Training OOD Detectors in their Natural Habitats

Julian Katz-Samuels [*1] **Julia Nakhleh** [*2] **Robert Nowak** [3] **Yixuan Li** [2]

Abstract

Out-of-distribution (OOD) detection is important for machine learning models deployed in the wild. Recent methods use auxiliary outlier data to regularize the model for improved OOD detection. However, these approaches make a strong distributional assumption that the auxiliary outlier data is completely separable from the in-distribution (ID) data. In this paper, we propose a novel framework that leverages wild mixture data—that naturally consists of both ID and OOD samples. Such wild data is abundant and arises freely upon deploying a machine learning classifier in their *natural habitats*. Our key idea is to formulate a constrained optimization problem and to show how to tractably solve it. Our learning objective maximizes the OOD detection rate, subject to constraints on the classification error of ID data and on the OOD error rate of ID examples. We extensively evaluate our approach on common OOD detection tasks and demonstrate superior performance.

1. Introduction

Out-of-distribution (OOD) detection has become a central challenge in safely deploying machine learning models in the wild, where test-time data can naturally arise from a mixture distribution of both knowns and unknowns (Bendale & Boult, 2015). Concerningly, modern neural networks are shown to produce overconfident and therefore untrustworthy predictions for unknown 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 (Hendrycks et al., 2019) or higher energy (Liu et al., 2020) on the outlier data. These methods have demonstrated encouraging OOD detection performance over their counterpart (without auxiliary data).

Despite this promise, methods utilizing outlier data impose a strong distributional assumption—the auxiliary outlier data has to be completely separable from the in-distribution (ID) data. This in practice can be restrictive and inflexible, as one needs to perform careful data collection and cleaning. On the other hand, unlabeled in-the-wild data can be collected almost for free upon deploying a machine learning classifier in the open world, and has been largely overlooked for OOD learning purposes. Such data is available in abundance, does not require any human annotation, and is often a much better match to the true test time distribution than data collected offline. While this setting naturally suits many real-world applications, it also poses unique challenges since the wild data distribution is noisy and consists of both ID data and OOD data.

In this paper, we propose a novel framework that enables effectively exploiting unlabeled data in the wild for OOD detection. Unlike the auxiliary outlier data in (Hendrycks et al., 2019), we make use of unlabeled "wild data" that is naturally encountered upon deploying an existing classifier. This can be viewed as training OOD detectors in their *natural habitats*. Our learning framework revolves around building the OOD classifier using only labeled ID data from $\mathbb{P}_{\text{in}}$ and unlabeled wild data from $\mathbb{P}_{\text{wild}}$, which can be considered to be a mixture of $\mathbb{P}_{\text{in}}$ and an unknown (OOD) distribution. To deal with the lack of a "clean" set of OOD examples, our key idea is to formulate a constrained optimization problem. In a nutshell, our learning objective aims to minimize the error of classifying data from $\mathbb{P}_{\text{wild}}$ as ID, subject to two constraints: *(i)* the error of declaring an ID data from $\mathbb{P}_{\text{in}}$ as OOD must be low, and *(ii)* the multi-class classification model must maintain the best-achievable accuracy (or close to it) of a baseline classifier designed without an OOD detection requirement. Even though our framework does not have access to a "clean" OOD dataset, we show both empirically and theoretically that it can learn to accurately detect OOD examples.

Beyond the mathematical framework, a key contribution of our paper is a constrained optimization implementation of the framework for deep neural networks. We propose a novel training procedure based on the augmented Lagrangian method, or ALM (Hestenes, 1969). While ALM

[*]Equal contribution [1]Institute for Foundations of Data Science, University of Wisconsin, Madison [2]Department of Computer Sciences, University of Wisconsin, Madison [3]Department of Electrical and Computer Engineering, University of Wisconsin, Madison. Correspondence to: Julian Katz-Samuels <katzsamuels@wisc.edu>.

is an established approach to optimization with functional constraints, its adaptation to modern deep learning is not straightforward or common. In particular, we adapt ALM to our problem setting with inequality constraints, obtaining an end-to-end training algorithm using stochastic gradient descent. Unlike methods that add a regularization term to the training objective, our method via constrained optimization offers strong guarantees (*c.f.* Section 3.1).

We extensively evaluate our approach on common OOD detection tasks and establish state-of-the-art performance. For completeness, we compare with two groups of approaches: (1) trained with only $\mathbb{P}_{\text{in}}$ data and (2) trained with both $\mathbb{P}_{\text{in}}$ and an auxiliary dataset. Compared to a strong baseline using only $\mathbb{P}_{\text{in}}$, our method outperforms by **6.98%** on FPR95, averaged across all test datasets. The performance gain precisely demonstrates the advantage of incorporating unlabeled wild data. Our method also outperforms Outlier Exposure (OE) (Hendrycks et al., 2019) by **8.80%** in FPR95. Our key contributions are summarized as follows:

- We propose a novel OOD detection framework via constrained optimization (dubbed WOODS, Wild OOD detection sans-Supervision), capable of exploiting unlabeled wild data. We show how to integrate constrained optimization into modern deep nets and solve it tractably.
- We provide novel theoretical insights that support WOODS, in particular for the choice of loss functions.
- We perform extensive ablations and comparisons under: (1) a diverse range of datasets, (2) different mixture ratios π of ID and OOD in $\mathbb{P}_{\text{wild}}$, and (3) different assumptions on the relationship between the wild distribution $\mathbb{P}_{\text{wild}}$ and the test-time distribution. Our method establishes *state-of-the-art* results, significantly outperforming existing methods.

2. Problem Setup

Labeled In-distribution Data Let $\mathcal{X} = \mathbb{R}^d$ denote the input space and $\mathcal{Y} = \{1, \dots, K\}$ denote the label space. We assume access to the labeled training set $\mathcal{D}_{\text{in}}^{\text{train}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, drawn *i.i.d.* from the joint data distribution $\mathbb{P}_{\mathcal{X}\mathcal{Y}}$. Let $\mathbb{P}_{\text{in}}$ denote the marginal distribution on $\mathcal{X}$, which is also referred to as the *in-distribution* (ID). Let $f_\theta : \mathcal{X} \mapsto \mathbb{R}^{|\mathcal{Y}|}$ denote a function for the classification task, which predicts the label of an input sample.

Out-of-distribution Detection When deploying machine learning models in the real world, a reliable classifier should not only accurately classify known ID samples, but also identify as "unknown" any *out-of-distribution* (OOD) input—samples from a different distribution $\mathbb{P}_{\text{out}}^{\text{test}}$ that the model has not been exposed to during training. This can be achieved through having an OOD classifier, in addition to the multi-class classifier f_θ. Samples detected as OOD will be rejected; samples detected as ID will be classified by f_θ.

OOD detection can be formulated as a binary classification problem. At test time, the goal is to decide whether a test-time input $\mathbf{x} \in \mathcal{X}$ is from the in-distribution $\mathbb{P}_{\text{in}}$ (ID) or not (OOD). We denote $g_\theta : \mathcal{X} \mapsto \{\text{in}, \text{out}\}$ as the function mapping for OOD detection.

Unlabeled in-the-wild Data A major challenge in OOD detection is the lack of labeled examples of OOD. In particular, the sample space for potential OOD data can be prohibitively large, making it expensive to collect labeled OOD data. In this paper, we incorporate unlabeled in-the-wild samples $\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_m$ into OOD detection. These samples consist of potentially both ID and OOD data, and can be collected almost for free upon deploying an existing classifier (say f_θ) in its natural habitats. We use the Huber contamination model (Huber, 1964) to model the marginal distribution of the wild data:

$$\mathbb{P}_{\text{wild}} := (1 - \pi)\mathbb{P}_{\text{in}} + \pi\mathbb{P}_{\text{out}},$$

where $\pi \in (0, 1]$.

Goal: Our learning framework revolves around building the OOD classifier g_θ and multi-class classifier f_θ by leveraging data from both $\mathbb{P}_{\text{in}}$ and $\mathbb{P}_{\text{wild}}$. We use the shared parameters θ to indicate the fact that they may share the neural network parameters. In testing, we measure the following errors:

$$\begin{aligned}
&\downarrow \text{FPR}(g_\theta) := \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\text{out}}^{\text{test}}}(\mathbb{1}\{g_\theta(\mathbf{x}) = \text{in}\}),\\
&\uparrow \text{TPR}(g_\theta) := \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\text{in}}}(\mathbb{1}\{g_\theta(\mathbf{x}) = \text{in}\}),\\
&\uparrow \text{Acc}(f_\theta) := \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}_{\mathcal{X}\mathcal{Y}}}(\mathbb{1}\{f_\theta(\mathbf{x}) = y\}),
\end{aligned}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function and the arrows indicate higher/lower is better. In reality, the test-time OOD distribution $\mathbb{P}_{\text{out}}^{\text{test}}$ may or may not be identical to $\mathbb{P}_{\text{out}}$, and we will consider both cases later in Sections 5 and A.2.

3. Method: Out-of-distribution Learning via Constrained Optimization

In this section, we present a novel framework that performs out-of-distribution learning with the unlabeled data in the wild. Our framework offers substantial advantages over the counterpart approaches that relies only on the ID data, and naturally suits many applications where machine learning models are deployed in the open world.

To exploit the in-the-wild data, our key idea is to formulate a constrained optimization problem (Section 3.1). Moreover,

we show how to integrate this constrained optimization problem into modern neural networks and solve it tractably using the Augmented Lagrangian Method (Section 3.2).

3.1. Learning Objective

In a nutshell, we formulate the learning objective by aiming to minimize the error of classifying data from $\mathbb{P}_{\text{out}}$ as ID, subject to *(i)* the error of declaring an ID as OOD is at most α, and *(ii)* the multi-class classification model meets some error threshold τ. Mathematically, this can be formalized as a constrained optimization problem:

Objective Overview Given $\alpha, \tau \in [0, 1]$, we aim to optimize:

$$\begin{aligned} &\inf_{\theta} \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\text{out}}}(\mathbb{1}\{g_\theta(\mathbf{x}) = \text{in}\}) \\ &\text{s.t. } \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\text{in}}}(\mathbb{1}\{g_\theta(\mathbf{x}) = \text{out}\}) \leq \alpha \\ &\quad \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}_{\mathcal{X}\mathcal{Y}}}(\mathbb{1}\{f_\theta(\mathbf{x}) \neq y\}) \leq \tau. \end{aligned} \tag{1}$$

However, we never observe a clean dataset from $\mathbb{P}_{\text{out}}$, making it difficult to directly solve (1). To sidestep this issue, we reformulate the learning objective as follows:

$$\begin{aligned} &\inf_{\theta} \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\text{wild}}}(\mathbb{1}\{g_\theta(\mathbf{x}) = \text{in}\}) \\ &\text{s.t. } \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\text{in}}}(\mathbb{1}\{g_\theta(\mathbf{x}) = \text{out}\}) \leq \alpha \\ &\quad \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}_{\mathcal{X}\mathcal{Y}}}(\mathbb{1}\{f_\theta(\mathbf{x}) \neq y\}) \leq \tau. \end{aligned} \tag{2}$$

where we replaced the OOD distribution $\mathbb{P}_{\text{out}}$ in the objective with $\mathbb{P}_{\text{wild}}$, a distribution that we observe an *i.i.d.* dataset from. As we explain in more detail shortly, optimizing (2) is sufficient under mild conditions to solve the original optimization problem (1).

Empirically, we can solve the optimization problem (2) by minimizing the number of samples $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_m$ from the wild distribution $\mathbb{P}_{\text{wild}}$ that are labeled as ID, subject to *(i)* labeling at least $1-\alpha$ of the ID samples $\mathbf{x}_1 \ldots, \mathbf{x}_n$ correctly, and *(ii)* achieving the classification performance threshold. Equivalently, we consider solving:

$$\begin{aligned} &\inf_{\theta} \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}\{g_\theta(\tilde{\mathbf{x}}_i) = \text{in}\} \\ &\text{s.t. } \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{g_\theta(\mathbf{x}_i) = \text{out}\} \leq \alpha \\ &\quad \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{f_\theta(\mathbf{x}_i) \neq y_i)\} \leq \tau. \end{aligned} \tag{3}$$

Surrogate Problem Note that the above objective in (3) is intractable due to the $0/1$ loss and here we propose a tractable relaxation, replacing the $0/1$ loss with a surrogate loss as follows:

$$\begin{aligned} &\inf_{\theta} \frac{1}{m}\sum_{i=1}^{m} \mathcal{L}_{\text{ood}}(g_\theta(\tilde{\mathbf{x}}_i), \text{in}) \\ &\text{s.t.} \frac{1}{n}\sum_{j=1}^{n} \mathcal{L}_{\text{ood}}(g_\theta(\mathbf{x}_j), \text{out}) \leq \alpha \\ &\quad \frac{1}{n}\sum_{j=1}^{n} \mathcal{L}_{\text{cls}}(f_\theta(\mathbf{x}_j), y_j) \leq \tau \end{aligned} \tag{4}$$

where $\mathcal{L}_{\text{ood}}$ denotes the loss of the binary OOD classifier and $\mathcal{L}_{\text{cls}}$ denotes a loss for the classification task. Our framework is general and can be instantiated by different forms of loss functions, for which we describe details later in Section 4.

Here, we state a theoretical result justifying the optimization problem (4) where we use tractable losses, specifically using the sigmoid loss $\sigma(t) = \frac{1}{1+e^{-t}}$ for $\mathcal{L}_{\text{ood}}$ and the hinge loss for $\mathcal{L}_{\text{cls}}$.[1] We suppose that the weights are p-dimensional and belong to a subset $\theta \in \Theta \subset \mathbb{R}^p$. Let $\texttt{opt}$ denote the value to the population-level optimization problem of interest:

$$\begin{aligned} &\inf_{\theta\in\Theta} \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\text{out}}}\mathcal{L}_{\text{ood}}(g_\theta(\tilde{\mathbf{x}}_i), \text{in}) \\ &\text{s.t. } \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\text{in}}}\mathcal{L}_{\text{ood}}(g_\theta(\mathbf{x}_j), \text{out}) \leq \alpha \\ &\quad \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}_{\mathcal{X}\mathcal{Y}}}\mathcal{L}_{\text{cls}}(f_\theta(\mathbf{x}_j), y_j) \leq \tau. \end{aligned} \tag{5}$$

Proposition 3.1. *Suppose* $K = 2$. *Suppose* $\mathcal{L}_{ood}(t, in) = \sigma(-t)$, $\mathcal{L}_{ood}(t, out) = \sigma(t)$, *and* $\mathcal{L}_{cls}(t, y) = max(1 - ty)$. *Define* $\epsilon_k := \sqrt{\frac{2\ln(6/\delta)}{k}} + 2 \max_{h\in\{f,g\}} \max_{\mathbb{P}\in\{\mathbb{P}_1,\mathbb{P}_{wild}\}} \mathbb{E}_{\tilde{\mathbf{x}}_1,\ldots,\tilde{\mathbf{x}}_m\sim\mathbb{P}} \mathbb{E}_{\eta_1,\ldots,\eta_m} \sup_{\theta\in\Theta} \frac{1}{k}\sum_{i=1}^{k} \eta_i h_\theta(\tilde{\mathbf{x}}_i)$ *where* $\eta_1, \ldots, \eta_k$ *are* i.i.d. *and* $\mathbb{P}(\eta_i = 1) = \mathbb{P}(\eta_i = -1) = 1/2$. *Let* $\widehat{\theta}_\epsilon$ *solve* (4) *with some tolerance* ϵ *(see the Appendix for a formal statement). Then, there exist universal positive constants* c_1, c_2, c_3 *such that under a mild condition (see the appendix), with probability at least* $1 - \delta$

1. $\mathbb{E}_{out}\sigma(-\widehat{g}_{\widehat{\theta}_\epsilon}(\mathbf{x})) \leq \texttt{opt} + c_1\pi^{-1}(\epsilon_n + \epsilon_m)$,
2. $\mathbb{E}_{in}\sigma(g_{\widehat{\theta}_\epsilon}(\mathbf{x})) \leq \alpha + c_2\epsilon_n$, *and*
3. $\mathbb{E}_{\mathcal{X}\mathcal{Y}}\mathcal{L}_{cls}(f_{\widehat{\theta}_\epsilon}(\mathbf{x}), y) \leq \tau + c_3\epsilon_n$.

The above Proposition shows that as long as the Rademacher complexities of the function classes $\{g_\theta : \theta \in \Theta\}$ and $\{f_\theta : \theta \in \Theta\}$ decay at a suitable rate, then solving (4) gives a solution that approaches feasibility and optimality for (5), the optimization problem of interest. Due to space

[1] In the experiments, we replace the hinge loss for classification with the cross-entropy loss as is standard in deep learning.

constraints, we defer a full Proposition statement with additional details and a proof to the Appendix. The above Proposition extends Theorem 1 of (Blanchard et al., 2010) from the computationally intractable $0/1$ loss to the sigmoid loss, a tractable, differentiable loss that we use in our experiments. In addition, we replace the VC dimension in their Theorem with the Rademacher complexity of the function class, a much more fine-grained measure of the function class complexity. We note that it is also possible to prove an analogue of Proposition 3.1 for the $0/1$ loss.

Having theoretically justified the optimization problem (4), we next show how to optimize it.

3.2. Solving the Constrained Optimization

In this subsection, we first provide background on the Augmented Lagrangian method, and then discuss how it can solve our constrained optimization problem.

Augmented Lagrangian Method (ALM) Augmented Lagrangian method (Hestenes, 1969) is an established approach to optimization with functional constraints. ALM improves over two other related methods for constrained optimization: the penalty method and the method of Lagrangian multipliers. While the penalty method suffers from ill-conditioning (Nocedal & Wright, 2006), the method of Lagrangian multipliers is specific to the convex case (Rockafellar, 1973). In this paper, we adapt ALM to our setting with inequality constraints, and later show that it can be optimized end-to-end with modern neural networks.

To provide some background, we consider the following constrained optimization problem as an example:

$$\min_{\theta \in \mathbb{R}^p} f(\theta) \quad \text{s.t. } c_i(\theta) \leq 0 \, \forall i \in [q], \tag{6}$$

where f and $c_1, \ldots, c_q$ are convex. ALM solves the constrained optimization problem in (6) by converting it into a sequence of unconstrained optimization problems.

Define the the classical augmented Lagrangian (AL) function

$$\mathcal{L}_\beta(\theta, \lambda) = f(\theta) + \sum_{i=1}^{q} \psi_\beta(c_i(\theta), \lambda_i)$$

where

$$\psi_\beta(u, v) = \begin{cases} uv + \frac{\beta}{2}u^2 & \beta u + v \geq 0 \\ -\frac{v^2}{2\beta} & \text{o/w} \end{cases},$$

$\lambda = (\lambda_1, \ldots, \lambda_q)^\top$, and $\beta > 0$. At iteration k, ALM minimizes the augmented Lagrangian function with respect to θ and then performs a gradient ascent update step on λ (Xu, 2017; 2021):

Algorithm 1 `WOODS` (Wild OOD detection sans-Supervision)

Input: $\theta^{(1)}_{(1)}$, $\lambda^{(1)}_{(1)}$ β_1, β_2, epoch length T, batch size B, learning rate μ_1, learning rate μ_2, penalty multiplier γ, `tol`
for epoch $= 1, 2, \ldots$ **do**
  for $t = 1, 2, \ldots, T-1$ **do**
    Sample a batch of data, calculate $\mathcal{L}_\beta^{\text{batch}}(\theta, \lambda)$
    $\theta^{(t+1)}_{(\text{epoch})} \longleftarrow \theta^{(t)}_{(\text{epoch})} - \mu_1 \nabla_\theta \mathcal{L}_\beta^{\text{batch}}(\theta, \lambda)$
  end for
  $\lambda^{(\text{epoch}+1)} \leftarrow \lambda^{(\text{epoch})} + \mu_2 \nabla_\theta \mathcal{L}_\beta(\theta^{(T)}_{(\text{epoch})}, \lambda^{(\text{epoch})})$
  if $\frac{1}{n}\sum_{i=1}^n \mathcal{L}_{\text{ood}}(g_{\theta^{(T)}_{(\text{epoch})}}(\mathbf{x}_i), \text{out}) > \alpha + \texttt{tol}$ **then**
    $\beta_1 \longleftarrow \gamma\beta_1$
  end if
  if $\frac{1}{n}\sum_{i=1}^n \mathcal{L}_{\text{cls}}(f_{\theta^{(T)}_{(\text{epoch})}}(\mathbf{x}_i), y_i) > \tau + \texttt{tol}$ **then**
    $\beta_2 \longleftarrow \gamma\beta_2$
  end if
  $\theta^{(1)}_{(\text{epoch}+1)} \longleftarrow \theta^{(T)}_{(\text{epoch})}$
end for

1. $\theta^{(k+1)} \longleftarrow \text{argmin}_\theta \mathcal{L}_{\beta_k}(\theta, \lambda^{(k)})$

2. $\lambda^{(k+1)} \longleftarrow \lambda^{(k)} + \rho \nabla_\lambda \mathcal{L}_{\beta_k}(\theta^{(k+1)}, \lambda)$

where ρ is a learning rate for the dual variable λ and $\{\beta_k\}_k$ is a sequence of penalty parameters. The sequence of penalty parameters $\{\beta_k\}_k$ may be chosen beforehand or adapted based on the optimization process.

Our Algorithm Algorithm 1 presents our approach to using ALM to solve (4). We define the augmented Lagrangian function as:

$$\begin{aligned}\mathcal{L}_\beta(\theta, \lambda) = &\frac{1}{m}\sum_{i=1}^m \mathcal{L}_{\text{ood}}(g_\theta(\tilde{\mathbf{x}}_i), \text{in}) \\ &+ \psi_{\beta_1}(\frac{1}{n}\sum_{j=1}^n \mathcal{L}_{\text{ood}}(g_\theta(\mathbf{x}_j), \text{out}) - \alpha, \lambda_1) \\ &+ \psi_{\beta_2}(\frac{1}{n}\sum_{j=1}^n \mathcal{L}_{\text{cls}}(f_\theta(\mathbf{x}_j), y_j) - \tau, \lambda_2),\end{aligned}$$

where $\beta = (\beta_1, \beta_2)^\top$.

Adaptation to Stochastic Gradient Descent We show that our framework can be adapted to the stochastic case, which is more amenable for training with modern neural networks. We outline the full process in Algorithm 1. In

each iteration, we calculate the per-batch loss as follows:

$$\mathcal{L}_{\beta}^{\text{batch}}(\theta, \lambda) = \frac{1}{B}\sum_{i\in I}\mathcal{L}_{\text{ood}}(g_\theta(\tilde{\mathbf{x}}_i), \text{in}) \tag{7}$$
$$+ \psi_{\beta_1}(\frac{1}{B}\sum_{j\in J}\mathcal{L}_{\text{ood}}(g_\theta(\mathbf{x}_j), \text{out}) - \alpha, \lambda_1^{(\text{epoch})})$$
$$+ \psi_{\beta_2}(\frac{1}{B}\sum_{j\in J}\mathcal{L}_{\text{cls}}(f_\theta(\mathbf{x}_j), y_j) - \tau, \lambda_2^{(\text{epoch})})],$$

where I and J denote the set of mini-batch of size B, randomly sampled from the wild data and ID data respectively. Since $\psi(u, v)$ is convex in u, by Jensen's inequality, the objective function in (7) is an upper bound on $\mathcal{L}_\beta(\theta, \lambda^{(\text{epoch})})$. This step therefore approximates ALM; indeed, it is not straightforward to adapt ALM to the stochastic case (Yan & Xu, 2020). At the end of the epoch, it performs a gradient ascent update on λ (see line 7). Finally, in lines 9 and 12, it increases the constraints weight penalties β_1 and β_2 by a penalty multiplier $\gamma > 1$.

4. Loss Functions with Neural Networks

In this section, we discuss how to realize our learning framework in the context of modern neural networks. Concretely, we address how to define the loss functions $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{ood}}$.

Classification Loss $\mathcal{L}_{\text{cls}}$ We consider a neural network parameterized by θ, which encodes an input $\mathbf{x} \in \mathbb{R}^d$ to a feature space with dimension r. We denote by $h_\theta(\mathbf{x}) \in \mathbb{R}^r$ the feature vector from the penultimate layer of the network. A weight matrix $\mathbf{W} \in \mathbb{R}^{r\times K}$ connects the feature $h_\theta(\mathbf{x})$ to the output $f_\theta(\mathbf{x}) = \mathbf{W}^\top h_\theta(\mathbf{x}) \in \mathbb{R}^K$. The per-sample classification loss $\mathcal{L}_{\text{cls}}$ can be defined using the cross-entropy (CE) loss:

$$\mathcal{L}_{\text{cls}}(f_\theta(\mathbf{x}), y) = -\log\frac{e^{f_\theta^{(y)}(\mathbf{x})}}{\sum_{j=1}^K e^{f_\theta^{(j)}(\mathbf{x})}}, \tag{8}$$

where $f_\theta^{(y)}(\mathbf{x})$ denotes the y-th element of $f_\theta(\mathbf{x})$ corresponding to the label y.

Binary Loss $\mathcal{L}_{\text{ood}}$ The loss function $\mathcal{L}_{\text{ood}}$ should ideally optimize for the separability between the ID vs. OOD data under some function that captures the data density. However, directly estimating $\log p(\mathbf{x})$ can be computationally intractable as it requires sampling from the entire space $\mathcal{X}$. We note that the log partition function $E_\theta(\mathbf{x}) := \log\sum_{j=1}^K e^{f_\theta^{(j)}(\mathbf{x})}$ is proportional to $\log p(\mathbf{x})$ with some unknown factor, which can be seen from the following:

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}, y)}{p(\mathbf{x})} = \frac{e^{f_\theta^{(y)}(\mathbf{x})}}{\sum_{j=1}^K e^{f_\theta^{(j)}(\mathbf{x})}}.$$

The negative log partition function is also known as the free energy, which was shown to be an effective uncertainty measurement for OOD detection (Liu et al., 2020).

Our idea is to explicitly optimize for a level-set estimation based on the energy function (threshold at 0), where the ID data has negative energy values and vice versa.

$$\begin{aligned}\text{argmin}_\theta \frac{1}{m}\sum_{i=1}^m \mathbb{1}\{E_\theta(\tilde{\mathbf{x}}_i) \le 0\}\\ \text{s.t. } \frac{1}{n}\sum_{j=1}^n \mathbb{1}\{E_\theta(\mathbf{x}_j) \ge 0\} \le \alpha\\ \frac{1}{n}\sum_{j=1}^n \mathcal{L}_{\text{cls}}(f_\theta(\mathbf{x}_j), y_j) \le \tau\end{aligned}$$

Since the $0/1$ loss is intractable, we replace it with the binary sigmoid loss, a smooth approximation of the $0/1$ loss, yielding the following optimization problem:

$$\begin{aligned}\text{argmin}_{\theta, w\in\mathbb{R}} \frac{1}{n}\sum_{i=1}^m \frac{1}{1+\exp(-w\cdot E_\theta(\tilde{\mathbf{x}}_i))}\\ \text{s.t. } \frac{1}{n}\sum_{j=1}^n \frac{1}{1+\exp(w\cdot E_\theta(\mathbf{x}_i))} \le \alpha\\ \frac{1}{n}\sum_{j=1}^n \mathcal{L}_{\text{cls}}(f_\theta(\mathbf{x}_j), y_j) \le \tau.\end{aligned} \tag{9}$$

Here w is a learnable parameter modulating the slope of the sigmoid function. Now we may apply the same approach as in section 3 to solve the constrained optimization problem (9). In effect, we have

$$\mathcal{L}_{\text{ood}}(g_\theta(\tilde{\mathbf{x}}_i), \text{in}) = \frac{1}{1+\exp(-w\cdot E_\theta(\tilde{\mathbf{x}}_i))}$$

This loss function is originally developed in (Du et al., 2022) for model regularization. Our approach has three notable advancements over energy-regularized learning (ERL) (Liu et al., 2020): (1) The loss function of ERL is based on the squared hinge loss and requires tuning two margin hyperparameters. In contrast, our loss with the binary logistic loss is hyperparameter-free, and is easier to use in practice. (2) We consider a more general unsupervised setting where the wild data distribution $\mathbb{P}_{\text{wild}}$ contains both ID and OOD data, whereas ERL assumes having access to an auxiliary outlier dataset that is completely separable from the ID data. Methods including `OE` (Hendrycks et al., 2019) require performing manual data collection and cleaning, which is more restrictive and inconvenient. (3) We formalize the learning objective as a constrained optimization, which offers strong guarantees. In contrast, ERL added an energy-based regularization term to the training objective.

Method	OOD Dataset												Average		Acc.
	SVHN		LSUN-R		LSUN-C		iSUN		Texture		Places365				
	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	
With $\mathbb{P}_{\text{in}}$ only															
MSP	48.49	91.89	52.15	91.37	30.80	95.65	56.03	89.83	59.28	88.50	59.48	88.20	51.04	90.90	**94.84**
ODIN	33.35	91.96	26.62	94.57	15.52	97.04	32.05	93.50	49.12	84.97	57.40	84.49	35.71	91.09	**94.84**
Energy	35.59	90.96	27.58	94.24	8.26	98.35	33.68	92.62	52.79	85.22	40.14	89.89	33.01	91.88	**94.84**
Mahalanobis	12.89	97.62	42.62	93.23	39.22	94.15	44.18	92.66	**15.00**	**97.33**	68.57	84.61	37.08	93.27	**94.84**
GODIN	13.55	97.61	17.93	96.86	17.68	96.93	22.94	96.05	29.43	94.87	41.27	91.49	17.67	96.37	94.48
CSI	17.30	97.40	12.15	98.01	1.95	99.55	8.30	98.61	20.45	95.93	34.95	93.64	15.85	97.19	94.17
With $\mathbb{P}_{\text{in}}$ and $\mathbb{P}_{\text{wild}}$															
OE	12.40	97.39	14.35	97.40	6.13	98.81	17.54	96.97	25.35	94.35	30.27	93.28	17.67	96.36	94.19
Energy (w/ OE)	6.49	98.48	9.58	98.03	2.85	99.35	11.19	97.78	22.68	94.72	23.35	94.32	12.69	97.11	94.67
WOODS (ours)	**5.23**	**98.63**	**4.41**	**99.01**	**1.38**	**99.65**	**4.82**	**98.93**	17.84	96.44	**19.50**	**95.71**	**8.87**	**98.06**	94.78

Table 1. **Main results.** Comparison with competitive OOD detection methods on `CIFAR-10`. For methods using $\mathbb{P}_{\text{wild}}$, we train under the same dataset and same $\pi = 0.1$. ↑ indicates larger values are better and vice versa.

5. Experiments

5.1. Experimental Setup

Datasets Following the common benchmarks in literature, we use `CIFAR-10` and `CIFAR-100` (Krizhevsky et al., 2009) as ID datasets ($\mathbb{P}_{\text{in}}$). For OOD test datasets $\mathbb{P}_{\text{out}}^{\text{test}}$, we use a suite of natural image datasets including `SVHN` (Netzer et al., 2011), `Textures` (Cimpoi et al., 2014), `Places365` (Zhou et al., 2018), `LSUN-Crop` (Yu et al., 2016), `LSUN-Resize` (Yu et al., 2016), and `iSUN` (Xu et al., 2015).

To simulate the wild data $\mathbb{P}_{\text{wild}}$, we mix a subset of ID data (as $\mathbb{P}_{\text{in}}$) with the auxiliary outlier dataset (as $\mathbb{P}_{\text{out}}$) under various $\pi \in \{0.05, 0.1, 0.2, 0.5, 1.0\}$. For $\mathbb{P}_{\text{out}}$, we use the publicly available 300K Random Images (Hendrycks et al., 2019), a subset of the original 80 Million Tiny Images dataset (Torralba et al., 2008). Note that we split `CIFAR` datasets into two halves: 25,000 images as ID training data, and remainder 25,000 used to create the mixture data.

Evaluation Metrics To evaluate the methods, we use the standard measures in the literature: the false positive rate of declaring OOD examples as ID when 95% of ID datapoints are declared as ID (FPR95) and the area under the receiver operating characteristic curve (AUROC).

Training Details For all experiments and methods, we use the Wide ResNet (Zagoruyko & Komodakis, 2016) architecture with 40 layers and widen factor of 2. The model is optimized using stochastic gradient descent with Nesterov momentum (Duchi et al., 2011). We set the weight decay coefficient to be 0.0005, and momentum to be 0.09. Models are initialized with a model pre-trained on the `CIFAR` data and trained for 50 epochs. Our initialization scheme from a pre-trained model naturally suits our setting (*c.f.* Section 2), where an existing classifier in deployment is available.

The initial learning rate is set to be 0.001 and decayed by a factor of 2 after 50%, 75%, and 90% of the epochs. We use a batch size of 128 and a dropout rate 0.3. All training is performed in PyTorch using NVIDIA GeForce RTX 2080 Ti GPUs. Code will be made publicly available online. For optimization in `WOODS`, we vary the penalty multiplier $\gamma \in \{1.1, 1.5\}$ and the dual update learning rate $\mu_2 \in \{0.1, 1, 2\}$. We set `tol` $= 0.05$, $\alpha = 0.05$, and set τ to be twice the loss of the pre-trained model. For validation, we use subsets of the ID data and of the 300K Random Images data. Further details are included in Appendix A.3.

5.2. Results

`WOODS` Achieves Superior Performance We present results in Table 1, where `WOODS` outperforms the state-of-the-art results. Our comparison covers an extensive collection of competitive OOD detection methods. For clarity, we divide the baseline methods into two categories: trained with and without in-the-wild data. For methods using ID data $\mathbb{P}_{\text{in}}$ only, we compare with methods such as `MSP` (Hendrycks & Gimpel, 2017), `ODIN` (Liang et al., 2018), `Mahalanobis` (Lee et al., 2018), and `Energy` (Liu et al., 2020), the model is trained with softmax CE loss, same as in Equation 8. `GODIN` (Hsu et al., 2020) is trained using a DeConf-C loss, which does not involve auxiliary data loss either. We also include the latest development based on self-supervised losses, namely `CSI` (Tack et al., 2020).

Closest to ours are Outlier Exposure (`OE`) (Hendrycks et al., 2019) and energy-based OOD learning method (Liu et al., 2020). These are among the strongest OOD detection baselines, which regularize the classification model by producing lower confidence or higher energy on the auxiliary outlier data. For a fair comparison, all the methods in this group are trained using the same ID and in-the-wild data, under the same mixture π.

We highlight several observations: (1) Methods using wild data $\mathbb{P}_{\text{wild}}$, in general, show strong OOD detection performance over the counterpart (without $\mathbb{P}_{\text{wild}}$). Compared to the strongest baseline `CSI` in the first group, our method

Method	OOD Dataset												Average		Acc.
	SVHN		LSUN-R		LSUN-C		iSUN		Texture		Places365				
	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	
$\pi = 0.05$															
OE	16.62	96.81	19.42	96.63	7.62	98.56	23.31	96.05	28.18	93.90	35.04	92.22	21.70	95.69	94.43
Energy (w/ OE)	13.72	97.07	12.53	97.43	3.82	99.17	15.54	96.95	28.95	92.87	27.39	93.16	16.99	96.11	94.76
WOODS (ours)	**8.98**	**97.98**	**6.81**	**98.56**	**1.79**	**99.56**	**8.52**	**98.32**	**22.84**	**94.95**	**21.80**	**95.03**	**11.79**	**97.40**	**94.90**
$\pi = 0.1$															
OE	12.40	97.39	14.35	97.40	6.13	98.81	17.54	96.97	25.35	94.35	30.27	93.28	17.67	96.36	94.19
Energy (w/ OE)	6.49	98.48	9.58	98.03	2.85	99.35	11.19	97.78	22.68	94.72	23.35	94.32	12.69	97.11	94.67
WOODS (ours)	**5.23**	**98.63**	**4.41**	**99.01**	**1.38**	**99.65**	**4.82**	**98.93**	**17.84**	**96.44**	**19.50**	**95.71**	**8.87**	**98.06**	**94.78**
$\pi = 0.2$															
OE	9.09	98.12	10.69	98.02	5.22	99.01	12.42	97.80	20.14	95.66	27.14	93.89	14.12	97.09	94.05
Energy (w/ OE)	4.32	98.73	5.96	98.67	2.32	99.46	6.64	98.54	17.25	96.21	20.91	95.01	9.57	97.77	94.49
WOODS (ours)	**3.27**	**98.86**	**3.92**	**99.14**	**1.41**	**99.62**	**3.81**	**99.12**	**13.36**	**97.45**	**17.95**	**96.08**	**7.29**	**98.38**	**94.77**
$\pi = 0.5$															
OE	4.03	98.86	6.14	98.73	3.16	99.31	6.49	98.68	11.53	97.58	19.82	95.40	8.53	98.09	94.13
Energy (w/ OE)	**2.00**	**99.26**	4.84	98.93	1.55	**99.59**	5.20	98.85	12.58	97.37	17.02	96.02	7.20	98.34	94.54
WOODS (ours)	2.00	99.00	**3.10**	**99.16**	**1.52**	99.54	**2.99**	**99.16**	**9.16**	**98.11**	**15.33**	**96.45**	**5.68**	**98.57**	**94.73**
$\pi = 1.0$															
OE	**1.60**	**99.33**	3.02	**99.16**	1.78	**99.49**	2.79	**99.21**	6.48	**98.54**	12.42	96.97	4.68	**98.78**	94.62
Energy (w/ OE)	7.07	98.10	2.83	99.02	1.62	99.37	2.85	99.08	**5.88**	98.50	**11.28**	**97.10**	5.26	98.53	94.16
WOODS (ours)	1.68	98.61	**2.29**	99.11	**1.35**	99.47	**2.05**	99.16	6.02	98.43	12.43	96.79	**4.30**	98.59	**94.83**

Table 2. **Effect of π.** A larger π indicates more OOD data in the mixture distribution $\mathbb{P}_{\text{wild}}$. ID dataset is `CIFAR-10`. ↑ indicates larger values are better and vice versa.

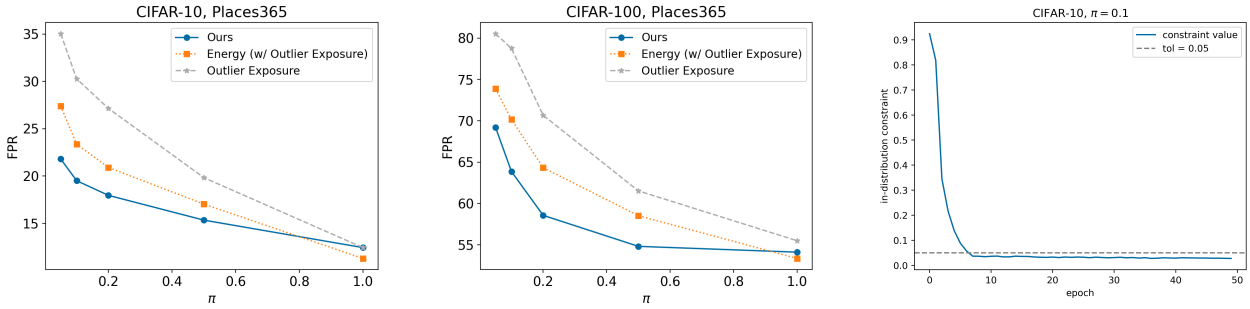

Figure 1: **Left and middle**: Ablation on π for the OOD setting, using `CIFAR-10` (left) and `CIFAR-100` (middle) as the ID dataset and `Places365` as the OOD dataset. Our method `WOODS` is more reliable as π decreases. **Right**: Value of the ID constraint term $\frac{1}{m}\sum_{j=1}^{m}\frac{1}{1+\exp(w\cdot E_\theta(x_i))} - \alpha$ from (9) over different training epochs. Our method is effective in satisfying this constraint, reducing it to zero (within a tolerance of 0.05).

outperforms by **6.98%** in FPR95, averaged across all test datasets. The performance gain precisely demonstrates the advantage of our setting, which incorporates in-the-wild data for effective OOD learning. (2) Compared to methods using $\mathbb{P}_{\text{wild}}$, `WOODS` outperforms `OE` by **8.80%** in FPR95. In particular, `OE` makes a strong distributional assumption that the auxiliary outlier data does not overlap with the ID data. This in practice requires carefully curating and cleaning the auxiliary outlier data. In contrast, our method does not impose such an assumption on $\mathbb{P}_{\text{wild}}$, which can be inherently mixed with ID and outliers. (3) Lastly, the ID accuracy of the model trained with our method is comparable to that using the CE loss alone. Due to space constraints, we provide results on `CIFAR-100` in the Appendix A.1, where our method's strong performance holds.

Effect of π In Table 2 and Figure 1, we ablate the effect under different π, which modulates the fraction of OOD data in the mixture distribution $\mathbb{P}_{\text{wild}}$. Recall our definition in Section 2, a smaller π indicates more ID data and less OOD data—this reflects the practical scenario that the majority of test data may remain ID. We highlight a few interesting observations: (1) The OOD detection performance for all methods (including `OE` and energy regularized learning) generally degrades as with decreasing π. In particular, a smaller π translates into a harder learning problem, because $\mathbb{P}_{\text{in}}$ and $\mathbb{P}_{\text{wild}}$ become largely overlapping. For example, the FPR95 of `OE` increases from 4.68% ($\pi = 1.0$) to 21.70% ($\pi = 0.05$). (3) Our method `WOODS` is overall more robust under small π settings than the baselines. In a challenging case with $\pi = 0.05$, our method outperforms `OE` by **9.91%** in FPR95. This demonstrates the benefits of `WOODS`

performing constrained optimization.

WOODS satisfies the constraint in optimization We also perform a sanity check on whether WOODS satisfies the constraints of the optimization objective in (9). As shown in Figure 1 (right), the ID constraint value $\frac{1}{m}\sum_{j=1}^{m}\frac{1}{1+\exp(w\cdot E_\theta(\mathbf{x}_i))}-\alpha$ is reduced to zero, within a specified tolerance of 0.05. This indeed verifies the efficacy of our constrained optimization framework.

Further Analysis We also perform experiments assessing the performance of our method in the special case where $\mathbb{P}_{\text{out}}=\mathbb{P}_{\text{out}}^{\text{test}}$ (Appendix A.2), a setting that may be suitable in some real-world deployment scenarios. Under this setting, our method achieves very low false positive rate. For example, under the mixing ratio with $\pi=0.1$ (where the mixture data contains only 10% of data from $\mathbb{P}_{\text{out}}$), WOODS obtains FPR95 of 3.13% and AUROC of 99.27%.

6. Related Work

OOD Detection OOD detection is an essential topic for safely deploying machine learning models in the open world, attracting much recent interest with several directions.

1) Some methods aim to design scoring functions for post-hoc detection, including OpenMax score (Bendale & Boult, 2015), Maximum Softmax Probability (Hendrycks & Gimpel, 2017), ODIN score (Liang et al., 2018; Hsu et al., 2020), Mahalanobis distance-based score (Lee et al., 2018), Energy score (Liu et al., 2020; Wang et al., 2021), and gradient norm (Huang et al., 2021). We show that by employing wild data that exists naturally in model's habitats, one can in fact build a stronger OOD detector.

2) Another line of work addresses the OOD detection problem by training-time regularization (Lee et al., 2017; Bevandić et al., 2018; Hendrycks et al., 2019; Malinin & Gales, 2018; Liu et al., 2020). For example, models are encouraged to give predictions with lower confidence (Hendrycks et al., 2019) or higher energies (Liu et al., 2020). These methods typically require access to auxiliary OOD data—a strong assumption in practice. In this work, we instead explore a more realistic setting by training OOD detectors using wild mixtures data containing both ID and OOD data. We formulate a novel constrained optimization problem and show how to solve it tractably with modern neural networks.

Anomaly Detection Anomaly detection has received much attention in recent years (e.g., (Ruff et al., 2018; Chalapathy et al., 2018; Ergen & Kozat, 2019; Perera & Patel, 2019; Song et al., 2017)). In anomaly detection, a dataset is drawn *i.i.d.* from $\mathbb{P}_{\text{in}}$ and the goal is to identify whether new data points are anomalous in the sense that they are not realizations from $\mathbb{P}_{\text{in}}$. In semi-supervised anomaly detection, an additional clean OOD dataset drawn from $\mathbb{P}_{\text{out}}$ is observed (e.g., (Ruff et al., 2019; Daniel et al., 2019; Hendrycks et al., 2019)). An important difference between anomaly detection and the OOD detection literature is that OOD detection additionally requires learning a classifier for the distribution $\mathbb{P}_{\mathcal{X}\mathcal{Y}}$. We refer the reader to (Ruff et al., 2021; Chalapathy & Chawla, 2019) for detailed surveys on anomaly detection.

A closely related paper to our work is (Blanchard et al., 2010), which studies the setting where samples from $\mathbb{P}_{\text{in}}$ and $\mathbb{P}_{\text{wild}}$ are observed and the goal is to find a θ minimizing $\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\text{out}}}(\mathbb{1}\{g_\theta(\mathbf{x})=\text{in}\})$ subject to the constraint that $\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\text{in}}}(\mathbb{1}\{g_\theta(\mathbf{x})=\text{out}\})\leq\alpha$. This work has several important differences with ours. First, they do not consider the out-of-distribution problem, that is, where the distribution at test time $\mathbb{P}_{\text{out}}^{\text{test}}$ differs from $\mathbb{P}_{\text{out}}$, whereas our energy-based approach (9) does. Second, their formulation only considers the task of distinguishing $\mathbb{P}_{\text{out}}$ and $\mathbb{P}_{\text{in}}$, not the task of doing classification simultaneously. Third, our work uses and theoretically analyzes the sigmoid loss for OOD detection, a differentiable loss that can be used in deep learning, whereas their work is mainly statistical, focusing on the computationally intractable 0-1 loss. Finally, their work is mainly statistical, only implementing their algorithm using a plug-in kernel-density estimator whereas we leverage neural networks using a computational approach based on ALM.

Constrained Optimization The augmented Lagrangian method is a popular approach to constrained optimization. It improves over two other related methods: the penalty method and the method of Lagrangian multipliers. While the penalty method suffers from ill-conditioning (Nocedal & Wright, 2006), the method of Lagrangian multipliers is specific to the convex case (Rockafellar, 1973). In this paper, we adapt ALM to our setting from a recent version proposed and analyzed for the case of nonlinear inequality constraints (Xu, 2017). There are only a limited number of examples of adapting ALM to modern neural networks (e.g., (Sangalli et al., 2021) for class imbalance).

7. Conclusion

In this paper, we propose a novel framework for OOD detection using wild data. Wild data has significant promise since it is abundant, can be collected essentially for free upon deploying a ML system, and is often a much better match to the test-time distribution than data collected offline. At the same time, it is challenging to leverage because it naturally consists of both ID and OOD examples. To overcome this challenge, we propose a framework based on constrained optimization and solve it tractably by adapting the augmented Lagrangian method to deep neural networks. We believe that wild data has the potential to dramatically advance OOD detection in practice, thereby helping to accelerate the deployment of safe and reliable machine learning.

References

Bendale, A. and Boult, T. Towards open world recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1893–1902, 2015.

Bevandić, P., Krešo, I., Oršić, M., and Šegvić, S. Discriminative out-of-distribution detection for semantic segmentation. *arXiv preprint arXiv:1808.07703*, 2018.

Blanchard, G., Lee, G., and Scott, C. Semi-supervised novelty detection. *The Journal of Machine Learning Research*, 11:2973–3009, 2010.

Chalapathy, R. and Chawla, S. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

Chalapathy, R., Menon, A. K., and Chawla, S. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, a. A. Describing Textures in the Wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Daniel, T., Kurutach, T., and Tamar, A. Deep variational semi-supervised novelty detection. *arXiv preprint arXiv:1911.04971*, 2019.

Du, X., Wang, Z., Cai, M., and Li, Y. Vos: Learning what you don't know by virtual outlier synthesis. *Proceedings of the International Conference on Learning Representations*, 2022.

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

Ergen, T. and Kozat, S. S. Unsupervised anomaly detection with lstm neural networks. *IEEE transactions on neural networks and learning systems*, 31(8):3127–3141, 2019.

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

Hendrycks, D., Mazeika, M., and Dietterich, T. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019.

Hestenes, M. R. Multiplier and gradient methods. *Journal of optimization theory and applications*, 4(5):303–320, 1969.

Hsu, Y.-C., Shen, Y., Jin, H., and Kira, Z. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10951–10960, 2020.

Huang, R., Geng, A., and Li, Y. On the importance of gradients for detecting distributional shifts in the wild. In *Advances in Neural Information Processing Systems*, 2021.

Huber, P. J. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35:73–101, March 1964.

Krizhevsky, A., Hinton, G., and others. Learning multiple layers of features from tiny images. 2009. Publisher: Citeseer.

Lee, K., Lee, H., Lee, K., and Shin, J. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017.

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*, pp. 7167–7177, 2018.

Liang, S., Li, Y., and Srikant, R. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

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

Malinin, A. and Gales, M. Predictive uncertainty estimation via prior networks. *arXiv preprint arXiv:1802.10501*, 2018.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading Digits in Natural Images with Unsupervised Feature Learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

Nguyen, A., Yosinski, J., and Clune, J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436, 2015.

Nocedal, J. and Wright, S. *Numerical optimization*. Springer Science & Business Media, 2006.

Perera, P. and Patel, V. M. Learning deep features for one-class classification. *IEEE Transactions on Image Processing*, 28(11):5450–5463, 2019.

Rockafellar, R. T. A dual approach to solving nonlinear programming problems by unconstrained optimization. *Mathematical programming*, 5(1):354–373, 1973.

Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., and Kloft, M. Deep one-class classification. In *International conference on machine learning*, pp. 4393–4402. PMLR, 2018.

Ruff, L., Vandermeulen, R. A., Görnitz, N., Binder, A., Müller, E., Müller, K.-R., and Kloft, M. Deep semi-supervised anomaly detection. *arXiv preprint arXiv:1906.02694*, 2019.

Ruff, L., Kauffmann, J. R., Vandermeulen, R. A., Montavon, G., Samek, W., Kloft, M., Dietterich, T. G., and Müller, K.-R. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 2021.

Sangalli, S., Erdil, E., Hötker, A., Donati, O. F., and Konukoglu, E. Constrained optimization to train neural networks on critical and under-represented classes. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Song, H., Jiang, Z., Men, A., and Yang, B. A hybrid semi-supervised anomaly detection model for high-dimensional data. *Computational intelligence and neuroscience*, 2017, 2017.

Tack, J., Mo, S., Jeong, J., and Shin, J. Csi: Novelty detection via contrastive learning on distributionally shifted instances. In *Advances in Neural Information Processing Systems*, 2020.

Torralba, A., Fergus, R., and Freeman, W. 80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(11):1958–1970, November 2008. ISSN 0162-8828. doi: 10.1109/TPAMI.2008.128.

Wang, H., Liu, W., Bocchieri, A., and Li, Y. Can multi-label classification networks know what they don't know? *Advances in Neural Information Processing Systems*, 34, 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:1504.06755 [cs]*, May 2015. arXiv: 1504.06755.

Xu, Y. First-order methods for constrained convex programming based on linearized augmented lagrangian function. *arXiv preprint arXiv:1711.08020*, 2017.

Xu, Y. Iteration complexity of inexact augmented lagrangian methods for constrained convex programming. *Mathematical Programming*, 185(1):199–244, 2021.

Yan, Y. and Xu, Y. Adaptive primal-dual stochastic gradient method for expectation-constrained convex stochastic programs. *arXiv preprint arXiv:2012.14943*, 2020.

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:1506.03365 [cs]*, June 2016. arXiv: 1506.03365.

Zagoruyko, S. and Komodakis, N. Wide Residual Networks. In *Procedings of the British Machine Vision Conference 2016*, pp. 87.1–87.12, York, UK, 2016. British Machine Vision Association. ISBN 978-1-901725-59-9. doi: 10.5244/C.30.87.

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, June 2018. ISSN 1939-3539. doi: 10.1109/TPAMI.2017.2723009. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Method	OOD Dataset												Average		Acc.
	SVHN		LSUN-R		LSUN-C		iSUN		Texture		Places365				
	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	
With $\mathbb{P}_{\text{in}}$ only															
MSP	84.59	71.44	82.42	75.38	66.54	83.79	82.80	75.46	83.29	73.34	82.84	73.78	80.41	75.53	**75.96**
ODIN	84.66	67.26	71.96	81.82	55.55	87.73	68.51	82.69	79.27	73.45	87.88	71.63	74.64	77.43	**75.96**
Energy	85.82	73.99	79.47	79.23	35.32	93.53	81.04	78.91	79.41	76.28	80.56	75.44	73.60	79.56	**75.96**
Mahalanobis	**57.52**	**86.01**	**21.23**	**96.00**	91.18	69.69	**26.10**	**94.58**	**39.39**	90.57	88.83	67.87	54.04	84.12	**75.96**
GODIN	83.38	84.05	62.24	88.22	72.86	83.84	69.16	86.44	83.83	78.91	80.56	76.14	75.34	82.93	75.33
CSI	64.70	84.97	91.55	63.42	38.10	92.52	90.10	65.18	74.70	**92.66**	82.25	73.63	73.57	78.73	69.90
With $\mathbb{P}_{\text{in}}$ and $\mathbb{P}_{\text{wild}}$															
OE	81.04	77.20	73.87	80.32	61.26	86.56	73.61	80.98	74.20	79.42	78.76	75.68	73.79	80.03	72.93
Energy (w/ OE)	80.22	80.12	61.87	86.72	28.00	94.87	64.50	86.04	69.15	82.36	70.16	80.61	62.32	85.12	75.34
WOODS (ours)	71.74	84.17	53.10	89.80	**13.72**	**97.51**	55.84	89.22	62.02	85.51	**63.86**	**83.35**	**53.38**	**88.26**	75.92

Table 3. **Main results.** Comparison with competitive OOD detection methods on `CIFAR-100`. For methods using $\mathbb{P}_{\text{wild}}$, we train under the same dataset and same $\pi = 0.1$. ↑ indicates larger values are better and vice versa.

Method	OOD Dataset												Average		Acc.
	SVHN		LSUN-R		LSUN-C		iSUN		Texture		Places365				
	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	
$\pi = 0.05$															
OE	81.19	78.35	74.54	81.11	62.73	86.48	72.85	82.03	77.39	78.03	80.52	74.97	74.87	80.16	73.18
Energy (w/ OE)	79.64	79.21	67.22	84.97	29.22	94.71	69.45	84.42	71.34	81.20	73.87	79.05	65.12	83.93	75.92
WOODS (ours)	**74.84**	**82.03**	**60.34**	**87.85**	**17.84**	**96.74**	**62.03**	**87.39**	**65.92**	**83.77**	**69.20**	**81.31**	**58.36**	**86.51**	**75.98**
$\pi = 0.1$															
OE	81.04	77.20	73.87	80.32	61.26	86.56	73.61	80.98	74.20	79.42	78.76	75.68	73.79	80.03	72.93
Energy (w/ OE)	80.22	80.12	61.87	86.72	28.00	94.87	64.50	86.04	69.15	82.36	70.16	80.61	62.32	85.12	75.34
WOODS (ours)	**71.74**	**84.17**	**53.10**	**89.80**	**13.72**	**97.51**	**55.84**	**89.22**	**62.02**	**85.51**	**63.86**	**83.35**	**53.38**	**88.26**	**75.92**
$\pi = 0.2$															
OE	72.08	81.44	60.12	84.74	48.52	89.69	59.53	85.02	64.69	82.77	70.66	79.10	62.60	83.79	72.57
Energy (w/ OE)	74.34	83.05	55.78	88.36	22.17	95.98	58.17	87.87	64.86	84.24	64.33	82.92	56.61	87.07	75.04
WOODS (ours)	**71.73**	**85.36**	**47.26**	**91.57**	**11.97**	**97.82**	**51.86**	**90.62**	**60.10**	**86.79**	**58.56**	**85.22**	**50.25**	**89.56**	**75.31**
$\pi = 0.5$															
OE	**68.58**	83.21	45.78	89.81	30.99	94.01	**47.18**	89.56	**55.95**	86.44	61.52	82.96	51.67	87.67	73.13
Energy (w/ OE)	69.94	85.61	51.66	89.56	16.48	96.98	55.79	88.72	57.51	86.84	58.52	85.16	51.65	88.81	74.44
WOODS (ours)	70.36	**86.63**	**41.32**	**92.69**	**12.57**	**97.75**	47.55	**91.59**	58.37	**87.57**	**54.81**	**86.94**	**47.50**	**90.53**	**75.76**
$\pi = 1.0$															
OE	**42.81**	**92.44**	48.35	89.83	20.90	96.32	52.07	88.75	51.60	88.76	55.49	87.34	**45.20**	90.57	74.88
Energy (w/ OE)	52.76	90.67	52.83	89.75	17.50	96.84	55.79	89.25	**50.86**	**89.70**	**53.35**	**88.43**	47.18	90.77	74.56
WOODS (ours)	59.34	89.80	**46.12**	**91.66**	**13.73**	**97.56**	**50.71**	**90.73**	56.33	88.26	54.11	87.61	46.72	**90.94**	**75.75**

Table 4. **Effect of π.** ID dataset is `CIFAR-100`. ↑ indicates larger values are better and vice versa.

A. Additional Experimental Results

A.1. Main Experiments: `CIFAR-100`

Table 3 shows a comparison of our method's performance with OOD baseline methods on `CIFAR-100`. On average, `WOODS` outperforms all of the other methods. It outperforms the fine-tuned methods by large margins, specifically, it outperforms `OE` on average FPR95 by more 20% and Energy by nearly 10%. It outperforms all of the other OOD baseline methods by about 20% except for Malananobis, which comes in a close second. Mahalanobis performs much better on CIFAR-100 than on CIFAR-10. On CIFAR-10, `WOODS` beat Mahalanobis by nearly 30%, but in CIFAR-100 the difference is small. This reflects the fact that the performance of fine-tuned methods such as `WOODS` depends largely on the quality of the auxiliary dataset. As we will see in Section A.2, the performance of `WOODS` improves dramatically when given a better auxiliary dataset, whereas Malananobis cannot make use of such an auxiliary dataset, limiting its potential for improvement.

Table 4 shows the results of ablation on π using `CIFAR-100` as the ID dataset. In general, `WOODS` outperforms OE and Energy by an even larger margin than on CIFAR-10 and across many values of π.

A.2. Special Case where $\mathbb{P}_{\text{out}} = \mathbb{P}_{\text{out}}^{\text{test}}$

Existing approaches for OOD detection typically assume access to a pure auxiliary datset of OOD examples at training time, where these samples are drawn from a different distribution ($\mathbb{P}_{\text{out}}$) than the OOD examples encountered at test time ($\mathbb{P}_{\text{out}}^{\text{test}}$). However, in a real-world deployment scenario, training data is often collected under similar conditions as those in which the

Method	OOD Dataset												Average		Acc.
	SVHN		LSUN-R		LSUN-C		iSUN		Texture		Places365				
	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	
$\pi = 0.1$															
OE	0.85	99.82	0.33	99.93	1.84	99.65	0.41	99.89	10.42	98.01	23.27	94.67	6.19	98.66	94.10
Energy (w/ OE)	4.95	98.92	5.04	98.83	1.93	99.49	7.16	98.50	17.94	95.50	17.04	95.75	9.01	97.83	94.83
WOODS (ours)	0.15	**99.97**	0.03	99.99	0.22	99.94	0.06	99.98	5.93	98.72	12.39	**97.00**	3.13	**99.27**	**94.86**
WOODS w/ NN class (ours)	**0.10**	99.96	**0.02**	**99.99**	**0.08**	**99.96**	**0.04**	**99.99**	**1.93**	**99.28**	**11.46**	96.07	**2.27**	99.21	94.72

Table 5. **Results when $\mathbb{P}_{\text{out}} = \mathbb{P}_{\text{out}}^{\text{test}}$.** ID dataset is `CIFAR-10`. ↑ indicates larger values are better and vice versa.

Method	OOD Dataset												Average		Acc.
	SVHN		LSUN-R		LSUN-C		iSUN		Texture		Places365				
	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	
$\pi = 0.1$															
OE	1.57	99.63	0.93	99.79	3.83	99.26	1.98	99.53	27.53	93.49	59.44	83.52	15.88	95.87	71.55
Energy (w/ OE)	1.47	99.68	2.68	99.50	2.52	99.44	5.64	99.01	33.55	92.23	53.05	86.40	16.48	96.05	73.62
WOODS (ours)	0.52	99.88	0.38	99.92	0.93	99.77	0.60	99.86	17.27	96.57	37.33	**91.07**	9.50	**97.84**	74.79
WOODS w/ NN class (ours)	**0.12**	**99.96**	**0.07**	**99.96**	**0.11**	**99.96**	**0.09**	**99.96**	**8.78**	**96.66**	**29.51**	90.52	**6.45**	97.84	**75.19**

Table 6. **Results when $\mathbb{P}_{\text{out}} = \mathbb{P}_{\text{out}}^{\text{test}}$.** ID dataset is `CIFAR-100`. ↑ indicates larger values are better and vice versa.

system is deployed. In this case, the "wild" auxiliary training data may consist of a mixture of OOD samples from the same distribution as those encountered at test time ($\mathbb{P}_{\text{out}}/\mathbb{P}_{\text{out}}^{\text{test}}$) and ID samples from $\mathbb{P}_{\text{in}}$. We perform an ablation showing that, under this benign setting in which $\mathbb{P}_{\text{out}} = \mathbb{P}_{\text{out}}^{\text{test}}$, `WOODS` (our method) outperforms existing baselines.

We also present results using `WOODS` w/ NN classifier, for which an OOD confidence score is not extracted directly from the output of the ID classifier, but rather learned by a separate neural network attached to the ID classifier's penultimate layer. The additional neural network has one fully-connected hidden layer with 300 neurons, followed by a ReLU activation and a single output logit, which provides an OOD confidence score, denoted $g_\theta(\cdot)$. With this architecture, we apply the same `WOODS` algorithm to solve the following constrained optimization problem:

$$\begin{aligned} \inf_{\theta} & \frac{1}{m}\sum_{i=1}^{m}\max(1-g_\theta(\tilde{\mathbf{x}}_i)),0) \\ \text{s.t. } & \frac{1}{n}\sum_{j=1}^{n}\max(1+g_\theta(\mathbf{x}_j)),0) \leq \alpha \\ & \frac{1}{n}\sum_{j=1}^{n}\mathcal{L}_{\text{cls}}(f_\theta(\mathbf{x}_j),y_j) \leq \tau. \end{aligned}$$

Table 5 and Table 6 show the results using `CIFAR-10` and `CIFAR-100` as the ID datasets, respectively. Here, we only consider the $\pi = 0.1$ setting. `WOODS` and `WOODS` w/ NN classifier substantially outperform Energy and OE. We note that the average FPR95 is misleading because the datasets `SVHN`, `LSUN-Crop`, `LSUN-Resize`, and `iSUN` are all fairly easy even with only access to the mixture $\mathbb{P}_{\text{wild}}$. `Textures` and `Places 365` are significantly more challenging and we see a larger gap between the methods. For example on `CIFAR-100`, `WOODS` outperforms the OE and Energy by about $10-16\%$ and `WOODS` w/ NN classifier outperforms OE and Energy by about $20-24\%$.

While `WOODS` w/ NN classifier has strong performance in the setting where $\mathbb{P}_{\text{out}} = \mathbb{P}_{\text{out}}^{\text{test}}$, we do not expect it to perform as well as `WOODS` in the setting studied previously where $\mathbb{P}_{\text{out}} \neq \mathbb{P}_{\text{out}}^{\text{test}}$. `WOODS` uses the energy score to build its classifier, which already has reasonable performance even without any additional auxiliary dataset. On the other hand, `WOODS` w/ NN classifier would do no better than random guessing without an auxiliary dataset. In this way, `WOODS` has a prior given by the energy score that we believe helps in the $\mathbb{P}_{\text{out}} \neq \mathbb{P}_{\text{out}}^{\text{test}}$ setting. Indeed, we observed this in some experiments.

We wish to also emphasize the substantial improvement of `WOODS` over the baselines using only $\mathbb{P}_{\text{in}}$ data, depicted in Tables 1 and 3. This shows that given access to a good wild distribution $\mathbb{P}_{\text{wild}}$, `WOODS` can use $\mathbb{P}_{\text{wild}}$ data to perform dramatically better than methods using only $\mathbb{P}_{\text{in}}$ data. We argue that a high-quality in-the-wild dataset can be collected almost for free upon deploying a machine learning classifier in the open world and, therefore, that this is a practically relevant setting for OOD detection.

A.3. Additional Experimental Details

Here we give additional experimental details, presented in Sections 5 and A.1. Energy and OE both optimize an objective of the form

$$\min_{\theta} L_{\text{classification}} + \lambda L_{\text{OOD}}$$

For energy, we varied $\lambda \in \{0.1, 1, 5\}$ and for OE we varied $\lambda \in \{0.1, 0.5, 1\}$.

We simulate the mixture distribution as follows. For each iteration at training, for the ID dataset we draw one batch of size 128 and for the wild dataset $\mathbb{P}_{\text{wild}}$ we draw another batch of size 128 where each example is drawn from $\mathbb{P}_{\text{out}}$ with probability π and from $\mathbb{P}_{\text{in}}$ with probability $1 - \pi$.

For the OOD experiments in Sections 5 and A.1, we repeated each experiment 3 times with 3 separate seeds. For the experiments on the setting where $\mathbb{P}_{\text{out}} = \mathbb{P}_{\text{out}}^{\text{test}}$ in Section A.2, we repeated each experiment 3 times with 3 separate seeds.

B. Proof of Proposition 3.1

In this Section, we prove Proposition 3.1. We begin by giving notation and proving an important Lemma. To ease notation, we write $\mathbb{P}_{\text{out}} = \mathbb{P}_1$ and $\mathbb{P}_{\text{in}} = \mathbb{P}_1$. Define the sigmoid loss for the OOD task

$$R_y(g_\theta) := \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\text{out}}}(\sigma(-g_\theta(\mathbf{x})) \cdot y).$$

Define

$$\begin{aligned} R_1^* := \inf_{\theta} \; & R_1(g_\theta) \\ \text{s.t.} & R_{-1}(g_\theta) \leq \alpha. \end{aligned}$$

Define

$$\begin{aligned} R_{\text{wild}} &:= \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\text{wild}}}(\sigma(-g_\theta(\mathbf{x})) \\ &= \pi R_1(g_\theta) + (1-\pi)\mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{-1}}(\sigma(-g_\theta(\mathbf{x})) \\ &= \pi R_1(g_\theta) + (1-\pi)(1 - R_{-1}(g_\theta)) \end{aligned}$$

where we used the symmetry of the sigmoid function, that is, $\sigma(z) + \sigma(-z) = 1$ for $z \in \mathbb{R}$. Now, define

$$\begin{aligned} R_{\text{wild}}^* := \inf_{\theta} \; & R_{\text{wild}}(g_\theta) \\ & \text{s.t. } R_{-1}(g_\theta) \leq \alpha. \end{aligned}$$

Next, we prove a key Lemma for our proof. This Lemma has a similar proof to Theorem 1 in (Blanchard et al., 2010), which applies to the $0/1$ loss. We establish an analogous result for the sigmoid loss. The key observation is that the symmetry property of the sigmoid loss enables a similar proof.

Lemma B.1. *Suppose that there exists θ^* such that $R_1(g_\theta) = R_1^*(g_{\theta^*})$ and $R_0(g_{\theta^*}) = \alpha$. Then,*

$$R_1(g_\theta) - R_1^* \leq \frac{1}{\pi}(R_{\text{wild}}(g_\theta) - R_1^* + (1-\pi)(R_0(g_\theta) - \alpha)).$$

Proof. We begin by showing that for any θ, $R_{\text{wild}}(g_\theta) = R_{\text{wild}}^*$ and $R_0(g_\theta) \leq \alpha$ if and only if $R_1(g_\theta) = R_1^*$ and $R_0(g_\theta) = \alpha$.

$\Longrightarrow$: First, suppose θ satisfies $R_{\text{wild}}(g_\theta) = R_{\text{wild}}^*$ and $R_0(g_\theta) \leq \alpha$. Suppose that either $R_{-1}(g_\theta) < \alpha$ or $R_1(g_\theta) > R_1^*$. By the assumption in the Proposition, there exists θ^* such that $R_1(g_\theta) = R_1^*(g_{\theta^*})$ and $R_0(g_{\theta^*}) = \alpha$. Then, we have that

$$\begin{aligned} R_{\text{wild}}^*(g_{\theta^*}) &= \pi R_1(g_{\theta^*}) + (1-\pi)(1 - R_{-1}(g_{\theta^*}) \\ &\quad + \pi R_1(g_{\theta^*}) + (1-\pi)(1-\alpha) \\ &< \pi R_1(g_\theta) + (1-\pi)(1 - R_{-1}(g_\theta) \\ &= R_{\text{wild}}(g_\theta). \end{aligned}$$

But, this contradicts the assumption that $R_{\text{wild}}(g_\theta) = R^*_{\text{wild}}$, completing this direction of the claim.

$\Longleftarrow$: Suppose θ satisfies $R_1(g_\theta) = R_1^*$ and $R_0(g_\theta) = \alpha$. By the assumption in the Lemma, there exists θ^* such that $R_1(g_\theta) = R_1^*(g_{\theta^*})$ and $R_0(g_{\theta^*}) = \alpha$. Towards a contradiction, suppose that $R_{\text{wild}}(g_{\theta^*}) < R_{\text{wild}}(g_\theta)$. Then, using $\pi > 0$, we have that

$$\begin{aligned} R_1(g_{\theta^*}) &= \frac{1}{\pi}(R_{\text{wild}}(g_{\theta^*}) - (1-\pi)(1 - R_{-1}(g_{\theta^*})) \\ &< \frac{1}{\pi}(R_{\text{wild}}(g_\theta) - (1-\pi)(1-\alpha) \\ &= R_1(g_\theta) \end{aligned}$$

but this contradicts our assumption on g_θ. This completes the proof of the claim.

The established claim implies that $R^*_{\text{wild}} = \pi R_1^* + (1-\pi)(1-\alpha)$. The result now follows by subtracting this equality from $R^*_{\text{wild}}(g_\theta) = \pi R_1(g_\theta) + (1-\pi)(1 - R_{-1}(g_\theta))$.

□

Here we restate Proposition 3.1 with all technical details included. Let $\Theta \subset \mathbb{R}^p$ where we have that $\theta \in \Theta$. Define

$$\begin{aligned} \widehat{\theta}_\epsilon \longleftarrow \text{argmin}_{\theta\in\Theta} & \frac{1}{m}\sum_{i=1}^m \sigma(-g_\theta(\tilde{\mathbf{x}}_i)) \\ \text{s.t.} & \frac{1}{n}\sum_{j=1}^n \sigma(g_\theta(\mathbf{x}_j) \le \alpha + \epsilon \\ & \frac{1}{n}\sum_{j=1}^n \max(1 - f_\theta(\mathbf{x}_j)y_j, 0) \le \tau + \epsilon \end{aligned} \tag{10}$$

Recall the optimization problem of interest:

$$\begin{aligned} & \inf_{\theta\in\Theta} \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\text{out}}}\sigma(-g_\theta(\tilde{\mathbf{x}}_i)) \\ & \text{s.t. } \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{\text{in}}}\sigma(g_\theta(\mathbf{x}_j)) \le \alpha \\ & \quad \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}_{\mathcal{X}\mathcal{Y}}}\max(1 - f_\theta(\mathbf{x}_j)\cdot y, 0) \le \tau. \end{aligned} \tag{11}$$

and let $\texttt{opt}$ denote its value.

We will make the following mild assumption and describe settings where it holds later.

Assumption B.2. There exists θ^* such that $R_1(g_\theta) = R_1^*(g_{\theta^*})$, $R_0(g_{\theta^*}) = \alpha$, and $\mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}_{\mathcal{X}\mathcal{Y}}}\max(1 - f_{\theta^*}(\mathbf{x}_j)\cdot y, 0) \le \tau$.

Proposition B.3. *Suppose $K = 2$. Suppose Assumption (B.2) holds. Define $\epsilon_k := \sqrt{\frac{2\ln(6/\delta)}{k}} + 2 \max_{h\in\{f,g\}}\max_{\mathbb{P}\in\{\mathbb{P}_1,\mathbb{P}_{wild}\}} \mathbb{E}_{\tilde{\mathbf{x}}_1,\ldots,\tilde{\mathbf{x}}_m\sim\mathbb{P}} \mathbb{E}_{\eta_1,\ldots,\eta_m} \sup_{\theta\in\Theta}\frac{1}{k}\sum_{i=1}^k \eta_i h_\theta(\tilde{\mathbf{x}}_i)$ where $\eta_1,\ldots,\eta_k$ are* i.i.d. *and $\mathbb{P}(\eta_i = 1) = \mathbb{P}(\eta_i = -1) = 1/2$. Let $\widehat{\theta}_\epsilon$ solve (10) with tolerance $\epsilon = c\epsilon_n$ where c is a universal positive constant. Then, with probability at least $1-\delta$*

1. $\mathbb{E}_{out}\sigma(-g_{\widehat{\theta}_\epsilon}(\mathbf{x})) \le \texttt{opt} + c_1\pi^{-1}(\epsilon_n + \epsilon_m)$,
2. $\mathbb{E}_{in}\sigma(g_{\widehat{\theta}_\epsilon}(\mathbf{x})) \le \alpha + c_2\epsilon_n$, *and*
3. $\mathbb{E}_{\mathcal{X}\mathcal{Y}}max(1 - f_{\widehat{\theta}_\epsilon}(\mathbf{x}_j)\cdot y, 0) \le \tau + c_3\epsilon_n$.

Proof. Define the following events

$$\Sigma_1 = \{|\frac{1}{m}\sum_{i=1}^{m}\sigma(-g_\theta(\tilde{\mathbf{x}}_i)) - R_{\text{wild}}(g_\theta)| \le 2\mathbb{E}_{\tilde{\mathbf{x}}_1,\ldots,\tilde{\mathbf{x}}_m\sim\mathbb{P}_{\text{wild}}}\mathbb{E}_{\eta_1,\ldots,\eta_m}\sup_{\theta\in\Theta}\frac{1}{m}\sum_{i=1}\eta_i g_\theta(\tilde{\mathbf{x}}_i) + c\sqrt{\frac{2\ln(6/\delta)}{m}} : \forall\theta\in\Theta\}$$

$$\Sigma_2 = \{|\frac{1}{n}\sum_{i=1}^{n}\sigma(g_\theta(\mathbf{x}_i)) - R_{-1}(g_\theta)| \le 2\mathbb{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_n\sim\mathbb{P}_{-1}}\mathbb{E}_{\eta_1,\ldots,\eta_n}\sup_{\theta\in\Theta}\frac{1}{n}\sum_{i=1}\eta_i g_\theta(\mathbf{x}_i) + c\sqrt{\frac{2\ln(6/\delta)}{n}} : \forall\theta\in\Theta\}$$

$$\Sigma_3 = \{|\frac{1}{n}\sum_{i=1}^{n}\max(1-f_\theta(\mathbf{x}_i)y_i,0) - \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{-1}}\max(1-f_\theta(\mathbf{x})y,0)| \le 2\mathbb{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_n\sim\mathbb{P}_{-1}}\mathbb{E}_{\eta_1,\ldots,\eta_n}\sup_{\theta\in\Theta}\frac{1}{n}\sum_{i=1}\eta_i g_\theta(\mathbf{x}_i)$$
$$+ c\sqrt{\frac{2\ln(6/\delta)}{n}} : \forall\theta\in\Theta\}$$

By Lemma B.4, we have that $\mathbb{P}(\Sigma_i)\ge 1-\delta/3$ for all $i=1,2,3$. Then, by the union bound, we have that $\Sigma := \Sigma_1\cap\Sigma_2\cap\Sigma_3$ holds with probability at least $1-\delta$. Assume Σ holds for the remainder of the proof.

Note that we have

$$\begin{aligned}
R_{-1}(g_{\widehat{\theta}_\epsilon}) - R_{-1}(g_{\theta^*}) &= R_{-1}(g_{\widehat{\theta}_\epsilon}) - \frac{1}{n}\sum_{i=1}^{n}\sigma(-g_{\widehat{\theta}_\epsilon}(\mathbf{x}_i)) \\
&\quad + \frac{1}{n}\sum_{i=1}^{n}\sigma(-g_{\widehat{\theta}_\epsilon}(\mathbf{x}_i)) - \frac{1}{n}\sum_{i=1}^{n}\sigma(-g_{\theta^*}(\mathbf{x}_i)) \\
&\quad + \frac{1}{n}\sum_{i=1}^{n}\sigma(-g_{\theta^*}(\mathbf{x}_i)) - R_{-1}(g_{\theta^*}) \\
&\le R_{-1}(g_{\widehat{\theta}_\epsilon}) - \frac{1}{n}\sum_{i=1}^{n}\sigma(-g_{\widehat{\theta}_\epsilon}(\mathbf{x}_i)) \\
&\quad + \frac{1}{n}\sum_{i=1}^{n}\sigma(-g_{\theta^*}(\mathbf{x}_i)) - R_{-1}(g_{\theta^*}) \qquad (12)\\
&\le 4\mathbb{E}_{\tilde{\mathbf{x}}_1,\ldots,\tilde{\mathbf{x}}_m\sim\mathbb{P}_{\text{wild}}}\mathbb{E}_{\eta_1,\ldots,\eta_m}\sup_{\theta\in\Theta}\frac{1}{m}\sum_{i=1}\sigma_i g_\theta(\tilde{\mathbf{x}}_i) + 2c\sqrt{\frac{2\ln(2/\delta)}{m}}. \qquad (13)
\end{aligned}$$

(12) follows since $\widehat{\theta}_\epsilon$ is feasible for the optimization problem (10) because, by the choice of ϵ and Σ,

$$\begin{aligned}
R_{-1}(g_{\theta^*}) &\le \alpha \\
\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_{-1}}\max(1-f_{\theta^*}(\mathbf{x})y,0) &\le \tau
\end{aligned}$$

Therefore, by definition of $\widehat{\theta}_\epsilon$ as the minimizer of (10), we have that

$$\frac{1}{n}\sum_{i=1}^{n}\sigma(-g_{\widehat{\theta}_\epsilon}(\mathbf{x}_i)) - \frac{1}{n}\sum_{i=1}^{n}\sigma(-g_{\theta^*}(\mathbf{x}_i)) \le 0.$$

(13) follows by the event Σ_1.

Similarly,

$$\begin{aligned}
\mathbb{E}_{\text{in}}\sigma(g_{\widehat{\theta}_\epsilon}(\mathbf{x})) &= \frac{1}{n}\sum_{i=1}^{n}\sigma(g_\theta(\mathbf{x}_i)) + \mathbb{E}_{\text{in}}\sigma(g_{\widehat{\theta}_\epsilon}(\mathbf{x})) - \frac{1}{n}\sum_{i=1}^{n}\sigma(g_\theta(\mathbf{x}_i)) \\
&\le \alpha + c\epsilon_n + \mathbb{E}_{\text{in}}\sigma(g_{\widehat{\theta}_\epsilon}(\mathbf{x})) - \frac{1}{n}\sum_{i=1}^{n}\sigma(g_\theta(\mathbf{x}_i)) \qquad (14)\\
&\le \alpha + c_2\epsilon_n \qquad (15)
\end{aligned}$$

where (14) follows from the definition of $\widehat{\theta}_\epsilon$ and (13) follows from Σ_2. This establishes claim 2 in the Proposition. Claim 3 follows by a similar argument to claim 2. Finally, claim 1 follows by (13), (15), and Lemma B.1.

□

Lemma B.4. *Let $\delta \in (0,1)$. Let $\mathbf{x}_1, \ldots, \mathbf{x}_k \sim \mathbb{P}$. Then,*

- *With probability at least $1-\delta$, for all $\theta \in \Theta$*

$$|\frac{1}{k}\sum_{i=1}^{k}\sigma(-g_\theta(\mathbf{x}_i)) - \mathbb{E}_{\mathbf{x}\sim\mathbb{P}}\sigma(-g_\theta(\mathbf{x}))| \leq 2\mathbb{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_k\sim\mathbb{P}}\mathbb{E}_{\eta_1,\ldots,\eta_m}\sup_{\theta\in\Theta}\frac{1}{k}\sum_{i=1}^{k}\eta_i g_\theta(\mathbf{x}_i) + c\sqrt{\frac{2\ln(2/\delta)}{k}}$$

- *With probability at least $1-\delta$, for all $\theta \in \Theta$*

$$|\frac{1}{k}\sum_{i=1}^{k}\sigma(g_\theta(\mathbf{x}_i)) - \mathbb{E}_{\mathbf{x}\sim\mathbb{P}}\sigma(g_\theta(\mathbf{x}))| \leq 2\mathbb{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_k\sim\mathbb{P}}\mathbb{E}_{\eta_1,\ldots,\eta_m}\sup_{\theta\in\Theta}\frac{1}{k}\sum_{i=1}^{k}\eta_i g_\theta(\mathbf{x}_i) + c\sqrt{\frac{2\ln(2/\delta)}{k}}$$

- *With probability at least $1-\delta$, for all $\theta \in \Theta$*

$$|\frac{1}{k}\sum_{i=1}^{k}max(1-f_\theta(\mathbf{x}_i)y_i,0) - \mathbb{E}_{\mathbf{x}\sim\mathbb{P}}max(1-f_\theta(\mathbf{x})y,0)| \leq 2\mathbb{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_k\sim\mathbb{P}}\mathbb{E}_{\eta_1,\ldots,\eta_n}\sup_{\theta\in\Theta}\frac{1}{k}\sum\eta_i g_\theta(\mathbf{x}_i) + c\sqrt{\frac{2\ln(2/\delta)}{k}}.$$

Proof. We show the first bullet point. The second and third bullet points follow by a similar argument. Using Mcdiarmid's inequality, we have that with probability at least $1-\delta$ for all $\theta \in \Theta$

$$|\frac{1}{k}\sum_{i=1}^{k}\sigma(-g_\theta(\mathbf{x}_i)) - \mathbb{E}_{\mathbf{x}\sim\mathbb{P}}\sigma(-g_\theta(\mathbf{x}))| \leq 2\mathbb{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_k\sim\mathbb{P}}\mathbb{E}_{\eta_1,\ldots,\eta_m}\sup_{\theta\in\Theta}\frac{1}{k}\sum_{i=1}^{k}\eta_i\sigma(g_\theta(\mathbf{x}_i)) + c\sqrt{\frac{2\ln(2/\delta)}{k}}$$

Then, using the contraction Lemma and the fact that the sigmoid function σ is 1-Lipschitz, we have that

$$\mathbb{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_k\sim\mathbb{P}}\mathbb{E}_{\eta_1,\ldots,\eta_m}\sup_{\theta\in\Theta}\frac{1}{k}\sum_{i=1}^{k}\eta_i\sigma(g_\theta(\mathbf{x}_i)) \leq \mathbb{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_k\sim\mathbb{P}}\mathbb{E}_{\eta_1,\ldots,\eta_m}\sup_{\theta\in\Theta}\frac{1}{k}\sum_{i=1}^{k}\eta_i g_\theta(\mathbf{x}_i).$$

The result follows by combining the above two inequalities. □

As an example where Assumption B.2 holds, consider for instance when $\theta = \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}$ with $w_1, w_2 \in \mathbb{R}^d$ and $g_\theta(x) = w_1^\top x$ and $f_\theta(x) = w_2^\top x$. Then if $\mathbb{P}_{-1}$ is absolutely continuous wrt the Lebesgue measure, Assumption B.2 holds. We could similar replace the linear maps g and f with neural networks that share a penultimate layer. See Blanchard et al. (2010) for a more detailed discussion and for more examples.

C. Validation using $\mathbb{P}_{\text{wild}}$ data

In this Section, we discuss how to use data from $\mathbb{P}_{\text{wild}}$ for a validation procedure and demonstrate its feasibility. For simplicity, we focus on the OOD task since it is standard to have a clean ID validation set for classification and therefore this captures the main difficulty. To ease notation, we write $\mathbb{P}_{\text{out}} = \mathbb{P}_1$ and $\mathbb{P}_{\text{in}} = \mathbb{P}_1$. Here, overloading notation, we assume access to a holdout set from $\mathbb{P}_{-1}$ and the mixture $\mathbb{P}_{\text{wild}}$:

- $\mathbf{x}_1, \ldots, \mathbf{x}_n \sim \mathbb{P}_{-1}$
- $\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_m \sim \mathbb{P}_{\text{wild}} := (1-\pi)\mathbb{P}_{-1} + \pi\mathbb{P}_1$ ($\pi \in (0,1]$ unknown)

We suppose that we have access to a small, finite set of models $\mathcal{G} \subset \{g : \mathbb{R}^d \mapsto \mathbb{R}\}$. $\mathcal{G}$ is typically obtained from training a model with a set of distinct hyperparameters, generating one $g \in \mathcal{G}$ for each hyperparameter configuration. Note that $\mathcal{G}$ is totally generic. As is typical in the OOD literature, we obtain from $\mathcal{G}$ a set of OOD predictors by thresholding each $g \in \mathcal{G}$ as follows:

$$\mathcal{H} := \{\text{sign}(g(x) - \tau) : g \in \mathcal{G}, \tau \in \mathbb{R}\}.$$

We now introduce some notation, overloading notation from Section B. Define

$$R_y(g_\theta) := \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\text{out}}}(\mathbb{1}\{h(\mathbf{x})) \neq y\}.$$

Define the optimization problem

$$\begin{aligned} R_1^* := \inf_{h \in \mathcal{H}} & R_1(h) \\ \text{s.t.} & R_{-1}(h) \leq \alpha. \end{aligned}$$

Define risk for $\mathbb{P}_{\text{wild}}$

$$\begin{aligned} R_{\text{wild}} &:= \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\text{wild}}}(\mathbb{1}\{h(\mathbf{x})) \neq 1\}) \\ &= \pi R_1(h) + (1 - \pi)(1 - R_{-1}(h)) \end{aligned}$$

Now, define another similar optimization, only changing the objective:

$$\begin{aligned} R_{\text{wild}}^* := \inf_{\theta} & R_{\text{wild}}(h) \\ & \text{s.t. } R_{-1}(h) \leq \alpha. \end{aligned}$$

We choose the $\widehat{h} \in \mathcal{H}$ that minimizes the FNR@95 on the holdout set:

$$\begin{aligned} \widehat{h}_\epsilon \in & \text{argmin}_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\{h(\tilde{\mathbf{x}}_i) \neq 1\} \\ & \text{s.t. } \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{h(\mathbf{x}_i) \neq -1\} \leq \alpha + \epsilon \end{aligned}$$

where we write $\widehat{h} := \widehat{h}_0$. We emphasize that this procedure is not only intuitive; it is also justified theoretically by applying Theorem 2 from (Blanchard et al., 2010). Theorem 2 requires that the following condition is satisfied:

Assumption C.1. For any $\alpha \in (0, 1)$, there exists $h^* \in \mathcal{G}$ such that $R_{-1}(h^*) = \alpha$ and $R_1(h^*) = R_{1,\alpha}^*(\mathcal{G})$.

Here, we show that $\mathcal{H}$ satisfies Assumption C.1 if P_0 is absolutely continuous with respect to the Lebesgue measure. Fix some $g \in \mathcal{G}$ and define $h_\tau(x) := \mathbb{1}\{h(x) > \tau\}$. Notice that if $\tau > \tau'$, then

$$h_\tau(x) \leq h_{\tau'}(x).$$

Thus, as discussed in the Remark of page 2978 in (Blanchard et al., 2010), we have that if for a given τ we have that $R_{-1}(h_\tau) < \alpha$, using the absolute continuity of P_0, we can find a τ' such that $R_0(h_{\tau'}) = \alpha$ and $R_1(h_{\tau'}) \leq R_1(h_\tau)$. Since this holds for any $g \in \mathcal{G}$, this implies that Assumption C.1 holds. Then, as a Corollary from Theorem 2 of (Blanchard et al., 2010), we obtain

Corollary C.2. *Let* $\epsilon_k := \sqrt{\frac{\mathcal{VC}(\mathcal{H}) - \log(\delta)}{k}}$ *where* $\mathcal{VC}(\mathcal{H})$ *denotes the VC dimension of* $\mathcal{H}$*. If* $\epsilon = c\epsilon_n$*, with probability at least* $1 - \delta$

1. $R_1(\widehat{h}_\epsilon) \leq R_1^* + c\pi^{-1}(\epsilon_n + \epsilon_m)$*, and*
2. $R_{-1}(\widehat{h}_\epsilon) \leq \alpha + c\pi^{-1}\epsilon_n$.