Human-in-the-Loop Out-of-Distribution Detection with False Positive Rate Control

A PREPRINT

Harit Vishwakarma
University of Wisconsin-Madison, WI
hvishwakarma@cs.wisc.edu

Heguang Lin
University of Pennsylvania, PA
hglin@seas.upenn.edu

Ramya Korlakai Vinayak
University of Wisconsin-Madison, WI
ramya@ece.wisc.edu

ABSTRACT

Robustness to out-of-distribution (OOD) samples is crucial for safely deploying machine learning models in the open world. Recent works have focused on designing scoring functions to quantify OOD uncertainty. Setting appropriate thresholds for these scoring functions for OOD detection is challenging as OOD samples are often unavailable up front. Typically, thresholds are set to achieve a desired true positive rate (TPR), e.g., 95% TPR. However, this can lead to very high false positive rates (FPR), ranging from 60 to 96%, as observed in the Open-OOD benchmark. In safety-critical real-life applications, e.g., medical diagnosis, controlling the FPR is essential when dealing with various OOD samples dynamically. To address these challenges, we propose a mathematically grounded OOD detection framework that leverages expert feedback to *safely* update the threshold on the fly. We provide theoretical results showing that it is guaranteed to meet the FPR constraint at all times while minimizing the use of human feedback. Another key feature of our framework is that it can work with any scoring function for OOD uncertainty quantification. Empirical evaluation of our system on synthetic and benchmark OOD datasets shows that our method can maintain FPR at most 5% while maximizing TPR.

1 Introduction

Deploying machine learning (ML) models in the open world makes them subject to out-of-distribution (OOD) inputs: an ML model trained to classify on K classes also encounters points that do not belong to any of the classes in the training data. Modern ML models, in particular deep neural networks, can fail silently with high confidence on such OOD points (1; 2). Such failures can have serious consequences in high-risk applications, e.g., medical diagnosis and autonomous driving. Safe deployment of ML models in an open world setting needs mechanisms that ensure robustness to OOD inputs. The importance of this problem has led to the development of many methods for OOD detection (3; 4; 5; 6) which aim to produce a *score* that can be used to decide on OOD vs in-distribution (ID) for a given point. For a detailed survey of literature in the area of generalized OOD detection, see (7).

ID data is usually plentiful, but we do not get to see different kinds of OOD samples before deployment. Consequently, many works in OOD detection are largely limited to *static settings* where the ID data is used to set a threshold on the scores used for detection (3; 5; 6). In these scenarios, this is usually done by setting a threshold that achieves a certain level of true positive rate (TPR), such as 95%. However, this can lead to a very high false positive rate (FPR), e.g., ranging between 60% to 96% as observed in the Open-OOD benchmark (8). Furthermore, even if the ID data distribution remains the same after deployment, the OOD data could vary, resulting in highly fluctuating FPR. Thus, having a small, fixed amount of OOD data collected a priori to validate the FPR at a given threshold would not help in guaranteeing the desired FPR.

In safety critical applications, the consequences of classifying an OOD point as ID (false positive) are more catastrophic than classifying an ID point as OOD (false negative). For example, in the medical diagnosis of brain scans, when the system is in doubt it is better to classify a scan as OOD and defer the decision to human experts rather than for the ML model to give it an ID label i.e., predicting an in-distribution disease or classifying it as a normal scan. Therefore, safely using ML models in such applications requires systems guaranteeing that the FPR is below a certain acceptable rate, e.g., FPR below 5%.

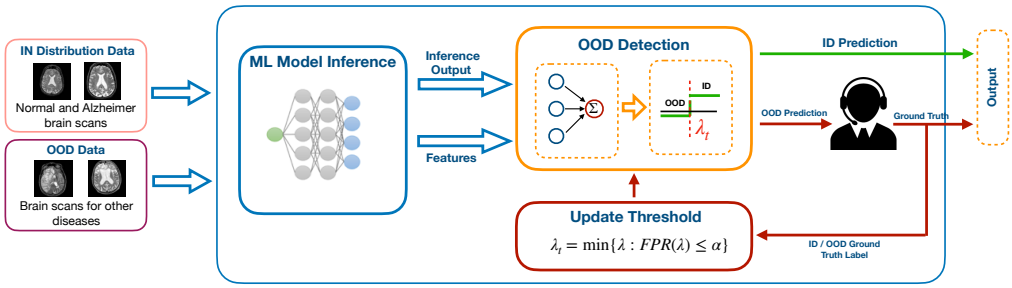


Figure 1: Illustration of OOD detection with human-in-the-loop with FPR control. In this example, the ID data is of brain scans of normal people and those with Alzheimer's disease. The OOD data could be anything other than these, e.g. brain scans of patients with some other diseases.

Furthermore, it is difficult to anticipate or collect the exact type of OOD data that the system can encounter during deployment. Thus it is crucial that such systems adapt to the OOD data while controlling the FPR. Motivated by these challenges we pose the following goal for safe OOD detection.

Goal: Develop a human-in-the-loop out-of-distribution detection system that has guaranteed false positive rate control while minimizing the amount of human intervention needed.

Our Contributions: Toward this goal, we make the following contributions:

1. **Human-in-the-loop OOD detection framework**: We propose a novel mathematically grounded framework that incorporates expert human feedback to safely update the OOD detection threshold, ensuring robustness to variations in OOD data encountered after deployment. Our framework can be used with any scoring function.
2. **Guaranteed FPR control**: For stationary settings, we provide theoretical guarantees for our framework on controlling FPR at the desired level at all times and also provide a bound on the time taken to reach a given level of optimality. Using the insight from this analysis, we also propose an approach for settings with change points that reduces the duration of violation of FPR control.
3. **Empirical validation on benchmark datasets**: We evaluate our framework through extensive simulations both in stationary and distribution shift settings. Through experiments on benchmark OOD datasets in image classification tasks with various scoring functions, we demonstrate the effectiveness of our proposed framework.

We emphasize that our aim is to develop a framework that can use any scoring function and safely adapt the threshold on the fly to enable the safe deployment of ML models. Therefore, our work is complementary to works that develop scoring functions for OOD detection.

2 Human-in-the-Loop OOD Detection

We propose a human-in-the-loop OOD detection system (Figure 1) that can work with any ML inference model and scoring function for OOD detection. We begin by describing the problem setting and then discuss each component of our proposed system in detail. See Algorithm 1 for step-by-step pseudocode.

2.1 Problem Setting

Data stream: Let $\mathcal{X} \subseteq \mathbb{R}^d$ denote the feature space and $\mathcal{Y} = \{-1, 1\}$ denote the label space for OOD detection with "1" denoting ID and "-1" denoting OOD. Let the distribution of ID and OOD data be denoted by $\mathcal{D}_{id}$ and $\mathcal{D}_{ood}$ respectively. Let $x_t \in \mathcal{X}$ denote the sample received at the time t. Let $y_t \in \{-1, 1\}$ denote the true label for x_t with respect to ID or OOD classification. We assume x_t are independent and drawn according to the following mixture model, $x_t \sim (1-\gamma)\,\mathcal{D}_{id} + \gamma\,\mathcal{D}_{ood}$, where $\gamma \in (0, 1)$ is the fraction of OOD points in the mixture. Note that $\mathcal{D}_{id}$, $\mathcal{D}_{ood}$ and γ are *unknown*.

Scoring function: After receiving data point x_t, the system uses a given scoring function, $g : \mathcal{X} \mapsto \mathcal{S} \subseteq \mathbb{R}$, to compute a score quantifying the uncertainty of the point being ID or OOD. Our system is designed to work with any scoring function based OOD uncertainty quantification. Let $s_t = g(x_t)$ denote the score computed for point x_t. To be consistent across various scoring functions, let a higher score indicate ID and a lower score indicate OOD points. After computing score s_t the system needs to decide whether x_t is OOD or ID, it is done using a threshold-based classifier $h_\lambda : \mathbb{R} \mapsto \{-1, +1\}$ parameterized with $\lambda \in \Lambda \subseteq \mathbb{R}$: $h_\lambda(g(x)) = \text{sign}(g(x) - \lambda)$. Here we assume $\Lambda = (\Lambda_{\min}, \Lambda_{\max})$.

FPR and TPR: The population level FPR and TPR for any $\lambda \in \Lambda$ are defined as follows,

$$\text{FPR}(\lambda) = \mathbb{E}_{x\sim\mathcal{D}_{ood}}[\mathbf{1}\{g(x) > \lambda\}] \quad \text{and}$$
$$\text{TPR}(\lambda) = \mathbb{E}_{x\sim\mathcal{D}_{id}}[\mathbf{1}\{g(x) > \lambda\}].$$

Note that the cumulative distribution function (CDF) of $\mathcal{D}_{ood}$, $\text{CDF}_{\mathcal{D}_{ood}}(\lambda) = \mathbb{E}_{x\sim\mathcal{D}_{ood}}[\mathbf{1}\{g(x) \leq \lambda\}]$. Therefore, $\text{FPR}(\lambda) = 1 - \text{CDF}_{\mathcal{D}_{ood}}(\lambda)$. Similarly, $\text{TPR}(\lambda) = 1 - \text{CDF}_{\mathcal{D}_{id}}(\lambda)$. Since the CDF of any distribution is a monotonic function, both the FPR and TPR are monotonic in λ.

Expert human feedback: Our goal is to tackle critical applications where a human expert examines the samples that are declared as OOD (instead of the ML model making automatic predictions on them). The feedback obtained from the human expert can be used to safely update the OOD detection threshold at each time step, λ_t, so that the FPR is maintained below the desired rate of α. One can trivially control FPR by setting $\lambda_t = \Lambda_{\max}$, i.e., always getting human feedback. This would of course be too expensive and defeat the purpose of using an ML model in the first place. Therefore, in addition to controlling the FPR, we aim to minimize the human feedback solicited by the system. In an ideal system, the points declared as ID are directly classified by the ML model, and only the points declared as OOD are examined by human expert. Thus, minimizing human feedback is equivalent to maximizing the TPR. This can be done by setting the threshold as, $\lambda_t := \arg\max_\lambda \text{TPR}(\lambda)$ subject to $\text{FPR}(\lambda) \leq \alpha$. Since the TPR is monotonic in λ, we can re-write this further as follows,

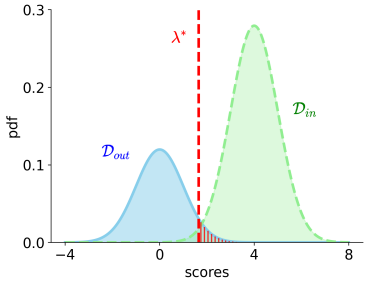


Figure 2: $\text{FPR}(\lambda^\star) = \alpha \longleftrightarrow \text{CDF}_{\mathcal{D}_{\text{out}}}(\lambda^\star) = 1 - \alpha$. Optimal $\lambda^\star$ for the optimization problem (P1) with $\alpha = 0.05$ and $x_t \overset{i.i.d}{\sim} 0.7\,\mathcal{D}_{in} + 0.3\,\mathcal{D}_{out}$, where $\mathcal{D}_{in}$ is $\mathcal{N}(4,1)$ and $\mathcal{D}_{out}$ is $\mathcal{N}(0,1)$.

$$\lambda_t^\star := \arg\min_{\lambda\in\Lambda} \lambda, \quad \text{s.t.} \quad \text{FPR}(\lambda) \leq \alpha. \tag{P1}$$

The optimal threshold, denoted by $\lambda^\star$, is the smallest λ such that $\text{FPR}(\lambda^\star) = \alpha = 1 - \text{CDF}_{\mathcal{D}_{out}}(\lambda^\star)$ (see Figure 2). When the distribution of the OOD points, $\mathcal{D}_{ood}$, is not changing, then setting $\lambda_t^\star = \lambda^\star$ for all t would be the optimal solution. Note that, changing the mixture ratio γ, or the distribution of the ID points $\mathcal{D}_{id}$ does not affect the value of the optimal threshold. As we do not have access to the true FPR and TPR values, we cannot solve the optimization problem (P1). Instead, we have to estimate the threshold at time t, denoted by $\hat{\lambda}_t$, using the observations until time t.

2.2 Adaptive Threshold Estimation

Ideally, we want to avoid human feedback for points that are determined as ID by the system, i.e., with a score greater than $\hat{\lambda}_t$. However, in order to have an unbiased estimate of the FPR and to detect potential changes in the distribution of OOD samples and therefore change in true FPR, we obtain human feedback with a small probability p for points predicted as ID by the system. We refer to this as *importance sampling*.

FPR estimation and adapting the threshold: At each time t, we observe $x_t \overset{i.i.d}{\sim} (1-\gamma)\mathcal{D}_{id} + \gamma\mathcal{D}_{ood}$, and $s_t = g(x_t)$ is the corresponding score. If $s_t \leq \hat{\lambda}_{t-1}$, where $\hat{\lambda}_{t-1}$ is the threshold determined in at time $t-1$, then it is considered an OOD point and hence gets a human label for it and we get to know whether it is in fact OOD or ID. If $s_t > \hat{\lambda}_{t-1}$, then x_t is considered an ID point and hence gets a human label only with probability p. So, we get to know whether it is truly ID or not with probability p. Now we have to update the threshold, $\hat{\lambda}_t$, such that the $\text{FPR}(\hat{\lambda}_t) \leq \alpha$, by finding the minimum λ that satisfies this constraint in order to maximize the TPR.

Our approach is based on using the feedback on the samples that are examined by human experts till time t to construct an unbiased estimator of $\text{FPR}(\lambda)$ (see Equation 3). We also construct an upper confidence interval for the estimated $\text{FPR}(\lambda)$ that is valid at all thresholds $\lambda \in \Lambda$ and for all times simultaneously with high probability (see Equation 2). This enables us to optimize for λ such that the upper bound on the true $\text{FPR}(\lambda)$ is at most α at each time t and thus safely update the threshold λ_t. Let $S_t^{(o)} = \{s_1^{(o)}, \ldots s_{N_t^{(o)}}^{(o)}\}$ denote the scores of the points that have been truly identified as OOD from human feedback so far, and $I_t^{(o)}$ be the corresponding time points. We estimate the FPR as follows,

$$\widehat{\text{FPR}}(\lambda, t) := \frac{1}{N_t^{(o)}} \sum_{u\in I_t^{(o)}} Z_u(\lambda), \tag{1}$$

$$Z_u(\lambda) := \begin{cases} \mathbf{1}(s_u^{(o)} > \lambda), & \text{if } s_u^{(o)} \le \hat{\lambda}_{u-1} \\ \frac{1}{p}\mathbf{1}(s_u^{(o)} > \lambda), & \text{w.p. } p \text{ if } s_u^{(o)} > \hat{\lambda}_{u-1} \\ 0, & \text{w.p. } 1-p \text{ if } s_u^{(o)} > \hat{\lambda}_{u-1} \end{cases}.$$

We show that our estimator for the FPR is unbiased (the proof is deferred to the Appendix).

Lemma 1. *Let $p > 0$, $\widehat{FPR}(\lambda, t)$ as defined in eq.* (3) *is an unbiased estimate of the true FPR(λ), i.e., $\mathbb{E}[\widehat{FPR}(\lambda, t)] =$ FPR(λ).*

Finding threshold using a UCB on FPR: We propose using our estimated FPR with an upper confidence bound (UCB), which we will describe soon, to obtain the following optimization problem (P2),

$$\hat{\lambda}_t := \arg\min_{\lambda \in \Lambda} \lambda \ \text{ s.t. } \ \widehat{\text{FPR}}(\lambda, t) + \psi(t, \delta) \le \alpha, \quad \text{(P2)}$$

where the term $\psi(t, \delta)$ is a time-varying upper confidence which is simultaneously valid for all λ for all time with probability at least $1 - \delta$ for any given $\delta \in (0, 1)$. The minimization problem can be solved in many ways. We use a binary search procedure where we search over a grid on $(\Lambda_{\min}, \Lambda_{\max})$ with grid-size ν. The procedure searches for a smallest λ such that $\widehat{\text{FPR}}(\lambda, t) + \psi(t, \delta) \le \alpha$. It uses eq. (3) to compute the empirical FPR at various thresholds and the confidence interval $\psi(t, \delta)$ given in eq. (2). Details of the binary search procedure are in the Appendix.

Algorithm 1 Human in the Loop OOD Detection

Input: FPR threshold α , sampling probability $p \in (0, 1)$, Scoring function $g : \mathcal{X} \mapsto \mathbb{R}$,

$S_0 = \Phi$, $\hat{\lambda}_0 = \infty$
for $t = 1, 2, \ldots$ **do**
  Receive data point x_t ; $s_t = g(x_t)$
  if $s_t \le \hat{\lambda}_{t-1}$ **then** $l_t = 1$
  else $l_t \sim$ `Bernoulli`(p)
  end if
  if $l_t = 1$ **then**
    $y_t =$ `GetExpertLabel`(x_t)
    $S_t = S_{t-1} \cup \{(s_t, y_t)\}$
  end if
  $\hat{\lambda}_t := \arg\min_{\lambda \in \Lambda} \lambda$ s.t. $\widehat{\text{FPR}}(\lambda, t) + \psi(t, \delta) \le \alpha$
  if $l_t = 1$ **then** Output y_t
  else Output $\hat{y}_t = \text{sign}(s_t - \hat{\lambda}_t)$
  end if
end for

Upper confidence bound (UCB): Our algorithm hinges on having confidence intervals on the FPR that are valid for all thresholds and for all times simultaneously. To construct such bounds, we use the confidence bounds based on Law of Iterated Logarithm(LIL) of (9). We note that at each time step t, whether the sample x_t gets human feedback or not depends on the previous threshold $\hat{\lambda}_{t-1}$ which is a function of data up to time $t-1$ and the importance sampling. Therefore, *the samples used to estimate the FPR are dependent* which prevents direct application of known results that are developed for i.i.d. samples (10). We build upon the LIL bounds for martingales (11) and derive a confidence interval bound that is valid in our setting, which is given by the following equation,

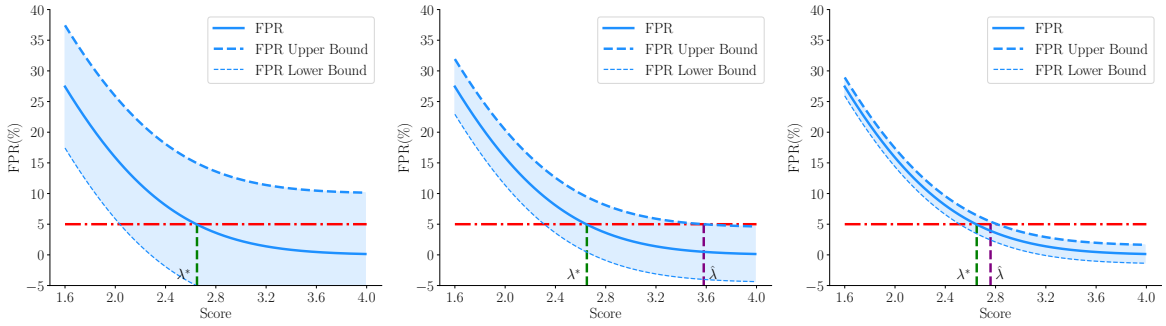
$$\psi(t, \delta) := \sqrt{\frac{3c_t}{N_t^{(o)}} \left[2 \log\log\left(\frac{3c_t N_t^{(o)}}{2}\right) + \log\left(\frac{2L}{\delta}\right) \right]}, \quad (2)$$

where $c_t = 1 - \beta_t + \frac{\beta_t}{p^2}$, $\beta_t = \frac{N_t^{(o,p)}}{N_t^{(o)}}$ and $N_t^{(o,p)}$ is the number of points sampled using importance sampling until time t and $\nu \in (0, 1)$ is a discretization parameter set by the user, $L = (\Lambda_{\max} - \Lambda_{\min})/\nu$.

3 Theoretical Guarantees

We would like three types of provable properties for our approach. First, it must have guaranteed FPR control—the safety property we set out to ensure. In addition, we would like to show a bound on the number of streamed observations (i.e., the time) taken to reach a point where every point does not need human feedback to ensure safety. Finally, we wish to have some notion of optimality, and a bound on the number of observations before it is reached. In this section, we provide a result that provides all of these properties under the following assumptions: (i) we are in the stationary setting, i.e., the distributions do not change over time, and (ii) the score distributions for the ID and OOD samples have sub-Gaussian tails.

To quantify how close to the optimal operating point the system is at any given time, we define the following notion of η-optimality.



(a) No feasible solution, in the beginning (b) Feasible solution, after sometime (c) Near optimal solution, eventually

Figure 3: Illustration of the confidence interval defined in eq. (2) on FPR and their effect on threshold estimation. As the system receives more OOD samples the confidence intervals will shrink and lead to better thresholds safely ($\hat{\lambda}_t \geq \lambda^\star$).

Definition 1. *(η-optimality) For any $\eta > FPR(\lambda^*) - \lim_{\epsilon \to 0^+} FPR(\lambda^* + \epsilon)$, the system is said to be operating in the η-optimal regime after some time point T_η, if $FPR(\lambda^*) - FPR(\hat{\lambda}_t) \leq \eta$ for all $t \geq T_\eta$.*

Using the estimated FPR in eq. (3) and the anytime valid confidence intervals on the FPR at all thresholds we obtained in eq. (2), we provide the following guarantees for Algorithm 1.

Theorem 1. *Let $\alpha, \delta, p, \gamma \in (0,1)$. Let $x_t \overset{i.i.d}{\sim} (1-\gamma)\mathcal{D}_{id} + \gamma \mathcal{D}_{ood}$ and let $c_t = 1 - \beta_t + \frac{\beta_t}{p^2}$, $\beta_t = \frac{N_t^{(o,p)}}{N_t^{(o)}}$ where $N_t^{(o,p)}$ is the number of OOD points sampled using importance sampling until time t and $N_t^{(o)}$ is the total number of OOD points observed till time t. Let $n_0 = \min\{u : c_u N_u^{(o)} \geq 173 \log(\frac{8}{\delta})\}$ and t_0 be such that $N_{t_0}^{(o)} \geq n_0$. If Algorithm 1 uses the optimization problem* (P2) *to find the thresholds with the upper confidence term $\psi(N_t^{(o)}, \delta/2)$ given by eq.* (2)*, then there exist constants $C_1, C_2, C_3 > 0$ such that with probability at least $1-\delta$,*

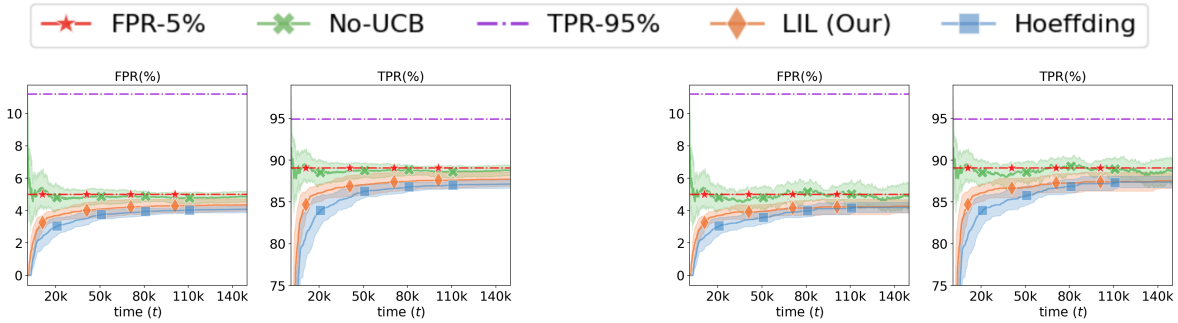
1. ***Controlled FPR:** For all $t \geq t_0$, $FPR(\hat{\lambda}_t) \leq \alpha$.*
2. ***Time to reach feasibility:** The algorithm will find a feasible threshold, $\hat{\lambda}_t$ such that $\widehat{FPR}(\hat{\lambda}_t) + \psi(N_t^{(o)}) \leq \alpha$, for all $t \geq \max(t_0, T_f)$, where, $T_f = \frac{2C_1}{\gamma \alpha^2} \log\left(\frac{4C_2}{\delta} \log(\frac{C_3}{\alpha})\right) + \frac{1}{\gamma^2} \log(\frac{4}{\delta})$.*
3. ***Time to reach $\eta-$optimality:** For all $t \geq \max(t_0, T_{\eta\text{-}opt})$, $\hat{\lambda}_t$ satisfy the η-optimality condition in definition 1, when $\widehat{FPR}(\hat{\lambda}_{T_{\eta\text{-}opt}}) \in [\alpha - \eta/2, \alpha]$ and $T_{\eta\text{-}opt} = \frac{8C_1}{\gamma \eta^2} \log\left(\frac{4C_2}{\delta} \log(\frac{2C_3}{\eta})\right) + \frac{1}{\gamma^2} \log(\frac{4}{\delta})$.*

We now discuss each property in detail, along with their implications.

Controlled false positive rate: We design our framework with the goal of safely updating the threshold. Our method guarantees that $\lambda_t \geq \lambda^\star$ at all times, i.e., we approach the optimal $\lambda^\star$ from above and therefore we never violate the FPR constraint. This property is crucial in applications where accurately controlling the FPR is essential.
Time to reach feasibility: Our algorithm begins with setting $\lambda_0 = \Lambda_{\max}$, and therefore obtaining human feedback on all the points until the time point when we can find a threshold that enables us to safely declare scores above it as ID (Fig 3b). Our analysis provides an upper bound on the time taken by the algorithm to find such a safe, $\hat{\lambda}_t < \Lambda_{\max}$ such that, $\text{FPR}(\hat{\lambda}_t) \leq \alpha$. We call this time as *time to reach feasibility*, T_f, and it is the time step at which a sufficient number of observations $N_{T_f}^{(o)}$ is obtained so that the confidence interval $\psi(T_f, \delta/2) \leq \alpha$. It is inversely proportional to the level α and the fraction of OOD samples γ.

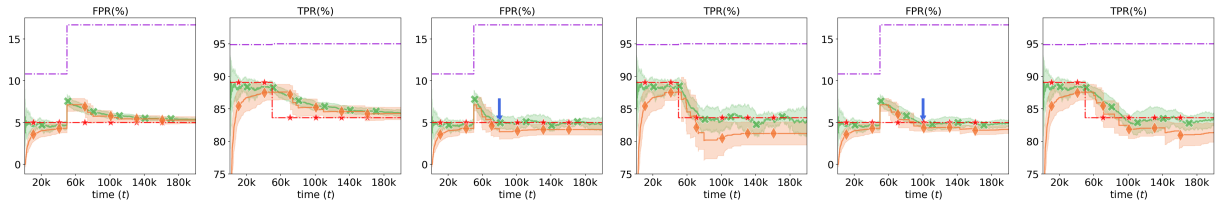
Time to reach η-optimality: As the time proceeds and the confidence intervals around the estimated FPR at different thresholds start to get smaller, the estimated safe threshold $\hat{\lambda}_t$ starts to approach $\lambda^\star$ (Fig 3c). If the estimated FPR at time step $T_{\eta\text{-opt}}$, denoted as $\widehat{\text{FPR}}(\hat{\lambda}_{T_{\eta\text{-opt}}})$, is within the range $[\alpha - \eta/2, \alpha]$ and the confidence interval $\psi(T_{\eta\text{-opt}}, \delta/2) \leq \eta/2$, then for all time points after $T_{\eta\text{-opt}}$ the algorithm will find a $\hat{\lambda}_t$ that satisfies the η-Optimality condition. In this regime, the algorithm operates in a state where the difference between the FPR at the true optimal threshold, $\text{FPR}(\lambda^*) = \alpha$, and the FPR at the estimated threshold $\text{FPR}(\hat{\lambda}_t)$, is bounded by η. Our analysis provides a bound on the time $T_{\eta\text{-opt}}$ which is the time point when the number of acquired OOD samples $N_{T_{\eta\text{-opt}}}^{(o)}$ becomes at least $\frac{4C_1}{\eta^2} \log\left(\frac{2C_2}{\delta} \log(\frac{2C_3}{\eta})\right)$. It is inversely proportional to the closeness to optimality η and the fraction of OOD samples γ.

The details of the proof are available in the appendix.



(a) No window, no distribution shift. (b) 5K window, no distribution shift

Figure 4: Results on the synthetic data with stationary distributions, $\gamma = 0.2$ and using no window. Each method is repeated 10 times. The mean and standard deviation are shown.



(a) Distribution shift, no window. (b) Distribution shift, 5k window. (c) Distribution shift, 10k window.

Figure 5: Effect of using various window sizes in synthetic data experiments. The distribution shift starts at $t = 50k$. The arrow indicates the time at which the mean FPR + std. deviation over 10 runs goes below 5% for LIL method.

4 Empirical Evaluation

We evaluate our method to verify the following claims:

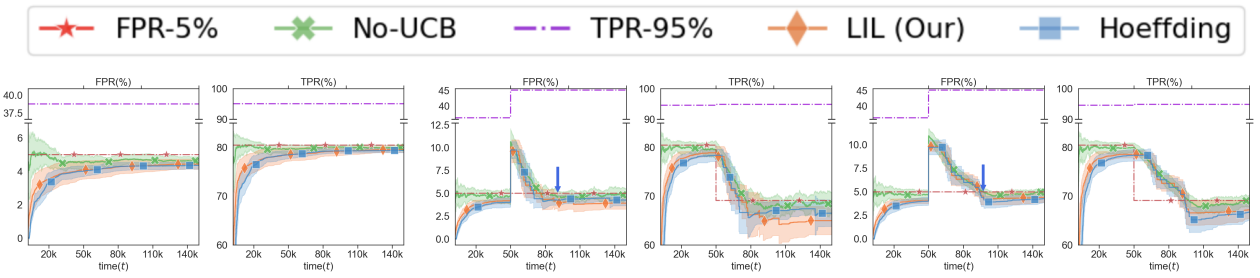
C1: Compared to non-adaptive baselines, our approach achieves lower FPR while maximizing the TPR.
C2: In the stationary setting, our adaptive method based on the LIL upper confidence bound satisfies the FPR constraint at all times and produces high TPR.
C3: The proposed framework is compatible with any OOD scoring functions.
C4: Our method continues to work even in distribution shift settings with a simple adaption using windowed approach described in Section 4.2.

Baselines: We compare our method against the non-adaptive baseline popularly used for OOD detection. This non-adaptive method (**TPR-95**) finds a threshold achieving 95% TPR using the ID data and uses it at all times. For our adaptive method, we consider three choices of confidence intervals **i) No-UCB**: does not use any confidence intervals, **ii) LIL**: Uses confidence interval from eq. (LIL-Heuristic), and **iii) Hoeffding**: uses the confidence intervals from Hoeffding's inequality (**?**). The confidence intervals from Hoeffding inequality are not valid simultaneously for all times but are a reasonable choice for a practitioner.

$$\tilde{\psi}(t,\delta) = C_1 \sqrt{\frac{c_t}{N_t^{(o)}} \left(\log\log\left(C_2 c_t N_t^{(o)}\right) + \log\left(\frac{C_3}{\delta}\right) \right)}. \qquad \text{(LIL-Heuristic)}$$

The theoretical LIL bound in eq. (2) has constants that can be pessimistic in practice. We get around this by using a LIL-Heuristic bound which has the same form as in eq. (2) but with different constants. We consider the form in eq. (LIL-Heuristic). We find the constants C_1, C_2, C_3 using a simulation on estimating the bias of a coin with different constants and picking the ones so that the observed failure probability is below 5%. We use $C_1 = 0.5$ and $C_2 = 0.75$, $C_3 = 1$. We use $\alpha = 0.05$, $\delta = 0.2$, and importance sampling probability $p = 0.2$ through all the empirical evaluations. More details are available in the appendix.

Synthetic data setup: We simulate the OOD and ID scores using a mixture of two Gaussians $\mathcal{N}_{id}(\mu = 5.5, \sigma = 4)$ and $\mathcal{N}_{ood}(\mu = -6, \sigma = 4)$. We randomly draw 100k samples with $\gamma = 0.2$ (see Figure 4).



(a) No distribution shift, no window. (b) Distribution shift, 5k window. (c) Distribution shift, 10k window.

Figure 6: Results with the KNN scores on Cifar-10 as ID dataset. For (b) and (c) the distribution shifts at $t = 50k$. The arrow indicates the time at which the mean FPR + std. deviation over 10 runs goes below 5% for LIL method.

Real data setup: We use ID and OOD datasets and scoring functions from the OpenOOD benchmark (8). Here we show the results on CIFAR-10 (12) as an ID dataset and show the results on CIFAR-100, and Imagenet-1K (13) in the appendix. To verify C3, we use various scoring functions: ODIN (3), Mahalanobis Distance (4), Energy Score (5), SSD (14), VIM (15), and KNN (16) scores for the evaluation. Due to space limitation, we present results for the KNN (16) score here. For more details on the datasets, scores, and results on the rest of the scores please see the appendix.

4.1 Stationary Distributions Setting

In the stationary setting the data distributions do not change over time. We use this setting to verify our theoretical claims as they are valid in such settings. We perform the experiments to verify claims C1 and C2 on synthetic and real data. See Figures 4(a) and 17(a) for the results. We make the following observations: (i) We see that the non-adaptive method (TPR-95) with the fixed threshold has a high FPR at all times and violates the FPR constraint by a big margin. On the other hand, the adaptive methods improve with time. (ii) We see that not using a UCB leads to violation of FPR constraints and the methods with LIL-Heuristic, Hoeffding based intervals are able to maintain the FPR below the user given threshold 5%. Moreover, all the methods improve as they acquire more samples with time and eventually reach very close to the optimal solution. We note that our method (LIL-heuristic) is faster in this regard than while maintaining safety.

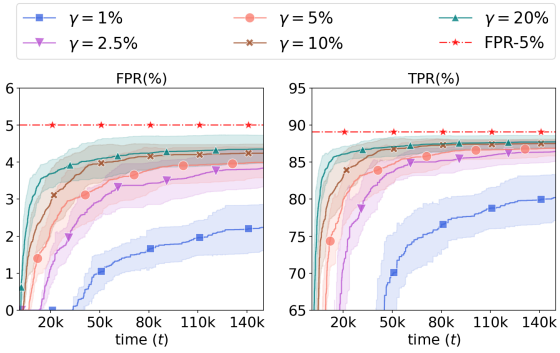


Figure 7: Results on the synthetic data with stationary distributions and different mixing ratio γ.

Time to reach feasibility and optimality: In our theoretical results, we derived bounds on the time to reach feasibility (T_f) and the time to reach η-optimality denoted by $T_{\eta-\text{opt}}$. These times are inversely proportional to the mixing ratio γ and the optimality level η. To verify this we run the LIL method on the synthetic data setup with different values of γ and observe T_f (corresponding to $\alpha = 5\%$) and $T_{\eta-\text{opt}}$. We report the mean and std. deviation of T_f and $T_{\eta-\text{opt}}$ over 10 runs with different random seeds (see Table 1). We see both T_f and $T_{\eta-\text{opt}}$ decrease as γ increases and $T_{\eta-\text{opt}}$ is also inversely proportional to the optimality level. The corresponding FPR and TPR trends for each γ are shown in Figure 7. These trends also corroborate our understanding of the effect of γ on the time for feasibility and optimality.

γ	T_f	$T_{\eta-\text{opt}}$			
		$\eta = 1.0\%$	$\eta = 1.5\%$	$\eta = 2.0\%$	$\eta = 2.5\%$
2.5%	14,167 ±602	93,011 ±27,387	71,089 ±25,654	70,559 ±35,056	37,534 ±9302
5%	7,054 ±301	53,971 ±20,816	47,143 ±24,004	39,864 ±22,328	32,473 ±20,262
10%	3,549 ±200	50,748 ±33,947	35,517 ±22,131	26,435 ±14,361	17,312 ±7,757
20%	1,770 ±72	40,240 ±37,751	28,943 ±31,138	9,004 ±3,383	6,500 ±2,495

Table 1: Time to reach feasibility T_f and optimality $T_{\eta-\text{opt}}$ in the stationary setting for different η and mixing ratios γ.

4.2 Distribution Shift Setting

We now proceed to investigate the case where the distributions change at a specific time point. One of the motivations for the proposed system is to be able to adapt to the variations of the OOD data. As long as $\mathcal{D}_{ood}$ does not change, any changes in the $\mathcal{D}_{id}$ or the mixing ratio γ do not affect the true FPR and therefore the optimal $\lambda^\star$. However, the true FPR does get affected when $\mathcal{D}_{ood}$ changes. When there is a change in $\mathcal{D}_{ood}$, estimating the FPR using all the acquired samples so far heavily biases the estimate towards scores that are far behind in time from the previous $\mathcal{D}_{ood}$. This leads to a long delay before our unbiased estimate of FPR can catch up to the change.

Windowed approach: To overcome this challenge, we propose a sliding window-based approach with the adaptive methods. The user can set a window size $N_w > 0$ and the system will estimate the FPR and the confidence intervals using only the most recent N_w samples that are determined as OOD by human feedback. This allows the system to more quickly adapt the threshold that is well aligned with the new distribution(s) of OOD samples.

Change detection: We use the following criteria to detect change, if $\widehat{\text{FPR}}(\hat{\lambda}_{t-1}, t) - \psi(t, \delta) > \alpha$ then it means the OOD distribution has changed. Here the $\widehat{\text{FPR}}$ and ψ are computed using the samples in the window. We use change detection only for the methods with confidence intervals. See the appendix for more details on this criteria.

The window size N_w has trade-offs, i.e., using a smaller window will enable faster change detection and adapting to the new distribution but imposes limitations on the optimality as the smallest width of the confidence interval possible is inversely proportional to the window size. We verify claim C4 and study these trade-offs using experiments on synthetic and real data.

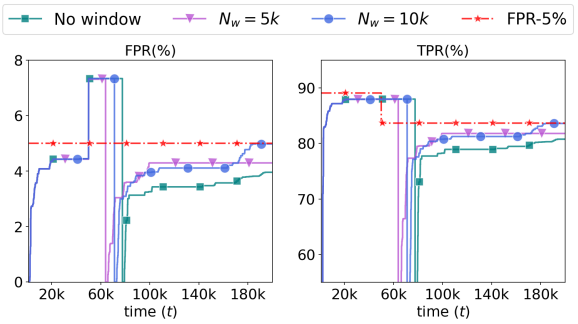


Figure 8: Change detection and restart.

For the synthetic data, we use the same ID and OOD distribution as above till $t = 50k$ and change the OOD distribution to $\mathcal{N}_{ood}(\mu = -5, \sigma = 4)$ at time $t = 50k$. In the real data setting, we use the CIFAR-10 as ID and a mixture of MNIST, SVHN, and Texture datasets as OOD till $t = 50k$ and a mixture of TinyImageNet, Places365, CIFAR-100 as OOD after $t = 50k$. We run TPR-95, and adaptive methods LIL, and Hoeffding in these settings with different choices of window sizes with 10 repeated runs using different random seeds and show the mean and std. deviations of the FPR and TPR in Figures 5 and 6. We find that the windowed approach adapts more quickly compared to the method without a window (see Figures 5(a),5(c)).

Effect of window size: The results with various window sizes are shown in Figures 5(b), 5(c) on synthetic data and in Figures 17(b), 17(c) on real data. We also show the box plots of change detection times with LIL in Figure 9. As expected, we see that with smaller window size the change is detected earlier and the method is able to adapt faster. We also see that while smaller window helps in faster adaption but limits how close to the optimal TPR is achieved.

To showcase the effect of **window-based approach in stationary setting**, we run the methods with a window size of 5k in the fixed distribution setting (see Figure 4(b)). We observe similar behavior to the case without a window but with higher variance as the confidence intervals are limited by the window size.

Restart after change detection: In the previous experiments the algorithm kept using all the samples in its window even after detecting the change. The window can contain samples from the previous distribution till some time which leads to prolonged violation of FPR constraint. In safety critical applications one might apply a conservative approach i.e., restart the algorithm after detecting the change by emptying the window and resetting the threshold to Λ_{max}. We run the LIL based method with this variation using different window sizes and show the FPR and TPR of a *median trend* in Figure 8. To get the median trend we run the algorithm with 10 random seeds and pick the FPR, and TPR trends corresponding to the run having the median change detection time. We see that the FPR and TPR drop to 0 immediately after the change is detected and then the method recovers gradually. The variation without a window took longer time to detect the change and hence lags behind the window-based methods in approaching optimality

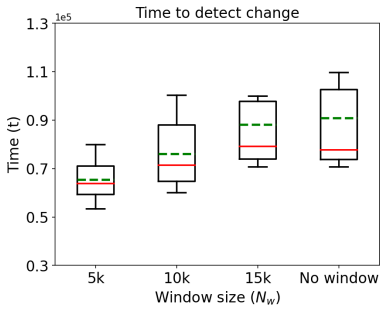


Figure 9: Box plot of change detection times with different window sizes on synthetic data. The median and mean are shown using solid and dashed lines respectively.

after restart. With 5k window it detected the change earlier but due to the small window, it is not able to reach optimality. The one with a window size of 10k appears to be a good trade-off as it is neither too late in detecting the change nor lags too far in approaching the optimality.

5 Related Works

Out-Of-Distribution detection: has been addressed in many recent works where the main contributions have been methods to quantify a score (uncertainty) which gives a better separation of OOD and ID data points. (3) proposed ODIN, which uses temperature scaling to separate the softmax score distributions between ID and OOD images. (5) proposed a framework using energy score to perform OOD detection on pre-trained neural classifiers. (4),(14), and (6) proposed Mahalanobis distance-based scores to detect OOD samples. While these methods perform well, the evaluation setup is rather static and does not reflect the real-world deployment scenario, wherein the system has to adapt as it sees OOD data and it is a priori not clear how to set the threshold for detection.

Online anomaly detection: There is a rich literature on anomaly (or outlier) detection in offline settings (17; 18; 19). However, our setting is akin to the online anomaly (outlier) detection – wherein the system receives samples one at a time and it has to figure out the outliers or anomalous behavior within a given window of time. Some of the notable works along this line are (20; 21; 22). The methods proposed are unsupervised and perform density or distance-based detection.

Outlier detection with human-in-the-loop: The notion of an outlier may not always be based on statistical rarity and might need input from humans to learn the notion of an outlier in the application of interest. Some of the recent works (23; 24) proposed methods for outlier detection in offline settings leveraging human inputs. They focused on minimizing human effort by figuring out some candidate outliers and designing good questions and context for getting human inputs. Our work is complementary to this literature. We propose a novel framework for safely adapting the threshold during deployment using human feedback on selected points.

OOD detection with test-time optimization. This area of work focuses on addressing the distribution shift problems because of the lack of training data. MEMO (25) proposed using multi-head models such that the trained model can be adapted with test time distribution shift. ETLT (26) proposed training a separate linear regression model during test time to calibrate the OOD scores as OOD scores mostly linearly related to their image features. While maintaining a separate model for score calibration could be inefficient, other works such as (27) and (28) address the issue in a post-hoc manner without altering the trained model. We consider these methods complementary to our work as our framework can adopt the calibrated OOD scores and adapt the threshold safely with FPR control.

6 Conclusion

We presented a mathematically grounded framework for human-in-the-loop OOD detection. By incorporating expert feedback and utilizing confidence intervals based on the Law of Iterated Logarithm (LIL), our approach maintains control over FPR while maximizing the TPR. The empirical evaluations on synthetic data and image classification tasks demonstrate the effectiveness of our method in maintaining FPR at or below 5% while achieving high TPR. Our theoretical guarantees are valid for stationary settings. We leave the extension to non-stationary settings as future work.

References

[1] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, pages 427–436. IEEE Computer Society, 2015.

[2] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016.

[3] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

[4] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

[5] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475, 2020.

[6] Yifei Ming, Yiyou Sun, Ousmane Dia, and Yixuan Li. Cider: Exploiting hyperspherical embeddings for out-of-distribution detection. *arXiv preprint arXiv:2203.04450*, 2022.

[7] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.

[8] Jingkang Yang, Pengyun Wang, Dejian Zou, Zitang Zhou, Kunyuan Ding, Wenxuan Peng, Haoqi Wang, Guangyao Chen, Bo Li, Yiyou Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, Dan Hendrycks, Yixuan Li, and Ziwei Liu. Openood: Benchmarking generalized out-of-distribution detection, 2022.

[9] Aleksandr Khinchine. Über einen satz der wahrscheinlichkeitsrechnung. *Fundamenta Mathematicae*, 6:9–20, 1924.

[10] Steven R. Howard and Aaditya Ramdas. Sequential estimation of quantiles with applications to A/B testing and best-arm identification. *Bernoulli*, 28(3):1704 – 1728, 2022.

[11] Akshay Balsubramani. Sharp finite-time iterated-logarithm martingale concentration, 2015.

[12] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[14] Vikash Sehwag, Mung Chiang, and Prateek Mittal. Ssd: A unified framework for self-supervised outlier detection. *arXiv preprint arXiv:2103.12051*, 2021.

[15] Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. Vim: Out-of-distribution with virtual-logit matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4921–4930, 2022.

[16] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning*, pages 20827–20840. PMLR, 2022.

[17] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), jul 2009.

[18] Guilherme O. Campos, Arthur Zimek, Jörg Sander, Ricardo J. Campello, Barbora Micenková, Erich Schubert, Ira Assent, and Michael E. Houle. On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study. *Data Min. Knowl. Discov.*, 30(4):891–927, 2016.

[19] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[20] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos. Online outlier detection in sensor data using non-parametric models. In *Proceedings of the 32nd International Conference on Very Large Data Bases*, VLDB '06, page 187–198. VLDB Endowment, 2006.

[21] Fabrizio Angiulli and Fabio Fassetti. Detecting distance-based outliers in streams of data. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, page 811–820, 2007.

[22] Yang Zhang, Nirvana Meratnia, and Paul J.M. Havinga. Distributed online outlier detection in wireless sensor networks using ellipsoidal support vector machine. *Ad Hoc Networks*, 11(3):1062–1074, 2013.

[23] Chengliang Chai, Lei Cao, Guoliang Li, Jian Li, Yuyu Luo, and Samuel Madden. Human-in-the-loop outlier detection. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, page 19–33, New York, NY, USA, 2020. Association for Computing Machinery.

[24] Md Rakibul Islam, Shubhomoy Das, Janardhan Rao Doppa, and Sriraam Natarajan. Glad: Glocalized anomaly detection via human-in-the-loop learning. *arXiv preprint arXiv:1810.01403*, 2018.

[25] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *Advances in Neural Information Processing Systems*, 35:38629–38642, 2022.

[26] Ke Fan, Yikai Wang, Qian Yu, Da Li, and Yanwei Fu. A simple test-time method for out-of-distribution detection. *arXiv preprint arXiv:2207.08210*, 2022.

[27] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[28] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021.

[29] Andrey Kolmogorov. Über das gesetz des iterierten logarithmus. *Mathematische Annalen*, 101:126–135, 1929.

[30] Nikolai Smirnov. Approximate laws of distribution of random variables from empirical data. *Uspekhi Matematicheskikh Nauk*, 10:179–206, 1944.

[31] Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil' ucb : An optimal exploration algorithm for multi-armed bandits, 2013.

[32] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

7 Appendix

The appendix is organized as follows. We summarize the notation in Table 2. Then we give the proof of the main theorem (Theorem 2) and the proofs of supporting lemmas. Further, we provide additional experiments and insights from them.

7.1 Glossary

The notation is summarized in Table 2 below.

Symbol	Definition
$\mathcal{X}$	feature space.
$\mathcal{Y}$	label space, $\{+1, -1\}$, +1 for ID and -1 for OOD .
$\mathcal{D}_{id}, \mathcal{D}_{ood}$	distributions of ID and OOD points.
γ	mixing ratio of OOD and ID distributions.
λ	threshold for OOD classification.
FPR(λ)	population level false positive rate with threshold λ.
TPR(λ)	population level true positive rate with threshold λ.
$\widehat{\text{FPR}}(\lambda, t)$	empirical FPR at time t, adjusted to account for importance sampling (see eq. (3)).
λ^*	the optimal threshold for OOD classification s.t. FPR(λ) $\leq \alpha$ and TPR(λ) is maximized.
$\hat{\lambda}_t$	the estimated threshold at round t.
x_t, y_t	sample and the true label at time t .
g	OOD uncertainty quantification (score) function.
$s_u^{(o)}$	score of u^{th} OOD sample.
$i_u^{(o)}$	indicator variable denoting whether $s_u^{(o)}$ was importance sampled or not.
$N_t^{(o)}$	number of OOD points till time t.
$N_t^{(o,p)}$	number of OOD points sampled using importance sampling until time t.
β_t	it is equal to $N_t^{(o,p)}/N_t^{(o)}$.
p	probability for importance sampling.
δ	failure probability.
α	user given upper bound on FPR that the algorithm needs to maintain.
η	the algorithm is in $\eta-$optimality if close FPR(λ^*) − FPR($\hat{\lambda}_t$) $\leq \eta$.
$\Lambda_{\min}, \Lambda_{\max}$	the minimum and maximum scores(thresholds) considered by the algorithm.
ν	discretization parameter for the interval $[\Lambda_{\min}, \Lambda_{\max}]$ set by the user.
$\psi(t, \delta)$	LIL based confidence interval at time t.

Table 2: Glossary of variables and symbols used in this paper.

7.2 Proofs

Summary of the setting: At each time t, we observe $x_t \overset{i.i.d}{\sim} (1-\gamma)\mathcal{D}_{id} + \gamma\mathcal{D}_{ood}$, and $s_t = g(x_t)$ is the corresponding score. If $s_t \leq \hat{\lambda}_{t-1}$, then it is considered an OOD point and hence gets a human label for it and we get to know whether it is in fact OOD or ID. If $s_t > \hat{\lambda}_{t-1}$, then it is considered an ID point and hence gets a human label only with probability p. So, we get to know whether it is truly ID or not with probability p. Now we have to update the threshold, $\hat{\lambda}_t$, such that the FPR($\hat{\lambda}_t$) $\leq \alpha$ for all t, while trying to maximize TPR($\hat{\lambda}_t$). Our approach is based on constructing an unbiased estimator of FPR(λ) using the OOD samples received till time t and in conjunction with confidence intervals for FPR(λ) for at all thresholds $\lambda \in \Lambda$ that is valid for all times simultaneously. Together, at each time t, these give us a reliable upper bound on the true FPR(λ) for all λ enabling us to find the smallest λ such that the upper bound on FPR(λ) is at most α. Let $S_t^{(o)} = \{s_1^{(o)}, \ldots s_{N_t^{(o)}}^{(o)}\}$ denote the scores of the points that have been truly identified as

OOD points from human feedback and $I_t^{(o)}$ be the corresponding time points. We estimate the FPR as follows,

$$\widehat{\text{FPR}}(\lambda, t) = \frac{1}{N_t^{(o)}} \sum_{u \in I_t^{(o)}} Z_u(\lambda), \text{ where } Z_u(\lambda) := \begin{cases} \mathbf{1}(s_u^{(o)} > \lambda), & \text{if } s_u^{(o)} \leq \hat{\lambda}_{u-1} \\ \frac{1}{p}\mathbf{1}(s_u^{(o)} > \lambda), & \text{w.p. } p \text{ if } s_u^{(o)} > \hat{\lambda}_{u-1} \\ 0, & \text{w.p. } 1-p \text{ if } s_u^{(o)} > \hat{\lambda}_{u-1} \end{cases}. \tag{3}$$

Proof outline: To obtain the guarantees in Theorem 2 we need confidence intervals $\psi(t, \delta)$ that are simultaneously valid with high probability for the FPR estimates at all time points and all thresholds. There is a rich line of work that provides tight confidence intervals valid for all times based on the Law of Iterated Logarithm (LIL) (9; 29; 30). Non-asymptotic versions of LIL have been proved in various settings e.g. multi-armed bandits (31) and for quantile estimation (10). Roughly speaking, these bounds provide confidence intervals that are $\mathcal{O}(\sqrt{\log\log(t)/t}$ and are known to be tight. However, most of them assume the samples to be i.i.d. In our setting, our treatment of observing the human feedback is dependent on whether the score is above or below $\hat{\lambda}_{t-1}$ which itself is estimated using all the past data which creates dependence and prevents us from utilizing results developed for i.i.d. samples (10). The main technical challenge is to show that the upper confidence bound in eq. (2) holds for all time and all thresholds with dependent samples used in estimating the FPR in eq. (3). We handle this by first showing that there is a martingale structure in our estimated FPR (eq. (3)). We then exploit this structure by using LIL results for martingales (11). A limitation of (11) is that it can only provide us confidence intervals valid for FPR estimate for a given threshold λ. However, we need intervals that are simultaneously valid for all λ as well. Building upon the work (11) we derive confidence intervals that are simultaneously valid for all t and finitely many thresholds. Equation (2) shows the $\psi(t, \delta)$ we obtain. Please see the Appendix for detailed proofs and discussion.

Next, we show that the above estimator $\widehat{\text{FPR}}(\lambda, t)$ is indeed an unbiased of false positive rate $\text{FPR}(\lambda)$.

Lemma 2. *Let $p > 0$, $\widehat{FPR}(\lambda, t)$ as defined in eq.* (3) *is an unbiased estimate of the true FPR(λ), i.e., $\mathbb{E}[\widehat{FPR}(\lambda, t)] =$ FPR(λ).*

Proof. Let $i_t^{(o)}$ be the indicator variable denoting whether $s_t^{(o)}$ was sampled using importance sampling (i.e. $i_t^{(o)} = 1$) or not (i.e. $i_t^{(o)} = 0$). Denote the pair as $r_t^{(o)} = (s_t^{(o)}, i_t^{(o)})$ for brevity. The proof is by induction. Since the first sample is drawn without any importance sampling so, we have $\mathbb{E}_{r_1^{(o)}}[\widehat{\text{FPR}}(\lambda, 1)] = \text{FPR}(\lambda)$. Now assume that $\mathbb{E}_{r_{t-1}^{(o)}, \ldots, r_1^{(o)}}[\widehat{\text{FPR}}(\lambda, t-1)] = \text{FPR}(\lambda)$.

$$\begin{aligned}
\mathbb{E}_{r_t^{(o)}, r_{t-1}^{(o)}, \ldots, r_1^{(o)}}[\widehat{\text{FPR}}(\lambda, t)] &= \mathbb{E}\Big[\frac{1}{N_t^{(o)}} \sum_{u \in I_t^{(o)}} Z_u(\lambda)\Big] \\
&= \mathbb{E}\Big[\frac{Z_t(\lambda)}{N_t^{(o)}} + \frac{N_t^{(o)} - 1}{N_t^{(o)}} \frac{1}{N_t^{(o)} - 1} \sum_{u \in I_{t-1}^{(o)}} Z_u(\lambda)\Big] \\
&= \mathbb{E}\Big[\frac{Z_t(\lambda)}{N_t^{(o)}} + \frac{N_t^{(o)} - 1}{N_t^{(o)}} \widehat{\text{FPR}}(\lambda, t-1)\Big] \\
&= \frac{1}{N_t^{(o)}}\Big[\mathbb{E}[Z_t(\lambda)] + (N_t^{(o)} - 1) \cdot \mathbb{E}[\widehat{\text{FPR}}(\lambda, t-1)]\Big] \\
&= \frac{1}{N_t^{(o)}}\Big[\mathbb{E}[Z_t(\lambda)] + (N_t^{(o)} - 1) \cdot \text{FPR}(\lambda)\Big] \\
&= \frac{1}{N_t^{(o)}}\Big[\mathbb{E}_{r_t^{(o)} | \hat{\lambda}_{t-1}}[Z_t(\lambda)] + (N_t^{(o)} - 1) \cdot \text{FPR}(\lambda)\Big] \\
&= \frac{1}{N_t^{(o)}}\Big[\text{FPR}(\lambda) + (N_t^{(o)} - 1) \cdot \text{FPR}(\lambda)\Big] \\
&= \text{FPR}(\lambda)
\end{aligned}$$

□

Having an unbiased estimator solves one part of the problem. In addition, we need confidence intervals on this estimate that are valid for anytime and for the choices of λ considered. Due to the dependence between the samples, we cannot directly apply similar results developed for quantile estimation in the i.i.d. setting (10). Fortunately, part of this problem has been addressed in (11), where they provide anytime valid confidence intervals when the estimators form a martingale sequence. We restate this result in the following lemma 3 and then building upon this result, in the next lemma 4 we derive such confidence intervals for our setting.

Lemma 3. *((11)) Let $\overline{M}_t$ be a martingale and suppose $|\overline{M}_t - \overline{M}_{t-1}| \leq \rho_t$ for constants $\{\rho_t\}_{t>1}$, let $m_0 = \min_{t\geq 1}|\overline{M}_t|$. Fix any $\delta \in (0,1)$, and let $t_0 = \min\{u : \sum_{t=1}^{u}\rho_t^2 \geq 173\log(\frac{4}{\delta})\}$ then,*

$$\mathbb{P}\Bigg(\exists t \geq t_0 : |\overline{M}_t| \geq \sqrt{3\Big(\sum_{i=1}^{t}\rho_i^2\Big)\Big(2\log\log\big(\frac{3\sum_{i=1}^{t}\rho_i^2}{m_0}\big) + \log\big(\frac{2}{\delta}\big)\Big)}\Bigg) \leq \delta \tag{4}$$

Proof. This lemma is a restatement of theorem 4 in (11). For proof details please see (11). □

In the next lemma, we show that the sums of $Z_u(\lambda)$ form a martingale sequence, allowing us to apply the results from the above lemma (3), and then we generalize it to all λ in some finite set.

Lemma 4. *(Anytime valid confidence intervals on FPR) Let $X_t^{(o)} = \{x_1^{(o)}, \dots x_{N_t^{(o)}}^{(o)}\}$ be the samples drawn from D_{ood} till round t and let $S_t^{(o)} = \{s_1^{(o)}, \dots s_{N_t^{(o)}}^{(o)}\}$ be the scores of these points, let $c_t = 1 - \beta_t + \frac{\beta_t}{p^2}$, $\beta_t = \frac{N_t^{(o,p)}}{N_t^{(o)}}$ and $N_t^{(o,p)}$ is the number of points sampled using importance sampling until time t and $\nu \in (0,1)$ is a discretization parameter set by the user. Let $\Lambda = \{\Lambda_{\min}, \Lambda_{\min}+\nu, \dots, \Lambda_{\max}\}$. Let $n_0 = \min\{u : c_u N_u^{(o)} \geq 173\log(\frac{4}{\delta})\}$ and t_0 be such that $N_{t_0}^{(o)} \geq n_0$. , then for any $\delta \in (0,1)$,*

$$\mathbb{P}\Bigg(\exists t \geq t_0 : \sup_{\lambda\in\Lambda}\widehat{\textit{FPR}}(\lambda,t) - \textit{FPR}(\lambda) \geq \psi(t,\delta)\Bigg) \leq \delta \tag{5}$$

for,

$$\psi(t,\delta) = \sqrt{\frac{3c_t}{N_t^{(o)}}\left[2\log\log\left(\frac{3c_t N_t^{(o)}}{2}\right) + \log\left(\frac{2|\Lambda|}{\delta}\right)\right]} \tag{6}$$

Proof. First, we show that we have a martingale sequence as follows,

Let $M_t(\lambda) = \sum_{u=1}^{N_t^{(o)}} Z_u(\lambda)$, and consider the centered random variables,

$$\overline{M}_t(\lambda) = M_t(\lambda) - \mathbb{E}[M_t(\lambda)] \quad \text{and} \quad \overline{Z}_t(\lambda) = Z_t(\lambda) - \text{FPR}(\lambda)$$

Let $\mathcal{F}_t$ be the $\sigma-$algebra of events till time t i.e. $(s_1^{(o)}, i_1^{(o)}), \dots, (s_{t-1}^{(o)}, i_{t-1}^{(o)}), (s_t^{(o)}, i_t^{(o)})$.

It is easy to see that $\mathbb{E}[\overline{M}_t] \leq \frac{1}{p} < \infty$ and $\overline{M}_t$ is $\mathcal{F}_t$-measurable for all $t > 1$. Further, we can see,

$$\mathbb{E}\big[\overline{M}_t(\lambda)|\mathcal{F}_{t-1}\big] = \mathbb{E}[\overline{Z}_t(\lambda) + \overline{M}_{t-1}(\lambda)|\mathcal{F}_{t-1}] = \mathbb{E}[\overline{Z}_t(\lambda)|\mathcal{F}_{t-1}] + \mathbb{E}[\overline{M}_{t-1}(\lambda)|\mathcal{F}_{t-1}] = \overline{M}_{t-1}(\lambda)$$

Since, $\mathbb{E}[\overline{Z}_t(\lambda)|\mathcal{F}_{t-1}] = 0$ and $\mathbb{E}[\overline{M}_{t-1}(\lambda)|\mathcal{F}_{t-1}] = \overline{M}_{t-1}(\lambda)$. Thus we have that $\overline{M}_t$ is a martingale sequence. Further, we also have the following,

$$|\overline{M}_t(\lambda) - \overline{M}_{t-1}(\lambda)| \leq \begin{cases} 1 & \text{if } i_t^{(o)} = 0 \\ \frac{1}{p} & \text{if } i_t^{(o)} = 1 \end{cases}$$

Let $\beta_t \in (0,1)$ be the fraction of OOD points sampled using probability p till round t. Let $N_t^{(o)}$ be the total number of points OOD points sampled till round t and $N_t^{(o,p)}$ be the points sampled from importance sampling, then $\beta_t = \frac{N_t^{(o,p)}}{N_t^{(o)}}$.

Let $c_t = 1 - \beta_t + \frac{\beta_t}{p^2}$. We know p and the number of points sampled with importance sampling, without importance sampling we knowβ_t, c_t are at time t. Applying lemma 3 we get the following result for a given λ,

$$\mathbb{P}\left(\exists t \geq t_0 : \overline{M}_t(\lambda) \geq \sqrt{3\left(c_t N_t^{(o)}\right)\left(2\log\log\left(3c_t N_t^{(o)}\right) + \log\left(\frac{2}{\delta}\right)\right)}\right) \leq \delta \tag{7}$$

$$\mathbb{P}\left(\exists t \geq t_0 : \widehat{\text{FPR}}(\lambda, t) - \text{FPR}(\lambda, t) \geq \sqrt{\frac{3c_t}{N_t^{(o)}}\left(2\log\log\left(3c_t N_t^{(o)}\right) + \log\left(\frac{2}{\delta}\right)\right)}\right) \leq \delta \tag{8}$$

Doing the union bound for the failure probability over all $\lambda \in \Lambda$, (where $|\Lambda| < \infty$) gives us the result.

□

Our performance guarantees in the main theorem 2 are based on $\psi(t, \delta)$ becoming smaller than certain values. In the next lemma we derive bound on $N_t^{(o)}$ such that $\psi(t, \delta)$ is at most μ and use it in the proof of the main theorem 2.

Lemma 5. *Let* $\psi(t,\delta) = \sqrt{\frac{3c_t}{N_t^{(o)}}\left(2\log\log\left(3c_t N_t^{(o)}\right) + \log\left(\frac{2|\Lambda|}{\delta}\right)\right)}$, *and let there be a constant* C_0 *and time* T_0, *such that* $\beta_t \leq C_0 p^2$ *for all* $t \geq T_0$ *(worst case* $T_0 = 1$ *and* $C_0 = 1/p^2$*). Then* $\psi(t,\delta) \leq \mu$ *for any* $t > T_\mu > T_0$ *such that* $N_{T_\mu}^{(o)} = \frac{10(C_0+1)}{\mu^2}\log\left(\frac{|\Lambda|}{\delta}\log(\frac{5(C_0+1)}{\mu})\right)$.

Proof. First we write a simplified form of $\psi(t, \delta)$ for all $t > T_0$ as follows,

$$\psi(t,\delta) = \sqrt{\frac{3(C_0+1)}{N_t^{(o)}}\left(2\log\log\left(3(C_0+1)N_t^{(o)}\right) + \log\left(\frac{2|\Lambda|}{\delta}\right)\right)}$$

In the above equation we used the bound on $\beta_t \leq C_0 p^2$ in the equation $c_t = 1 - \beta_t + \beta_t/p^2$ leading to $c_t \leq C_0 + 1$, Now, for brevity let $a_1 = 3(C_0 + 1)$ and $a_2 = 2|\Lambda|$ and rewrite $\psi(t, \delta)$ as follows,

$$\psi^2(t,\delta) = \frac{a_1}{N_t^{(o)}}\left(2\log\log\left(a_1 N_t^{(o)}\right) + \log\left(\frac{a_2}{\delta}\right)\right) \leq \frac{2a_1}{N_t^{(o)}}\left(\log\left(\frac{a_2}{\delta}\log\left(a_1 N_t^{(o)}\right)\right)\right)$$

We want to find $N_t^{(o)}$ such that $\psi^2(t, \delta) \leq \mu^2$. It is difficult to directly invert this function. To get a bound on $N_t^{(o)}$ we first assume the following form for it with unknown constants $b_1, b_2, b_3 > 0$ and then figure out the constants by simplifying $\psi^2(N_t^{(o)})$ and constraining it to be at most μ^2.

$$\text{Let } N_{T_\mu}^{(o)} = \frac{b_1}{\mu^2}\log\left(\frac{a_2}{b_3\delta}\log\left(\frac{b_2}{\mu}\right)\right)$$

$$\begin{aligned}
\psi^2(T_\mu,\delta) &\leq \frac{2a_1}{N_{T_\mu}^{(o)}}\log\left[\frac{a_2}{\delta}\log(a_1 N_{T_\mu}^{(o)})\right] \\
&= \frac{2a_1\mu^2}{b_1\log\left(\frac{a_2}{b_3\delta}\log\left(\frac{b_2}{\mu}\right)\right)}\log\left[\frac{a_2}{\delta}\log\left\{\frac{a_1 b_1}{\mu^2}\log\left(\frac{a_2}{b_3\delta}\log\left(\frac{b_2}{\mu}\right)\right)\right\}\right] \\
&\overset{(i)}{\leq} \frac{2a_1\mu^2}{b_1\log\left(\frac{a_2}{b_3\delta}\log\left(\frac{b_2}{\mu}\right)\right)}\log\left[\frac{a_2}{\delta}\log\left\{\frac{a_1 b_1}{\mu^2}\log\left(\frac{a_2}{b_3\delta}\frac{b_2}{\mu}\right)\right\}\right] \\
&= \frac{2a_1\mu^2}{b_1\log\left(\frac{a_2}{b_3\delta}\log\left(\frac{b_2}{\mu}\right)\right)}\log\left[\frac{a_2}{\delta}\log\left\{\frac{a_1 b_1}{\mu^2}\log\left(\frac{a_2 b_2}{b_3\delta\mu}\right)\right\}\right] \\
&\overset{(ii)}{\leq} \frac{2a_1\mu^2}{b_1\log\left(\frac{a_2}{b_3\delta}\log\left(\frac{b_2}{\mu}\right)\right)}\log\left[\frac{a_2}{\delta}\log\left\{\frac{a_1 b_1}{\mu^2}\frac{a_2 b_2}{b_3\delta\mu}\right\}\right] \\
&= \frac{2a_1\mu^2}{b_1\log\left(\frac{a_2}{b_3\delta}\log\left(\frac{b_2}{\mu}\right)\right)}\log\left[\frac{a_2}{\delta}\log\left\{\frac{a_1 b_1}{\mu^2}\frac{a_2 b_2}{b_3\delta\mu}\right\}\right]
\end{aligned}$$

$$
\begin{aligned}
&= \frac{2a_1\mu^2}{b_1 \log\left(\frac{a_2}{b_3\delta}\log\left(\frac{b_2}{\mu}\right)\right)} \log\left[\frac{a_2}{\delta}\log\left\{\frac{a_1b_1a_2}{b_3b_2^2\delta}\left(\frac{b_2}{\mu}\right)^3\right\}\right] \\
&\overset{(iii)}{\leq} \frac{2a_1\mu^2}{b_1 \log\left(\frac{a_2}{b_3\delta}\log\left(\frac{b_2}{\mu}\right)\right)} \log\left[\frac{a_2}{\delta}\frac{a_1b_1a_2}{b_3b_2^2\delta}\log\left\{\left(\frac{b_2}{\mu}\right)^3\right\}\right] \\
&= \frac{2a_1\mu^2}{b_1 \log\left(\frac{a_2}{b_3\delta}\log\left(\frac{b_2}{\mu}\right)\right)} \log\left[\frac{a_2}{\delta}\frac{3a_1b_1a_2}{b_3b_2^2\delta}\log\left(\frac{b_2}{\mu}\right)\right] \\
&= \frac{2a_1\mu^2}{b_1 \log\left(\frac{a_2}{b_3\delta}\log\left(\frac{b_2}{\mu}\right)\right)} \log\left[\left(\frac{a_2}{b_3\delta}\right)^2\frac{3a_1b_1b_3}{b_2^2}\log\left(\frac{b_2}{\mu}\right)\right] \\
&\overset{(iv)}{\leq} \frac{2a_1\mu^{2\,\frac{3a_1b_1b_3}{b_2^2}}}{b_1 \log\left(\frac{a_2}{b_3\delta}\log\left(\frac{b_2}{\mu}\right)\right)} \log\left[\left(\frac{a_2}{b_3\delta}\right)^2\log\left(\frac{b_2}{\mu}\right)\right] \\
&\overset{(v)}{\leq} \frac{2a_1\mu^2\cdot 2^{\frac{3a_1b_1b_3}{b_2^2}}}{b_1 \log\left(\frac{a_2}{b_3\delta}\log\left(\frac{b_2}{\mu}\right)\right)} \log\left[\frac{a_2}{b_3\delta}\log\left(\frac{b_2}{\mu}\right)\right] \\
&= \frac{12\mu^2a_1^2b_3}{b_2^2}.
\end{aligned}
$$

The inequalities $(i), (ii)$ follow from $\log(x) \leq x$ for any $x > 0$.

The inequality (iii) comes from $\log(ax) \leq a\log(x)$ for any $a > 2, x > 2$. We use $a = \frac{a_1b_1a_2}{b_3b_2^2\delta}$ and $x = \left(\frac{b_2}{\mu}\right)^3$, this enforces the following constraints,

$$\frac{b_2}{\mu} > 2^{1/3} \tag{9}$$

$$\frac{a_1b_1a_2}{b_3b_2^2\delta} > 2 \tag{10}$$

For (iv) we again use $\log(ax) \leq a\log(x)$ with $a = \frac{3a_1b_1b_3}{b_2^2}$ and $x = \left(\frac{a_2}{b_3\delta}\right)^2\log\left(\frac{b_2}{\mu}\right)$, this enforces the following constraints,

$$\frac{3a_1b_1b_3}{b_2^2} > 2 \tag{11}$$

$$\left(\frac{a_2}{b_3\delta}\right)^2\log\left(\frac{b_2}{\mu}\right) > 2 \tag{12}$$

Lastly, (v) follows by using $\log(x^ay) \leq a\log(xy)$ for any $x > 0, a > 1, y > 1$. For this we use $x = \frac{a_2}{b_3\delta}$ and $y = \log(\frac{b_2}{\mu})$, leading the following constraints,

$$\log(\frac{b_2}{\mu}) > 1 \tag{13}$$

For $\psi^2(T_\mu) \leq \mu^2$, we need

$$12a_1^2b_3 \leq b_2^2 \tag{14}$$

Let $b_3 = 2, b_1 = 10a_1, b_2 = 5a_1$ then the constraints 9,10,11,12,13 and 14 are satisfied (when $|\Lambda| \geq 10$) for any $\mu \in (0,1), \delta \in (0,1)$. Thus we have,

$$\psi(T_\mu, \delta) \leq \mu \text{ for } N_{T_\mu} = \frac{10(C_0+1)}{\mu^2}\log\left(\frac{|\Lambda|}{\delta}\log(\frac{5(C_0+1)}{\mu})\right) \tag{15}$$

□

Lemma 6. *Let the data points $\{x_t\}_{t\geq 1}$ be independent draws from the mixture distribution $(1-\gamma)\mathcal{D}_{id}+\gamma\mathcal{D}_{ood}$, and $N_t^{(o)}$ be the number of OOD points received till time t from this distribution, then for any $\delta\in(0,1)$ for any $t\geq T_k$ we have $N_t^{(o)}\geq k$ w.p. $1-\delta$, where T_k is given as follows,*

$$T_k=\frac{2k}{\gamma}+\frac{1}{\gamma^2}\log(\frac{1}{\delta}). \tag{16}$$

Proof. We want to find t such that $N_t^{(o)}\geq k$ w.p. $1-\delta$. This is the same as finding the number of coin tosses of a coin with bias γ so that the number of heads observed is at least k. Applying Hoeffding's inequality gives us the following w.p. $1-\delta$,

$$N_t^{(o)}\geq \gamma t-\sqrt{\frac{t}{2}\log(\frac{1}{\delta})}.$$

Equating the r.h.s. above with k and solving for t will give us the desired bound on t. Note that it is enough to have an upper bound on t that satisfies the following and then use that upper bound as T_k.

$$\gamma t-\sqrt{\frac{t}{2}\log(\frac{1}{\delta})}=k.$$

To simplify the calculations, let $c=\sqrt{\frac{1}{2}\log\frac{1}{\delta}}$ and let $t=u^2$ then we have the following quadratic equation,

$$\gamma u^2-cu-k=0.$$

Considering the larger of the two solutions,

$$u=\frac{c+\sqrt{c^2+4k\gamma}}{2\gamma}.$$

Using the fact that for any $a,b\geq 0$, $\sqrt{a+b}\leq\sqrt{a}+\sqrt{b}$,

$$u\leq\frac{c+\sqrt{c^2}+\sqrt{4k\gamma}}{2\gamma}=\frac{2c+2\sqrt{k\gamma}}{2\gamma}=\frac{c}{\gamma}+\sqrt{\frac{k}{\gamma}}.$$

Lastly, using $(a+b)^2\leq 2a^2+2b^2$ for any $a,b\in\mathbb{R}$ we get the following upper bound on t,

$$t=u^2\leq\frac{2c^2}{\gamma^2}+\frac{2k}{\gamma}=\frac{2k}{\gamma}+\frac{1}{\gamma^2}\log(\frac{1}{\delta}).$$

□

Theorem 2. *Let $\alpha,\delta,p,\gamma\in(0,1)$. Let $x_t\overset{i.i.d}{\sim}(1-\gamma)\mathcal{D}_{id}+\gamma\mathcal{D}_{ood}$ and let $c_t=1-\beta_t+\frac{\beta_t}{p^2}$, $\beta_t=\frac{N_t^{(o,p)}}{N_t^{(o)}}$ where $N_t^{(o,p)}$ is the number of OOD points sampled using importance sampling until time t and $N_t^{(o)}$ is the total number of OOD points observed till time t. Let $n_0=\min\{u: c_uN_u^{(o)}\geq 173\log(\frac{8}{\delta})\}$ and t_0 be such that $N_{t_0}^{(o)}\geq n_0$. If Algorithm 1 uses the optimization problem* (P2) *to find the thresholds with the upper confidence term $\psi(N_t^{(o)},\delta/2)$ given by eq.* (2)*, then there exist constants $C_1,C_2,C_3>0$ such that with probability at least $1-\delta$,*

1. ***Controlled FPR:** For all $t\geq t_0$, FPR$(\hat{\lambda}_t)\leq\alpha$.*
2. ***Time to reach feasibility:** The algorithm will find a feasible threshold, $\hat{\lambda}_t$ such that $\widehat{FPR}(\hat{\lambda}_t)+\psi(N_t^{(o)})\leq\alpha$, for all $t\geq\max(t_0,T_f)$, where, $T_f=\frac{2C_1}{\gamma\alpha^2}\log\left(\frac{4C_2}{\delta}\log(\frac{C_3}{\alpha})\right)+\frac{1}{\gamma^2}\log(\frac{4}{\delta})$.*
3. ***Time to reach $\eta-$optimality:** For all $t\geq\max(t_0,T_{\eta\text{-}opt})$, $\hat{\lambda}_t$ satisfy the η-optimality condition in definition 1, when $\widehat{FPR}(\hat{\lambda}_{T_{\eta\text{-}opt}})\in[\alpha-\eta/2,\alpha]$ and $T_{\eta\text{-}opt}=\frac{8C_1}{\gamma\eta^2}\log\left(\frac{4C_2}{\delta}\log(\frac{2C_3}{\eta})\right)+\frac{1}{\gamma^2}\log(\frac{4}{\delta})$.*

Proof. To prove this we first obtain confidence intervals on FPR valid w.p. $1-\delta/2$ using Lemma 4. Then applying Lemma 5 on these confidence intervals gives us the number of OOD samples that are sufficient to guarantee certain width of the confidence intervals and lastly we use Lemma 6 to bound the time point such that we observe a certain

number of OOD points till that time. We do this for the second and third points separately each time invoking Lemma 6 with failure probability $\delta/4$ and then union bound over them.

Controlled FPR: This follows from Lemma 4 (with probability $1 - \frac{\delta}{2}$) and the fact the algorithm uses confidence intervals on FPR estimate that are valid for all $t \geq t_0$ for the choices of λ it considers.

Time to reach feasibility: Applying Lemma 5 with $\mu = \alpha$ gives bound on N_{T_f} with $C_1 = 10(C_0 + 1), C_2 = |\Lambda|, C_3 = 5(C_0 + 1)$. Then using Lemma 6 with $k = N_{T_f}$ gives us the desired T_f.

Time to reach η-optimality : We know, $\text{FPR}(\lambda^*) = \alpha$, and it is given that $\widehat{\text{FPR}}(\hat{\lambda}_t) \in [\text{FPR}(\lambda^*) - \eta/2, \alpha]$

$$\text{FPR}(\hat{\lambda}_t) \in [\text{FPR}(\lambda^*) - \eta/2 - \psi(t, \delta), \alpha]$$

this means $\text{FPR}(\hat{\lambda}_t) \geq \text{FPR}(\lambda^*) - \eta/2 - \psi(t, \delta)$

$$\text{FPR}(\lambda^*) - \text{FPR}(\hat{\lambda}_t) \leq \eta/2 + \psi(t, \delta)$$

If $\psi(t, \delta) \leq \eta/2$ we have, $\text{FPR}(\lambda^*) - \text{FPR}(\hat{\lambda}_t) \leq \eta$. Thus applying we want to find t for which $\psi(t, \delta) = \eta/2$. Applying lemma 5 with $\mu = \eta/2$ gives bound on N_{T_f} with $C_1 = 40(C_0 + 1), C_2 = |\Lambda|, C_3 = 10(C_0 + 1)$. Then using Lemma 6 with $k = N_{T_{opt}}$ gives us the desired T_{opt}. □

This concludes the proofs of the main results. Next, we present details of the procedure used to solve the optimization problem P2 and additional experiments on synthetic and real datasets.

7.3 Additional Details of the Algorithm

We use the following algorithm (Algorithm 2) based on binary search to solve the optimization problem P2 i.e., find $\hat{\lambda}_t$. In addition to the best estimate of current threshold $\hat{\lambda}_t$ it also returns a flag $feasible$ that indicates whether the procedure found a threshold satisfying the constraint in P2 or not.

Algorithm 2 SolveOptForLambda

Input: FPR threshold α , S_t
$low = 1, high = \frac{\Lambda_{\max} - \Lambda_{\min}}{\nu}, feasible = False$
while $low < high$ **do**
  $mid = \lceil (low + high)/2 \rceil$
  $\lambda_{\text{mid}} = \Lambda_{\min} + k\nu$
  if $\widehat{\text{FPR}}(\lambda_{\text{mid}}, t) + \psi(t, \delta) \leq \alpha$ **then**
    $feasible = True$
    $high = mid$
  else
    $low = mid$
  end if
end while
Output $feasible, \lambda_{\text{mid}}$

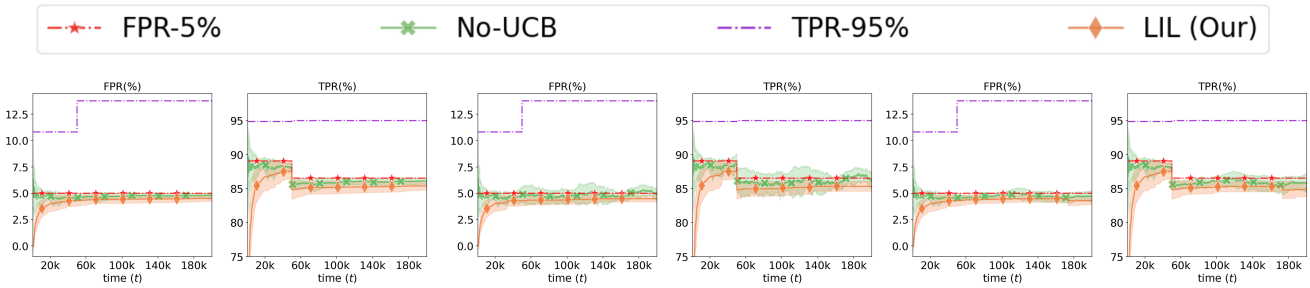
7.4 Additional Experiments and Details

In the simulations we study the effect of changing γ, using different window sizes and the case when the In-Distribution shifts. For the real data experiments, we study the performance of the methods under different settings with different scoring functions on CIFAR-10 and CIFAR-100 as In-Distribution datasets.

7.4.1 Searching for Constants in LIL-Heuristic

The theoretical LIL bound in eq. (2) has constants that can be pessimistic in practice. We get around this by using a LIL-Heuristic bound which has the same form as in eq. (2) but with different constants in particular we consider the form in eq. (LIL-Heuristic). We find the constants C_1, C_2, C_3 using a simulation on estimating the bias of a coin with different constants and picking the ones so that the observed failure probability is below 5%.

$$\tilde{\psi}(t, \delta) = C_1 \sqrt{\frac{c_t}{N_t^{(o)}} \left(\log\log\left(C_2 c_t N_t^{(o)}\right) + \log\left(\frac{C_3}{\delta}\right) \right)}. \qquad \text{(LIL-Heuristic)}$$



(a) No window. (b) 5k window. (c) 10k window.

Figure 10: Changing ID distribution in synthetic data.

Specifically, we keep $C_3 = 1$, and run for $\delta \in \{0.01, 0.05, 0.1, 0.2, 0.3, 0.4\}$ with varying C_1 from 0.1 to 0.9 and C_2 from 1.5 to 4.75. For each choice of δ, C_1, C_2, we toss an unbiased coin (mean $p = 0.5$) for $T = 10k$ times. For each choice of $t = 1, 2, \cdots, T$, we compute the empirical mean $\hat{p}$ of the coin and define it as a failure if $p \notin [\hat{p} - \tilde{\psi}(t, \delta), \hat{p} + \tilde{\psi}(t, \delta)]$. We run this process for 100 times and compute the average failure probability for each choice of $t = 1, 2, \cdots, T$. We then pick the constant so that the observed average probability is below 5%. Throughout the paper, we use $C_1 = 0.5$ and $C_2 = 0.75$.

7.4.2 Additional Experiments on Synthetic Data

In-Distribution shift : We study the scenario when the ID distribution changes and the OOD distribution remains fixed. In this setting the FPR for any threshold does not change since the OOD does not change but the TPR changes due to the change in ID distribution. Thus, we expect that with this type of change our method will not violate FPR constraint and it will gradually adapt to the threshold achieving the new TPR.

We simulate the OOD and ID scores using a mixture of two Gaussians $\mathcal{N}_{id}(\mu = 5.5, \sigma = 4)$ and $\mathcal{N}_{ood}(\mu = -5, \sigma = 4)$ with $\gamma = 0.2$. To simulate distribution change we change the ID distribution to $\mathcal{N}_{id}(\mu = 5, \sigma = 4)$ at time $t = 50k$. We run the methods 10 times with different random seeds. The results are shown in Figure 10. We can clearly see that changing ID distribution(ID scores getting closer to the OOD scores) leads to a decrease in the TPR at the threshold with 5% FPR. Since the estimation of threshold only depends on the FPR estimates and hence only on OOD samples, changing ID distribution does not affect this estimation so the methods perform the same as in the setting of no-distribution shift but get a reduction in the TPR at FPR 5%.

7.4.3 Additional Real OOD Datasets Experiments

We run our proposed system on real ID and OOD datasets with various OOD scoring methods. We use $\alpha = 0.05$, $\delta = 0.2$, and importance sampling probability $p = 0.2$ through all the experiments.

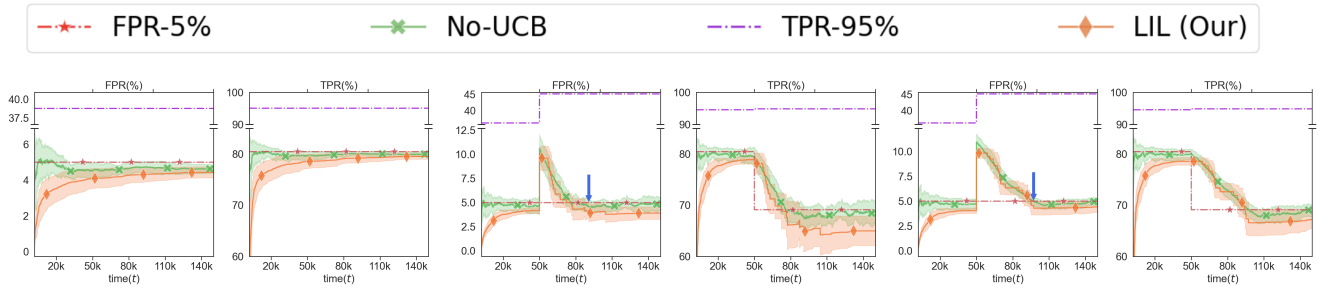
ID and OOD datasets. We use CIFAR-10 or CIFAR-100 as ID datasets. In the distribution shift setting, if not specified, we use MNIST, SVHN, and Texture as the first mixture of OOD datasets, and TinyImageNet, Places365, and CIFAR-10/100 as the second mixture of OOD datasets by default. We use a pre-trained Resnet-50 model for SSD method, and Resnet-18 for the rest of the methods.

Scoring functions: We use the following scoring functions,

1. **ODIN**: ODIN (3) takes the soft-max score from DNNs, and scales the score with temperature. A gradient-based input perturbation is also used for better performance. We choose temperature 1000 and input perturbation noise 0.0014, as discussed in (3). The results with this scoring function on CIFAR-10 and CIFAR-100 ID data setting are shown in Figures 15 and 22 respectively.
2. **Mahalanobis Distance (MDS):** For a given point x, the Mahalanobis Distance (MDS) based score is its MD from the closest class conditional distribution. We use the MD-based score as given in (4) for detecting OOD and adversarial samples. They compute the scores using representations from various layers of DNNs and combine them to get a better scoring function. We choose input perturbation noise to be 0.0014. The results with this scoring function on CIFAR-10 and CIFAR-100 ID data setting are shown in Figures 13 and 19 respectively.
3. **Energy Score (EBO):** This score was proposed in (5) and it is well aligned with the probability density of the samples, with low energy implying ID and high energy implying OOD. We choose the temperature parameter to be 1. The results with EBO scoring function on CIFAR-10 and CIFAR-100 ID data setting are shown in Figures 12 and 18 respectively.
4. **SSD**. It is based on computing the Mahalanobis distance in the feature space of the model trained on the unlabeled in-distribution data using self-supervised learning. We use the official implementation of (14). For CIFAR-10, we use the pre-train model they released. For CIFAR-100, We train a Resnet-50 using a contrastive self-supervised

learning loss, SimCLR (32). When calculating the distance-based OOD scores, we use one unsupervised clustering center as the only center for ID distribution for both CIFAR-10 and CIFAR-100. The results with this scoring function on CIFAR-10 and CIFAR-100 ID data setting are shown in Figures 16 and 21 respectively.

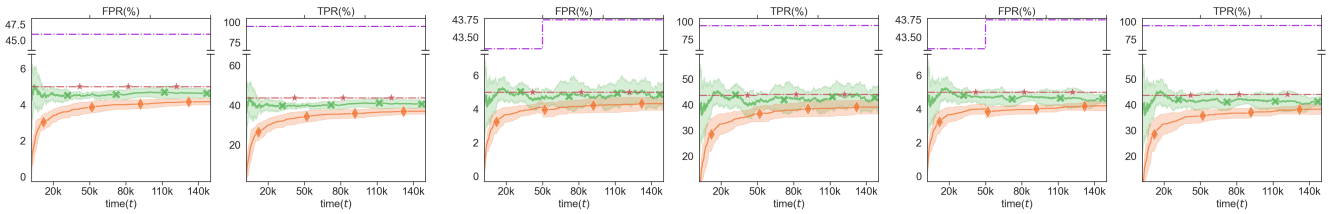
5. **Virtual-logit Match**. Virtual-logit Match (VIM) (15) combines the class-agnostic score from feature space and ID class-dependent logits. Specifically, an additional logit representing the virtual OOD class is generated from the residual of the feature against the principal space and then matched with the original logits by a constant scaling. We set the dimension of the principal space $D = 100$. The results with VIM scoring function on CIFAR-10 and CIFAR-100 ID data setting are shown in Figures 14 and 20 respectively.
6. **K-Nearest-Neighborhood**. Unlike other methods that impose a strong distributional assumption of the underlying feature space, the KNN-based method (16) explores the efficacy of non-parametric nearest-neighbor distance for OOD detection. The distance between the test sample and its k-nearest training IN sample will be used as the score for a threshold based OOD detection. We choose neighbor number $k = 50$. The results with this scoring function on CIFAR-10 and CIFAR-100 ID data setting are shown in Figures 11 and 17 respectively.



(a) No distribution shift, no window. (b) Distribution shift, 5k window. (c) Distribution shift, 10k window.

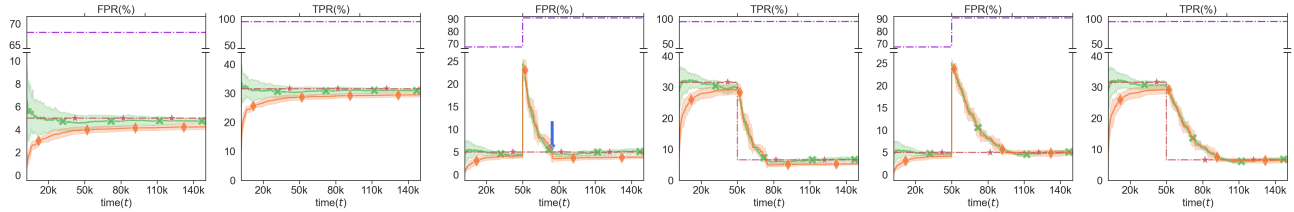
Figure 11: Results with the KNN scores on Cifar-10 as ID dataset. For (b) and (c) the distribution shifts at $t = 50k$. The arrow indicates the time at which the mean FPR + std. deviation over 10 runs goes below 5% for LIL method.

We also provide visualizations showing the distributions of the scores obtained using these scoring functions. Please see Figures 23 to 34.



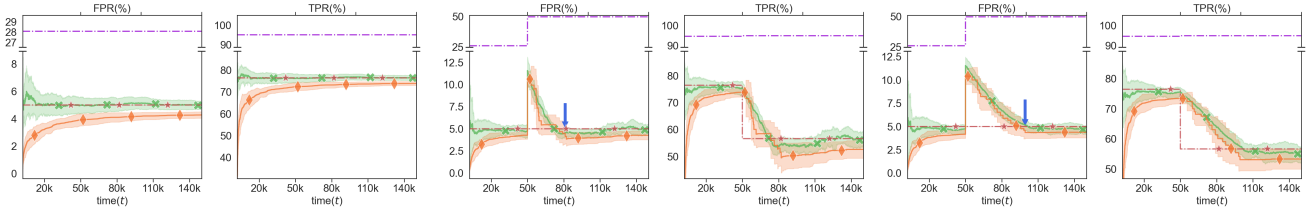
(a) No distribution shift, no window. (b) Distribution shift, 5k window. (c) Distribution shift, 10k window.

Figure 12: Results with the EBO scores on Cifar-10 as ID dataset. For (b) and (c) the distribution shifts at $t = 50k$. The arrow indicates the time at which the mean FPR + std. deviation over 10 runs goes below 5% for LIL method.



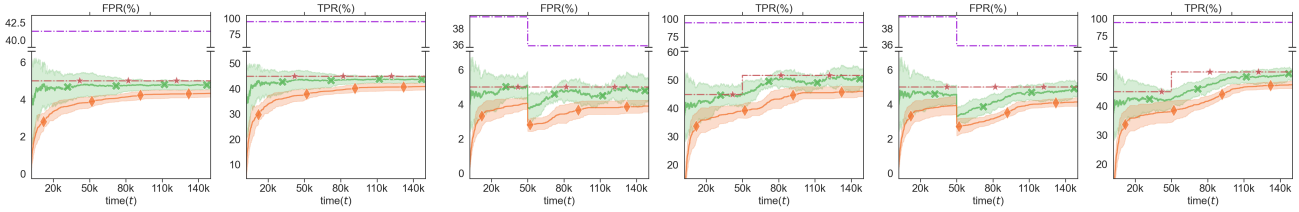
(a) No distribution shift, no window. (b) Distribution shift, 5k window. (c) Distribution shift, 10k window.

Figure 13: Results with the MDS scores on Cifar-10 as ID dataset. For (b) and (c) the distribution shifts at $t = 50k$. The arrow indicates the time at which the mean FPR + std. deviation over 10 runs goes below 5% for LIL method.



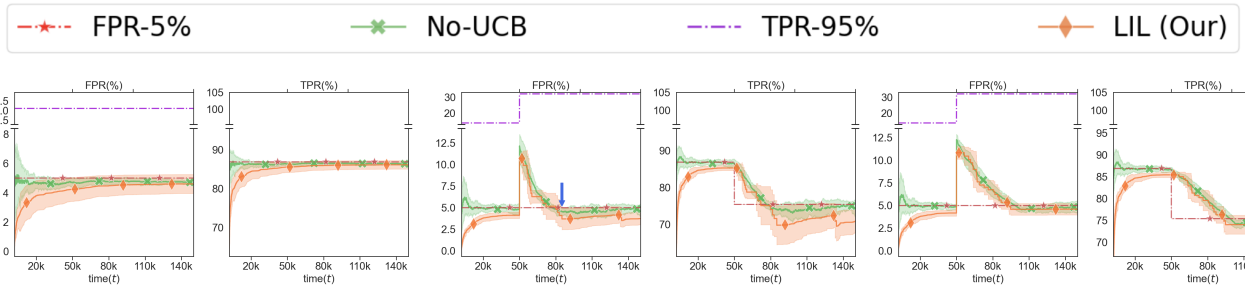
(a) No distribution shift, no window. (b) Distribution shift, 5k window. (c) Distribution shift, 10k window.

Figure 14: Results with the VIM scores on Cifar-10 as ID dataset. For (b) and (c) the distribution shifts at $t = 50k$. The arrow indicates the time at which the mean FPR + std. deviation over 10 runs goes below 5% for LIL method.



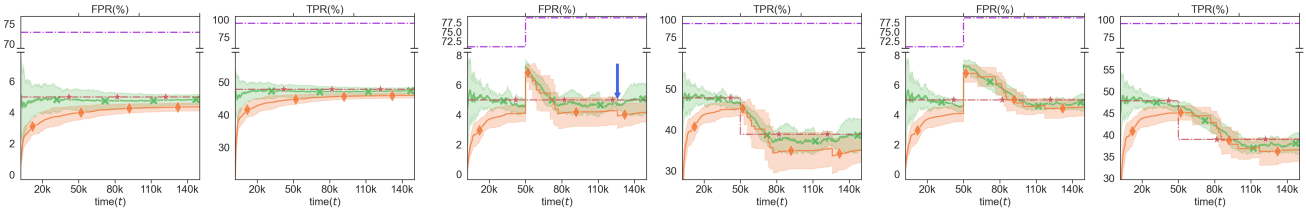
(a) No distribution shift, no window. (b) Distribution shift, 5k window. (c) Distribution shift, 10k window.

Figure 15: Results with the ODIN scores on Cifar-10 as ID dataset. For (b) and (c) the distribution shifts at $t = 50k$. The arrow indicates the time at which the mean FPR + std. deviation over 10 runs goes below 5% for LIL method.



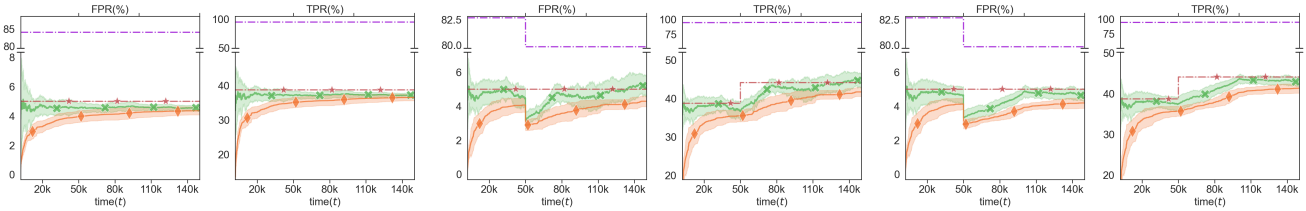
(a) No distribution shift, no window. (b) Distribution shift, 5k window. (c) Distribution shift, 10k window.

Figure 16: Results with the SSD scores on Cifar-10 as ID dataset. For (b) and (c) the distribution shifts at $t = 50k$. The arrow indicates the time at which the mean FPR + std. deviation over 10 runs goes below 5% for LIL method.



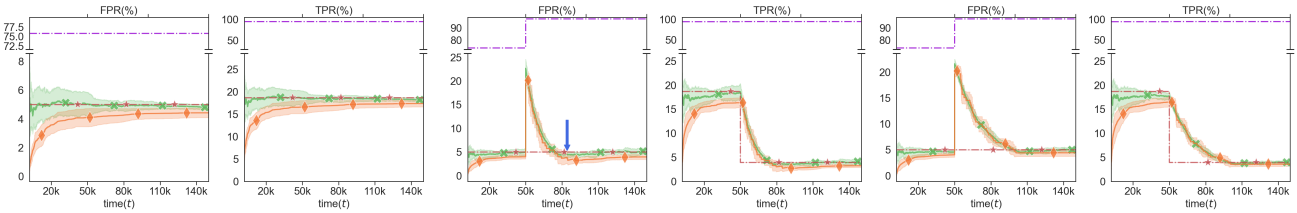
(a) No distribution shift, no window. (b) Distribution shift, 5k window. (c) Distribution shift, 10k window.

Figure 17: Results with the KNN scores on Cifar-100 as ID dataset. For (b) and (c) the distribution shifts at $t = 50k$. The arrow indicates the time at which the mean FPR + std. deviation over 10 runs goes below 5% for LIL method.



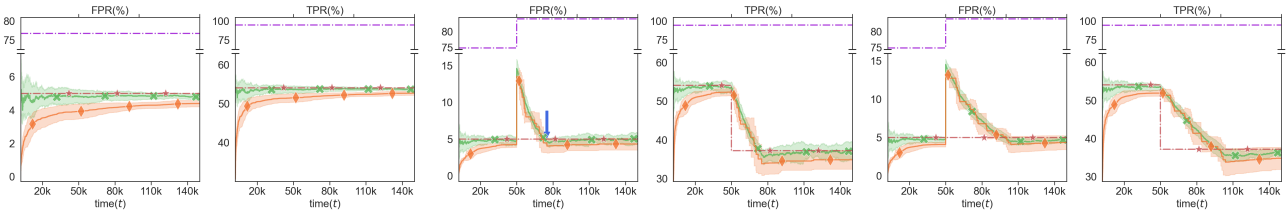
(a) No distribution shift, no window. (b) Distribution shift, 5k window. (c) Distribution shift, 10k window.

Figure 18: Results with the EBO scores on Cifar-100 as ID dataset. For (b) and (c) the distribution shifts at $t = 50k$. The arrow indicates the time at which the mean FPR + std. deviation over 10 runs goes below 5% for LIL method.



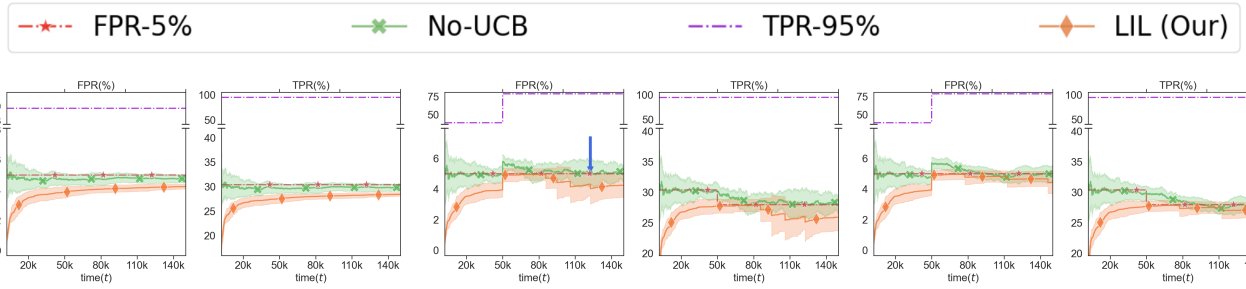
(a) No distribution shift, no window. (b) Distribution shift, 5k window. (c) Distribution shift, 10k window.

Figure 19: Results with the MDS scores on Cifar-100 as ID dataset. For (b) and (c) the distribution shifts at $t = 50k$. The arrow indicates the time at which the mean FPR + std. deviation over 10 runs goes below 5% for LIL method.



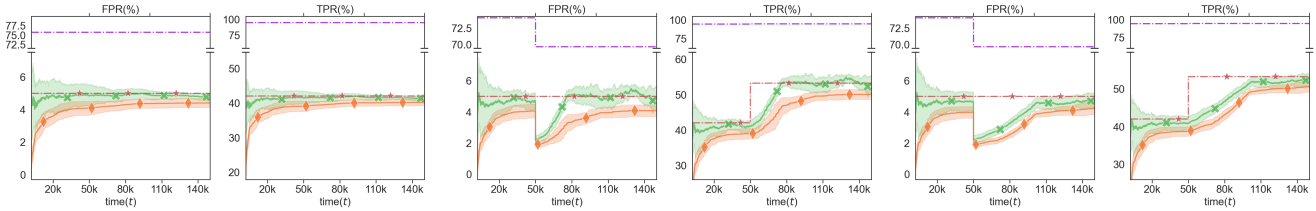
(a) No distribution shift, no window. (b) Distribution shift, 5k window. (c) Distribution shift, 10k window.

Figure 20: Results with the VIM scores on Cifar-100 as ID dataset. For (b) and (c) the distribution shifts at $t = 50k$. The arrow indicates the time at which the mean FPR + std. deviation over 10 runs goes below 5% for LIL method.



(a) No distribution shift, no window. (b) Distribution shift, 5k window. (c) Distribution shift, 10k window.

Figure 21: Results with the SSD scores on Cifar-100 as ID dataset. For (b) and (c) the distribution shifts at $t = 50k$. The arrow indicates the time at which the mean FPR + std. deviation over 10 runs goes below 5% for LIL method.



(a) No distribution shift, no window. (b) Distribution shift, 5k window. (c) Distribution shift, 10k window.

Figure 22: Results with the ODIN scores on Cifar-100 as ID dataset. For (b) and (c) the distribution shifts at $t = 50k$. The arrow indicates the time at which the mean FPR + std. deviation over 10 runs goes below 5% for LIL method.

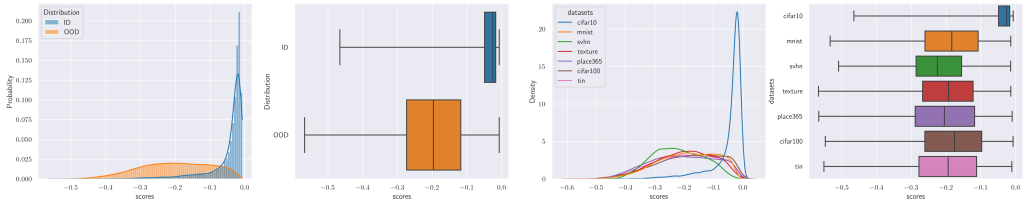


Figure 23: Scores distribution for KNN with CIFAR-10 as In-Distribution.

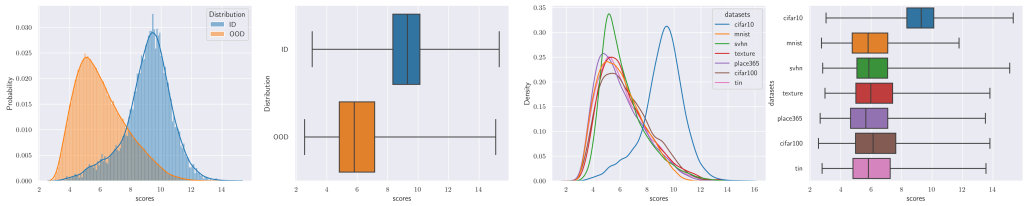


Figure 24: Scores distribution for EBO with CIFAR-10 as In-Distribution.

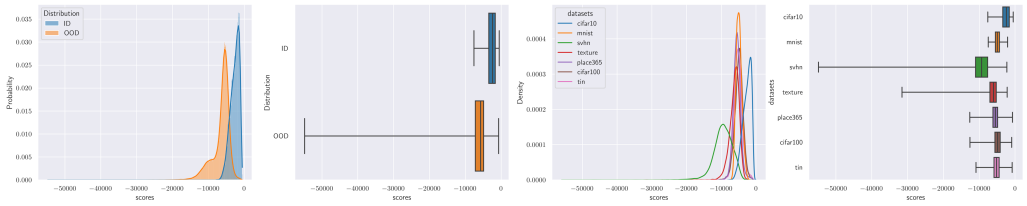


Figure 25: Scores distribution for SSD with CIFAR-10 as In-Distribution.

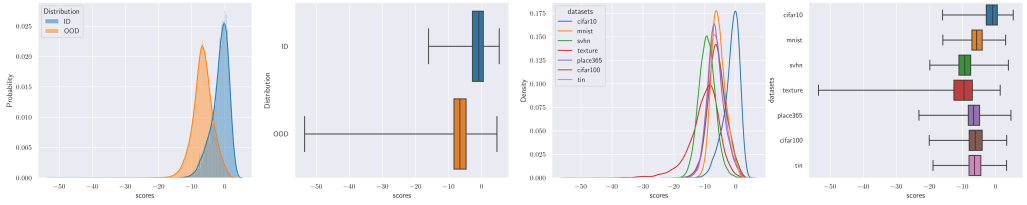


Figure 26: Scores distribution for VIM with CIFAR-10 as In-Distribution.

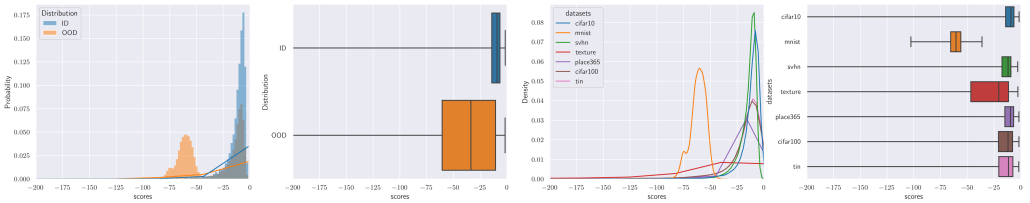


Figure 27: Scores distribution for MDS with CIFAR-10 as In-Distribution.

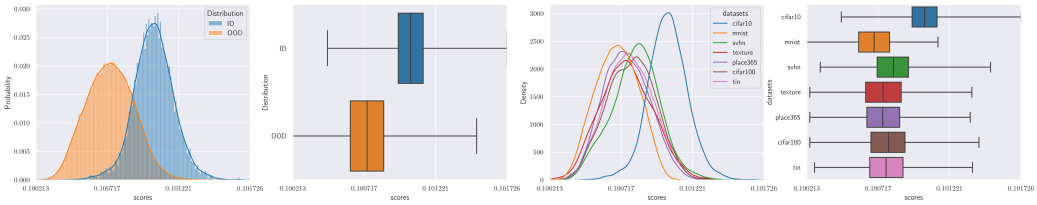


Figure 28: Scores distribution for ODIN with CIFAR-10 as In-Distribution.

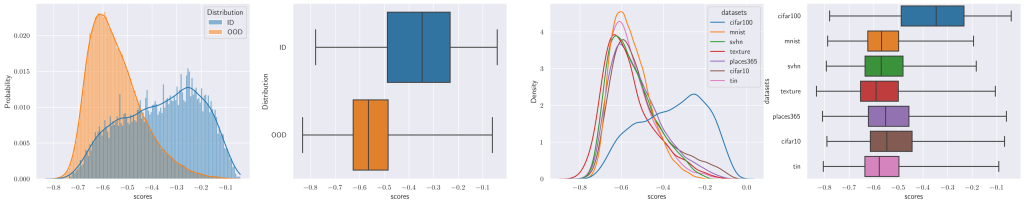


Figure 29: Scores distribution for KNN with cifar-100 as In-Distribution.

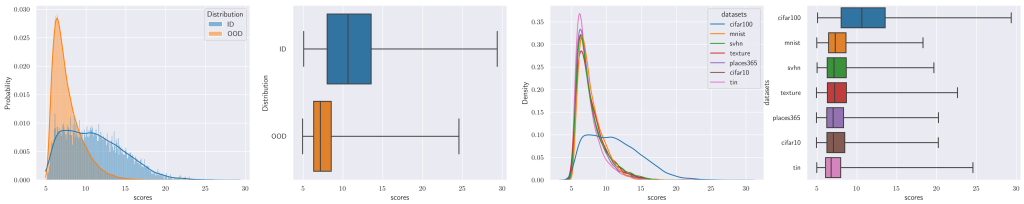


Figure 30: Scores distribution for EBO with cifar-100 as In-Distribution.

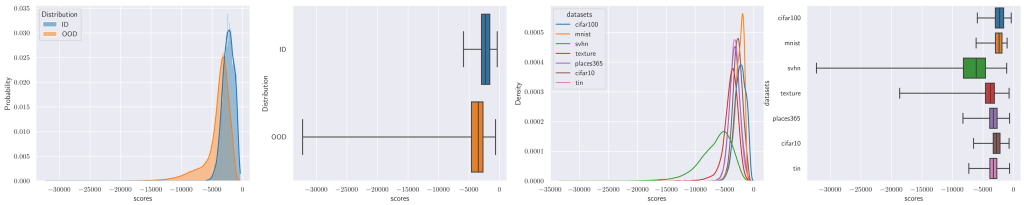


Figure 31: Scores distribution for SSD with cifar-100 as In-Distribution.

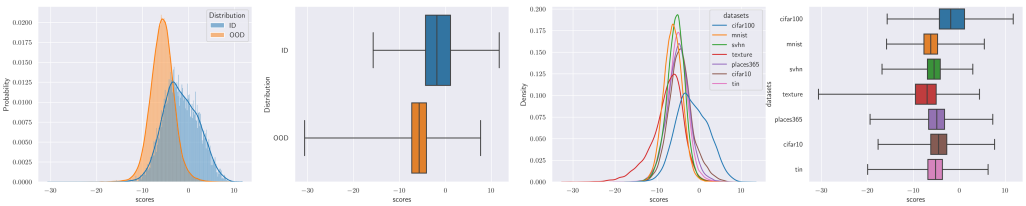


Figure 32: Scores distribution for VIM with cifar-100 as In-Distribution.

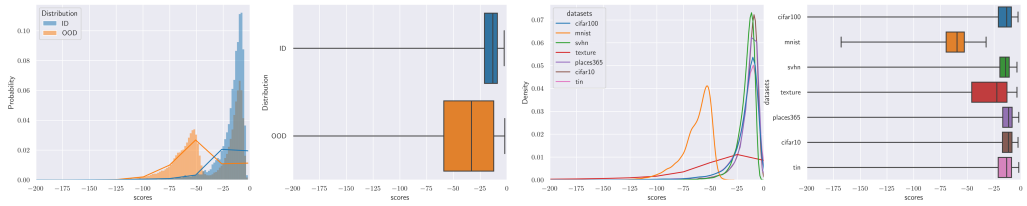


Figure 33: Scores distribution for MDS with cifar-100 as In-Distribution.

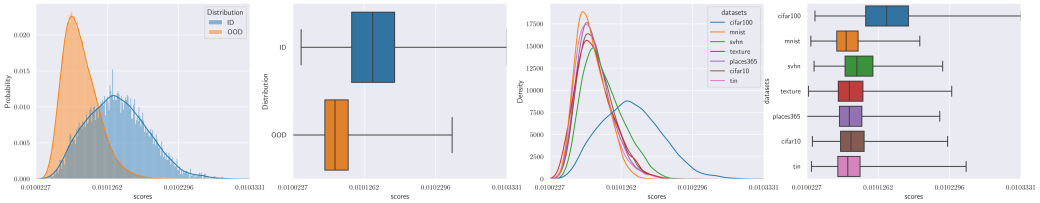


Figure 34: Scores distribution for ODIN with cifar-100 as In-Distribution.