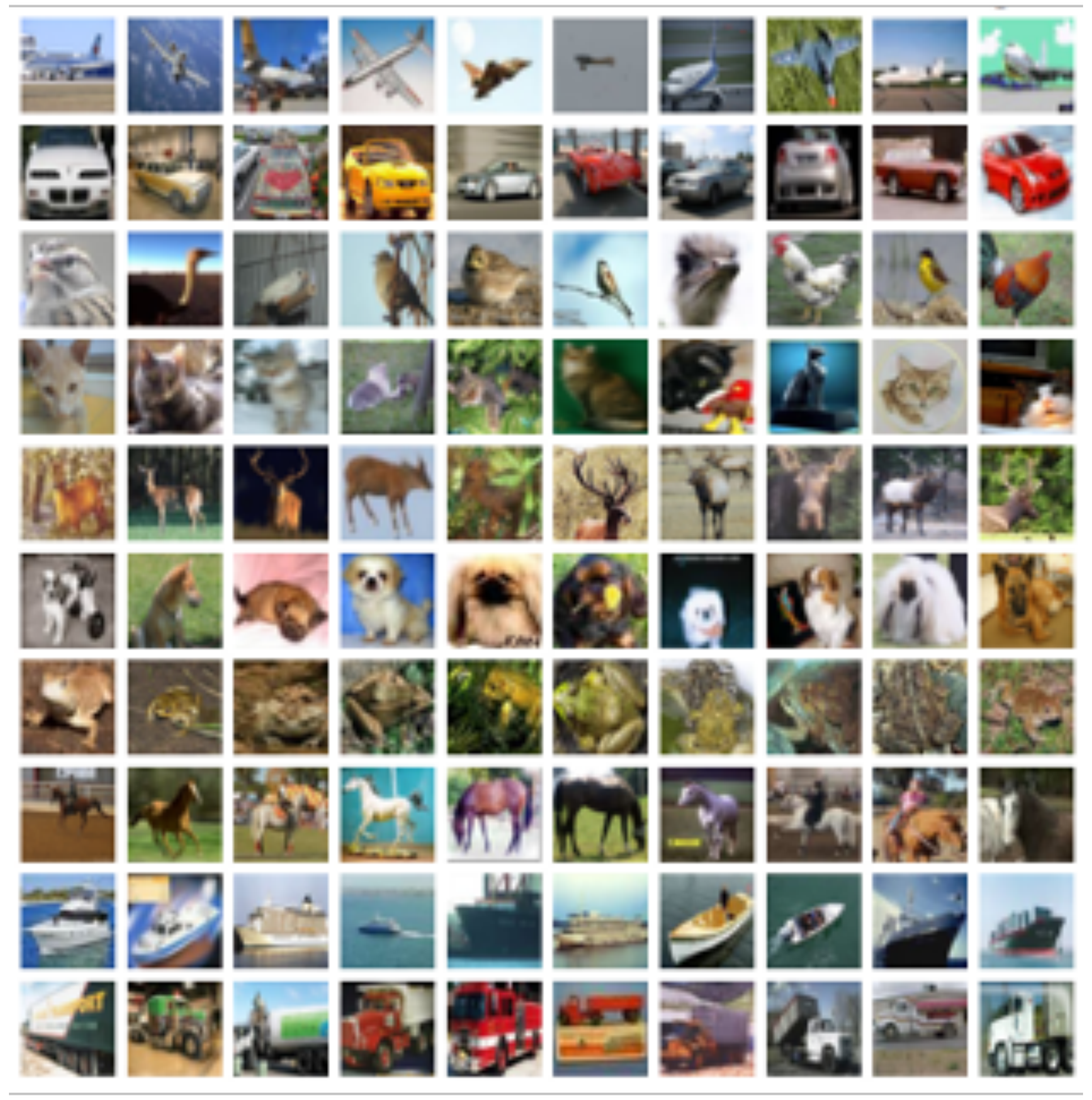
Department of Computer Sciences
University of Wisconsin-Madison

* Equal contribution

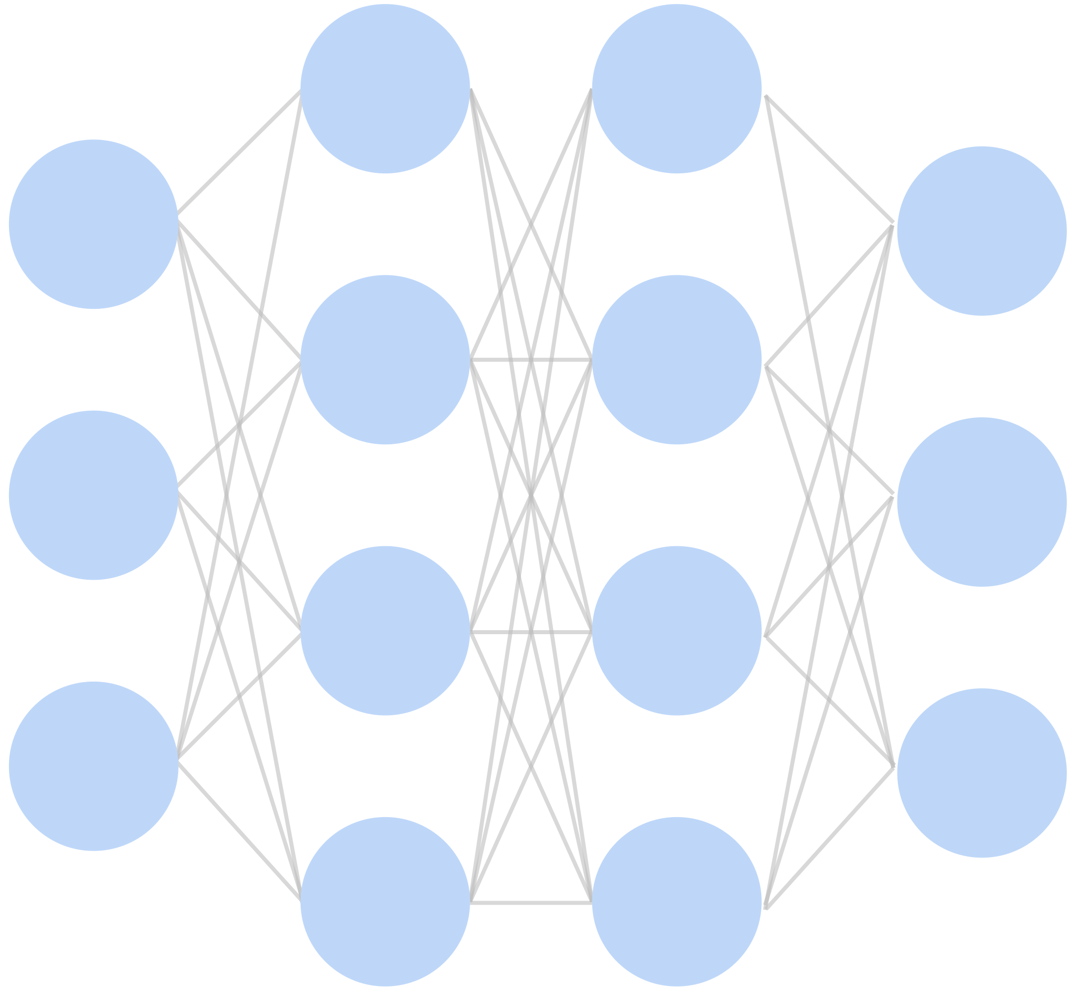
Outline

- Introduction: out-of-distribution (OOD) detection
- OOD detection with outlier exposure
- Outlier mining: a Thompson sampling view
- POEM: posterior sampling-based outlier mining
- Results and analysis

The Task of OOD Detection



CNN $f(x;\theta)$



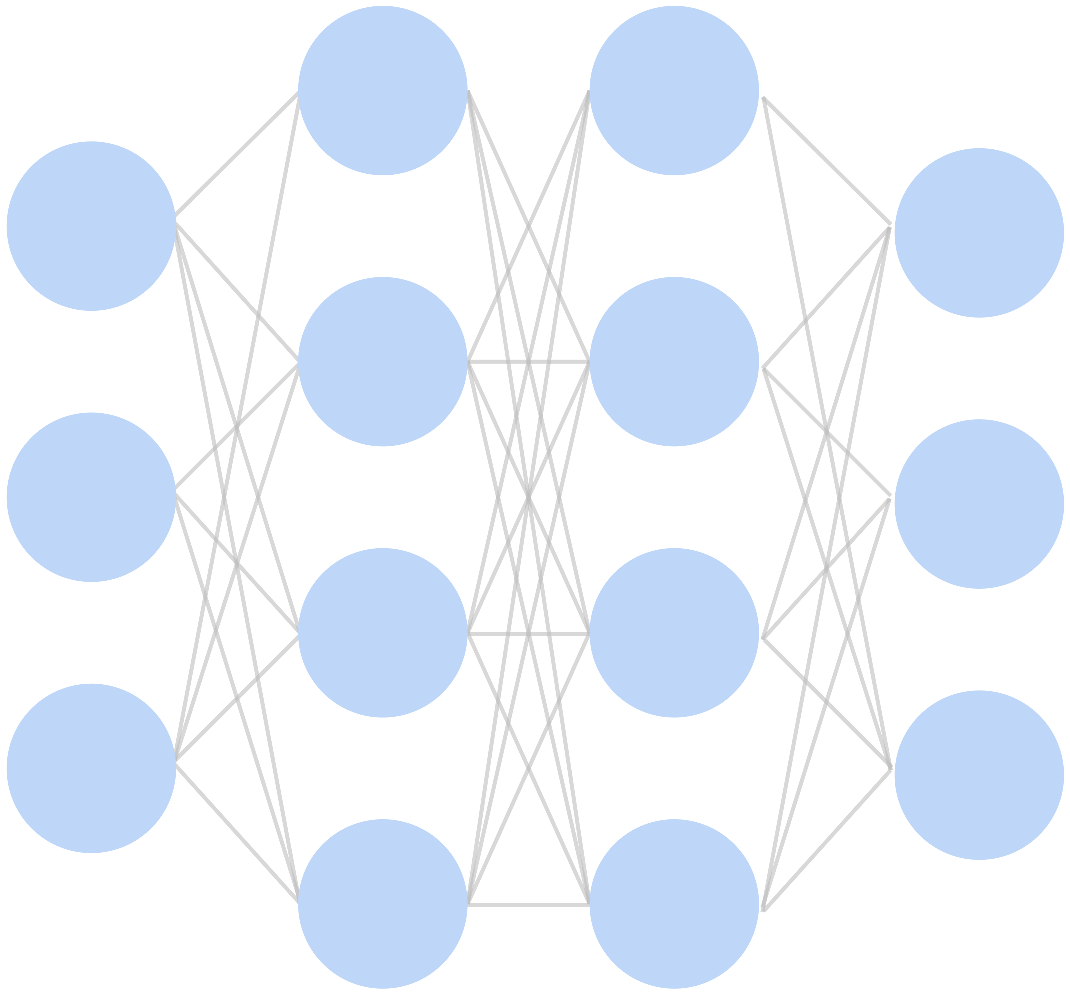
Empirical risk minimization:

$$\hat{\mathcal{R}}(\theta) = \mathbb{E}_{(\mathbf{x},y)\sim\hat{P}}[\ell(\theta;(\mathbf{x},y))].$$

Trained on in-distribution data
(e.g., CIFAR-10)

The Task of OOD Detection

Model $f(x;\theta)$

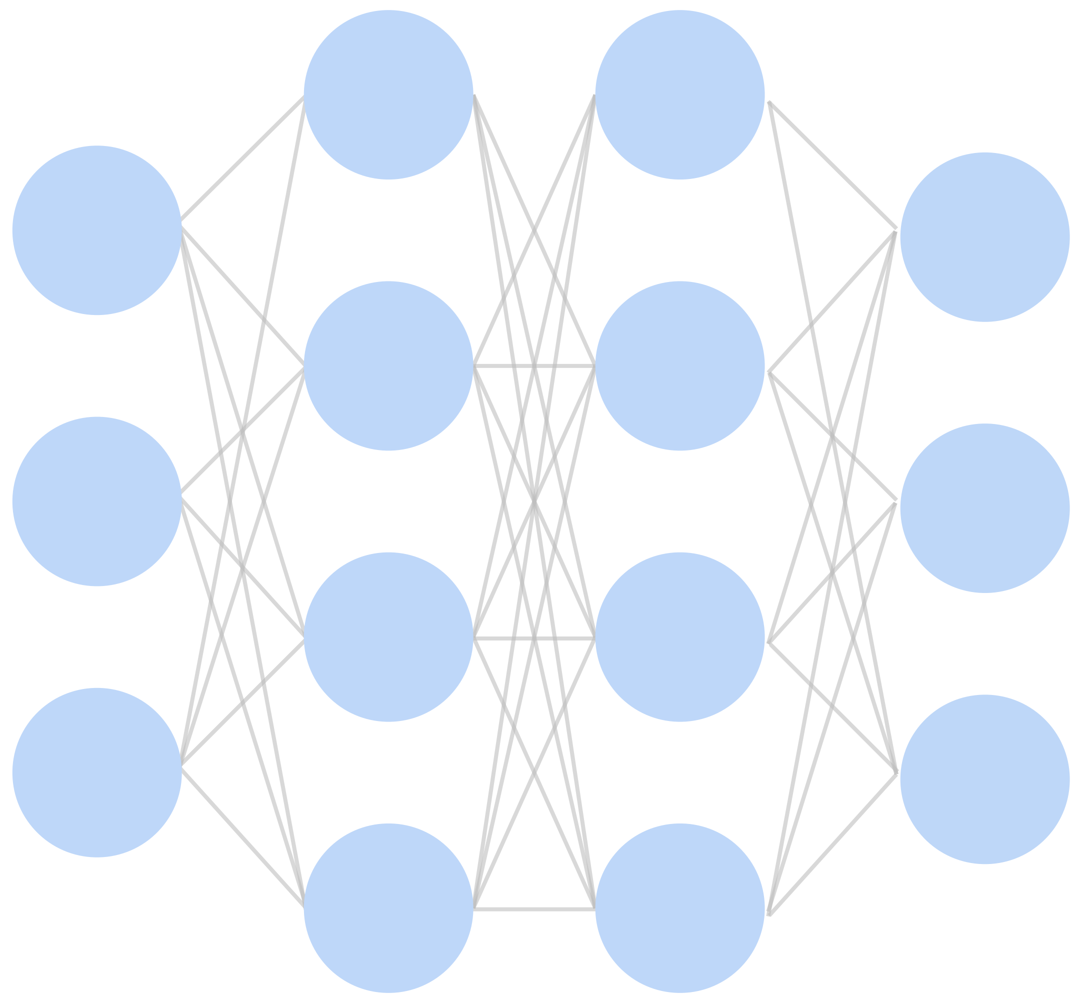


Trained on in-distribution data

The Task of OOD Detection

Model $f(x;\theta)$

Test input $x \longrightarrow$



$\longrightarrow$

Based on the model f

$$G_\lambda(\mathbf{x};\boxed{f}) = \begin{cases} \text{in} & S(\mathbf{x};f) \geq \lambda \\ \text{out} & S(\mathbf{x};f) < \lambda \end{cases}$$

S: Scoring function

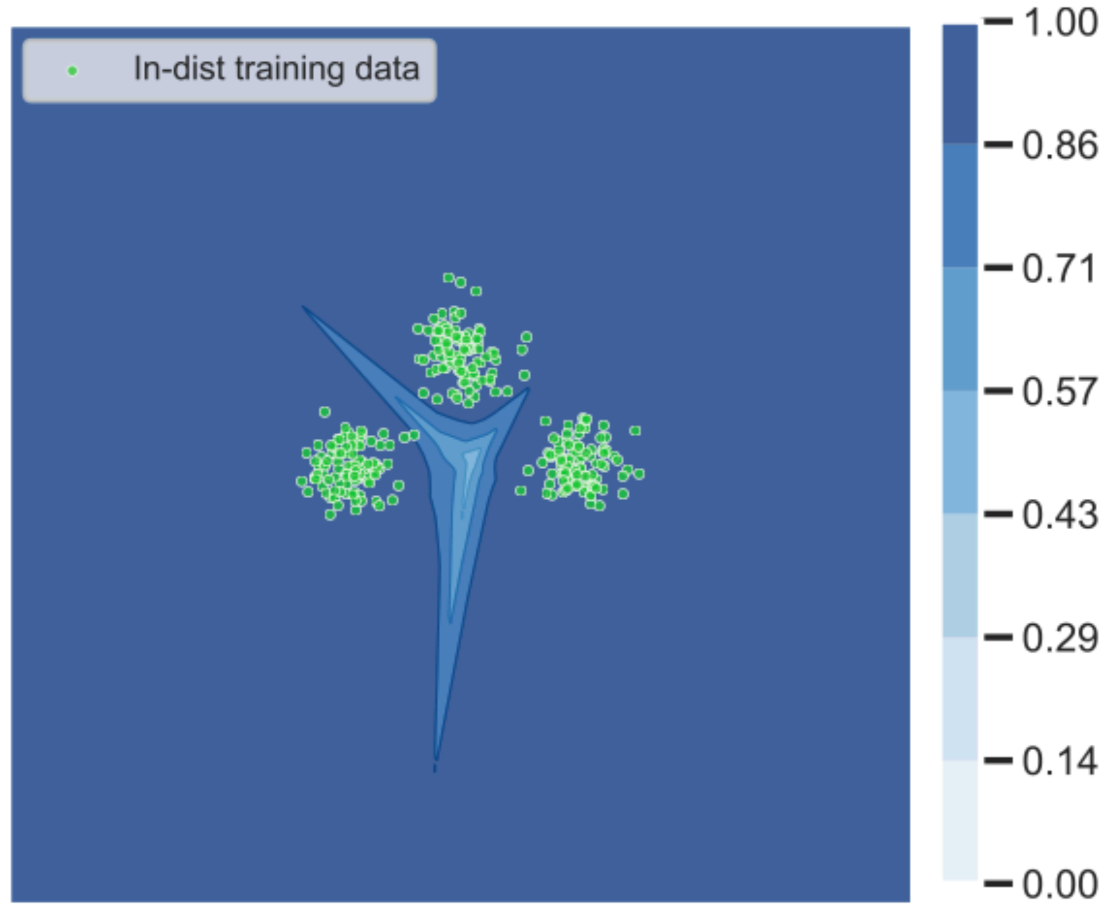
Trained on in-distribution data

OOD Detection with Outlier Exposure

- Motivation: modern neural networks tend to be over-confident for OOD inputs
  (Due to limited supervision with only ID data during training)

OOD Detection with Outlier Exposure

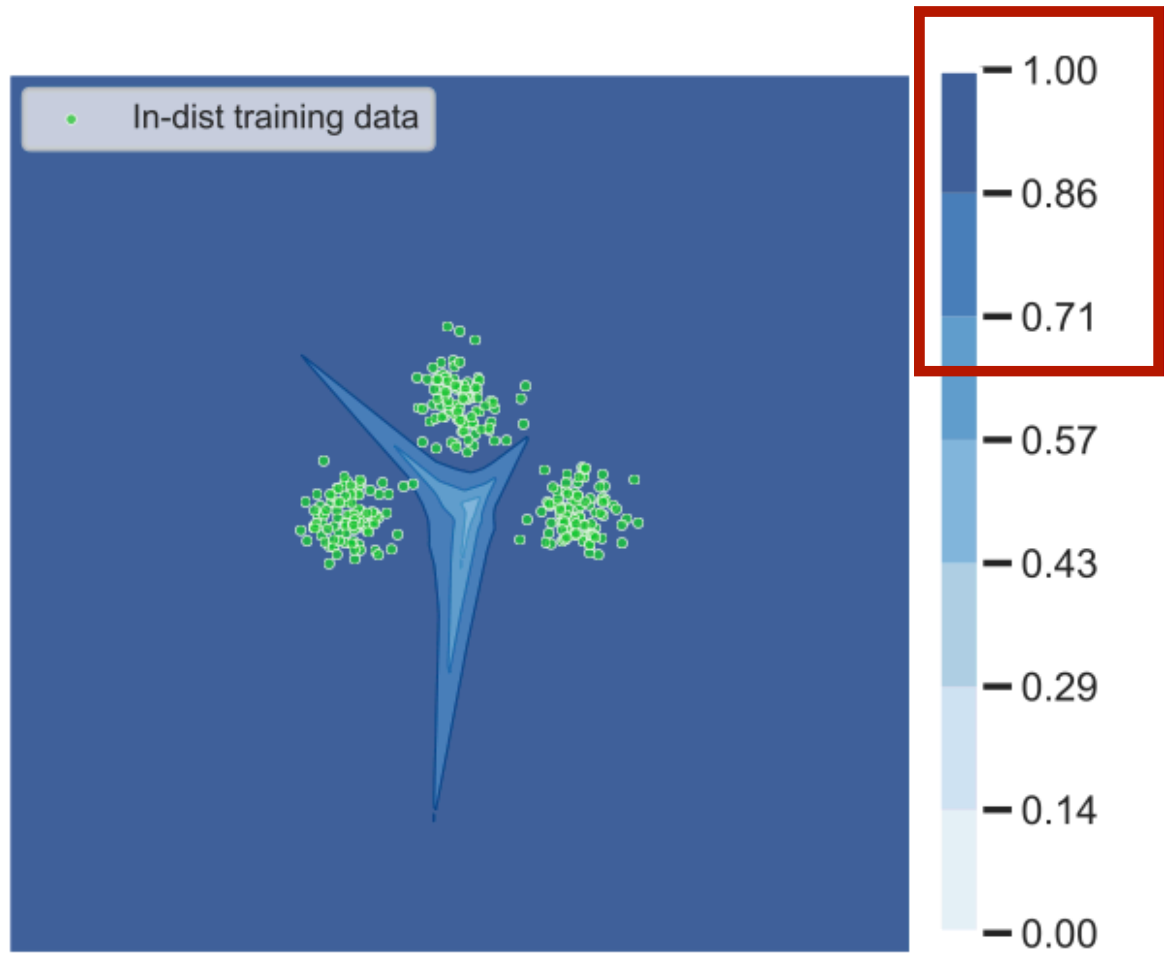
- Motivation: modern neural networks tend to be over-confident for OOD inputs
  (Due to limited supervision with only ID data during training)



Maximum Softmax probability over ID categories
when the network is trained with only ID data (green)

OOD Detection with Outlier Exposure

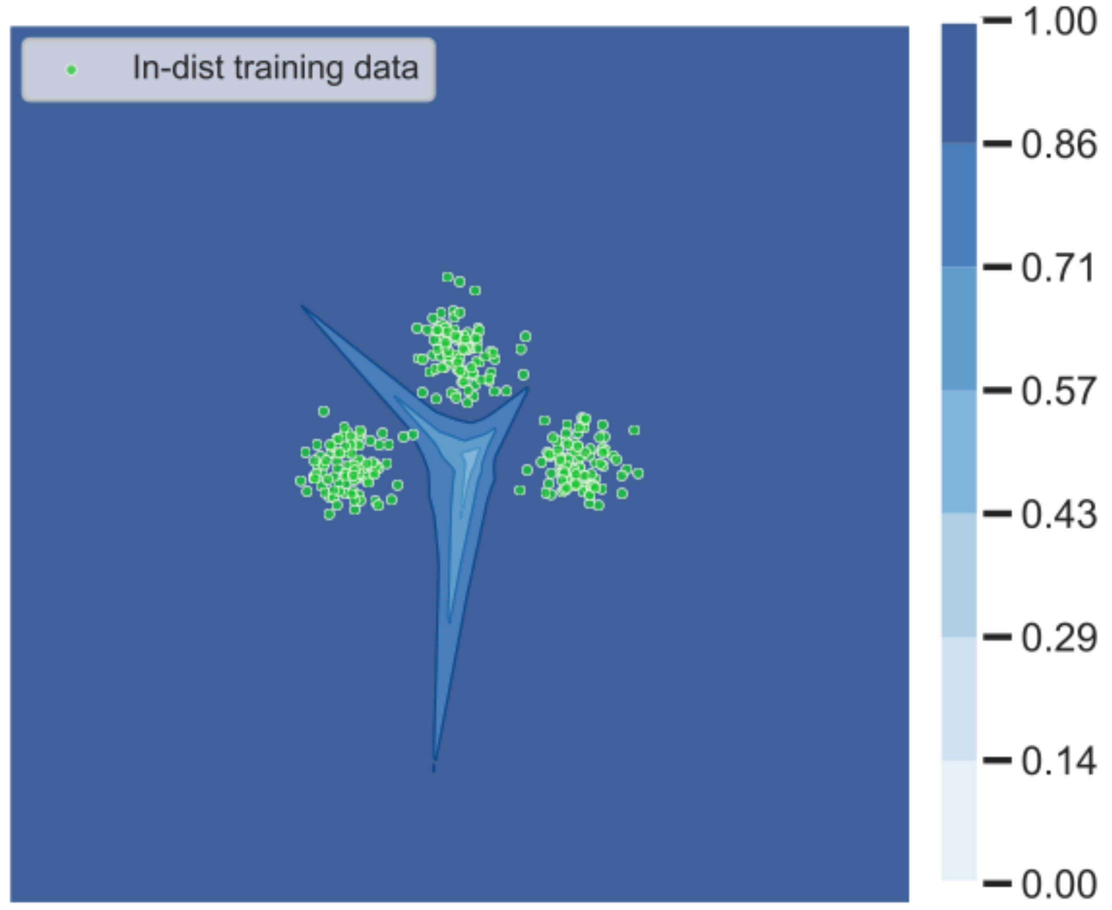
- Motivation: modern neural networks tend to be over-confident for OOD inputs
  (Due to limited supervision with only ID data during training)



Maximum Softmax probability over ID categories
when the network is trained with only ID data (green)

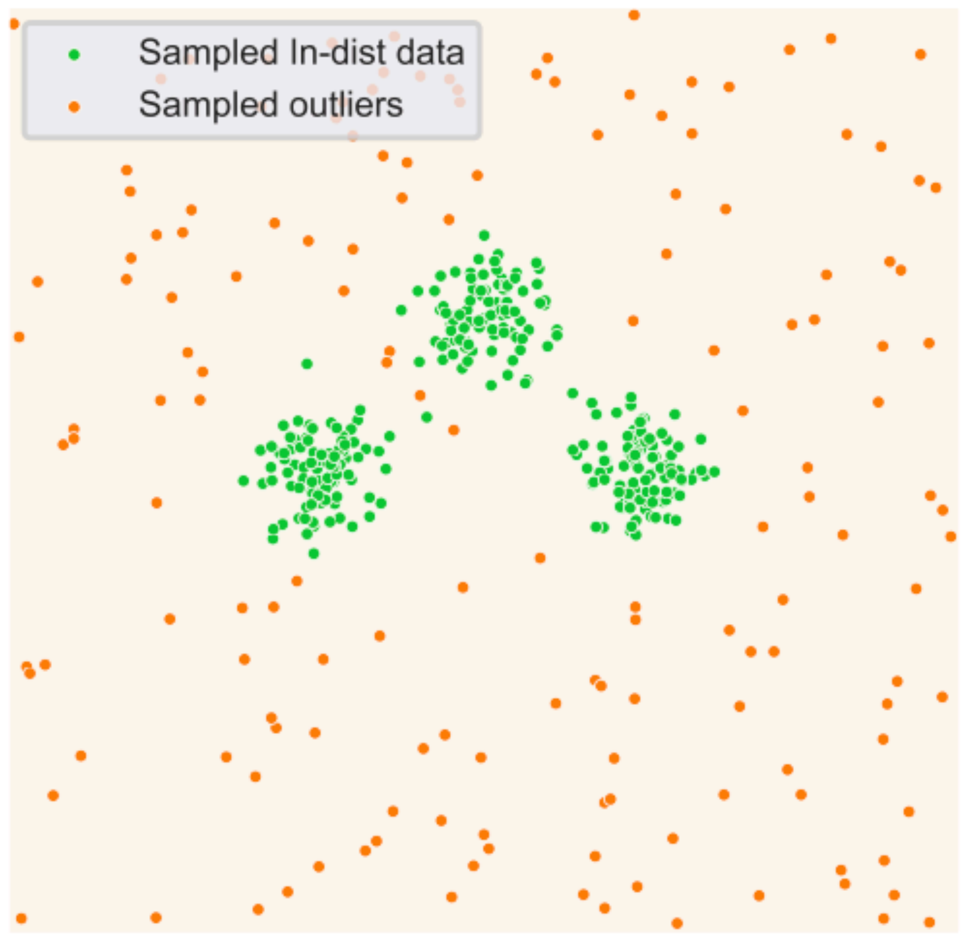
OOD Detection with Outlier Exposure

- Motivation: modern neural networks tend to be over-confident for OOD inputs
  (Due to limited supervision with only ID data during training)



Maximum Softmax probability over ID categories
when the network is trained with only ID data (green)

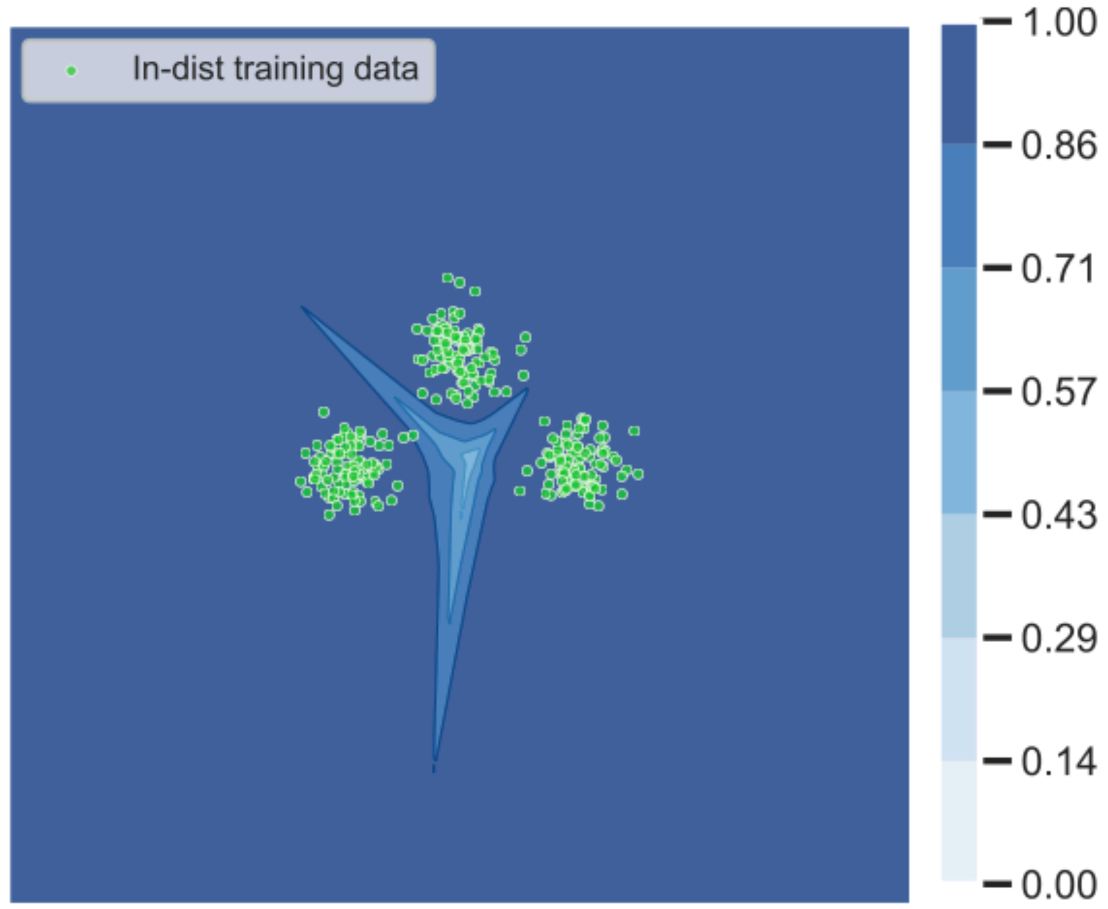
If we have a large auxiliary outlier dataset



Sample space: green (ID) vs. orange (OOD)

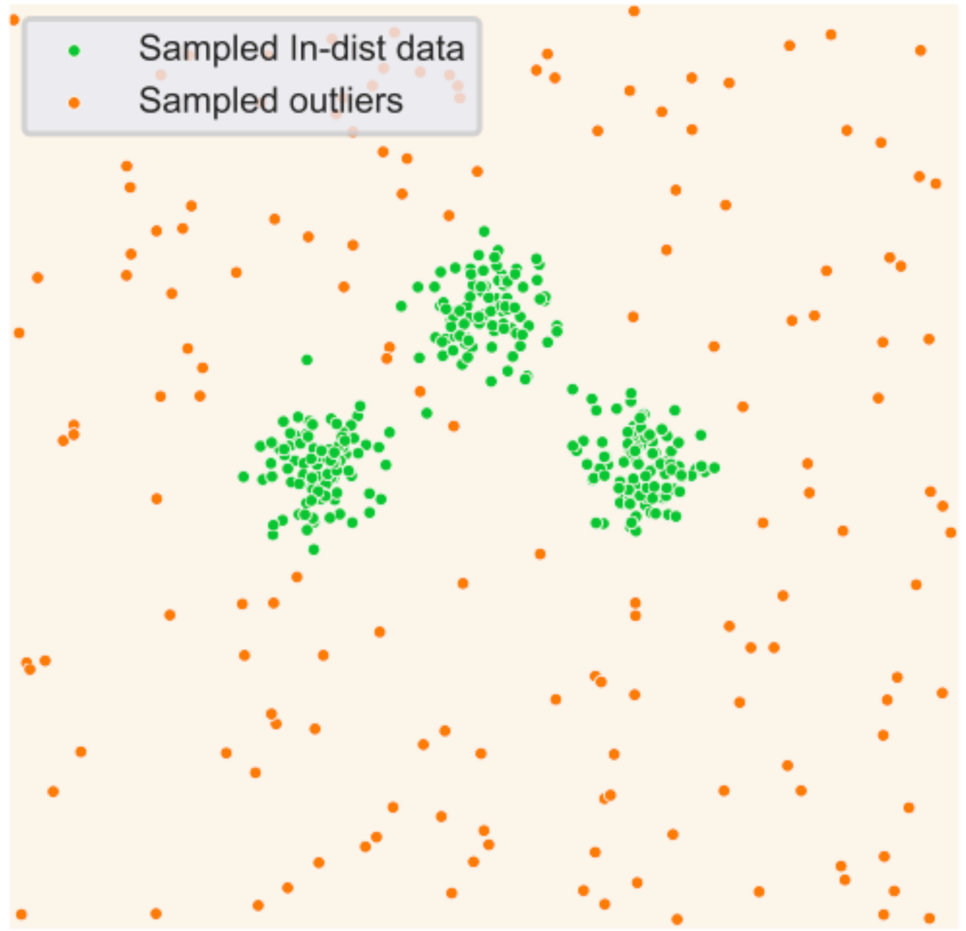
OOD Detection with Outlier Exposure

- Motivation: modern neural networks tend to be over-confident for OOD inputs
  (Due to limited supervision with only ID data during training)



Maximum Softmax probability over ID categories
when the network is trained with only ID data (green)

If we have a large auxiliary outlier dataset

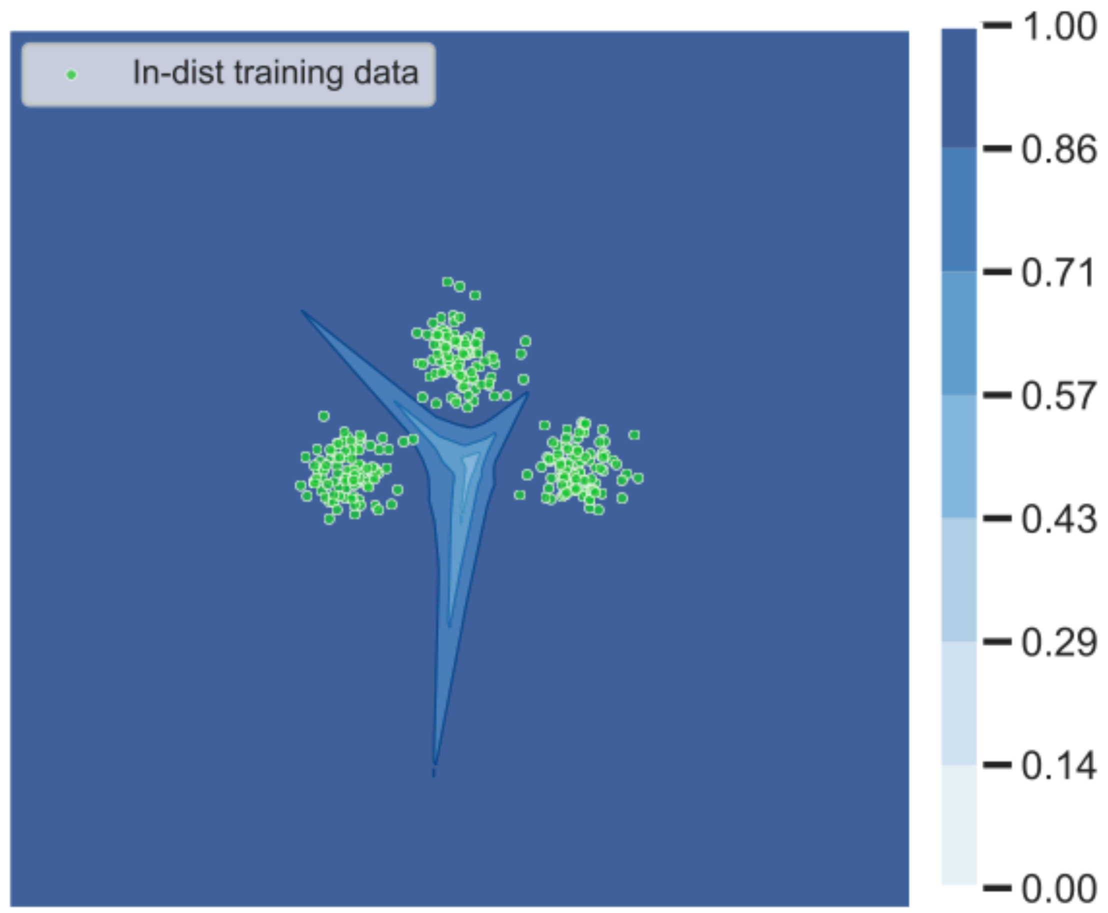


Sample space: green (ID) vs. orange (OOD)

- Challenge: the space of potential OOD data can be extremely large for high-dim feature space

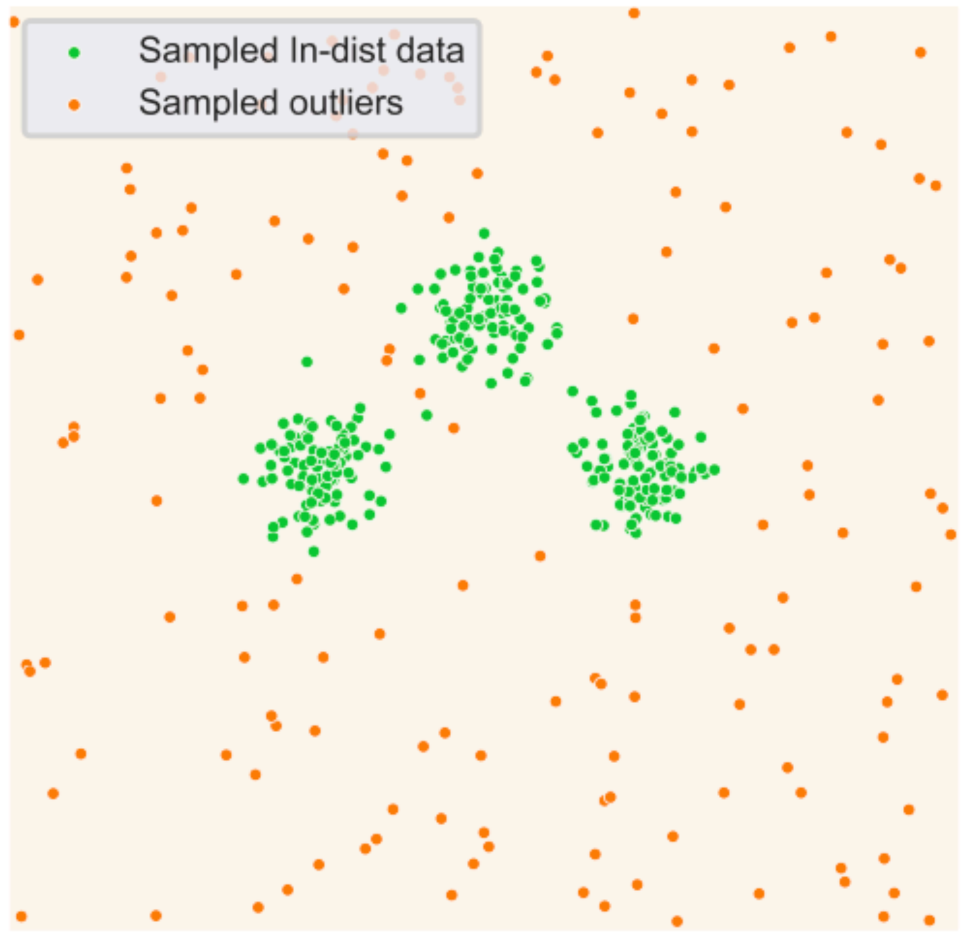
OOD Detection with Outlier Exposure

- Motivation: modern neural networks tend to be over-confident for OOD inputs
  (Due to limited supervision with only ID data during training)



Maximum Softmax probability over ID categories
when the network is trained with only ID data (green)

If we have a large auxiliary outlier dataset

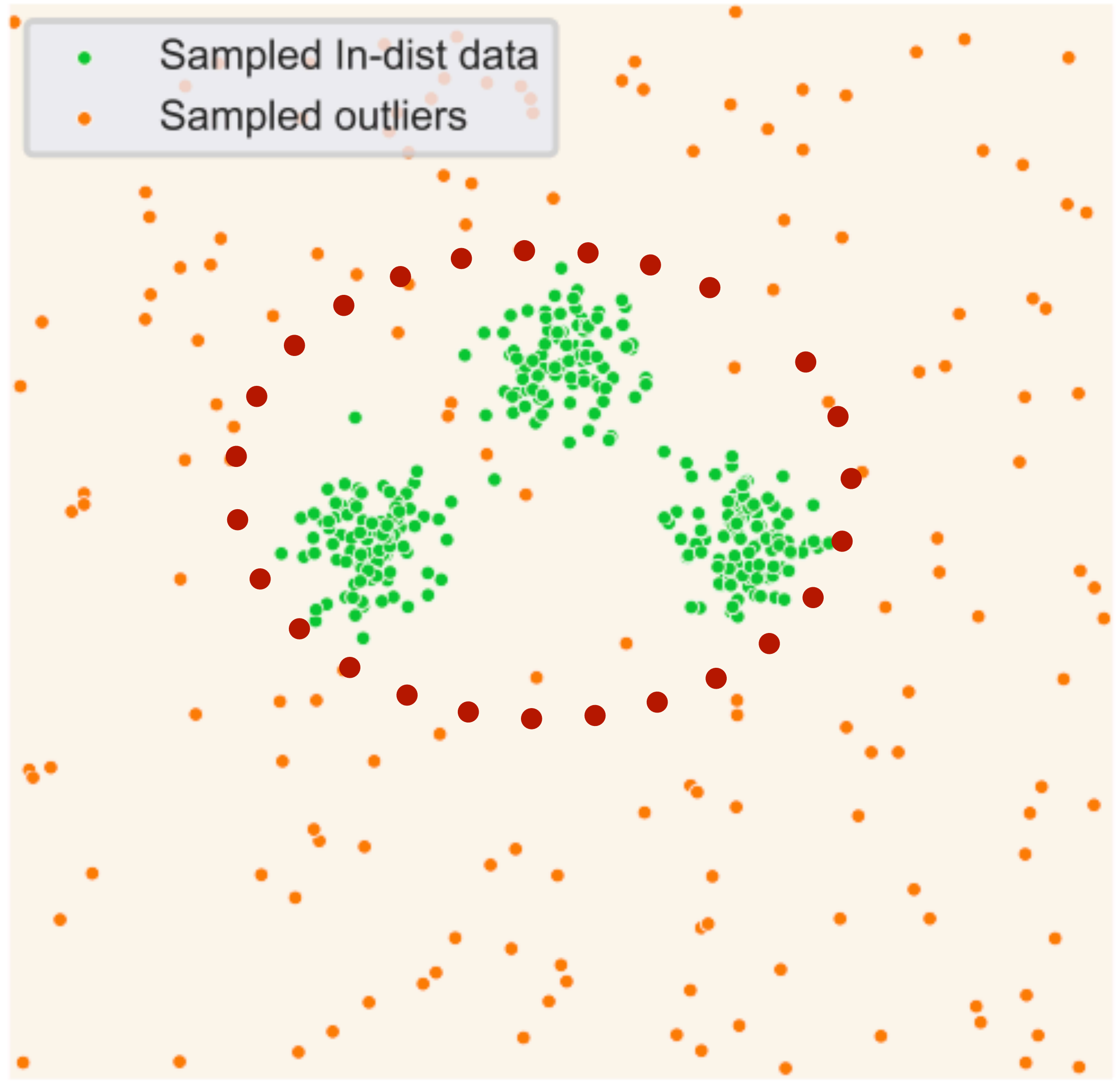


Sample space: green (ID) vs. orange (OOD)

- Challenge: the space of potential OOD data is extremely large for high-dim feature space
- Requirement: data-efficient solution to learn a compact ID-OOD decision boundary

Illustration of Outlier Mining

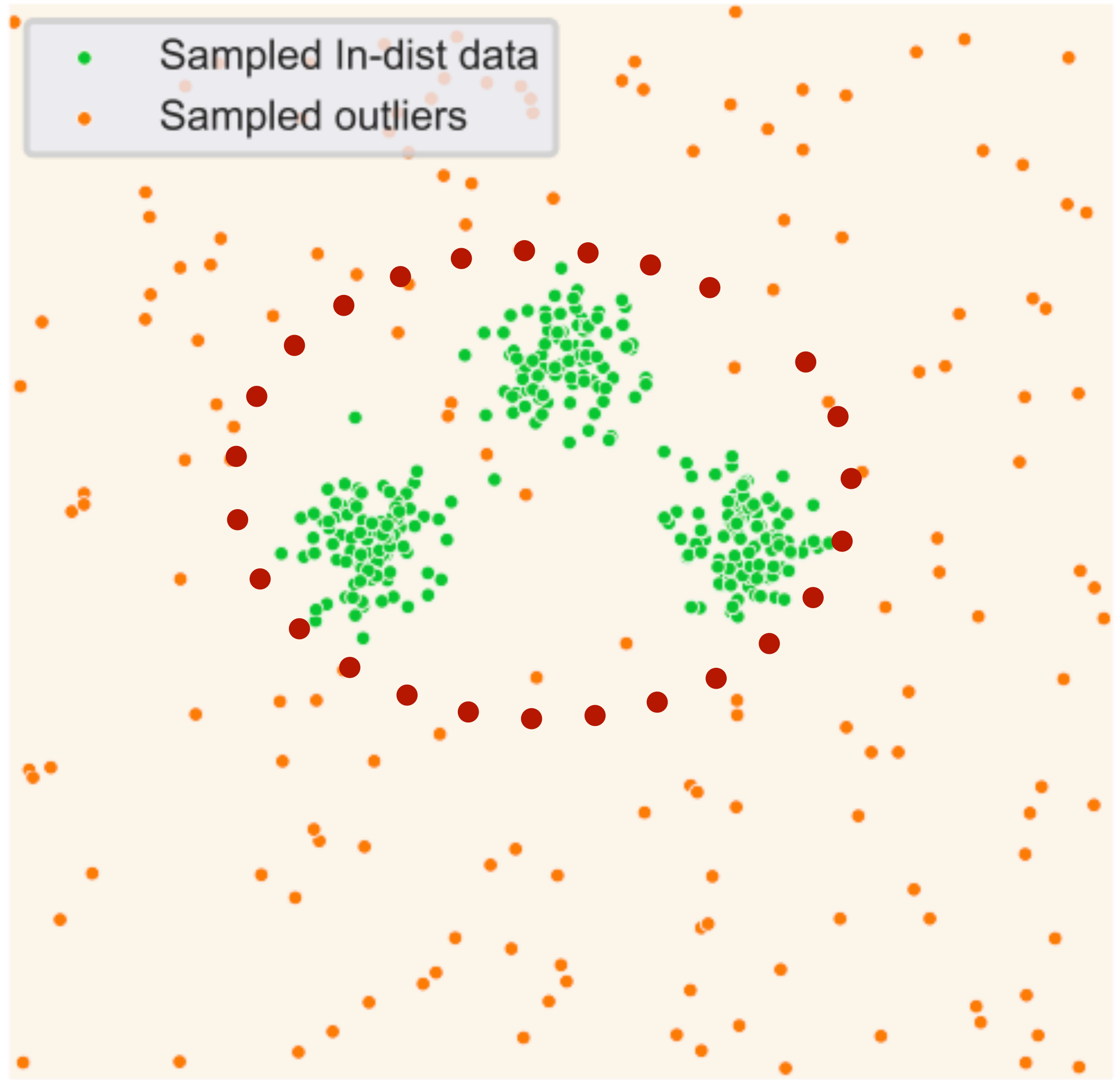
- Outlier Mining: to identify the most informative outlier samples close to the ID-OOD boundary



Sample space:
green (ID) vs. orange (OOD)

Illustration of Outlier Mining

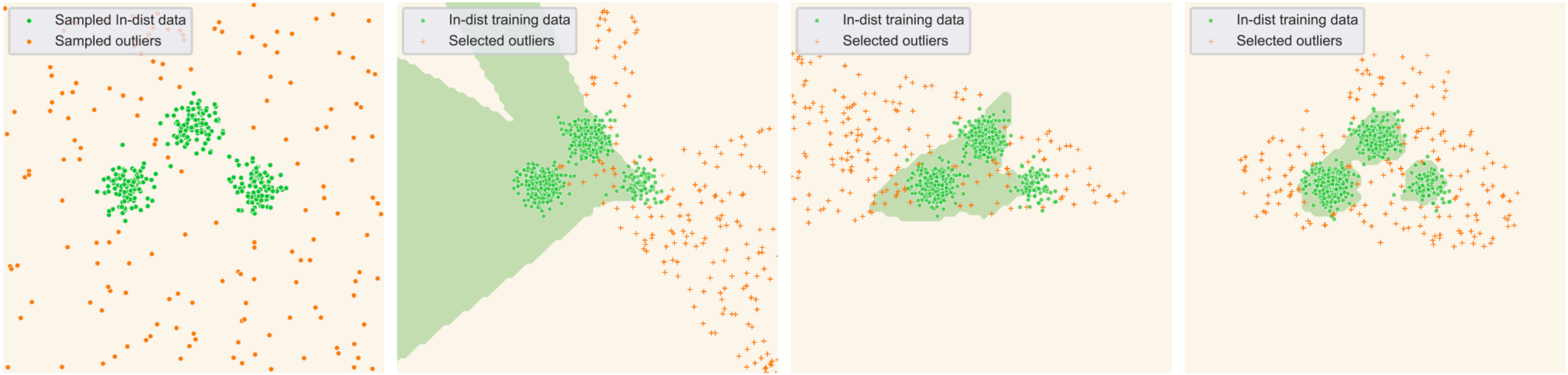
- Outlier Mining: to identify the most informative outlier samples close to the ID-OOD boundary



Sample space:
green (ID) vs. orange (OOD)

Illustration of Outlier Mining

- Outlier Mining: to identify the most informative outlier samples close to the ID-OOD boundary



Sample space: green (ID) vs. orange (OOD) | Epoch 1 | Epoch 4 | Epoch 30

Outlier Mining: A Thompson Sampling View (informally)

- Our main novelty: framing outlier mining as a ***sequential decision making problem***:
  - Objective: to identify most informative outliers (i.e., close to the *unknown* ID-OOD boundary)
  - At each timestep,
    - Action: outlier selection
    - Reward: based on the closeness to the unknown ID-OOD boundary

- To summarize, finding outliers close to the boundary given an auxiliary set
  → can be formulated as **optimizing an unknown function by selecting samples**

- Exploration vs. exploitation trade-off is crucial for efficient optimization!
  → **Thompson Sampling** (sampling from posterior distribution to take action)

Outlier Mining: A Thompson Sampling View (formally)

- TS for outlier mining: maintaining and modeling the distribution of $\mathbf{w}^*$, and using this model to select near-boundary outliers over time via posterior sampling

- At each step t, the model parameter $\mathbf{w^t}$ is sampled from the posterior distribution of $\mathbf{w}^*$, then the learner takes an action a_t by choosing outlier $\mathbf{x} \sim \mathscr{P}_{\text{aux}}$ that **maximize the estimated boundary score (to be defined next) according to $\mathbf{w_t}$**

Algorithm 1 Outlier Mining via Thompson Sampling

Input: A prior distribution $P_0^{\mathbf{w}}$ over $\mathbf{w}$.

for step $t = 0, 1, \cdots, T$ **do**

  Sample $\mathbf{w}_t \sim P_t^{\mathbf{w}}$.

  Take action a_t by choosing outliers $\mathbf{x} \sim \mathcal{P}_{\text{aux}}$ based on the sampled model $\mathbf{w}_t$.

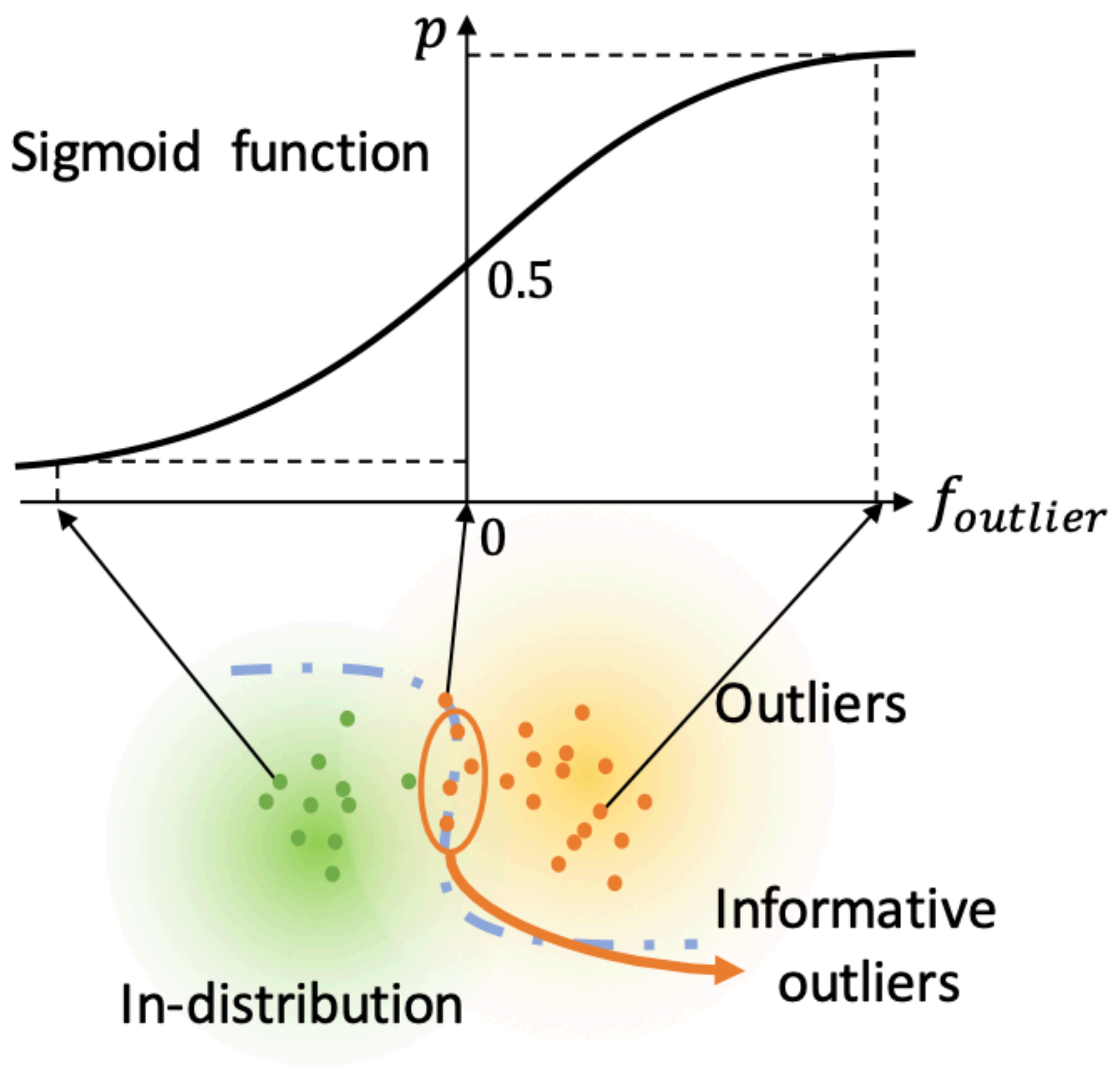
  Receive some reward $G(\mathbf{x})$.

  Update the posterior distribution $P_{t+1}^{\mathbf{w}}$ for model.

end for

Outlier Mining: Boundary Score

- Q: How to measure the distance to the boundary for outlier samples?
  - Modeling ***Boundary Score***: $G(\mathbf{x}) = -\,|f_{\text{outlier}}(\mathbf{x}; \mathbf{w}^*)|$
  - f_{outlier} is a function parameterized by $\mathbf{w}^*$ that maps input $\mathbf{x}$ to the ***logit*** space: $p(\text{outlier}\,|\,\mathbf{x}) = \text{Sigmoid}(f_{\text{outlier}}(\mathbf{x}; \mathbf{w}^*))$
  - Near-boundary outliers correspond to $|f_{\text{outlier}}(\mathbf{x}; \mathbf{w}^*)| \approx 0$



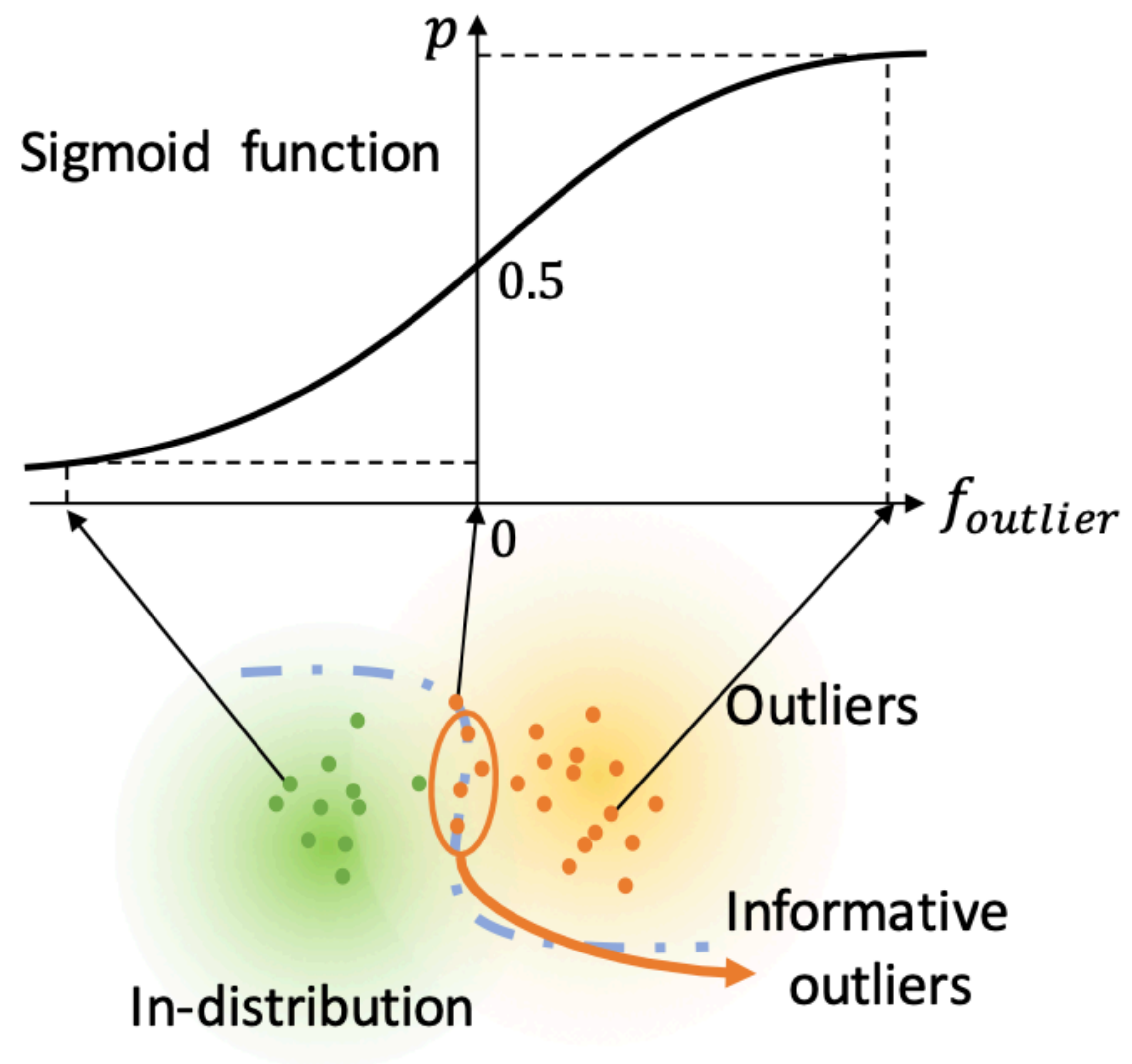
(b) Boundary Score & Density

Outlier Mining: Insights for Boundary Score

- **Intuitively**, outliers with the highest boundary scores are more desirable for model regularization to learn a compact ID-OOD boundary

- **Theoretically**, we show that outliers with high boundary scores *benefit sample complexity* for OOD detection:

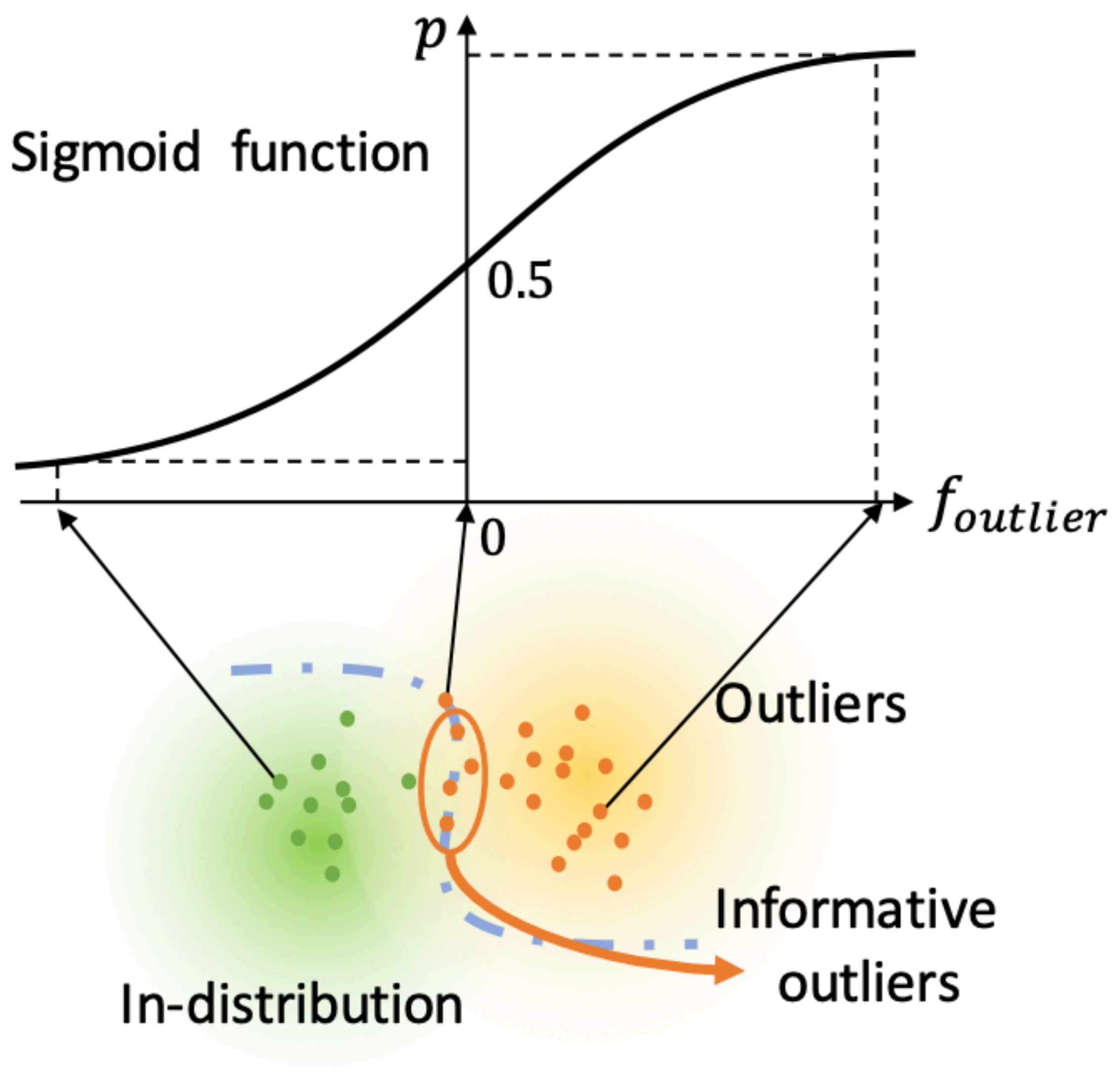
- (Informal version of Thm 6.1) We show that FPR is a **decreasing** function of the average boundary score of the selected outlier under Gaussian mixture assumptions



(b) Boundary Score & Density

Outlier Mining: Estimating Boundary Score

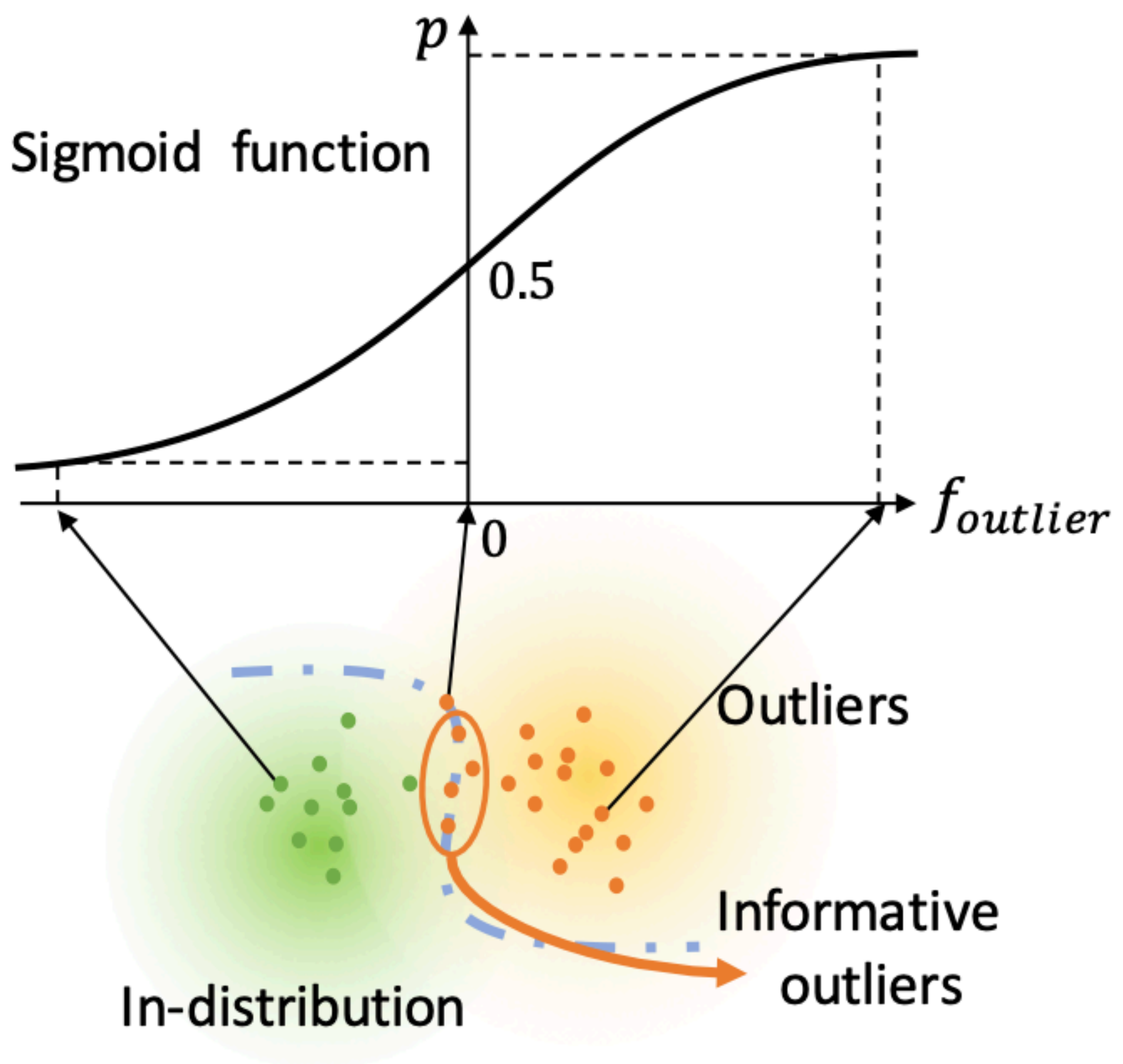
- Recall $G(\mathbf{x}) = -\,|f_{\text{outlier}}(\mathbf{x};\mathbf{w}^*)|$, $\mathbf{w}^*$ is unknown
- Given any $\mathbf{x}$ labeled as OOD/ID, we can infer a target logit y_{tar} as an approximate target value of $f_{\text{outlier}}(\mathbf{x};\mathbf{w}^*)$



(b) Boundary Score & Density

Outlier Mining: Estimating Boundary Score

- Recall $G(\mathbf{x}) = -\,|f_{\text{outlier}}(\mathbf{x}; \mathbf{w}^*)|$, $\mathbf{w}^*$ is unknown
- Given any $\mathbf{x}$ labeled as OOD/ID, we can infer a target logit y_{tar} as an approximate target value of $f_{\text{outlier}}(\mathbf{x}; \mathbf{w}^*)$
- Q: how to find the most informative outliers?
- Use the approximate target value to build a **regression model** with uncertainty measurement
  → Choose outliers close to the sampled decision boundary via TS



(b) Boundary Score & Density

Outlier Mining: Modeling f_{outlier} with Neural Networks

- We perform **Bayesian linear regression (BLR)** on top of the penultimate layer as feature $\phi(\mathbf{x})$ of a deep neural network to model the boundary score:

- At each timestep, estimate $\hat{f}_{\text{outlier}}(\mathbf{x}; \mathbf{w_t}) = \mathbf{w_t}^\top \phi(\mathbf{x})$

Outlier Mining: Modeling f_{outlier} with Neural Networks

- We perform **Bayesian linear regression (BLR)** on top of the penultimate layer as feature $\phi(\mathbf{x})$ of a deep neural network to model the boundary score:

- At each timestep, estimate $\hat{f}_{\text{outlier}}(\mathbf{x}; \mathbf{w_t}) = \mathbf{w_t}^\top \phi(\mathbf{x})$

- To get $\mathbf{w}_t$, **maintain and update the posterior distribution of $\mathbf{w}^*$**:
  - Build a Gaussian prior of $\mathbf{w}_0 \sim \mathcal{N}(0,\Sigma)$
  - Sample $\mathbf{w_t} \sim \mathcal{N}\left(\sigma^{-2}\Sigma_{\mathrm{p}}^{-1}\Phi\mathbf{y}_{\mathrm{tar}}, \Sigma_{\mathrm{p}}^{-1}\right)$
    - $\Sigma_{\mathrm{p}} := \sigma^{-2}\Phi\Phi^\top + \Sigma^{-1}$ posterior covariance matrix
    - Φ: concatenation of feature representations $\{\phi(\mathbf{x}_i)\}$
    - $\mathbf{y}_{\mathrm{tar}}$: concatenation of target logit values
    - σ^2: variance of i.i.d. noises for target logit values

Outlier Mining: Modeling f_{outliter} with Neural Networks

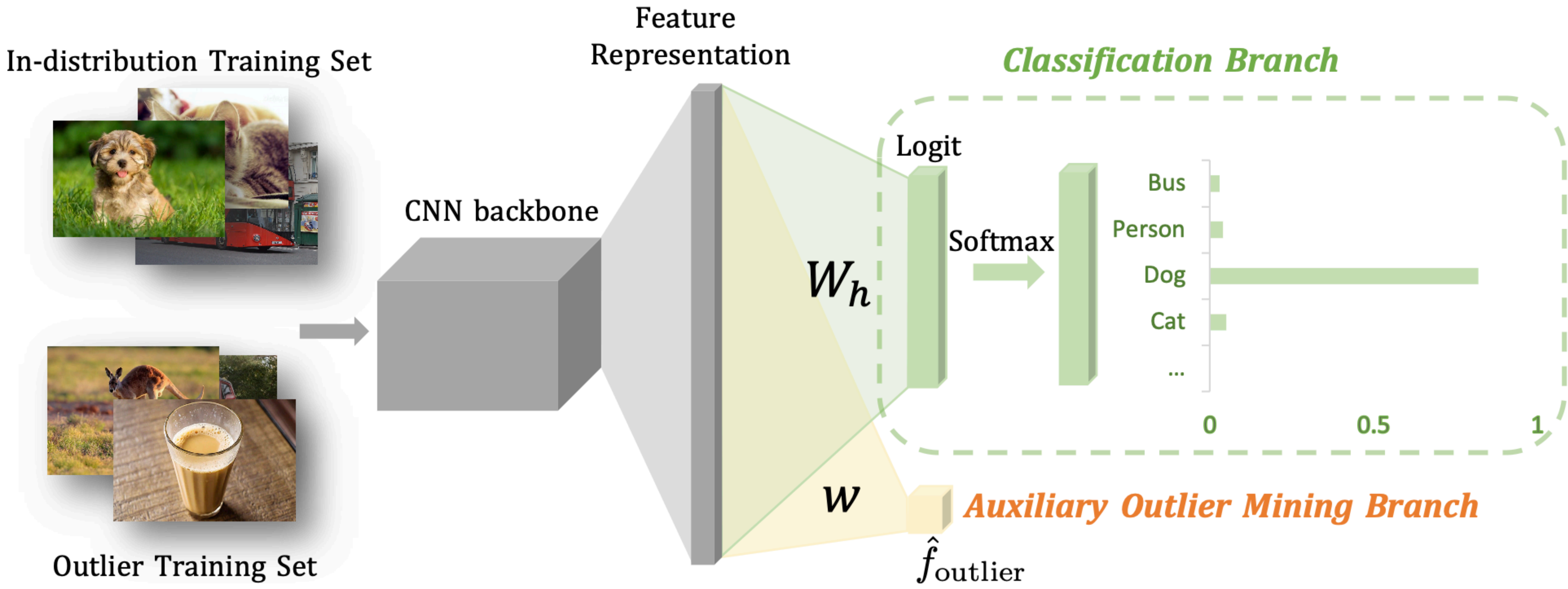
- We perform **Bayesian linear regression (BLR)** on top of the penultimate layer feature $\phi(\mathbf{x})$ of a deep neural network to model the boundary score:

- At each timestep, estimate $\hat{f}_{\text{outlier}}(\mathbf{x}; \mathbf{w_t}) = \mathbf{w_t}^\top \phi(\mathbf{x})$

- To get $\mathbf{w}_t$, **maintain and update the posterior distribution of $\mathbf{w}^*$**:
  - Build a Gaussian prior of $\mathbf{w}_0 \sim \mathcal{N}(0, \Sigma)$
  - Sample $\mathbf{w_t} \sim \mathcal{N}\left(\sigma^{-2}\Sigma_{\text{p}}^{-1}\Phi\mathbf{y}_{\text{tar}}, \Sigma_{\text{p}}^{-1}\right)$
    - $\Sigma_{\text{p}} := \sigma^{-2}\Phi\Phi^\top + \Sigma^{-1}$ posterior covariance matrix
    - Φ: concatenation of feature representations $\{\phi(\mathbf{x}_i)\}$
    - $\mathbf{y}_{\text{tar}}$: concatenation of target logit values
    - σ^2: variance of i.i.d. noises for target logit values

- TS with BLR is a good trade-off between *computational tractability and OOD detectability*

Putting Together: Framework Overview



POEM: **P**osterior Sampling-based **O**utlier **M**ining

Putting Together: Training and Inference

- Training loops:
  - Step 1: Constructing an auxiliary outlier training set by selecting outliers with the highest sampled boundary scores from a large candidate pool
  - Step 2: The classification branch, together with the network backbone are trained using a mixture of ID and selected outlier data with energy regularization (Liu et al. [1])
  - Step 3: Based on the updated feature representation, we perform the posterior update of the weights in the outlier mining branch

Putting Together: Training and Inference

- Training loops:
  - Step 1: Constructing an auxiliary outlier training set by selecting outliers with the highest sampled boundary scores from a large candidate pool
  - Step 2: The classification branch together with the network backbone are trained using a mixture of ID and selected outlier data with energy regularization (Liu et al. [1])
  - Step 3: Based on the updated feature representation, we perform the posterior update of the weights in the outlier mining branch
- Inference:
  - At test time, OOD detection is based on the energy of the input: $D_\lambda(\mathbf{x}) = \mathbf{1}\{-E(\mathbf{x}) \geq \gamma\}$

- Remark: threshold γ is typically chosen so that a high fraction of ID data (e.g., 95%) is correctly classified

Experimental Setup

Datasets

- ID datasets:
  - CIDER-10 and CIFAR-100
- Auxiliary outlier dataset:
  - ImageNet-RC (Chrabaszcz et al.) [2], a downsampled version of ImageNet1K
- OOD test sets:
  - SVHN (Netzer et al.) [3], Textures (Cimpoi et al.) [4], Places365 (Zhou et al.) [5]. LSUN-crop, LSUN-resize (Yu et al.) [6], iSUN (Xu et al.) [7]

Evaluation Metrics

- FPR95: the false positive rate (of OOD samples) when the true positive rate of ID samples is at 95%
- AUROC: the area under the receiver operating characteristic curve
- AUPR: the area under the precision-recall curve
- ID-ACC: ID classification accuracy.

Main Results: Overview

$\mathcal{D}_{in}$	**Method**	**FPR95↓**	**AUROC↑**	**AUPR↑**	ID-ACC	w./w.o. $\mathcal{D}_{aux}$	Sampling Method
CIFAR-10	MSP (Hendrycks & Gimpel, 2017)	58.98	90.63	93.18	**94.39**	✗	NA
	ODIN (Liang et al., 2018)	26.55	94.25	95.34	94.39	✗	NA
	Mahalanobis (Lee et al., 2018b)	29.47	89.96	89.70	94.39	✗	NA
	Energy (Liu et al., 2020)	28.53	94.39	95.56	94.39	✗	NA
	SSD+ (Sehwag et al., 2021)	7.22	98.48	98.59	NA	✗	NA
	OE (Hendrycks et al., 2018)	9.66	98.34	98.55	94.12	✓	random
	SOFL (Mohseni et al., 2020)	5.41	98.98	99.10	93.68	✓	random
	CCU (Meinke & Hein, 2020)	8.78	98.41	98.69	93.97	✓	random
	NTOM (Chen et al., 2021)	4.38	99.08	99.24	94.11	✓	greedy
	Energy (w. $\mathcal{D}_{aux}$) (Liu et al., 2020)	4.62	98.93	99.12	92.92	✓	random
	POEM (ours)	**2.54**$^{\pm 0.56}$	**99.40**$^{\pm 0.05}$	**99.50**$^{\pm 0.07}$	93.49$^{\pm 0.27}$	✓	Thompson

Observations:

- POEM achieves SoTA OOD detection performance and maintains comparable ID classification accuracy

POEM Outperforms Other OE-based Methods

$\mathcal{D}_{\text{in}}$	**Method**	**FPR95**↓	**AUROC**↑	**AUPR**↑	ID-ACC	w./w.o. $\mathcal{D}_{\text{aux}}$	Sampling Method
CIFAR-10	MSP (Hendrycks & Gimpel, 2017)	58.98	90.63	93.18	**94.39**	✗	NA
	ODIN (Liang et al., 2018)	26.55	94.25	95.34	94.39	✗	NA
	Mahalanobis (Lee et al., 2018b)	29.47	89.96	89.70	94.39	✗	NA
	Energy (Liu et al., 2020)	28.53	94.39	95.56	94.39	✗	NA
	SSD+ (Sehwag et al., 2021)	7.22	98.48	98.59	NA	✗	NA
	OE (Hendrycks et al., 2018)	9.66	98.34	98.55	94.12	✓	random
	SOFL (Mohseni et al., 2020)	5.41	98.98	99.10	93.68	✓	random
	CCU (Meinke & Hein, 2020)	8.78	98.41	98.69	93.97	✓	random
	NTOM (Chen et al., 2021)	4.38	99.08	99.24	94.11	✓	greedy
	Energy (w. $\mathcal{D}_{\text{aux}}$) (Liu et al., 2020)	4.62	98.93	99.12	92.92	✓	random
	POEM (ours)	**2.54**$^{\pm 0.56}$	**99.40**$^{\pm 0.05}$	**99.50**$^{\pm 0.07}$	93.49$^{\pm 0.27}$	✓	Thompson

Observations:

- POEM achieves SoTA OOD detection performance and maintains comparable ID classification accuracy
- POEM utilizes outliers more effectively than other Outlier Exposure-based (w. D_{aux}) methods

Thompson Sampling vs. Greedy Sampling

$\mathcal{D}_{\text{in}}$	**Method**	**FPR95**↓	**AUROC**↑	**AUPR**↑	ID-ACC	w./w.o. $\mathcal{D}_{\text{aux}}$	Sampling Method
CIFAR-10	MSP (Hendrycks & Gimpel, 2017)	58.98	90.63	93.18	**94.39**	✗	NA
	ODIN (Liang et al., 2018)	26.55	94.25	95.34	94.39	✗	NA
	Mahalanobis (Lee et al., 2018b)	29.47	89.96	89.70	94.39	✗	NA
	Energy (Liu et al., 2020)	28.53	94.39	95.56	94.39	✗	NA
	SSD+ (Sehwag et al., 2021)	7.22	98.48	98.59	NA	✗	NA
	OE (Hendrycks et al., 2018)	9.66	98.34	98.55	94.12	✓	random
	SOFL (Mohseni et al., 2020)	5.41	98.98	99.10	93.68	✓	random
	CCU (Meinke & Hein, 2020)	8.78	98.41	98.69	93.97	✓	random
	NTOM (Chen et al., 2021)	4.38	99.08	99.24	94.11	✓	greedy
	Energy (w. $\mathcal{D}_{\text{aux}}$) (Liu et al., 2020)	4.62	98.93	99.12	92.92	✓	random
	POEM (ours)	**2.54**$^{\pm 0.56}$	**99.40**$^{\pm 0.05}$	**99.50**$^{\pm 0.07}$	93.49$^{\pm 0.27}$	✓	Thompson

Observations:

- POEM achieves SoTA OOD detection performance and maintains comparable ID classification accuracy
- POEM utilizes outliers more effectively than other Outlier Exposure-based (w. D_{aux}) methods
- Thompson Sampling (POEM) is ***better than greedy sampling*** (NTOM chen et al. [8])

Similar Trends Also Hold for CIFAR-100

$\mathcal{D}_{in}$	Method	FPR95↓	AUROC↑	AUPR↑	ID-ACC	w./w.o. $\mathcal{D}_{aux}$	Sampling Method
CIFAR-100	MSP (Hendrycks & Gimpel, 2017)	80.30	73.13	76.97	**74.05**	✗	NA
	ODIN (Liang et al., 2018)	56.31	84.89	85.88	74.05	✗	NA
	Mahalanobis (Lee et al., 2018b)	47.89	85.71	87.15	74.05	✗	NA
	Energy (Liu et al., 2020)	65.87	81.50	84.07	74.05	✗	NA
	SSD+ (Sehwag et al., 2021)	38.32	88.91	89.77	NA	✗	NA
	OE (Hendrycks et al., 2018)	19.54	94.93	95.26	74.25	✓	random
	SOFL (Mohseni et al., 2020)	19.32	96.32	96.99	73.93	✓	random
	CCU (Meinke & Hein, 2020)	19.27	95.02	95.41	74.49	✓	random
	NTOM (Chen et al., 2021)	19.96	96.29	97.06	73.86	✓	greedy
	Energy (w. $\mathcal{D}_{aux}$) (Liu et al., 2020)	19.25	96.68	97.44	72.39	✓	random
	POEM (ours)	**15.14**$^{\pm 1.16}$	**97.79**$^{\pm 0.17}$	**98.31**$^{\pm 0.12}$	73.41$^{\pm 0.21}$	✓	Thompson

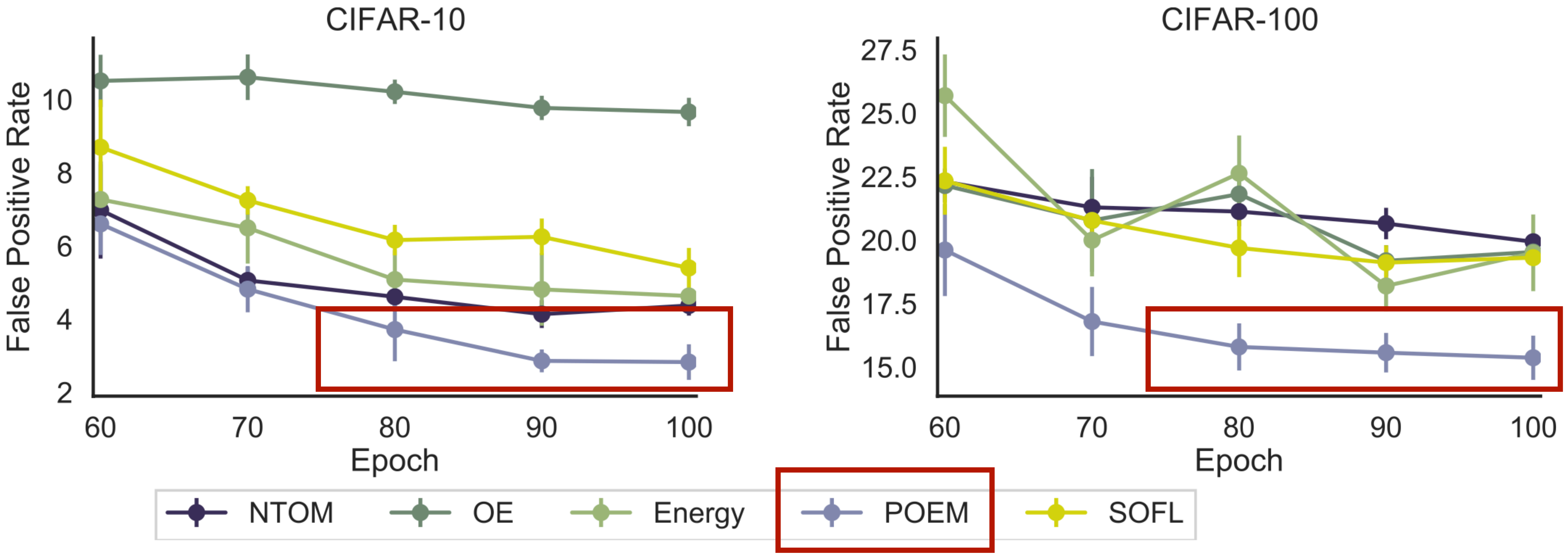
Observations:

- POEM achieves SoTA OOD detection performance and maintains comparable ID classification accuracy
- POEM utilizes outliers more effectively than other Outlier Exposure-based (w. D_{aux}) methods
- Thompson Sampling (POEM) is better than greedy sampling (NTOM chen et al. [8])

A Closer Look at Benefits of Thompson Sampling

Observations

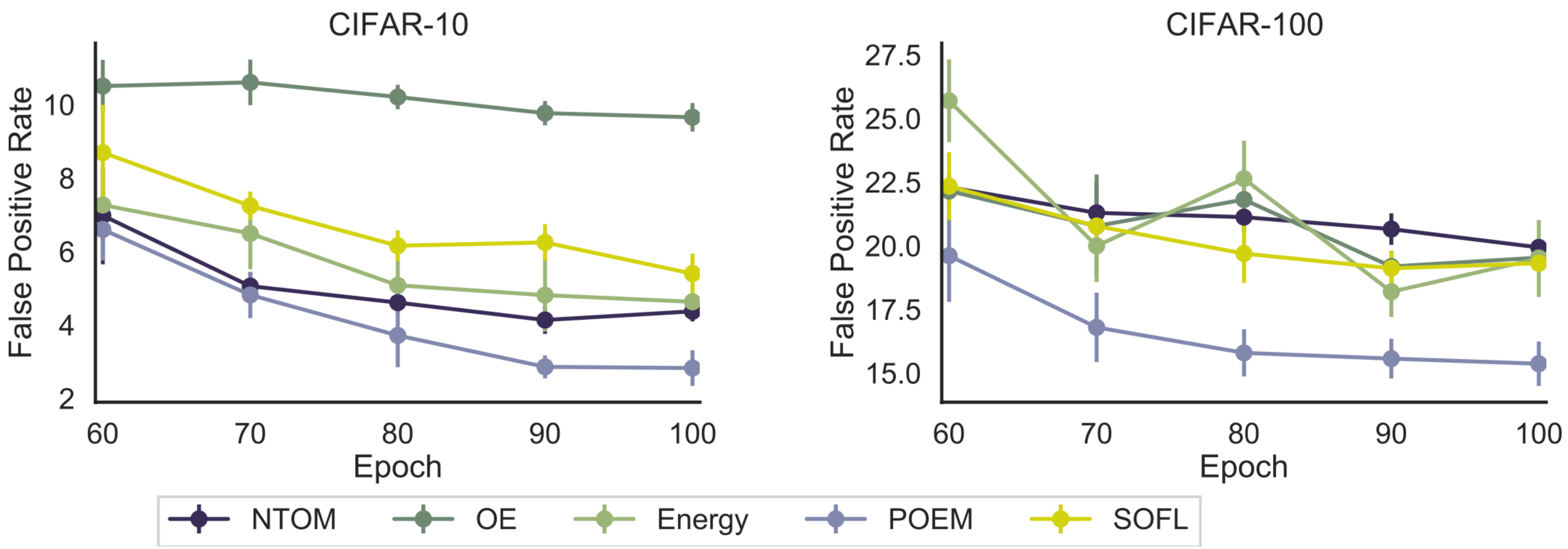
- POEM utilizes outliers more efficiently than other OE based methods



A Closer Look at Benefits of Thompson Sampling

Observations

- POEM utilizes outliers more efficiently than other OE based methods



- Training with more randomly sampled outliers does not improve the performance of Energy score

Method (CIFAR-100 as $\mathcal{D}_{\text{in}}$)	**FPR95** ↓	**AUROC** ↑	**Time**↓
1x outliers (rand. sampling)	19.25	96.68	5.0h
3x outliers (rand. sampling)	19.19	97.18	8.9h

Summary

Our contributions

- We propose a novel Posterior Sampling-based Outlier Mining framework (POEM), which facilitates efficient use of outlier data and promotes learning a compact ID-OOD decision boundary
- Theoretically: We provide insights on why outlier mining with high boundary scores benefits sample efficiency
- Empirically:
  - POEM established SoTA on common benchmarks
  - Thompson Sampling is better than greedy sampling
  - POEM utilizes outliers more effectively than other OE-based methods

GitHub https://github.com/deeplearning-wisc/poem