POEM: Out-of-Distribution Detection with Posterior Sampling

Yifei Ming [*1] **Ying Fan** [*1] **Yixuan Li** [1]

Abstract

Out-of-distribution (OOD) detection is indispensable for machine learning models deployed in the open world. Recently, the use of an auxiliary outlier dataset during training (also known as outlier exposure) has shown promising performance. As the sample space for potential OOD data can be prohibitively large, sampling informative outliers is essential. In this work, we propose a novel posterior sampling-based outlier mining framework, POEM, which facilitates efficient use of outlier data and promotes learning a compact decision boundary between ID and OOD data for improved detection. We show that POEM establishes *state-of-the-art* performance on common benchmarks. Compared to the current best method that uses a greedy sampling strategy, POEM improves the relative performance by 42.0% and 24.2% (FPR95) on CIFAR-10 and CIFAR-100, respectively. We further provide theoretical insights on the effectiveness of POEM for OOD detection.

1. Introduction

Out-of-distribution (OOD) detection has become a central challenge in safely deploying machine learning models in the open world, where the test data can naturally arise from a different distribution. Concerningly, modern neural networks are shown to produce overconfident and therefore untrustworthy predictions for OOD inputs (Nguyen et al., 2015). To mitigate the issue, recent works have explored training with a large auxiliary outlier dataset, where the model is regularized to produce lower confidence (Lee et al., 2018a; Hendrycks et al., 2018; Meinke & Hein, 2020; Chen et al., 2021) or higher energy (Liu et al., 2020) on the outlier training data. These methods have demonstrated promising OOD detection performance over the counterpart (without auxiliary data).

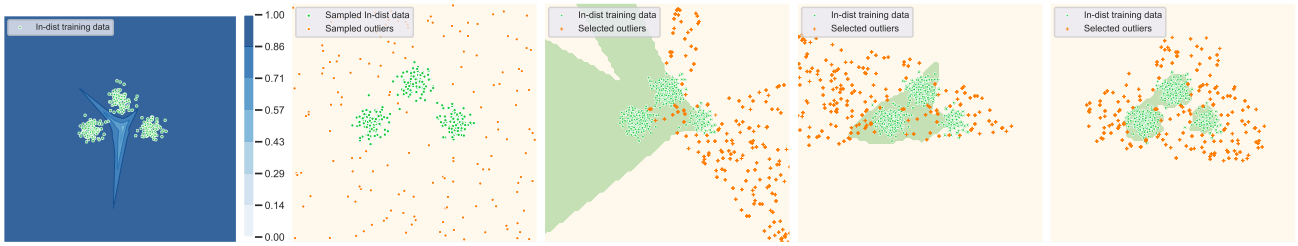
Despite encouraging results, existing methods suffer from ineffective use of outliers, and have largely underlooked the importance of outlier mining. In particular, the sample space for potential outlier data can be prohibitively large, making the majority of outliers uninformative for model regularization. As recently observed by Chen et al. (2021), *randomly* selecting outlier samples during training yields a large portion of uninformative samples that do not meaningfully improve the estimated decision boundary between in-distribution (ID) and OOD data. The above limitations motivate the following important yet underexplored question: *how can we efficiently utilize the auxiliary outlier data for model regularization*?

In this work, we propose a novel Posterior Sampling-based Outlier Mining **(POEM)** framework for OOD detection — selecting the most informative outlier data from a large pool of auxiliary data points, which can help the model estimate a compact decision boundary between ID and OOD for improved OOD detection. For example, Figure 1 shows the evolution of decision boundaries, as a result of outlier mining. Our novel idea is to formalize outlier mining as sequential decision making, where actions correspond to outlier selection, and the reward function is based on the closeness to the boundary between ID and OOD data. In particular, we devise a novel reward function termed *boundary score*, which is higher for outliers close to the boundary. Key to our framework, we leverage Thompson sampling (Thompson, 1933), also known as posterior sampling, to balance exploration and exploitation during reward optimization.

As an integral part of our framework, `POEM` maintains and updates the posterior over models to encourage better exploration. For efficient outlier mining, `POEM` chooses samples with the highest boundary scores based on the posterior distribution. Because the exact posterior is often intractable, we approximate it by performing Bayesian linear regression on top of feature representations extracted via a neural network (Riquelme et al., 2018). Posterior updates are performed periodically during training, which help shape the estimated decision boundary between ID and OOD accordingly (see Figure 1). Unlike the greedy mining approach (Chen et al., 2021) that focuses on exploitation,

[*]Equal contribution [1]Department of Computer Sciences, University of Wisconsin-Madison, USA. Correspondence to: Yifei Ming <ming5@wisc.edu>, Ying Fan <yfan87@wisc.edu>, Yixuan Li <sharonli@cs.wisc.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.



(a) ID data only (b) Sample Space (c) Epoch 1 (d) Epoch 4 (e) Epoch 30

Figure 1. Illustration of our *Posterior Sampling-based Outlier Mining* **(POEM)** framework. (a) Samples from in-distribution $\mathcal{P}_{\text{in}}$ (green dots). Blue shades represent the maximum predicted probability of a neural network classifier trained *without* an auxiliary outlier dataset. The resulting classifier is overly confident for regions (dark blue) far away from the ID data. (b) Some samples drawn from $\mathcal{P}_{\text{in}}$ (green dots) and auxiliary outlier data $\mathcal{P}_{\text{aux}}$ (orange dots). (c) to (e): Selected outliers (orange dots) from posterior distributions at different training epochs, along with the decision boundary (green for ID and beige for OOD) of the classifier.

`POEM` selects outliers via posterior sampling and allows for better balancing between exploration and exploitation. We show that our framework is computationally tractable and can be trained efficiently end-to-end in the context of modern deep neural networks.

We show that `POEM` demonstrates state-of-the-art OOD detection performance and enjoys good theoretical properties. On common OOD detection benchmarks, `POEM` significantly outperforms competitive baselines using randomly sampled outliers (Hendrycks et al., 2018; Liu et al., 2020; Mohseni et al., 2020; Meinke & Hein, 2020). Compared to the greedy sampling strategy (Chen et al., 2021), `POEM` improves the relative performance by 42.0% and 24.2% (FPR95) on CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009) respectively, evaluated across six diverse OOD test datasets. Lastly, we provide theoretical analysis for `POEM`, and explain why outliers with high boundary scores benefit sample complexity. Our main contributions are:

- We propose a novel Posterior Sampling-based Outlier Mining framework (dubbed `POEM`), which facilitates efficient use of outlier data and promotes learning a compact decision boundary between ID and OOD data for improved OOD detection.
- We perform extensive experiments comparing `POEM` with competitive OOD detection methods and various sampling strategies. We show that `POEM` establishes **state-of-the-art** results on common benchmarks. `POEM` also displays strong performance with comparable computation as baselines via a tractable algorithm for maintaining and updating the posterior.
- We provide theoretical insights on why outlier mining with high boundary scores benefits sample efficiency.

2. Preliminaries

We consider supervised multi-class classification, where $\mathcal{X} = \mathbb{R}^d$ denotes the input space and $\mathcal{Y} = \{1, 2, ..., K\}$ denotes the label space. The training set $\mathcal{D}_{\text{in}}^{\text{train}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ is drawn *i.i.d.* from $P_{\mathcal{X}\mathcal{Y}}$. Let $\mathcal{P}_{\text{in}}$ denote the marginal (ID) distribution on $\mathcal{X}$.

2.1. Out-of-distribution Detection

When deploying a model in the real world, a reliable classifier should not only accurately classify known in-distribution (ID) samples, but also identify as "unknown" any OOD input. This can be achieved through having dual objectives: OOD identification and multi-class classification of ID data (Bendale & Boult, 2016).

OOD detection can be formulated as a binary classification problem. At test time, the goal of OOD detection is to decide whether a sample $\mathbf{x} \in \mathcal{X}$ is from $\mathcal{P}_{\text{in}}$ (ID) or not (OOD). In literature, OOD distribution $\mathcal{P}_{\text{out}}$ often simulates unknowns encountered during deployment time, such as samples from an irrelevant distribution whose label set has no intersection with $\mathcal{Y}$ and therefore should not be predicted by the model. The decision can be made via a thresholding comparison:

$$D_\lambda(\mathbf{x}) = \begin{cases} \text{ID} & S(\mathbf{x}) \geq \gamma \\ \text{OOD} & S(\mathbf{x}) < \gamma \end{cases},$$

where samples with higher scores $S(\mathbf{x})$ are classified as ID and vice versa, and γ is the threshold.

Auxiliary outlier data. While the test-time OOD distribution $\mathcal{P}_{\text{out}}$ remains unknown, recent works (Hendrycks et al., 2018) make use of an auxiliary unlabeled dataset $\mathcal{D}_{\text{aux}}$ drawn from $\mathcal{P}_{\text{aux}}$ for model regularization. The model is encouraged to be less confident in the auxiliary outliers. For terminology clarity, we refer to training-time examples as *auxiliary outliers* and exclusively use *OOD data* to refer to test-time unknown inputs.

Although previous works showed that utilizing an auxiliary outlier dataset during training improves OOD detection, they suffer from ineffective use of outliers due to the prohibitively large sample space of OOD data. This calls for a sample-efficient way of utilizing the auxiliary outlier data.

3. Method

In this section, we present our novel framework, *Posterior sampling-based outlier mining* (dubbed **POEM**) — selecting the most informative outlier data from a large pool of unlabeled data points, which helps the model estimate a compact decision boundary between ID and OOD for improved OOD detection. Unlike using random sampling (Hendrycks et al., 2018; Mohseni et al., 2020; Liu et al., 2020) or greedy sampling (Chen et al., 2021), our framework allows balancing exploitation and exploration, which enjoys both strong empirical performance and theoretical properties.

In designing POEM, we address three key challenges. We will first introduce how to model outlier mining via the Thompson sampling process (Section 3.1), and then describe how to tractably maintain and update the posterior in the context of modern neural networks (Section 3.2). Lastly, we describe the training objective that incorporates the sampled outliers for model regularization (Section 3.3).

3.1. Outlier Mining: A Thompson Sampling View

One novelty of this work is to formalize outlier mining as sequential decision making, where the actions correspond to outliers selection, and the reward function is based on the closeness to the (unknown) decision boundary between ID and OOD data. The goal of outlier mining is to identify the most informative outliers that are closer to the decision boundary between ID and OOD data. Key to our framework, we leverage the classic and elegant Thompson sampling algorithm (Thompson, 1933) — also known as posterior sampling — to balance between exploitation (*i.e.*, maximizing immediate performance) and exploration (*i.e.*, investing to accumulate new information that may improve future performance).

A conceptual example of outlier mining. To help readers understand the role of outlier mining, we provide a simple example in Figure 1. The in-distribution data (green) consists of three class-conditional Gaussians. Outlier training data (orange) is drawn from a uniform distribution and is at least two standard deviations away from the mean of every in-distribution class. Figure 1 (c)-(d) shows the evolution of decision boundaries between ID and outliers, as a result of outlier mining. We can see that outlier mining significantly reduces the uninformative outliers selected from epoch 4 to epoch 30. These near-boundary outliers can help the network become aware of the real decision boundary between ID and outliers, which improves the OOD detection.

Formalize outlier mining. We now formalize the outlier mining process (see Algorithm 1 for an overview). At each step t, the model parameter $\mathbf{w}_t$ is sampled from the posteior distribution, then the learner takes an action a_t by choosing outlier $\mathbf{x} \sim \mathcal{P}_{\text{aux}}$ based on $\mathbf{w}_t$. The learner receives a reward

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

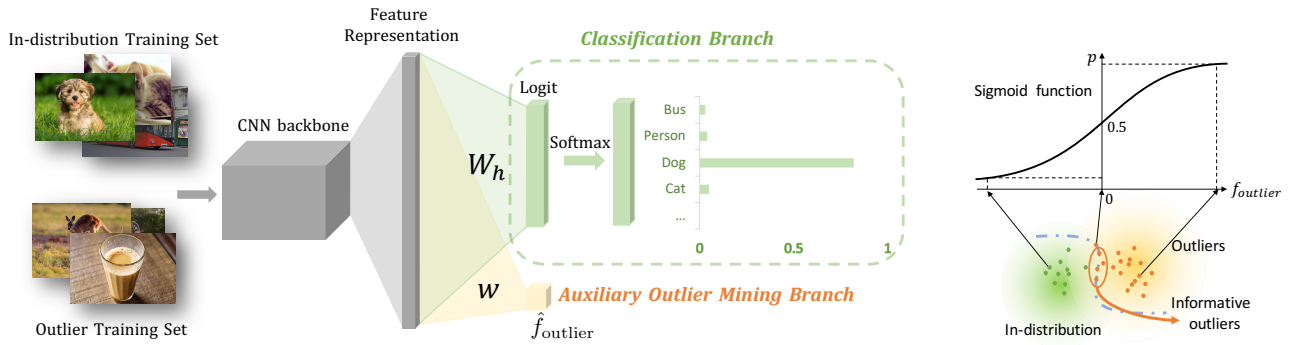
$G(\mathbf{x})$ termed **boundary score**, where a high boundary score indicates outliers being close to the boundary. The goal of outlier mining is to find the most informative outliers, *i.e.*, those with the highest boundary scores, since they are more desirable for model regularization to learn a compact boundary between ID and OOD. A key component in Thompson sampling is to maintain a posterior over models to encourage better exploration. The posterior distribution is then updated after the reward is observed.

Concretely, we define $G(\mathbf{x}) = -|f_{\text{outlier}}(\mathbf{x}; \mathbf{w}^*)|$, where f_{outlier} is a function parameterized by some unknown ground truth weights $\mathbf{w}^*$ and maps a high-dimensional input $\mathbf{x}$ into a scalar. One can easily convert the logit to its probabilistic form by using the sigmoid function: $p(\text{outlier}|\mathbf{x}) = \text{Sigmoid}(f_{\text{outlier}}(\mathbf{x}; \mathbf{w}^*))$. As shown in Figure 2b, the near-boundary outliers would correspond to $|f_{\text{outlier}}(\mathbf{x}; \mathbf{w}^*)| \approx 0$. In a nutshell, we are optimizing an unknown function by selecting samples, maintaining and modeling the distribution of $\mathbf{w}$, and using this model to select near-boundary outliers over time. Unlike greedy algorithms that focus on exploitation, the Thompson sampling framework allows balancing the exploration and exploitation trade-off during the optimization. We proceed to describe details of maintaining and updating the posterior distribution for the model.

3.2. Approximate Posterior with Neural Networks

A key challenge in our outlier mining framework is *how to maintain and update the posterior over models*? Because the exact posterior is often intractable, we need ways to approximate them efficiently. Recent advances in approximate Bayesian methods have made posterior approximation for flexible neural network models practical. As shown in Riquelme et al. (2018), performing Bayesian linear regression on top of feature representations extracted via a neural network is tractable, easy-to-tune, and enjoys strong performance compared to a wide range of other Bayesian approximations. This also circumvents the problem of linear algorithms due to their lack of representational power.

Bayesian linear regression based on deep features. Give the above considerations, in this work, we choose to perform Bayesian linear regression (BLR) on top of the penultimate



(a) Method Overview (b) Boundary Score & Density

Figure 2. (a) Overview of **POEM**. We use a two-branch architecture with shared feature representation. During training, the outlier mining branch (beige) selects the near-boundary outliers. A mixture of outliers and ID data is used to train the classifier (green) and update the weights in the CNN backbone (grey). Based on the updated feature representation, the posterior update is applied to the weights $\mathbf{w}$ of the Bayesian linear layer to refine the decision boundary between ID and outlier data. (b) The connection between the boundary score $G(\mathbf{x}) = -|f_{\text{outlier}}(\mathbf{x}; \mathbf{w}^*)|$ and density. Outliers close to the boundary tend to have higher boundary scores.

layer feature $\phi(\mathbf{x}) \in \mathbb{R}^m$ of a deep neural network to model the boundary score:

$$\hat{f}_{\text{outlier}}(\mathbf{x}; \mathbf{w}_t) = \mathbf{w}_t^\top \phi(\mathbf{x}),$$

where $\mathbf{w}_t \in \mathbb{R}^m$ is the weight parameter sampled from a posterior distribution (*c.f.* Equation 1). Based on the sampled model $\mathbf{w}_t$, we can perform outlier mining by selecting a set of outliers from the auxiliary dataset according to the estimated boundary score $\hat{G}(\mathbf{x}) = -|\hat{f}_{\text{outlier}}(\mathbf{x}; \mathbf{w}_t)|$. We use the notation $\hat{f}_{\text{outlier}}(\mathbf{x}; \mathbf{w}_t)$ to distinguish model estimation from the groundtruth function $f_{\text{outlier}}(\mathbf{x}; \mathbf{w}^*)$ where $\mathbf{w}^*$ is unknown. The target logits are set similar to Weber et al. (2018); see Section 4.1 for details.

Besides simplicity and tractability, we will later show in Section 4.2 that the above model based on deep features demonstrates strong empirical performance across a wide range of tasks and enjoys good theoretical properties.

Tractable and efficient posterior update. We now describe details on the distribution of $\mathbf{w}$. One benefit of Bayesian linear regression is the existence of closed-form formulas for posterior update (Rasmussen, 2003). We assume a Gaussian prior of $\mathbf{w}_0 \sim \mathcal{N}(0, \Sigma)$. The posterior distribution of $\mathbf{w}_t$ is a multivariate Gaussian with the following closed form:

$$\mathbf{w}_t \sim \mathcal{N}\left(\sigma^{-2}\Sigma_{\text{p}}^{-1}\Phi\mathbf{y}_{\text{tar}}, \Sigma_{\text{p}}^{-1}\right), \tag{1}$$

where $\Sigma_{\text{p}} := \sigma^{-2}\Phi\Phi^\top + \Sigma^{-1}$ is the posterior covariance matrix, $\Phi \in \mathbb{R}^{m\times M}$ is the concatenation of feature representations $\{\phi(\mathbf{x}_i)\}_{i=1}^M$, and $\mathbf{y}_{\text{tar}} \in \mathbb{R}^M$ is the concatenation of target logit values, and σ^2 the variance of *i.i.d.* noises in target logit values. In practice, we use a fix-size queue to store M most recent feature vectors Φ to save computation. M can be much smaller than the auxiliary dataset size $M \ll |\mathcal{D}_{\text{aux}}|$. Similar techniques are used in Azizzadenesheli et al. (2018). We show in Section 5 that our method overall achieves strong performance with computational efficiency.

Interleaving posterior update with feature update. During training, we interleave posterior update with the feature representation update, which are mutually beneficial as the former helps select near-boundary outliers that facilitate learning a good representation, which in turn improves the decision boundary estimation. Specifically, at the start of each epoch, we first sample the model from the posterior, based on which we select a set of outliers from the auxiliary dataset according to the *estimated* boundary score. Then we train the neural network for one epoch and update a fix-size queue with new features, based on which we update the posterior distribution of the model. We present the pseudo-code of POEM in Algorithm 2.

Remark: Justification for choosing Thompson sampling with BLR. We choose Thompson sampling with BLR because it is simple and effective in practice, and also enjoys good theoretical guarantees. **(1)** Empirically, due to the stochastic nature of Thomspon sampling, there is no need for extra hyper-parameters except for the prior distribution. In other statistical optimization methods like upper confidence bound (UCB (Auer, 2002)), the upper bound is another critical hyper-parameter, and choosing an upper bound to cover the actual value with high probability can be hard in some cases (Russo & Van Roy, 2014). Sub-optimal construction of such upper bound can lead to a lack of statistical efficiency (Osband & Van Roy, 2017). **(2)** Theoretically, Thompson sampling with BLR matches the regret bound of linear UCB from a Bayesian view (Russo & Van Roy, 2014), while it also enjoys computational tractability. Other

Algorithm 2 Posterior Sampling-based Outlier Mining (POEM)

Input: In-distribution training set $\mathcal{D}_{\text{in}}^{\text{train}} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$, a large outlier dataset $\mathcal{D}_{\text{aux}}$, neural network encoder ϕ, weights $\mathbf{w}_{\text{OM}}$ for Bayesian linear regression with prior covariance Σ, the variance of target logits noise σ^2, margin hyperparameters m_{in} and m_{out}, regularizer weight β, pool size S, queue size M, prior distribution $\mathcal{N}(0, \Sigma)$

Sample $\mathbf{w}_0 \sim \mathcal{N}(0, \Sigma)$
for $t = 0, 1, 2, \ldots$ **do**
  Sample a subset of $\mathcal{D}_{\text{aux}}$ as the outlier pool $\mathcal{D}_{\text{pool}} = \{\mathbf{x}_i\}_{i=1}^{S}$
  for $\mathbf{x}_i \in \mathcal{D}_{pool}$ **do**
    Calculate the estimated boundary score $\hat{G}(\mathbf{x}_i) = -|\hat{f}_{\text{outlier}}(\mathbf{x}_i; \mathbf{w}_t)| = -|\mathbf{w}_t^\top \phi(\mathbf{x}_i)|$
  // select informative outliers
  Construct $\mathcal{D}_{\text{out}}^{\text{OM}}$ by selecting the top N samples with the largest $\hat{G}(\mathbf{x})$ from $\mathcal{D}_{\text{pool}}$
    // update weights in the backbone and multi-class classification branch
  Train for one epoch with energy-regularized cross-entropy loss $L_{\text{cls}} + \beta L_{\text{reg}}$ defined in Equation (2)
    // use queue to save computational cost
  Update the feature queue $Q = \{\phi(\mathbf{x}_i)\}_{i=1}^{M}$ based on $\mathcal{D}_{\text{out}}^{\text{OM}}$ and $\mathcal{D}_{\text{in}}^{\text{train}}$
    // update feature representation matrix based on the queue
  Update $\Phi = [\phi(\mathbf{x}_1)|\phi(\mathbf{x}_2)|\ldots|\phi(\mathbf{x}_M)]$ by concatenation
    // the posterior is updated to facilitate exploration
  Update the posterior covariance matrix $\Sigma_p = \sigma^{-2}\Phi\Phi^\top + \Sigma^{-1}$
    // sample from the posterior distribution
  Sample $\mathbf{w}_{t+1} \sim \mathcal{N}\left(\sigma^{-2}\Sigma_p^{-1}\Phi\mathbf{y}_{\text{tar}}, \Sigma_p^{-1}\right)$

methods like entropy-based methods, probability matching, and Bayesian neural networks do not enjoy such strong guarantees to our knowledge. In one sentence, Thompson sampling with BLR is a good trade-off between computational tractability and OOD detectability.

3.3. Putting Together: Learning and Inference

Lastly, we address the third challenge: *how can we leverage the mined outliers for model regularization*? We present the full training and evaluation algorithm for multi-class classification and OOD detection.

Architecture. As shown in Figure 2a, our framework consists of two branches: a *classification branch* (in green) with K outputs of class labels, and an *outlier mining* branch (in beige) where Bayesian neural linear regression is performed. Two branches share the same feature representation. The feature vector $\phi(\mathbf{x}) \in \mathbb{R}^m$ goes through a linear transformation with weight matrix $\mathbf{W}_h \in \mathbb{R}^{m\times K}$, followed by a softmax function. The final softmax prediction is a vector $F(\mathbf{x}) = \text{softmax}(\mathbf{W}_h^\top \cdot \phi(\mathbf{x})) \in \mathbb{R}^K$.

Training procedure. The overall training workflow consists of three steps: **(1)** constructs an auxiliary outlier training set by selecting outliers with the highest *estimated* boundary scores from a large candidate pool. **(2)** The classification branch together with the network backbone are trained using a mixture of ID and selected outlier data. **(3)** Based on the updated feature representation, we perform the posterior update of the weights in the outlier mining branch. The pseudo-code is provided in Algorithm 2. We proceed by describing the training objective for the classification branch.

Training objective. Our overall training objective is a combination of standard cross-entropy loss, together with a regularization term L_{reg}:

$$\min_{\theta} \quad \mathbb{E}_{(\mathbf{x},c)\sim\mathcal{D}_{\text{in}}^{\text{train}}}[-\log F_c(\mathbf{x};\theta)] + \beta \cdot L_{\text{reg}}, \tag{2}$$

where $\mathcal{D}_{\text{in}}^{\text{train}}$ is the in-distribution training data and θ is the parameterization of the neural network. We leverage the energy-regularized learning (Liu et al., 2020), which performs the classification task while regularizing the model to produce *lower* energy for ID data and *higher* energy for the auxiliary outliers. This helps create a strong energy gap that facilitates test-time OOD detection. The regularization is defined by:

$$L_{\text{reg}} = \mathbb{E}_{\mathbf{x}_{\text{in}}\sim\mathcal{D}_{\text{in}}^{\text{train}}}(\max(0, E(\mathbf{x}_{\text{in}};\theta) - m_{\text{in}}))^2 \tag{3}$$

$$+ \mathbb{E}_{\mathbf{x}_{\text{out}}\sim\mathcal{D}_{\text{out}}^{\text{OM}}}(\max(0, m_{\text{out}} - E(\mathbf{x}_{\text{out}};\theta)))^2, \tag{4}$$

where $\mathcal{D}_{\text{out}}^{\text{OM}}$ is the auxiliary outliers selected by our outlier mining procedure (Section 3.1). m_{in} and m_{out} are margin hyperparameters, and the energy $E(\mathbf{x};\theta) := -\log\sum_{i=1}^{K} e^{f_i(\mathbf{x};\theta)}$ as defined in Liu et al. (2020). While Liu et al. (2020) randomly sampled outliers from the pool, we perform outlier mining based on posterior sampling. We will show in Section 4 that OOD detection performance can be improved significantly with our novel outlier selection.

OOD inference. At test time, the OOD detection is based on the energy of the input: $D_\lambda(\mathbf{x}) = \mathbb{1}\{-E(\mathbf{x}) \geq \gamma\}$, where a threshold mechanism is exercised to distinguish

between ID and OOD. Note that we negate the sign of the energy $E(\mathbf{x})$ to align with the convention that samples with higher scores are classified as ID and vice versa. The threshold γ is typically chosen so that a high fraction of ID data (*e.g.,* 95%) is correctly classified.

4. Experiments

In this section, we present extensive experiments to validate the superiority of POEM. We also provide comparisons with three broad categories of OOD detection methods based on: (1) pre-trained models, (2) models trained with randomly sampled outliers, and (3) models trained with greedily sampled outliers. Code is publicly available at: `https://github.com/deeplearning-wisc/poem`.

4.1. Experimental Setup

Datasets. Following the common benchmarks, we use `CIFAR-10` and `CIFAR-100` (Krizhevsky et al., 2009) as in-distribution datasets. A downsampled version of ImageNet (`ImageNet-RC`) (Chrabaszcz et al., 2017) is used as the auxiliary outlier dataset. For OOD test sets, we use a suite of diverse image datasets including `SVHN` (Netzer et al., 2011), `Textures` (Cimpoi et al., 2014), `Places365` (Zhou et al., 2017), `LSUN-crop`, `LSUN-resize` (Yu et al., 2015), and `iSUN` (Xu et al., 2015).

Training details. The pool of outliers consists of randomly selected 400,000 samples from ImageNet-RC, and only 50,000 samples (same size as the ID training set) are selected for training based on the boundary score. For efficiency, we follow common practice and maintain a queue with data in the previous 4 epochs for the posterior update (Weber et al., 2018). In other words, the size of our queue (which is also the action space we select outliers from) is $M = 50,000 \times 4$. We use DenseNet-101 as the backbone for all methods and train the model using stochastic gradient descent with Nesterov momentum (Duchi et al., 2011). We set the momentum to be 0.9 and the weight decay coefficient to be 10^{-4}. The batch size is 64 for both in-distribution and outlier training data. Models are trained for 100 epochs. For a fair comparison, the above setting is the *same* for all methods trained with outliers. SSD+ (Sehwag et al., 2021) employed the supervised contrastive loss (Khosla et al., 2020), which requires a larger batch size (1024) and longer training time (400 epochs) than the softmax cross-entropy loss. For margin hyperparameters, we use the default as in Liu et al. (2020): $m_{\text{in}} = -7$, $m_{\text{out}} = -25$ and $\beta = 0.1$.

In Bayesian regression, we follow the practice in Weber et al. (2018), and regress on target logit values of 3 (for $\mathbf{x} \in \mathcal{D}_{\text{aux}}$, corresponding to a target probability of 0.95) and -3 (for $\mathbf{x} \in \mathcal{D}_{\text{in}}^{\text{train}}$ with a target probability of 0.05) with *i.i.d.* noise $\upsilon \sim \mathcal{N}(0, \sigma^2)$. Note that we avoid regressing on 0-1 labels since it incurs infinite logits. As a result, the reward incurred is $|3+\upsilon|$ for $\mathbf{x} \in \mathcal{D}_{\text{aux}}$ and $|-3+\upsilon|$ for $\mathbf{x} \in \mathcal{D}_{\text{in}}^{\text{train}}$. In this setting, Thompson sampling can still find new OOD data that are closer to the boundary (see detailed explanation in Appendix C).

Evaluation metrics. Following common practice in the literature, we report: (1) the false positive rate (FPR95) of OOD samples when the true positive rate of in-distribution samples is at 95%; (2) the area under the receiver operating characteristic curve (AUROC); (3) the area under the precision-recall curve (AUPR). We also report the ID classification accuracy (ID-ACC).

4.2. Results and Discussion

POEM achieves SOTA performance. Our method outperforms existing competitive methods, establishing *state-of-the-art* performance on both CIFAR-10 and CIFAR-100. Table 1 summarizes detailed comparison with methods that **(1)** directly use a pre-trained network for OOD detection: `MSP` (Hendrycks & Gimpel, 2017), `ODIN` (Liang et al., 2018), `Mahalanobis` (Lee et al., 2018b), `Energy` (Liu et al., 2020); **(2)** use an auxiliary outlier dataset but randomly select outliers during training: `OE` (Hendrycks et al., 2018), `SOFL` (Mohseni et al., 2020), `CCU` (Meinke & Hein, 2020), energy-regularized learning (*i.e.*, Energy (with outlier)); **(3)** involve outlier mining: `NTOM` (Chen et al., 2021). Compared to the best baseline energy-regularized learning, `POEM` reduces the FPR95 by 4.1% on CIFAR-100, which is a relative **21.4**% reduction in error. This highlights the benefits of posterior-sampling-based outlier mining, as opposed to randomly selecting outliers as in Liu et al. (2020).

Thompson sampling v.s. greedy sampling. Chen et al. (2021) also perform outlier mining for OOD detection, which is the most relevant baseline[1]. `NTOM` can be viewed as a simple greedy sampling strategy, where outliers are selected based on the estimated confidence without considering the uncertainty of model parameters. Despite the simplicity, this method only exploits what is currently available to the model, while falling short of proper exploration. In contrast, our method achieves a better exploration-exploitation trade-off by maintaining a posterior distribution over models. Empirically, under the same configurations, our method outperforms `NTOM` by 4.82% (FPR95) on CIFAR-100. Moreover, while the performance of NTOM can be sensitive to the confidence hyperparameter (see Table 5 in Appendix), our framework does not suffer from this issue. Since the estimated decision boundary is adjusted

[1]There are two variants in Chen et al. (2021): NTOM (for standard training) and ATOM (for adversarial training). NTOM has comparable performance as ATOM on natural datasets, and is a more fair comparison for our setting.

Table 1. **Main Results**. Comparison with competitive OOD detection methods trained with the same DenseNet backbone. All values are percentages and are averaged over six OOD test datasets described in Section 4.1. Bold numbers indicate superior results. We report the performance of POEM based on 5 independent training runs. Standard deviations for competitive baselines can be seen in Figure 3.

$\mathcal{D}_{\text{in}}$	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
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. $\mathcal{D}_{\text{aux}}$) (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

through the posterior update, POEM obviates the need for a dataset-dependent hyperparameter.

POEM utilizes outliers more effectively than existing approaches. Figure 3 contrasts the OOD detection performance of various methods at different training epochs. For a fair comparison, we mainly consider methods that use auxiliary outlier datasets. We report the average FPR95 (with standard deviation) at varying epochs. On both CIFAR-10 and CIFAR-100, POEM achieves lower FPR95 with fewer training epochs. Our experiments suggest that POEM utilizes outliers more effectively than existing approaches. This also highlights the importance of selectively choosing the outliers near the decision boundary between ID and OOD.

POEM improves OOD detection while maintaining comparable classification accuracy. We compare the multi-class classification accuracy in Table 1. When trained on DenseNet with CIFAR-100 as ID, POEM achieves a test error of 26.59%, compared to the NTOM fine-tuned model's 26.14% and the pre-trained model's 25.95%. Overall our training method leads to improved OOD detection performance, and at the same time maintains comparable classification accuracy on in-distribution data.

Due to space constraints, additional ablation studies such as the impact of *alternative auxiliary datasets*, and the *pool size of auxiliary datasets* are included in Appendix A.

5. Further Discussion on Computation

As with most Bayesian methods, there is no "free lunch": keeping track of the uncertainty of model parameters comes at a cost of additional computation. In this section, we discuss the computation and performance trade-off, and we demonstrate that our method POEM achieves strong performance with moderate computational cost. We also provide some practical techniques to further reduce computation. The estimated average runtime for each method (utilizing outliers) is summarized in Table 2. Methods utilizing auxiliary data generally take longer to train, as a trade-off for superior detection performance. Among those, SOFL (Mohseni et al., 2020) is the most computationally expensive baseline (14 hours). The training time of POEM is shorter than SOFL. Moreover, the performance gain is substantial, establishing state-of-the-art performance. For example, the average FPR95 is reduced from 19.25% to 15.14% on CIFAR-100. To put the numbers in context, compared with methods using outlier mining such as NTOM (Chen et al., 2021) (19.96%), POEM still yields superior performance under comparable computations.

We can improve the training efficiency of POEM using early stopping, *i.e.*, no outlier mining and posterior update after certain epochs but regular training continues. The insight is taken from Figure 3, where POEM establishes SOTA performance around epoch 80. This further saves the average training time by 2.4 hours with a marginal performance

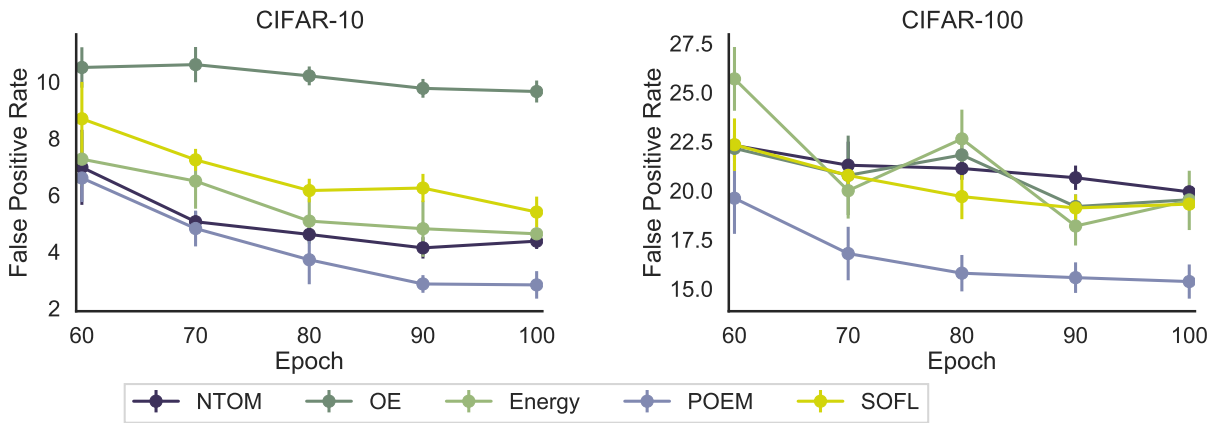


Figure 3. Performance comparison of methods that utilize an auxiliary outlier dataset on CIFAR-10 (left) and CIFAR-100 (right) at different epochs during training. **Mean** and **standard deviations** are estimated across five independent training runs.

Table 2. The estimated average runtime for each method. Models are trained with DenseNet as the architecture. ImageNet-RC (Chrabaszcz et al., 2017) is used for methods that use an auxiliary outlier dataset. h denotes hour.

Method	**Training Time**
Energy (w. $\mathcal{D}_{\text{aux}}$) (Liu et al., 2020)	5.0 h
CCU (Meinke & Hein, 2020)	6.7 h
NTOM (Chen et al., 2021)	7.4 h
SOFL (Mohseni et al., 2020)	14.0 h
POEM	10.3 h
POEM (early stopping)	7.9 h

decrease (with AUROC 97.33% on CIFAR-100).

Training with more randomly sampled outliers does not outperform POEM. We verify that random sampling with more auxiliary samples does not yield more competitive performance compared to POEM. As shown in Table 3, training (Liu et al., 2020) with 3 times more auxiliary samples only marginally improved the performance (FPR95 from 19.25% to 19.19%, AUROC from 96.68% to 97.18%) on CIFAR-100. Here we refer to training with a larger outlier buffer with randomly sampled outliers. The overall size of the auxiliary dataset ImageNet-RC is kept the same. However, simply training with more randomly sampled outliers incurred a significant computational burden, where the training time increases from 5h to 8.9h. The above observations further highlight the benefit of outlier mining with POEM.

Table 3. Training with more *random* outliers does not improve the performance of Energy (w. $\mathcal{D}_{\text{aux}}$) (Liu et al., 2020). The results reported are the average of 5 independent runs.

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

6. Theoretical Insights: Sample Complexity with High Boundary Scores

In this section, we provide insights on how selecting data points with high boundary scores benefits sample efficiency under a simple Gaussian mixture model in binary classification. Due to space constraints, a discussion on more general models for OOD detection can be found in Appendix F.

Setup. As our method works on the feature space, to simplify notations, we use $\mathbf{x}$ to denote the extracted feature and the following distributions are defined on the feature space. We assume $\mathcal{P}_{\text{in}} = \mathcal{N}(\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$, and $\mathcal{P}_{\text{aux}} = \mathcal{N}(-\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$, with $\boldsymbol{\mu} \in \mathbb{R}^d$. The hypothesis class $\mathcal{H} = \{\text{sign}(\boldsymbol{\theta}^\top \mathbf{x}), \boldsymbol{\theta} \in \mathbb{R}^d\}$, where a classifier outputs 1 if it predicts $\mathbf{x} \sim \mathcal{P}_{\text{in}}$ and -1 if it predicts $\mathbf{x} \sim \mathcal{P}_{\text{aux}}$. The overall feature distribution is a Gaussian mixture model with equal class priors: $\frac{1}{2}\mathcal{N}(\boldsymbol{\mu}, \sigma^2\boldsymbol{I}) + \frac{1}{2}\mathcal{N}(-\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$).

Outliers with high boundary scores benefit sample complexity. We can write the false negative rate (FNR) and false positive rate (FPR) of our data model: $\text{FNR}(\boldsymbol{\theta}) = erf(\frac{\boldsymbol{\mu}^\top\boldsymbol{\theta}}{\sigma||\boldsymbol{\theta}||}) = \text{FPR}(\boldsymbol{\theta})$, where $erf(x) = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-t^2/2}dt$. Consider the classifier given by $\hat{\boldsymbol{\theta}}_{n,n'} = \frac{1}{n'+n}(\Sigma_{i=1}^{n'}\mathbf{x}_i' - \Sigma_{i=1}^{n}\mathbf{x}_i)$ where each $\mathbf{x}_i$ is drawn *i.i.d.* from $\mathcal{P}_{\text{aux}}$, each $\mathbf{x}_i'$ is drawn *i.i.d.* from $\mathcal{P}_{\text{in}}$. We have the following result:

Theorem 6.1. *Assume the signal/noise ratio is large:* $\frac{||\boldsymbol{\mu}||}{\sigma} = r_0 \gg 1$, *the dimensionality/sample ratio is* $r_1 = \frac{d}{n}$, *and we have access to data points* $\mathbf{x} \sim \mathcal{P}_{aux}$ *that satisfy the following constraint of high boundary scores (on average):*

$$\frac{-\Sigma_{i=1}^{n} G(\mathbf{x}_i)}{n} \leq \epsilon.$$

There exists a constant c *that*

$$\frac{\boldsymbol{\mu}^\top\hat{\boldsymbol{\theta}}_{n,n'}}{\sigma||\hat{\boldsymbol{\theta}}_{n,n'}||} \geq \frac{||\boldsymbol{\mu}||^2 - \sigma^{1/2}||\boldsymbol{\mu}||^{3/2} - \frac{\sigma^2\epsilon}{2}}{2\sqrt{\frac{\sigma^2}{n}(d+\frac{1}{\sigma}) + ||\boldsymbol{\mu}||^2}} \tag{5}$$

with probability at least $1-(1+c)e^{-r_1 n/8\sigma^2}-2e^{-\frac{1}{2}n||\boldsymbol{\mu}||/\sigma}$.

Remark. The above result suggests that when ϵ decreases, the lower bound of $\frac{\boldsymbol{\mu}^\top \hat{\boldsymbol{\theta}}_{n,n'}}{\sigma||\hat{\boldsymbol{\theta}}_{n,n'}||}$ will increase. As $\text{FNR}(\hat{\boldsymbol{\theta}}_{n,n'}) = \text{FPR}(\hat{\boldsymbol{\theta}}_{n,n'}) = erf(\frac{\boldsymbol{\mu}^\top \hat{\boldsymbol{\theta}}_{n,n'}}{\sigma||\hat{\boldsymbol{\theta}}_{n,n'}||})$, then the upper bound of $\text{FNR}(\hat{\boldsymbol{\theta}}_{n,n'})$ and $\text{FPR}(\hat{\boldsymbol{\theta}}_{n,n'})$ will decrease, which shows the benefit of outlier mining with high boundary scores (see Appendix E for details).

7. Related Works

Thompson sampling. Thompson Sampling (Thompson, 1933), also known as posterior sampling, is an elegant framework that balances exploration and exploitation for bandit problems. It is easy to implement and can be combined with neural networks (Riquelme et al., 2018; Zhang et al., 2021). Recently, Thompson sampling enjoys popularity in a wide range of applications such as recommendation systems (Kawale et al., 2015), marketing (Schwartz et al., 2017), and web optimization (Hill et al., 2017). While the framework is widely known for bandits and reinforcement learning problems, we are the first to establish a formal connection between OOD detection and Thompson sampling.

Gaussian process with Thompson sampling. Optimizing an unknown reward function has been well studied. We follow a line of works that use Gaussian processes to model continuous functions. Russo & Van Roy (2014) explore Thompson sampling under the Gaussian process assumptions and develop a Bayesian regret bound. Riquelme et al. (2018) investigate a range of bandit problems and show that linear Bayesian models on top of a two-layer neural network representation serve as a satisfying extension of Gaussian processes in higher dimensional spaces. Azizzadenesheli et al. (2018) and Fan & Ming (2021) demonstrate the success of such methods in reinforcement learning.

Out-of-distribution detection without auxiliary data. In Bendale & Boult (2016), the OpenMax score is first developed for detecting samples from outside training categories, using the extreme value theory (EVT). Subsequent work (Hendrycks & Gimpel, 2017) proposed a baseline using maximum softmax probability, which is not suitable for OOD detection as theoretically shown in Morteza & Li (2022). Advanced techniques are proposed to improve the detection performance exploiting the logit space, including ODIN (Liang et al., 2018), energy score (Liu et al., 2020; Lin et al., 2021; Wang et al., 2021), ReAct (Sun et al., 2021), and logit normalization (Wei et al., 2022). Compared to max logit (Vaze et al., 2021), the energy score enjoys theoretical interpretation from a log-likelihood perspective. OOD detection based on the feature space such as Mahalanobis distance-based score (Lee et al., 2018b) and non-parametric KNN-based score (Sun et al., 2022) also demonstrates promises, especially on OOD with spurious correlation (Ming et al., 2022). Zhang et al. (2020) find that jointly learning a classifier with a flow-based density estimator is effective for OOD detection. Without pre-training on a large dataset (Fort et al., 2021), the performance of post hoc OOD detection methods is generally inferior to those that use auxiliary datasets for model regularization.

Out-of-distribution detection with auxiliary data. Another line of work incorporates an auxiliary outlier dataset during training, which may consist of natural (Hendrycks et al., 2018; Mohseni et al., 2020; Liu et al., 2020; Chen et al., 2021; Katz-Samuels et al., 2022) or synthesized OOD training samples (Lee et al., 2018a; Du et al., 2022b;a). Recently, Chen et al. (2021) propose a greedy confidence-based outlier mining method. Despite the simplicity, it falls short of exploration. Unlike existing methods using random or greedy sampling, our posterior sampling-based framework allows balancing exploitation and exploration, which enjoys both empirical benefits and theoretical properties.

Out-of-distribution detection in natural language processing. Distribution shifts in NLP can occur due to changes in topics and domains, unexpected user utterances, etc. Compared to early language models such as ConvNets and LSTMs, pre-trained Transformers (Vaswani et al., 2017) are shown robust to distribution shifts and more effective at identifying OOD instances (Hendrycks et al., 2020; Podolskiy et al., 2021; Xu et al., 2021). Various algorithmic solutions are proposed to handle OOD detection, including model ensembling (Li et al., 2021), data augmentation (Chen & Yu, 2021; Zhan et al., 2021), and contrastive learning (Zhou et al., 2021; Zeng et al., 2021; Jin et al., 2022). Since our method does not depend on any assumption about the task domain, it also has the potential to be applied to NLP tasks.

8. Conclusion and Outlook

In this paper, we propose a novel posterior sampling-based learning framework (POEM) that facilitates learning a more compact decision boundary between ID and OOD for improved detection. A key to our framework is finding the near-boundary outlier training examples for model regularization. We conduct extensive experiments and show that POEM establishes the state-of-the-art among competitive OOD detection methods. We also provide theoretical insights on why selecting outliers with high boundary scores improves OOD detection. We hope our research can raise more attention to a broader view of using posterior sampling approaches for OOD detection.

Acknowledgements

Research was supported by funding from the Wisconsin Alumni Research Foundation (WARF).

References

Auer, P. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3(Nov):397–422, 2002.

Azizzadenesheli, K., Brunskill, E., and Anandkumar, A. Efficient exploration through bayesian deep q-networks. In *Information Theory and Applications Workshop (ITA)*. IEEE, 2018.

Bendale, A. and Boult, T. E. Towards open set deep networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Chen, D. and Yu, Z. Gold: improving out-of-scope detection in dialogues using data augmentation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Chen, J., Li, Y., Wu, X., Liang, Y., and Jha, S. Atom: Robustifying out-of-distribution detection using outlier mining. *In Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2021.

Chrabaszcz, P., Loshchilov, I., and Hutter, F. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Du, X., Wang, X., Gozum, G., and Li, Y. Unknown-aware object detection: Learning what you don't know from videos in the wild. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022a.

Du, X., Wang, Z., Cai, M., and Li, Y. Vos: Learning what you don't know by virtual outlier synthesis. In *International Conference on Learning Representations (ICLR)*, 2022b.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Fan, Y. and Ming, Y. Model-based reinforcement learning for continuous control with posterior sampling. In *International Conference on Machine Learning (ICML)*, 2021.

Fort, S., Ren, J., and Lakshminarayanan, B. Exploring the limits of out-of-distribution detection. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *International Conference on Learning Representations (ICLR)*, 2017.

Hendrycks, D., Mazeika, M., and Dietterich, T. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations (ICLR)*, 2018.

Hendrycks, D., Liu, X., Wallace, E., Dziedzic, A., Krishnan, R., and Song, D. Pretrained transformers improve out-of-distribution robustness. In *Association for Computational Linguistics (ACL)*, 2020.

Hill, D. N., Nassif, H., Liu, Y., Iyer, A., and Vishwanathan, S. An efficient bandit algorithm for realtime multivariate optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1813–1821, 2017.

Jin, D., Gao, S., Kim, S., Liu, Y., and Hakkani-Tur, D. Towards textual out-of-domain detection without in-domain labels. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2022.

Katz-Samuels, J., Nakhleh, J., Nowak, R., and Li, Y. Training ood detectors in their natural habitats. In *International Conference on Machine Learning (ICML)*, 2022.

Kawale, J., Bui, H. H., Kveton, B., Tran-Thanh, L., and Chawla, S. Efficient thompson sampling for online matrix-factorization recommendation. In *Advances in neural information processing systems (NeurIPS)*, 2015.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lee, K., Lee, H., Lee, K., and Shin, J. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *International Conference on Learning Representations (ICLR)*, 2018a.

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018b.

Li, X., Li, J., Sun, X., Fan, C., Zhang, T., Wu, F., Meng, Y., and Zhang, J. kfolden: k-fold ensemble for out-of-distribution detection. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Liang, S., Li, Y., and Srikant, R. Enhancing the reliability of out-of-distribution image detection in neural networks. *International Conference on Learning Representations (ICLR)*, 2018.

Lin, Z., Roy, S. D., and Li, Y. Mood: Multi-level out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15313–15323, June 2021.

Liu, W., Wang, X., Owens, J., and Li, Y. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Meinke, A. and Hein, M. Towards neural networks that provably know when they don't know. *International Conference on Learning Representations (ICLR)*, 2020.

Ming, Y., Yin, H., and Li, Y. On the impact of spurious correlation for out-of-distribution detection. In *The AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

Mohseni, S., Pitale, M., Yadawa, J., and Wang, Z. Self-supervised learning for generalizable out-of-distribution detection. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Morteza, P. and Li, Y. Provable guarantees for understanding out-of-distribution detection. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2022.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. *Advances in Neural Information Processing Systems Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Nguyen, A., Yosinski, J., and Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Osband, I. and Van Roy, B. Why is posterior sampling better than optimism for reinforcement learning? In *International Conference on Machine Learning (ICML)*, 2017.

Podolskiy, A., Lipin, D., Bout, A., Artemova, E., and Piontkovskaya, I. Revisiting mahalanobis distance for transformer-based out-of-domain detection. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Rasmussen, C. E. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pp. 63–71. Springer, 2003.

Riquelme, C., Tucker, G., and Snoek, J. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *International Conference on Learning Representations (ICLR)*, 2018.

Russo, D. and Van Roy, B. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39 (4):1221–1243, 2014.

Schwartz, E. M., Bradlow, E. T., and Fader, P. S. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science*, 36(4): 500–522, 2017.

Sehwag, V., Chiang, M., and Mittal, P. Ssd: A unified framework for self-supervised outlier detection. In *International Conference on Learning Representations (ICLR)*, 2021.

Sun, Y., Guo, C., and Li, Y. React: Out-of-distribution detection with rectified activations. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Sun, Y., Ming, Y., Zhu, X., and Li, Y. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning (ICML)*, 2022.

Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Torralba, A., Fergus, R., and Freeman, W. T. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Vaze, S., Han, K., Vedaldi, A., and Zisserman, A. Open-set recognition: A good closed-set classifier is all you need. *International Conference on Learning Representations (ICLR)*, 2021.

Wang, H., Liu, W., Bocchieri, A., and Li, Y. Can multi-label classification networks know what they don't know? In *Advances in Neural Information Processing Systems*, 2021.

Weber, N., Starc, J., Mittal, A., Blanco, R., and Màrquez, L. Optimizing over a bayesian last layer. In *NeurIPS workshop on Bayesian Deep Learning*, 2018.

Wei, H., Xie, R., Cheng, H., Feng, L., An, B., and Li, Y. Mitigating neural network overconfidence with logit normalization. In *International Conference on Machine Learning (ICML)*, 2022.

Xu, K., Ren, T., Zhang, S., Feng, Y., and Xiong, C. Unsupervised out-of-domain detection via pre-trained transformers. In *Association for Computational Linguistics (ACL)*, 2021.

Xu, P., Ehinger, K. A., Zhang, Y., Finkelstein, A., Kulkarni, S. R., and Xiao, J. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015.

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and Xiao, J. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Zeng, Z., He, K., Yan, Y., Liu, Z., Wu, Y., Xu, H., Jiang, H., and Xu, W. Modeling discriminative representations for out-of-domain detection with supervised contrastive learning. *Association for Computational Linguistics (ACL)*, 2021.

Zhan, L.-M., Liang, H., Liu, B., Fan, L., Wu, X.-M., and Lam, A. Out-of-scope intent detection with self-supervision and discriminative training. *Association for Computational Linguistics (ACL)*, 2021.

Zhang, H., Li, A., Guo, J., and Guo, Y. Hybrid models for open set recognition. In *European Conference on Computer Vision (ECCV)*, 2020.

Zhang, W., Zhou, D., Li, L., and Gu, Q. Neural thompson sampling. In *International Conference on Learning Representations (ICLR)*, 2021.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2017.

Zhou, W., Liu, F., and Chen, M. Contrastive out-of-distribution detection for pretrained transformers. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

A. Discussion on the Choice of Auxiliary Dataset and Pool Size

Discussion on the use of auxiliary outlier dataset. The assumption of having access to outliers has been adopted in a large body of OOD detection work (*c.f.* baselines considered). The key premise of outlier exposure approaches is the availability of a large and diverse auxiliary dataset. Inheriting this setting, the precise challenge lies in how to use them in a more effective way that facilitates OOD detection. Thus, our goal is to propose a more **data-efficient** way to identify the most informative outliers which outperforms greedy approaches (NTOM) and random sampling (OE, CCU, SOFL). As shown in Figure 3, the performance gap between POEM and all baselines with outlier exposure plateaus around epoch 85. Training even longer might lead to overfitting with diminishing gains w.r.t. OOD detection performance. This further signifies the importance of effective outlier mining of POEM, even when computation is unconstrained. The sample efficiency is also theoretically justified (*c.f.* Theorem 6.1). When the auxiliary dataset is limited or very different from ID data, outlier exposure approaches *in general* may not be the most suitable; there exist other approaches to consider such as synthesizing outliers (Du et al., 2022b).

The choice of auxiliary outlier datasets. POEM is consistently competitive when using a different auxiliary outlier dataset. Following common practice, with TinyImages (Torralba et al., 2008) as the auxiliary dataset, the average FPR95 of POEM is **1.56**% on CIFAR-10, as shown in Table 4.

Table 4. The effect of training with TinyImages as the auxiliary outlier dataset. All methods are trained with the same DenseNet backbone. All values are percentages and are averaged over six OOD test datasets described in Section 4.1. Bold numbers indicate superior results. Each number is based on the average of 5 independent training runs.

Method (CIFAR-10 as $\mathcal{D}_{\text{in}}$)	FPR95 ↓
POEM	**1.56**
NTOM (Chen et al., 2021)	3.27
OE (Hendrycks et al., 2018)	4.33
Energy (w. $\mathcal{D}_{\text{aux}}$) (Liu et al., 2020)	3.38

Effects of the pool size. Our experiments suggest that POEM's performance is **insensitive** to the auxiliary pool size. The average AUROC slightly decreased from 99.40 to 99.29 on CIFAR-10 (FPR95 increased from 2.54 to 4.23), as a result of decreasing the pool size by half. POEM remains competitive compared to other baseline methods.

B. Discussion on Modeling the Outlier Class as the K+1-th Class

Ablation on the training loss. As discussed in Section 4.2 (POEM vs. NTOM), our method differs from NTOM (Chen et al., 2021) in terms of both training objectives as well as outlier mining strategy. To isolate the effect of the training objective, we conduct an ablation by using the same training objective as in NTOM, however replacing the confidence-based outlier mining with POEM. In particular, f_{outlier} is taken as the $K+1$-th component of the classification output $f_{K+1}(\mathbf{x})$. The performance comparison is shown in Table 5. Under the alternative training scheme (dubbed $K+1$ scheme), POEM outperforms NTOM on both CIFAR-10 and CIFAR-100. Our ablation also indicates the superiority of using the energy-regularized training objective, compared to using the $K+1$ training scheme. In particular, on CIFAR-100, the average FPR95 decreases by 1.97% with our method than using cross-entropy loss with $K+1$ classes.

Table 5. Comparison between NTOM and POEM under different schemes. POEM (K+1 scheme) refers to POEM with the same training and inference scheme as in NTOM. All values are percentages and are averaged over six natural OOD test datasets. Bold numbers indicate superior results.

$\mathcal{D}_{in}$	**Method**	**FPR95**↓	**AUROC**↑	**AUPR**↑
CIFAR-10	NTOM (q=0.125)	7.21	98.64	98.52
	NTOM (q=0)	4.38	99.08	99.24
	POEM (K+1 scheme)	2.81	99.28	99.41
	POEM (ours)	**2.54**	**99.40**	**99.50**
CIFAR-100	NTOM (q=0.125)	23.06	95.16	96.25
	NTOM (q=0)	19.96	96.29	97.06
	POEM (K+1 scheme)	17.11	96.87	97.73
	POEM (ours)	**15.14**	**97.79**	**98.31**

C. The Choice of Logit Values in BLR

In Thompson sampling, the key is to find data with the highest estimated boundary score, not that from target values. There is no easy way to get the ground-truth ID/OOD probability, so we adopt a fixed target value from $p = 0.95$ with noise for all ID/OOD data. But it does not prevent finding new OOD data closer to the estimated boundary: at each iteration, we select data with the estimated highest boundary score $\hat{G}(x)$ via Thompson sampling from the auxiliary set, which means selected query samples are closer to the estimated boundary than any OOD data in the training pool, *i.e.*, the newly selected OOD data is always "in the middle" of current ID and OOD data in the training pool. As we update the OOD training pool with the newly selected OOD data, the selected OOD becomes closer to ID, and the estimated boundary approaches the real one via Thompson sampling (see Figure 1 (c) - (e) for visualization).

D. Results on Individual Datasets

We provide reference results of POEM on each OOD dataset in Table 6, based on the publicly available checkpoints.

ID Dataset	OOD Dataset												Average	
	LSUN-crop		Places365		LSUN-resize		iSUN		Texture		SVHN			
	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑
CIFAR-10	13.36	97.52	1.47	99.40	0.00	100.00	0.00	100.00	0.12	99.90	0.37	99.63	2.55	99.41
CIFAR-100	49.85	92.87	5.92	98.29	0.00	100.00	0.00	100.00	1.10	99.57	15.52	97.36	12.06	98.01

Table 6. OOD detection results of POEM on each OOD dataset (based on DenseNet-101).

E. Theoretical Insights on Sample Complexity with High Boundary Scores

In this section, we provide details on how sampling with high boundary scores benefits the sample complexity. We begin with a brief review of some of the key definitions and notations.

Definitions and notations. Due to the representational power of deep neural networks, similar to prior works (Lee et al., 2018b; Sehwag et al., 2021), we assume the extracted feature approximately follow a Gaussian mixture model (GMM) with equal class priors: $\frac{1}{2}\mathcal{N}(\boldsymbol{\mu}, \sigma^2\boldsymbol{I}) + \frac{1}{2}\mathcal{N}(-\boldsymbol{\mu}, \sigma^2\boldsymbol{I}))$, where $\mathcal{P}_{\text{in}} = \mathcal{N}(\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$, and $\mathcal{P}_{\text{aux}} = \mathcal{N}(-\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$. The hypothesis class $\mathcal{H} = \{\text{sign}(\boldsymbol{\theta}^\top \mathbf{x}), \boldsymbol{\theta} \in \mathbb{R}^d\}$, where a classifier outputs 1 if it predicts $\mathbf{x} \sim \mathcal{P}_{\text{in}}$ and -1 if it predicts $\mathbf{x} \sim \mathcal{P}_{\text{aux}}$. The boundary score $G(\mathbf{x}) = -|f_{\text{outlier}}(\mathbf{x})|$ (the ground truth weight $\mathbf{w}^*$ is omitted for clarity). The probability of being an outlier is given by $p(\text{outlier}|\mathbf{x}) = \text{Sigmoid}(f_{\text{outlier}}(\mathbf{x}))$.

Lemma E.1. *Assume the selected data points $\mathbf{x} \sim \mathcal{P}_{aux}$ satisfy the following constraint of high boundary scores (on average):*

$$\frac{-\Sigma_{i=1}^n G(\mathbf{x}_i)}{n} \leq \epsilon \tag{6}$$

Then we have

$$\Sigma_{i=1}^n |2\mathbf{x}_i^\top \boldsymbol{\mu}| \leq n\sigma^2\epsilon \tag{7}$$

Proof of Lemma E.1. We first obtain the expression for $G(\mathbf{x})$ under the Gaussian mixture model described above.

By Bayes' rule, $p(\text{outlier}|\mathbf{x})$ can be expressed as:

$$\begin{aligned} p(\text{outlier}|\mathbf{x}) &= \frac{p(\mathbf{x}|\text{outlier})p(\text{outlier})}{p(\mathbf{x})} \\ &= \frac{\frac{1}{\sqrt{(2\pi\sigma^2)^d}} e^{-\frac{1}{2\sigma^2} d_{\text{outlier}}(\mathbf{x})}}{\frac{1}{\sqrt{(2\pi\sigma^2)^d}} e^{-\frac{1}{2\sigma^2} d_{\text{in}}(\mathbf{x})} + \frac{1}{\sqrt{(2\pi\sigma^2)^d}} e^{-\frac{1}{2\sigma^2} d_{\text{outlier}}(\mathbf{x})}} \\ &= \frac{1}{1 + e^{-\frac{1}{2\sigma^2}(d_{\text{outlier}}(\mathbf{x}) - d_{\text{in}}(\mathbf{x}))}}, \end{aligned} \tag{8}$$

where $d_{\text{in}}(\mathbf{x}) := (\mathbf{x} - \boldsymbol{\mu})^\top(\mathbf{x} - \boldsymbol{\mu})$ and $d_{\text{outlier}}(\mathbf{x}) := (\mathbf{x} + \boldsymbol{\mu})^\top(\mathbf{x} + \boldsymbol{\mu})$.

Note that $p(\text{outlier}|x) = \frac{1}{1+e^{-f_{\text{outlier}}(\mathbf{x})}}$. Thus, $f_{\text{outlier}}(\mathbf{x}) = \frac{1}{2\sigma^2}(d_{\text{outlier}}(\mathbf{x}) - d_{\text{in}}(\mathbf{x}))$.

Then we have:

$$G(\mathbf{x}) = -|f_{\text{outlier}}(\mathbf{x})| = -\frac{1}{2\sigma^2}|d_{\text{outlier}}(\mathbf{x}) - d_{\text{in}}(\mathbf{x})| = -\frac{1}{2\sigma^2}|(\mathbf{x}-\boldsymbol{\mu})^\top(\mathbf{x}-\boldsymbol{\mu}) - (\mathbf{x}+\boldsymbol{\mu})^\top(\mathbf{x}+\boldsymbol{\mu})| = -\frac{1}{2\sigma^2}|4\mathbf{x}^\top\boldsymbol{\mu}|. \tag{9}$$

Therefore, the boundary score constraint $\frac{-\Sigma_{i=1}^n G(\mathbf{x_i})}{n} \leq \epsilon$ is translated to:

$$\Sigma_{i=1}^n |2\mathbf{x}_i^\top \boldsymbol{\mu}| \leq n\sigma^2\epsilon \tag{10}$$

As $\max_{i\in n} |\mathbf{x}_i^\top\boldsymbol{\mu}| \leq \Sigma_{i=1}^n |\mathbf{x}_i^\top\boldsymbol{\mu}|$, given a fixed n, the selected samples can be seen as generated from $\mathcal{P}_{\text{aux}}$ with the constraint that all samples lie within the two hyperplanes in (10).

Sample Complexity Analysis. Now we show the benefit of such constraint in controlling the sample complexity. Assume the signal/noise ratio is large: $\frac{||\boldsymbol{\mu}||}{\sigma} = r_0 \gg 1$, the dimensionaility/sample size ratio is $r_1 = \frac{d}{n}$, and $\epsilon \leq 1$ is some constant.

Recall that we consider the classifier given by $\hat{\boldsymbol{\theta}}_{n,n'} = \frac{1}{n'+n}(\Sigma_{i=1}^{n'}\mathbf{x}_i' - \Sigma_{i=1}^n \mathbf{x}_i)$ where each $\mathbf{x}_i$ is drawn *i.i.d.* from $\mathcal{P}_{\text{aux}}$ and $\mathbf{x}_i'$ is drawn *i.i.d.* from $\mathcal{P}_{\text{in}}$. We can decompose $\hat{\boldsymbol{\theta}}_{n,n'}$ as:

$$\hat{\boldsymbol{\theta}}_{n,n'} = \boldsymbol{\mu} + \frac{n'}{n+n'}\boldsymbol{\theta}_1 + \frac{n}{n+n'}\boldsymbol{\theta}_2, \tag{11}$$

where $\boldsymbol{\theta}_1 = \frac{1}{n'}(\Sigma_{i=1}^{n'}\mathbf{x}_i') - \boldsymbol{\mu} \sim \mathcal{N}(0, \frac{\sigma^2}{n'}I)$ and $\boldsymbol{\theta}_2 = \frac{1}{n}(-\Sigma_{i=1}^n \mathbf{x}_i) - \boldsymbol{\mu}$.

For $\boldsymbol{\theta}_1$, we have that $||\boldsymbol{\theta}_1||^2 \sim \frac{\sigma^2}{n'}\chi_d^2$, and $\frac{\boldsymbol{\mu}^\top\boldsymbol{\theta}_1}{||\boldsymbol{\mu}||} \sim \mathcal{N}(0, \frac{\sigma^2}{n'})$. Then from standard concentration bounds:

$$\mathbb{P}(||\boldsymbol{\theta}_1||^2 \geq \frac{\sigma^2}{n'}(d + \frac{1}{\sigma})) \leq e^{-d/8\sigma^2},$$
$$\mathbb{P}(\frac{|\boldsymbol{\mu}^\top\boldsymbol{\theta}_1|}{||\boldsymbol{\mu}||} \geq (\sigma||\boldsymbol{\mu}||)^{1/2}) \leq 2e^{-\frac{1}{2}n'||\boldsymbol{\mu}||/\sigma}.$$

Since we have the equal prior probability for each class, we assume that $n = n'$. For $||\boldsymbol{\theta}_2||$, since all $\mathbf{x}_i$ is drawn *i.i.d.* from $\mathcal{P}_{\text{aux}}$ under the constraint (10), so the distribution of $\boldsymbol{\theta}_2$ can be seen as a truncated distribution of $\boldsymbol{\theta}_1$. Thus, we have $\mathbb{P}(||\boldsymbol{\theta}_2||^2 \geq \frac{\sigma^2}{n}(d + \frac{1}{\sigma})) \leq c\mathbb{P}(||\boldsymbol{\theta}_1||^2 \geq \frac{\sigma^2}{n}(d + \frac{1}{\sigma})) \leq ce^{-d/8\sigma^2}$, where c is some finite positive constant, n' is replaced with n in the above inequality.

The benefit of high boundary scores is brought by our constraint about $\boldsymbol{\theta}_2$: for $\boldsymbol{\mu}^\top\boldsymbol{\theta}_2$, unlike the analysis of $\boldsymbol{\mu}^\top\boldsymbol{\theta}_1$, we use results from Lemma E.1 to obtain that $\frac{1}{n}\Sigma_{i=1}^n|\mathbf{x}_i^\top\boldsymbol{\mu}| \leq \frac{\sigma^2\epsilon}{2}$ always holds. So $|\boldsymbol{\mu}^\top\boldsymbol{\theta}_2| \leq ||\boldsymbol{\mu}||^2 + \frac{\sigma^2\epsilon}{2}$.

Now we can develop a lower bound for $\frac{\boldsymbol{\mu}^\top\hat{\boldsymbol{\theta}}_{n,n'}}{\sigma||\hat{\boldsymbol{\theta}}_{n,n'}||}$. Let

$$||\boldsymbol{\theta}_1||^2 \leq \frac{\sigma^2}{n}(d + \frac{1}{\sigma}), ||\boldsymbol{\theta}_2||^2 \leq \frac{\sigma^2}{n}(d + \frac{1}{\sigma}), \frac{|\boldsymbol{\mu}^\top\boldsymbol{\theta}_1|}{||\boldsymbol{\mu}||} \leq (\sigma||\boldsymbol{\mu}||)^{1/2} \tag{12}$$

hold simultaneously, we have $||\hat{\boldsymbol{\theta}}_{n,n'}||^2 \leq \frac{\sigma^2}{n}(d + \frac{1}{\sigma}) + ||\boldsymbol{\mu}||^2$, and $|\boldsymbol{\mu}^\top\hat{\boldsymbol{\theta}}_{n,n'}| \geq \frac{1}{2}(||\boldsymbol{\mu}||^2 - \sigma^{1/2}||\boldsymbol{\mu}||^{3/2} - \frac{\sigma^2\epsilon}{2})$.

Recall that $\frac{||\boldsymbol{\mu}||}{\sigma} = r \gg 1$, $\epsilon \leq 1$, then it satisfies that $||\boldsymbol{\mu}||^2 - \sigma^{1/2}||\boldsymbol{\mu}||^{3/2} - \frac{\sigma^2\epsilon}{2} = \sigma^2(r^2 - r^{3/2} - \frac{1}{2}) > 0$.

Thus via union bound, we have that

$$\frac{\boldsymbol{\mu}^\top\hat{\boldsymbol{\theta}}_{n,n'}}{\sigma||\hat{\boldsymbol{\theta}}_{n,n'}||} \geq \frac{||\boldsymbol{\mu}||^2 - \sigma^{1/2}||\boldsymbol{\mu}||^{3/2} - \frac{\sigma^2\epsilon}{2}}{2\sqrt{\frac{\sigma^2}{n}(d + \frac{1}{\sigma}) + ||\boldsymbol{\mu}||^2}} \tag{13}$$

with probability at least $1 - (1+c)e^{-d/8\sigma^2} - 2e^{-\frac{1}{2}n||\boldsymbol{\mu}||/\sigma} = 1 - (1+c)e^{-r_1 n/8\sigma^2} - 2e^{-\frac{1}{2}n||\boldsymbol{\mu}||/\sigma}$.

Interpretations. The above results suggests that when ϵ decreases, the lower bound of $\frac{\boldsymbol{\mu}^\top \hat{\boldsymbol{\theta}}_{n,n'}}{\sigma||\hat{\boldsymbol{\theta}}_{n,n'}||}$ will increase. Recall that $\text{FNR}(\hat{\boldsymbol{\theta}}_{n,n'}) = \text{FPR}(\hat{\boldsymbol{\theta}}_{n,n'}) = erf(\frac{\boldsymbol{\mu}^\top \hat{\boldsymbol{\theta}}_{n,n'}}{\sigma||\hat{\boldsymbol{\theta}}_{n,n'}||})$, so the upper bound of $\text{FNR}(\hat{\boldsymbol{\theta}}_{n,n'})$ and $\text{FPR}(\hat{\boldsymbol{\theta}}_{n,n'})$ will decrease accordingly, which shows the benefit of outlier mining with high boundary scores.

F. Extension: Towards a More General Data Model for OOD Detection

In this section, we extend the data model in Appendix E to a more general case where both the auxiliary and in-distribution data points are generated from mixtures of Gaussian distributions. Moreover, we consider a broader test OOD distribution $\mathcal{Q}_{\boldsymbol{v}}$ beyond $\mathcal{P}_{\text{aux}}$. We aim to provide more insights into the benefits of sampling with high boundary scores.

Specifically, the auxiliary data is generated with the following procedure: first draw a scalar g from a uniform distribution with support $\{g : |g| \leq \frac{1}{4}||\boldsymbol{\mu}||_2\}$, then draw $\mathbf{x}$ from $\mathcal{N}((-1+g)\boldsymbol{\mu}, \sigma^2\boldsymbol{I}))$. In addition, the in-distribution data points are generated as follows: first draw a scalar s from a uniform distribution with support $\{s : |s| \leq \frac{1}{4}||\boldsymbol{\mu}||_2\}$, then draw $\mathbf{x}$ from $\mathcal{N}((1+s)\boldsymbol{\mu}, \sigma^2\boldsymbol{I}))$. For the test OOD distribution, consider any $\mathcal{Q}_{\boldsymbol{v}:\mathcal{N}(-\boldsymbol{\mu}+\boldsymbol{v},\sigma^2\boldsymbol{I}))}$, where $\boldsymbol{v} \in \mathbb{R}^d$ and $||\boldsymbol{v}||_2 \leq \frac{1}{4}||\boldsymbol{\mu}||_2$.

We can express the FPR as follows:

$$\begin{aligned}
\text{FPR}(\boldsymbol{\theta}) &= \mathbb{P}_{\mathbf{x}\sim\mathcal{Q}_{\boldsymbol{v}}}(\boldsymbol{\theta}^\top \mathbf{x} \geq 0) \\
&= \mathbb{P}(\mathcal{N}((-\boldsymbol{\mu}+\boldsymbol{v})^\top\boldsymbol{\theta}, \sigma||\boldsymbol{\theta}||_2) \geq 0) \\
&= erf(\frac{(\boldsymbol{\mu}-\boldsymbol{v})^\top\boldsymbol{\theta}}{\sigma||\boldsymbol{\theta}||_2}) \\
&\leq erf(\frac{\boldsymbol{\mu}^\top\boldsymbol{\theta}}{\sigma||\boldsymbol{\theta}||_2} - \frac{||\boldsymbol{\mu}||_2}{4\sigma}).
\end{aligned} \tag{14}$$

Consider any $\mathcal{P}_{\text{in}} = \mathcal{N}((1+s)\boldsymbol{\mu}, \sigma^2\boldsymbol{I}))$ where $|s| \leq \frac{1}{4}||\boldsymbol{\mu}||_2$ as the ID test set where we calculate the FNR. Similarly, we have $\text{FNR}(\boldsymbol{\theta}) \leq erf(\frac{\boldsymbol{\mu}^\top\boldsymbol{\theta}}{\sigma||\boldsymbol{\theta}||_2} - \frac{||\boldsymbol{\mu}||_2}{4\sigma})$.

However, under such general data model, unlike in Appendix E, there is no clean solution for $G(\mathbf{x})$ as equation (9): the probability density functions for $\mathcal{P}_{\text{in}}$ and $\mathcal{P}_{\text{aux}}$ require integrating Gaussian densities on w.r.t. the probability measure of s and g, respectively. Thus when calculating the inverse of the Sigmoid function in $p(\text{outlier}|x) = \frac{1}{1+e^{-f_{\text{outlier}}(\mathbf{x})}}$ using the new probabilistic model, we cannot directly disentangle $\mathbf{x}^\top\mu$ from the expression of f_{outlier} as equation (9). Nevertheless, we can expect that the decision boundary for $\mathcal{P}_{\text{in}}$ and $\mathcal{P}_{\text{aux}}$ is still a hyperplane under this setting. Recall that $p(\text{outlier}|\mathbf{x}) = \frac{1}{1+e^{-f_{\text{outlier}}(\mathbf{x})}}$, which indicates that the property of data points with high boundary scores ($p(\text{outlier}|\mathbf{x}) \approx 0.5$) can generally bound the distances between selected data and the decision hyperplane, so there exists two hyperplanes that can bound those auxiliary points we select with constraint (6). As a result, we will have a revised version of Lemma E.1 with a different bound on the right hand side of equation (10) which will still be positively correlated with ϵ. Using the subgaussian property of $\boldsymbol{v}$ and following similar steps in Appendix E, we can expect similar sample complexity results with a different (but still negative) weight on ϵ compared to the results in Theorem 6.1 which still shed light on the benefits of sampling with high boundary scores.

G. Hardware and Software

We run all the experiments on NVIDIA GeForce RTX-2080Ti GPU. Our implementations are based on Ubuntu Linux 20.04 with Python 3.8.