TUNING OOD DETECTORS WITHOUT GIVEN OOD DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Many existing out-of-distribution (OOD) detectors are tuned using a separate dataset deemed OOD with respect to the training distribution, often ad hocly chosen. In practical settings, such OOD data may be unavailable or difficult to obtain; even when available, they may not represent actual OOD data, which often include "unknown unknowns." As such, there is significant variance in detector performance depending on the ad hoc OOD dataset used for tuning. Hence, we introduce and formalize the often neglected problem of tuning OOD detectors without a "given" OOD dataset. We propose and evaluate two strong baseline methods based on related literature and found that they too yield inconsistent performance across detectors and datasets. We address this by introducing simulated hold-out tuning (SHOT), a generic approach that requires no additional data other than the ID training set. SHOT repeatedly holds out subsets of ID classes and treats them as simulated OOD samples for detector tuning. We evaluate these approaches on six OOD detectors across the OpenOOD and BROAD datasets, spanning both semantic and covariate shifts. SHOT achieves the highest number of wins across datasets and detectors compared with the baselines, particularly excelling at higher parameter detectors. These results show that effective OOD detector tuning need not rely on collecting a separate OOD dataset, removing a significant practical barrier to OOD detection in real-world deployments.

1 INTRODUCTION

When machine learning models encounter inputs that differ in distribution from their training distribution, their predictions can become unreliable or unsafe, making accurate *OOD detection* crucial, especially in safety-critical systems (Yang et al., 2024a). As a result, a large body of research has focused on designing increasingly sophisticated OOD detectors, including score-based methods (Liu et al., 2020), density models (Zong et al., 2018; He et al., 2016b), and learned shaping functions (Sun et al., 2021; Djurisic et al., 2023; Mondal et al., 2025).

Despite this progress, an important practical challenge has received surprisingly little systematic attention: how to tune the parameters of OOD detectors. In most state-of-the-art OOD detection methods, performance depends critically on one or more parameters that control, for example, thresholds or shaping functions. Existing benchmarks and experimental protocols typically assume access to a held-out dataset deemed OOD with respect to the training data of a neural network (NN) for a task of interest, which we refer to as a "task NN". This OOD tuning dataset is used to determine the parameters of a OOD detector before deployment onto the task NN for evaluation. Because it is difficult to foresee the data to be encountered at test time, the choice of this tuning dataset is typically ad hoc. While this is convenient for benchmarking, it masks a serious gap between experimental practice and real-world deployment.

In practice, a suitable tuning dataset is rarely available. What constitutes "out-of-distribution" depends not only on the data domain but also on the specific trained model, making it difficult to predefine representative OOD samples. Moreover, collecting or generating realistic OOD data can be costly, sensitive, or operationally infeasible in many applications: Collecting and curating a sufficiently diverse dataset requires numerous man hours and financial costs (Papadopoulos et al., 2021); this could amount to several months of labor and tens of thousands of dollars (see Appendix G for details). Even when some OOD samples are available, there is no guarantee that they adequately

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

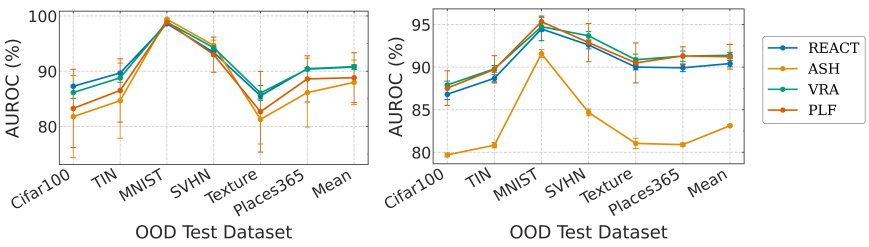

Figure 1: **Rankings of OOD detectors vary with tuning set.** The same OOD detector exhibits markedly different performance depending on the tuning dataset selected (vertical bars). Variability is large enough to alter the relative ranking of detectors. Left: DenseNet-101. Right: ResNet-18.

cover the space of possible OOD inputs, leading to substantial variability in the subsequent detection outcome. Figure 1 illustrates this issue on the OpenOOD benchmark (Zhang et al., 2024), where the relative ranking of state-of-the-art OOD detectors changes depending on which OOD tuning set is used. Hence, despite having a tuning set, whether collected or from a publicly available source, one would remain uncertain regarding the performance of the OOD detector. Instead, we advocate for an algorithmic approach that would reduce cost to a fixed number of GPU hours without need for data collection at all – by simulating OOD data from ID training data.

In this work, we investigate the problem of tuning OOD detectors without being given a dataset. To the best of our knowledge, the only attempts of tuning OOD detectors without an ad hoc tuning set are through the use of Gaussian noise images (Kirichenko et al., 2020) and adversarial perturbations (Lee et al., 2018) – the former has been specifically used to tune OOD detectors on ImageNet (Sun et al., 2021; Xu et al., 2023) and the latter has been extensively used in adaptation methods to simulate OOD examples Chung et al. (2025); Zhang et al. (2025). Hence, we adopt them as strong baselines for generating synthetic or simulated tuning sets for determining the parameters of OOD detectors. However, neither have been systematically benchmarked and studied across covariate and semantic shifts. We begin by benchmarking these baseline methods as means of generating tuning sets in place of a predefined dataset; we found that their performance is highly detector- and shift-dependent with neither consistently outperforming the other. Hence, we address this gap by introducing a new paradigm for OOD detector tuning that eliminates the need for a tuning dataset altogether. Our key insight is that the structure of the in-distribution data (used to train the task NN) itself contains sufficient information to calibrate OOD detectors, if it is exploited in the right way. This leads to a principled and fully self-contained tuning procedure that depends only on the data already available during the training of the NN on which the OOD detector will be deployed.

Contributions:

1. We introduce and formalize the problem of the determining the parameters of OOD detectors without access to a separate "given" tuning dataset.
2. We adopt two strong baselines from related literature and systematically evaluate them for this problem; we find no consistent best performer across benchmark datasets for multiple architectures.
3. We present a novel OOD parameter tuning method that uses only in-distribution training data of the task NN and does not require any ad hoc tuning dataset.
4. Amongst the evaluated methods, we show that our approach consistently outperforms baselines – out of 180 tests, our method has greater than 2 times more wins than any other baseline.

2 Related Work

Our focus is on tuning *post-hoc inference methods* for OOD detection (Zhang et al., 2024), which is a widely studied OOD detection approach that operates on pre-trained models without requiring training modifications to the underlying classification model. These methods assign an OOD score to an input based on the model's outputs or intermediate representations, and detect OOD samples via thresholding the score. Early approaches include confidence-based methods such as MSP (Hendrycks & Gimpel, 2017) and ODIN (Liang et al., 2018), which rely on softmax probabilities with calibration and input perturbations. Energy-based methods (Liu et al., 2020) replace softmax confidence with a Helmholtz free energy-based score computed from the logits. Energy scoring is widely used in feature shaping OOD detectors, which transform the feature outputs from the penultimate layer of

the network prior to scoring. Several approaches exist in this class of detectors, such as ReAct (Sun et al., 2021) which applies feature rectification by clipping under a threshold, ASH (Djurisic et al., 2023), which sparsifies intermediate layer activations, and other optimization-driven feature shaping approaches such as VRA (Xu et al., 2023) and PLF (Mondal et al., 2025). Other approaches derive OOD scores directly from feature representations without explicitly modifying them. KNN (Sun et al., 2022) computes distances to training features in the penultimate layer, while VIM (Wang et al., 2022) combines logits with a feature residual signal based on deviations from the in-distribution feature subspace. In this work, we focus on representative post-hoc detectors spanning these categories: ReAct, ASH, KNN, VIM, VRA, and PLF, which have been reported to achieve state-of-the-art OOD detection results across various benchmarks (Zhang et al., 2024). These methods differ in how they construct OOD scores making them suitable for evaluating the robustness of tuning methods.

Across the broad literature, we find that OOD detectors often rely on ad hoc tuning dataset(s) for determining their parameters. These OOD tuning datasets are typically chosen from a held-out set of a specific test dataset (Zhang et al., 2024). Attempts to depart from the reliance of these tuning datasets have been limited, and are also adhoc and specific to the target dataset, e.g., Gaussian noise images for ImageNet-1K (Sun et al., 2021; Xu et al., 2023); we treat these as baselines. Our work aims to present the first general framework for OOD tuning detectors without the need for any predefined OOD tuning dataset. Our method, SHOT, tunes OOD detectors by repeatedly holding out subsets of ID classes and treats them as simulated OOD samples for detector tuning. Our approach is related to (Vyas et al., 2018), which ensembles held-out networks for OOD detection, and has proposed a specific OOD detector by running such an ensemble at inference. SHOT, in contrast, tunes arbitrary detector families offline rather than defining a specific detector. Related to the problem of tuning OOD detectors without a tuning set are earlier post-hoc parameter-free OOD detectors, e.g., Hendrycks & Gimpel (2017); Liu et al. (2020); Lee et al. (2018); Ren et al. (2021); Sastry & Oore (2020). These methods have been largely superseded by parametric methods, which out-perform parameter-free methods with the exception of one (Ren et al., 2021) on small scale benchmarks (Hendrycks et al., 2022) (see also Appendix Table 8).

3 Baseline OOD Detector Tuning Methods Without Given OOD

To the best of our knowledge, there is no existing method for tuning an OOD detector without a tuning set. Hence, we begin by considering Gaussian noise images and adversarial perturbations, which past literature Kirichenko et al. (2020); Lee et al. (2018) assumed to be representative of OOD examples and have been applied outside detector tuning Sun et al. (2021); Xu et al. (2023); Chung et al. (2025); Zhang et al. (2025). As they do not rely on "true" or "given" OOD data, we adapt them as strong baselines for tuning OOD detectors.

Baseline #1 - Gaussian Noise: Using ID validation data from the validation set of the network training set for ID tuning samples, we treat *Gaussian noise images* as OOD validation data. The rationale is that Gaussian noise images are obviously OOD for practical tasks, where ID data will typically contain structure. This has been mentioned in Sun et al. (2021). Gaussian noise images are created by sampling Gaussian noise $\mathcal{N}(\mu, \sigma)$ iid at each pixel Kirchheim et al. (2022). The OOD detector parameters are trained with this simulated OOD dataset.

Baseline #2 - Adversarial Perturbations: We consider *adversarially perturbed ID data* (Lee et al., 2018) as OOD validation. Adversarial perturbations (Goodfellow et al., 2015) are signals that when added to data are imperceptible and shift the sample into low-density regions of the training data manifold (Song et al., 2018; Stutz et al., 2019). Thus, adversarial perturbed data serve as proxy OOD examples as they exhibit lower likelihood under the training distribution. ID data is sampled from the training data of the task NN. OOD data is adversarially perturbed sampled ID data, i.e., given an input image x and label y, adversarial examples are generated by the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2015) in our experiments as $x_{\text{adv}} = x + \epsilon \cdot \text{sign}\big(\nabla_x \mathcal{L}(x, y)\big)$, where $\mathcal{L}$ denotes the Binary Cross-Entropy (BCE) loss and ϵ controls the perturbation magnitude.

Hyper-parameter Tuning: Both baselines have a hyper-parameter. These choices are not discussed in the literature and are often chosen by hand. Our results show performance variation across their choice. In practical applications, one rarely has access to test data to choose the hyper-parameter to maximize performance on the test set. Therefore, we propose a hyper-parameter tuning approach. We generate balanced ID and OOD tuning data of size T by any of the two baselines. For robustness,

we sample S tuning sets, denoted $\mathcal{D}_{id,sim,j}$ and $\mathcal{D}^h_{ood,sim,j}$, where h is the hyper-parameter, and j denotes index of the random tuning set. If d_ϕ denotes an OOD detector with parameters ϕ, we define a loss as $\ell(\phi \mid h) = \frac{1}{S}\sum_{j=1}^{S} L(d_\phi; f_\theta, \mathcal{D}_{id,sim,j}, \mathcal{D}^h_{ood,sim,j})$, where L is a chosen metric, e.g., negative area under the receiver operating characteristic curve (AUROC). We optimize this loss by Bayesian optimization (Frazier, 2018) to obtain parameters ϕ^*_h. For each h, we resample tuning data, then select the hyperparameter whose ϕ^*_h minimizes the loss. This aligns with the hyperparameter tuning procedure introduced for our method, ensuring consistent selection across methods.

4 Our Method: Simulated Hold-Out Tuning (SHOT)

In this section, we present our new method (SHOT) for tuning OOD detectors without a given tuning dataset. We illustrate our approach on the classification task. We denote the in-distribution training dataset $\mathcal{D}^t = \{\mathcal{C}^t_1, \ldots, \mathcal{C}^t_n\}$ where n is the number of classes and $\mathcal{C}^t_i$ is the set of data for category i.

Method Construction and Rationale: OOD data may lie anywhere outside of the training (ID) distribution within some latent feature space. Thus, the goal of SHOT is to determine the parameters of the OOD detector that approximate the boundary between ID and OOD (Fig. 2 (a)). To approximate the ID/OOD boundary, it would be sufficient to tune the detector with OOD samples near and along the ID/OOD boundary and ID samples. SHOT estimates the ID/OOD boundary of surrogate networks (trained on a subset of held-in ID data), from which (held-out) near-boundary OOD samples are by construction available. Also, by construction, the surrogate networks are close to the original network, and thus the ID/OOD boundary of the surrogates and original network are close. Therefore, the surrogate boundaries approximate the true ID/OOD boundary.

Let f be the pretrained task NN. We construct the surrogate networks and OOD data as follows. Consider a subset of classes $\mathcal{D}_{ho}$ held-out from the training set $\mathcal{D}^t$, and the held-in data $\mathcal{D}_{hi} = \mathcal{D}^t \backslash \mathcal{D}_{ho}$. Notice $\mathcal{D}_{ho}$ is OOD to a network f_{hi} trained on $\mathcal{D}_{hi}$. Furthermore, as $\mathcal{D}_{ho}$ is from the original training set $\mathcal{D}^t$, it shares statistical properties of the ID training set $\mathcal{D}^t$ and so can be considered close to ID data of f_{hi}. Therefore, $\mathcal{D}_{ho}$ is OOD to f_{hi} because the held-out semantic classes were not observed, yet $\mathcal{D}_{ho}$ is near the ID/OOD boundary of f_{hi} (being close to ID). Hence, an OOD detector tuned to separate f_{hi} features of $\mathcal{D}_{hi}$ and $\mathcal{D}_{ho}$ would approximate part of the ID/OOD boundary of f_{hi} near $\mathcal{D}_{ho}$. Since f_{hi} and the original network f are close, which is true provided $\mathcal{D}_{ho}$ is a small enough subset $\mathcal{D}^t$, the ID/OOD boundary of f_{hi} approximates the ID/OOD boundary of the original network f. We note that only a portion of the ID/OOD boundary may be approximated with a detector tuned on $\mathcal{D}_{ho}$ as $\mathcal{D}_{ho}$ may only be near to a portion of the true ID/OOD boundary. Hence, SHOT tunes the detector on multiple such surrogate networks f_{hi} and held-out surrogate OOD data $\mathcal{D}_{ho}$ from multiple class selections. This better approximates the full ID/OOD boundary (Fig. 2(c)).

As SHOT holds out semantic categories, a natural question is whether SHOT would generalize to non-semantic OOD, e.g., covariate shifts. As noted by Yang et al. (2024b), semantic and covariate shifts frequently occur together since changes in semantic content coincide with changes in data source, acquisition process, domain, and visual style. Thus, holding out semantic classes may also induce covariate shifts, enabling SHOT to generalize to covariate shifts; we validate this empirically.

Overview of SHOT: We now provide details of the prior construction. A schematic of SHOT is shown in Figure 3. SHOT is given the original task training set $\mathcal{D}^t$ and the original trained task NN f. First, simulated ID/OOD datasets and tuning sets are generated and held-out networks are trained on the held-in datasets, as described in Algorithm 1. SHOT then uses these simulated ID/OOD datasets, tuning sets and networks to find the optimal detector parameters as described in Algorithm 2. We elaborate on these offline algorithms next. During inference, the optimal detector is used with the original task NN. As the optimal parameters are designed to optimize each held-out network, and all are close to the original network, these parameters generalize to the original network.

Simulated ID/OOD Datasets (Algorithm 1): Given a choice of M held-out categories, we simulate N ID/OOD dataset pairs from the training set $\mathcal{D}^t$ and train surrogates of the task NN, f_θ, where θ represents the parameters of the network, on each of the N simulated ID datasets. Each of the simulated ID/OOD datasets is denoted $\mathcal{D}^M_{hi,i}, \mathcal{D}^M_{ho,i}$ where $i = 1, \ldots, N$, and is generated by randomly sampling M categories from the training dataset $\mathcal{D}^t$ forming simulated OOD and the remaining $n - M$ categories form the simulated ID dataset. We train $f^M_{\theta_i}$ on held-in data $\mathcal{D}^M_{hi,i}$, and so $\mathcal{D}^M_{ho,i}$ is OOD to f_{θ_i} and the held-in data $\mathcal{D}^M_{hi,i}$ is ID. This is repeated for multiple M.

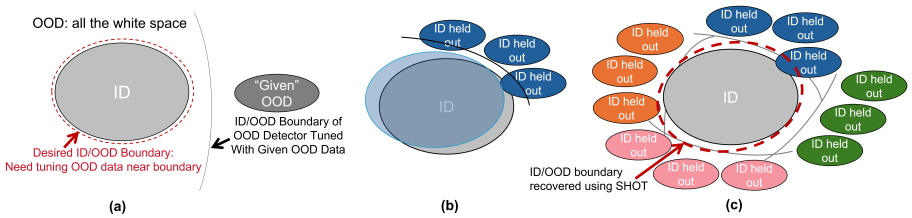

Figure 2: **Intuition for SHOT.** (a) A visual depiction of the space of all data. The large ellipse indicates ID data. The small ellipse indicates "given" OOD data, which may only cover a small part of all OOD. Tuning an OOD detector may result in ID/OOD boundary depicted by the black curve. If all true OOD data is to the right of the black curve, this tuned detector is adequate. An ideal boundary robust to any OOD would be the dashed red line. (b) A visual depiction of SHOT is shown, in this example $M = 3$. Held-out classes are depicted in solid blue, and the blue transparent ellipse indicates the corresponding held-in classes. Held-in data may approximate true ID data (solid gray ellipse). The boundary (black solid curve) between one held-out set and the held-in data may approximate a portion of the ideal ID/OOD boundary. (c) By holding out multiple sets of random classes, SHOT approximates other parts of the ideal ID/OOD boundary. t-SNE embedding plots on CIFAR-10 benchmark (Appendix Figure 4) provide evidence of this intuition.

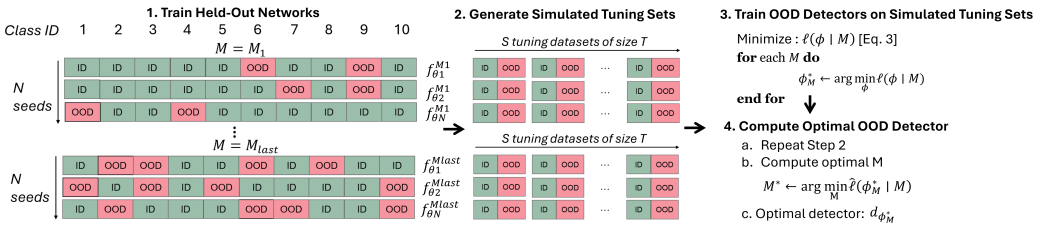

Figure 3: **Schematic of SHOT.** 1) Training Held-Out Networks: For each $M = M_1, \ldots, M_{last}$, M classes are randomly chosen to be held-out (red) from the training set and a network is trained on the remaining (held-in - green) classes; this is repeated N times to give networks $f^M_{\theta_i}$. 2) Each held-out/held-in dataset pairs are then sampled to generate T ID/OOD samples. This is repeated S times to give multiple simulated tuning sets. 3) OOD detectors are now tuned for each M on the simulated tuning sets by minimizing the loss (Equation 1). 4) The optimal OOD detector is found by optimizing M over the same loss but over a newly generated random set of validation tuning sets.

Simulated ID/OOD Tuning Sets (Algorithm 1): We generate the simulated *tuning* datasets for OOD detector tuning. Note that the simulated ID/OOD datasets specified in the previous paragraph are typically imbalanced with more ID than OOD data. We address this issue by sampling a fixed number T samples from each simulated ID and OOD set, typically T is far fewer than the training set size as more data is not needed as typical OOD detectors do not contain more than 10 parameters. To provide robustness to random sample selection, we sample S simulated ID/OOD tuning sets, denoted $\mathcal{D}^M_{hi,i,j}$ and $\mathcal{D}^M_{ho,i,j}$ for $j = 1, \ldots, S$ from each simulated ID/OOD dataset $\mathcal{D}^M_{hi,i}, \mathcal{D}^M_{ho,i}$.

OOD Detector Tuning (Algorithm 2): Let d_ϕ denote the OOD detector to be tuned where ϕ denotes its parameters. We formulate a loss function for ϕ that measures the sum of losses over all tuning sets to be robust to all sources of random choices, and then we use Bayesian optimization (Frazier, 2018) to efficiently globally minimize the loss, but any other optimization procedure can be used. Specifically, let $L(d_\phi; f^M_{\theta_i}, \mathcal{D}^M_{hi,i,j}, \mathcal{D}^M_{ho,i,j})$ denote the loss function associated with the loss of d_ϕ for detecting simulated OOD data with NN $f^M_{\theta_i}$ from the tuning set $\mathcal{D}^M_{hi,i,j}, \mathcal{D}^M_{ho,i,j}$. L could be a metric such as negative AUROC or FPR95 or a combination, etc. We minimize the loss ℓ:

$$\ell(\phi|M) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{S}\sum_{j=1}^{S} L(d_\phi; f^M_{\theta_i}, \mathcal{D}^M_{hi,i,j}, \mathcal{D}^M_{ho,i,j}), \tag{1}$$

where i indexes over sampled held-out datasets and j indexes over the simulated tuning sets.

Choosing the Number of Simulated OOD Categories (Algorithm 2): ℓ depends on the number M of simulated OOD categories, a hyper-parameter. Holding out too few categories will not learn enough about the possible OOD distribution, while holding out too many would mean that $f^M_{\theta_i}$ differs too

Algorithm 1 Simulated ID/OOD Training and Detector Tuning Data from ID Training Data

Require: Network training set $\mathcal{D}^t = \{\mathcal{C}_1^t, \ldots, \mathcal{C}_n^t\}$, neural net architecture f
Require: $\{M\}$ - a set of M where M is the # of held-out classes, N - # of trials for held-out class selection, S - # simulated tuning sets, T - size of tuning set
Ensure: Trained held-out networks $\{f_{\theta_1}^M, \ldots, f_{\theta_N}^M\}$, simulated tuning ID/OOD datasets $\mathcal{D}_{\text{hi},i,j}^M$ and $\mathcal{D}_{\text{ho},i,j}^M$, simulated ID/OOD datasets $\mathcal{D}_{\text{hi},i}^M$ and $\mathcal{D}_{\text{ho},i}^M$ for each M

for each M and each $i = 1, \ldots, N$ **do**
  Randomly sample M simulated OOD categories from $\mathcal{D}^t$ to form $\mathcal{D}_{\text{ho},i}^M$
  Define remaining categories as simulated ID dataset $\mathcal{D}_{\text{hi},i}^M$
  Train task network $f_{\theta_i}^M$ on $\mathcal{D}_{\text{hi},i}^M$
  for $j = 1, \ldots, S$ **do**
    Sample sim. tuning ID dataset $\mathcal{D}_{\text{hi},i,j}^M$ of size $T/2$ from validation set of $\mathcal{D}_{\text{hi},i}^M$
    Sample simulated tuning OOD dataset $\mathcal{D}_{\text{ho},i,j}^M$ of size $T/2$ from $\mathcal{D}_{\text{ho},i}^M$
  end for
end for

Algorithm 2 Optimal Detector Tuning

Require: Simulated ID/OOD datasets $\mathcal{D}_{\text{hi},i}^M$, $\mathcal{D}_{ho,i}^M$ and tuning sets $\mathcal{D}_{\text{hi},i,j}^M$ and $\mathcal{D}_{\text{ho},i,j}^M$
Ensure: Optimal detector $\phi_{M^*}^*$ to be used with original task NN

for each M **do**
  $\phi_M^* \leftarrow \arg\min_\phi \ell(\phi \mid M)$ (Bayesian Optimization)
  Resample $\mathcal{D}_{\text{hi},i,j}^M$ and $\mathcal{D}_{\text{ho},i,j}^M$ from $\mathcal{D}_{\text{hi},i}^M$ and $\mathcal{D}_{\text{ho},i}^M$ and compute validation loss $\hat{\ell}(\phi_M^* \mid M)$
end for
$M^* \leftarrow \arg\min_M \hat{\ell}(\phi_M^* \mid M)$

much from the original task NN. The optimal M is in-between. To choose M, we optimize $\ell(\phi|M)$ in ϕ to obtain ϕ_M^* for each M. We then re-sample datasets $\mathcal{D}_{hi,i,j}^M$ and $\mathcal{D}_{ho,i,j}^M$ from $\mathcal{D}_{hi,i}^M$ and $\mathcal{D}_{ho,i}^M$ and re-compute the loss of ϕ_M^*. This estimates the loss on independently generated tuning datasets from the same distribution as the simulated ID/OOD dataset, hence it can be used for hyper-parameter validation. Thus, we choose M^* to minimize this validation loss.

5 Experiments

Tuning of OOD detectors without given OOD data is a problem that does not have existing experimental protocols for evaluation. Therefore, we propose one in this section.

Datasets: We evaluate tuning methods using data from the OpenOOD benchmark (Zhang et al., 2024) and the BROAD benchmark (Guille-Escuret et al., 2024). OpenOOD includes four in-distribution (ID) datasets: CIFAR-10 (Krizhevsky et al., 2009a), CIFAR-100 (Krizhevsky et al., 2009b), ImageNet-200, and ImageNet-1K (Deng et al., 2009), each paired with 5-6 predefined semantic shift OOD datasets and two covariate shift datasets (resampling and style shifts) for ImageNet-1K. BROAD captures diverse covariate shifts, including adversarial attacks, synthetic images and multi-labels.

Networks: We use ResNet-18/50 (He et al., 2016a;b) as specified by the benchmarks.

Baseline Tuning Methods: For Gaussian noise images, we sample from $\mathcal{N}(128, \sigma)$ iid for each pixel where $\sigma \in \{32, 64, 128\}$, clip to $[0, 255]$, and normalize to $[0, 1]$. For FGSM, we apply perturbations with $\epsilon \in \{0.005, 0.01, 0.1\}$ to images on the $[0, 1]$ scale and clip the perturbed images to $[0, 1]$.

SHOT Tuning Method: We choose the range of the number of simulated OOD categories (M) to be a portion of total range of classes. For the smaller scale benchmarks - CIFAR-10 and CIFAR-100, we consider holdout fractions from 10% up to 40% (e.g. 10 classes up to 40 classes held-out for CIFAR-100), and use $N = 10$ random seeds. For larger benchmarks ImageNet-200 and ImageNet-1K, we use holdout class proportions in the range of 5% up to 30% and 6 held-out sets, and $N = 5$ seeds. Ablations that follow show the results are not sensitive to these choices.

Bayesian Optimization: We tune detector parameters using Gaussian Process-Based Bayesian Optimization (Frazier, 2018), widely used for hyperparameter tuning. The optimization and search

Method	CIFAR-10 (ResNet-18)							CIFAR-100 (ResNet-18)						
	cifar100	tin	mnist	svhn	texture	places365	Best (#)	cifar10	tin	mnist	svhn	texture	places365	Best (#)
ASH (Gauss.)	74.19 ± 3.31	76.69 ± 4.16	87.78 ± 2.94	73.33 ± 5.45	75.33 ± 6.18	80.25 ± 2.18	0	78.42 ± 1.28	**80.76 ± 0.83**	78.71 ± 5.54	81.10 ± 1.60	77.36 ± 1.71	78.30 ± 1.41	1
ASH (Adv.)	**77.03 ± 2.11**	**79.36 ± 1.19**	**89.08 ± 3.29**	**80.98 ± 3.03**	**79.55 ± 1.27**	79.72 ± 1.28	5	**79.17 ± 0.24**	80.67 ± 0.68	**83.07 ± 0.40**	80.00 ± 2.42	75.05 ± 1.46	77.41 ± 0.74	2
ASH (Ours)	76.77 ± 3.12	79.20 ± 2.67	85.82 ± 3.99	76.28 ± 6.71	78.37 ± 4.84	**82.36 ± 1.97**	1	76.49 ± 0.22	80.29 ± 0.23	73.81 ± 1.84	**84.38 ± 1.82**	**79.32 ± 0.47**	**79.80 ± 0.10**	3
VIM (Gauss.)	87.80 ± 0.26	89.67 ± 0.34	94.71 ± 0.54	**94.60 ± 0.56**	95.19 ± 0.31	89.63 ± 0.38	1	71.90 ± 0.13	**77.71 ± 0.23**	81.43 ± 1.94	**83.30 ± 3.83**	86.39 ± 0.37	75.59 ± 0.41	2
VIM (Adv.)	87.50 ± 0.14	89.44 ± 0.29	94.43 ± 0.35	94.12 ± 0.67	95.04 ± 0.24	89.04 ± 0.42	0	71.32 ± 0.82	77.60 ± 0.20	80.81 ± 1.48	81.75 ± 3.32	**86.58 ± 1.10**	73.79 ± 1.02	1
VIM (Ours)	**87.92 ± 0.26**	**89.71 ± 0.34**	**94.72 ± 0.50**	94.56 ± 0.57	**95.20 ± 0.30**	**89.81 ± 0.36**	5	**72.03 ± 0.32**	77.60 ± 0.12	**82.10 ± 0.72**	81.92 ± 2.91	85.87 ± 0.88	**75.77 ± 0.43**	3
ReAct (Gauss.)	85.80 ± 0.81	88.30 ± 0.49	93.20 ± 3.51	88.92 ± 3.66	88.92 ± 1.15	90.51 ± 0.57	0	69.80 ± 2.25	77.28 ± 2.19	66.34 ± 4.46	**83.97 ± 6.05**	**84.95 ± 0.72**	78.28 ± 2.60	2
ReAct (Adv.)	85.51 ± 1.02	88.06 ± 0.69	92.90 ± 3.55	88.14 ± 3.92	88.59 ± 1.43	**90.67 ± 0.71**	1	75.13 ± 2.74	80.87 ± 1.38	72.96 ± 5.39	83.25 ± 4.17	82.92 ± 3.79	80.20 ± 0.69	0
ReAct (Ours)	**86.42 ± 0.82**	**88.64 ± 0.64**	**93.87 ± 2.44**	**91.02 ± 1.38**	**89.13 ± 1.36**	89.87 ± 1.18	5	**78.16 ± 0.05**	**82.88 ± 0.07**	**77.28 ± 1.94**	83.68 ± 1.28	81.57 ± 0.35	**80.49 ± 0.08**	4
KNN (Gauss.)	89.55 ± 0.12	91.34 ± 0.24	93.97 ± 0.36	92.42 ± 0.30	92.91 ± 0.24	91.52 ± 0.23	0	76.24 ± 0.27	82.91 ± 0.18	**83.00 ± 1.44**	84.00 ± 1.34	**84.21 ± 0.80**	78.66 ± 0.49	2
KNN (Adv.)	89.55 ± 0.12	91.34 ± 0.24	93.97 ± 0.36	92.42 ± 0.30	92.91 ± 0.24	91.52 ± 0.23	0	76.24 ± 0.27	82.92 ± 0.18	83.00 ± 1.44	84.00 ± 1.34	84.20 ± 0.80	78.67 ± 0.49	0
KNN (Ours)	**89.60 ± 0.13**	**91.39 ± 0.25**	**94.04 ± 0.36**	**92.48 ± 0.30**	**92.97 ± 0.24**	**91.58 ± 0.23**	6	**76.65 ± 0.27**	**83.16 ± 0.18**	82.64 ± 1.47	**84.03 ± 1.18**	83.98 ± 0.81	**79.11 ± 0.48**	4
VRA (Gauss.)	71.82 ± 9.74	72.49 ± 11.63	78.23 ± 8.32	74.71 ± 8.09	77.96 ± 8.99	71.74 ± 16.20	0	61.08 ± 3.13	68.07 ± 3.29	57.16 ± 4.97	74.79 ± 9.37	80.59 ± 2.28	70.28 ± 2.38	0
VRA (Adv.)	85.68 ± 0.81	87.94 ± 0.83	92.50 ± 3.37	88.88 ± 3.36	89.46 ± 0.97	**89.71 ± 1.75**	1	75.07 ± 3.38	80.73 ± 2.17	73.30 ± 5.07	83.40 ± 3.38	**83.42 ± 3.20**	78.94 ± 0.91	1
VRA (Ours)	**86.22 ± 0.62**	**88.57 ± 0.37**	**94.10 ± 2.46**	**91.49 ± 1.06**	**89.71 ± 0.78**	88.98 ± 0.66	5	**77.40 ± 0.24**	**82.71 ± 0.05**	**76.89 ± 2.43**	**84.05 ± 1.52**	83.37 ± 0.17	**80.15 ± 0.21**	5
PLF (Gauss.)	82.47 ± 3.29	84.88 ± 3.26	89.78 ± 3.51	87.80 ± 4.15	87.61 ± 3.46	83.55 ± 2.24	0	65.44 ± 3.16	72.79 ± 3.60	51.75 ± 7.18	67.61 ± 11.72	74.47 ± 11.72	73.55 ± 4.91	0
PLF (Adv.)	84.70 ± 0.23	87.02 ± 0.17	90.69 ± 3.66	87.17 ± 4.67	**89.17 ± 1.39**	86.55 ± 3.20	1	77.99 ± 0.64	82.42 ± 0.31	76.61 ± 3.65	81.84 ± 2.94	80.94 ± 4.19	**80.62 ± 1.66**	1
PLF (Ours)	**85.55 ± 0.80**	**87.95 ± 0.45**	**94.30 ± 2.38**	**91.59 ± 1.34**	88.53 ± 0.65	**88.11 ± 0.83**	5	**78.72 ± 0.10**	**83.13 ± 0.07**	**78.71 ± 1.70**	**83.18 ± 0.57**	**81.26 ± 0.54**	80.58 ± 0.14	5

Method	ImageNet-200 (ResNet-18)						ImageNet-1K (ResNet-50)					
	SSB-hard	NINCO	iNaturalist	Textures	OpenImage-O	Best (#)	SSB-hard	NINCO	iNaturalist	Textures	OpenImage-O	Best (#)
ASH (Gauss.)	77.06 ± 0.43	81.46 ± 0.76	93.31 ± 1.02	94.61 ± 0.06	89.68 ± 0.85	0	**73.61**	82.06	95.62	95.03	91.71	1
ASH (Adv.)	**79.69 ± 0.15**	**85.40 ± 0.57**	**95.06 ± 0.63**	**94.97 ± 0.06**	**91.93 ± 0.39**	5	73.00	**83.38**	**96.88**	**96.71**	**93.20**	4
ASH (Ours)	79.38 ± 0.29	84.51 ± 0.43	94.52 ± 0.41	93.14 ± 0.17	90.42 ± 0.09	0	73.27	83.17	96.60	96.29	92.90	0
VIM (Gauss)	72.41 ± 5.44	81.93 ± 3.31	92.87 ± 1.63	93.42 ± 1.11	88.43 ± 1.77	0	64.35	78.25	87.22	97.87	89.14	0
VIM (Adv.)	74.42 ± 5.77	83.17 ± 3.83	**93.51 ± 1.86**	93.63 ± 1.62	89.05 ± 2.09	1	64.00	78.08	89.17	**98.27**	90.30	1
VIM (Ours)	**74.47 ± 0.29**	**83.90 ± 0.22**	93.31 ± 0.07	**95.06 ± 0.04**	**89.42 ± 0.15**	4	**71.88**	**81.78**	**94.11**	91.40	**90.97**	4
ReAct (Gauss.)	77.85 ± 2.81	83.59 ± 2.13	**94.03 ± 0.51**	**93.44 ± 0.15**	**90.88 ± 0.30**	3	73.15	**81.94**	**96.07**	**92.73**	**91.93**	4
ReAct (Adv.)	79.25 ± 0.33	84.97 ± 0.28	92.85 ± 0.90	91.50 ± 0.86	88.93 ± 0.16	0	72.48	80.46	92.40	89.73	89.85	0
ReAct (Ours)	**79.83 ± 0.02**	**85.17 ± 0.12**	92.56 ± 0.50	90.87 ± 0.16	89.25 ± 0.25	2	**73.18**	81.69	94.88	91.72	91.33	1
KNN (Gauss.)	**77.67 ± 0.16**	85.67 ± 0.13	91.71 ± 0.48	94.86 ± 0.06	88.70 ± 0.36	1	58.03	75.88	85.03	**97.33**	85.26	1
KNN (Adv.)	77.55 ± 0.18	**86.12 ± 0.10**	**93.24 ± 0.39**	**95.10 ± 0.03**	**89.63 ± 0.32**	4	62.64	79.67	**86.37**	97.07	**87.02**	2
KNN (Ours)	77.65 ± 0.16	85.52 ± 0.13	91.23 ± 0.51	94.78 ± 0.08	88.45 ± 0.38	0	**63.09**	**79.87**	85.99	96.91	86.90	2
VRA (Gauss.)	73.73 ± 8.46	81.57 ± 6.39	92.26 ± 1.82	**94.31 ± 0.36**	88.65 ± 3.65	1	64.44	71.15	**92.30**	**96.03**	87.72	2
VRA (Adv.)	**79.44 ± 0.41**	85.05 ± 0.49	92.83 ± 0.19	91.57 ± 2.43	89.64 ± 0.93	1	**72.61**	79.35	89.97	86.00	87.99	1
VRA (Ours)	79.28 ± 0.04	**85.37 ± 0.13**	**93.37 ± 0.48**	93.59 ± 0.14	**90.73 ± 0.32**	3	70.74	**79.85**	92.16	92.82	**90.57**	2
PLF (Gauss.)	74.74 ± 0.37	80.45 ± 0.19	84.30 ± 1.38	85.38 ± 0.57	82.64 ± 0.80	0	64.50	77.97	93.72	**95.36**	91.07	1
PLF (Adv.)	76.26 ± 0.30	79.61 ± 0.46	85.60 ± 0.80	78.23 ± 0.31	82.00 ± 0.68	0	71.89	78.11	91.56	90.16	89.06	0
PLF (Ours)	**80.95 ± 0.04**	**86.52 ± 0.16**	**94.16 ± 0.46**	**92.25 ± 0.13**	**90.66 ± 0.20**	5	**74.44**	**82.95**	**95.41**	91.55	**91.72**	4

Table 1: Semantic OOD benchmark results on CIFAR-10/100 (top) and ImageNet-200/1K (bottom) using OpenOOD test sets. We report AUROC. Bold indicates the best-performing tuning method for each detector. Best (#) is the number of test OOD datasets on which a method achieves highest AUROC. Gauss denotes Gaussian noise and Adv is adversarial tuning, and Ours indicates SHOT.

ranges are applied consistently across all detectors and tuning methods. We minimize the loss L (Equation 1) being negative AUROC. Further details are provided in Appendix B.

OOD Detectors Tuned: We evaluate tuning approaches on top performing post-hoc OOD detectors: ReAct, ASH, KNN, VIM, VRA, and PLF. Out of these ReAct, ASH, KNN, and VIM each have a single tuning parameter: a clipping threshold for ReAct, sparsification percentile for ASH, number of neighbors k for KNN, and subspace dimension D for VIM. VRA and PLF are higher-parameter detectors with three and seven parameters, respectively.

Results (Semantic Shift OOD): We report the results on OpenOOD data in Table 1. SHOT generally performs better than Gaussian and adversarial tuning on the higher parameter OOD detectors - VRA and PLF. Across CIFAR and ImageNet benchmarks, SHOT achieves 15/22 wins for VRA and 19/22 wins for PLF, substantially outperforming the other baselines. These results indicate that SHOT can be particularly effective for detectors with more complex parameterizations, whereby the detectors may overfit to Gaussian noise or adversarial perturbations as OOD tuning data.

For lower parameter detectors - ReAct, ASH, KNN, and VIM, the gains are more modest but remain consistently competitive. On CIFAR benchmarks, SHOT achieves 9/12 wins for ReAct, 10/12 for KNN, and 8/12 wins for VIM, while ASH shows mixed results but generally matches or improves upon Gaussian tuning. On ImageNet, although SHOT attains fewer outright wins, it consistently ranks among the top two methods across most datasets, often outperforming Gaussian tuning and closely matching adversarial baselines. Overall, these results suggest that SHOT yields the largest and most consistent improvements for higher parameter detectors such as PLF and VRA, while providing more modest and less uniform gains for lower parameter methods.

Results (Covariate Shift OOD): Table 2 shows that SHOT performs strongly under covariate shift, achieving the highest number of wins for KNN and PLF (7/8 datasets each), and 4/8 wins for both ASH and VRA. For ReAct, SHOT and Gaussian tuning tie with 3/8 wins. For VIM, adversarial tuning performs best with 5/8 wins and SHOT ranks second with 3/8. Overall, SHOT performs best on 28 tests compared to 10 for adversarial, and 10 for Gaussian noise. These results suggest that SHOT generalizes well to covariate shifts, delivering consistent improvements for several detectors while remaining robust across diverse distributional changes.

Summarized Results: Table 3 summarizes the results of the previous experiments, reporting the number of OOD datasets that each tuning method performed the best on across datasets and architec-

Method	ImageNet-1K (ResNet-50)								
	ImageNet-C	ImageNet-V2	ImageNet-R	AutoAttack	PGD	Stable Diff.	Cond. BigGAN	Multi-Label	Best (#)
ASH (Gauss.)	**84.28**	58.32	85.23	86.29	**61.15**	69.79	**56.90**	65.07	3
ASH (Adv.)	84.04	58.34	85.55	86.34	60.55	70.36	54.03	**69.40**	1
ASH (Ours)	84.07	**58.35**	**85.63**	**86.38**	60.64	**70.38**	54.08	69.05	**4**
VIM (Gauss.)	84.00	57.37	88.13	84.44	63.57	76.12	38.15	63.33	0
VIM (Adv.)	**84.21**	57.76	**88.81**	**85.29**	**64.33**	**76.92**	35.21	62.89	**5**
VIM (Ours)	82.06	**58.87**	87.69	84.69	60.84	75.03	**38.32**	**67.94**	3
ReAct (Gauss.)	82.46	58.58	86.30	84.81	**59.06**	**73.66**	40.17	**73.62**	**3**
ReAct (Adv.)	82.54	58.67	**86.92**	85.00	57.75	73.09	**42.68**	71.92	2
ReAct (Ours)	**82.55**	**58.69**	86.68	**85.05**	58.47	73.31	42.21	73.11	**3**
KNN (Gauss.)	83.80	54.87	86.14	84.63	62.74	70.61	**42.50**	49.06	1
KNN (Adv.)	84.97	56.48	87.67	86.04	64.98	73.07	37.78	53.99	0
KNN (Ours)	**85.15**	**56.70**	**87.85**	**86.21**	**65.29**	**73.44**	37.27	**54.64**	**7**
VRA (Gauss.)	**82.54**	57.28	87.56	84.52	61.15	**74.98**	36.72	65.58	2
VRA (Adv.)	82.39	57.47	87.60	84.69	**62.01**	74.40	36.80	**68.93**	2
VRA (Ours)	82.48	**58.41**	**87.74**	**84.92**	58.29	74.18	**41.16**	66.93	**4**
PLF (Gauss.)	81.99	57.83	86.63	84.58	**60.91**	73.74	40.62	69.03	1
PLF (Adv.)	82.42	57.92	86.79	84.97	60.80	73.59	42.92	68.88	0
PLF (Ours)	**83.01**	**58.91**	**86.87**	**85.46**	59.34	**74.42**	**43.59**	**69.17**	**7**

Table 2: Performance comparison for tuning detectors for detecting covariate shift OOD data from OpenOOD covariate shift and the BROAD datasets. Best values within each method block are bolded.

Method	CIFAR-10 ResNet-18	CIFAR-100 ResNet-18	ImageNet-200 ResNet-18	ImageNet-1K ResNet-50	ImageNet-1K (Cov.) ResNet-50	Total
ASH (Gauss)	0	1	0	1	3	5
ASH (Adv)	**5**	2	**5**	**4**	1	**17**
ASH (Ours)	1	**3**	0	0	**4**	8
VIM (Gauss)	1	2	0	0	0	3
VIM (Adv)	0	1	1	1	**5**	8
VIM (Ours)	**5**	**3**	**4**	**4**	3	**19**
ReAct (Gauss)	0	2	**3**	**4**	**3**	12
ReAct (Adv)	1	0	0	0	2	3
ReAct (Ours)	**5**	**4**	2	1	**3**	**15**
KNN (Gauss)	0	2	1	1	1	5
KNN (Adv)	0	0	**4**	**2**	0	6
KNN (Ours)	**6**	**4**	0	**2**	**7**	**19**
VRA (Gauss)	0	0	1	**2**	2	5
VRA (Adv)	1	1	1	1	2	6
VRA (Ours)	**5**	**5**	**3**	**2**	**4**	**19**
PLF (Gauss)	0	0	0	1	1	2
PLF (Adv)	1	1	0	0	0	2
PLF (Ours)	**5**	**5**	**5**	**4**	**7**	**26**

Table 3: Number of benchmark wins across datasets and architectures for OOD detector tuning strategies. Bold indicates the best performance within each detector family block for each column.

tures. Overall, SHOT is the best tuning method for each OOD method except one, winning 106/180 tests, which is more than twice as many as Adversarial (42/180) or Gauss (32/180).

Comparison to 'Given' OOD Tuning Set: We compare SHOT to tuning with existing datasets specified in the OpenOOD benchmark, which are datasets specifically chosen by a human, and deemed as realistic tuning OOD data. Results are shown in Table 4, which shows that SHOT's performance is comparable to a given tuning dataset, and better in almost half of the tests. This implies that SHOT is able to approximate the ID/OOD boundary of the task NN.

Computational Cost: The majority of the cost of SHOT is (offline) training of the held-out networks. The number of networks to train is equal to the cardinality ($|\{M\}|$) of the chosen set of M times N (the number of seeds). On 4 NVIDIA GeForce RTX 3090 GPUs, this gives 20-24 hours per network for ResNet-50 trained on ImageNet-1K. The Bayesian optimization cost is quantified as: number of Bayesian steps $\times$ cost of each step (proportional to evaluating $|\{M\}| \times N$ networks) $\times$ number of ID/OOD tuning sets. In our experiments, number of Bayesian steps = 100, and for ImageNet-1K, number of ID/OOD splits = 10. This gives 30,000 $\times$ the cost of a Bayesian step. For ImageNet-1K, this was about 42 minutes total on Intel i9-10980XE (18 cores) CPU.

Compute-Performance Ablation: We ablate the number of random holdout splits N and the number of held-out categories M on ImageNet-1K with ResNet-50, and with ReAct and PLF (detectors with the lowest and highest number of parameters). Results in Table 5 show that performance is generally stable as N is reduced; in many settings, *a single held-out network achieves AUROC comparable to using all five*. More extensive ablations over M for all detectors on ImageNet-200 and ImageNet-1K are provided in Tables 6 and 7 of Appendix D, and similarly show limited sensitivity to the holdout size. We emphasize that the training cost of SHOT is fixed and incurred entirely offline. In cases

Method	**CIFAR-10** (ResNet-18)		**CIFAR-100** (ResNet-18)		**ImageNet-200** (ResNet-18)		**ImageNet-1K** (ResNet-50)	
	Near-OOD	Far-OOD	Near-OOD	Far-OOD	Near-OOD	Far-OOD	Near-OOD	Far-OOD
ReAct	87.11±0.61	90.42±1.41	**80.77±0.05**	80.39±0.49	81.87±0.98	**92.31±0.56**	77.38	**93.67**
ReAct (SHOT)	**87.53±0.73**	**90.97±1.59**	80.52±0.06	**80.76±0.91**	**82.50±0.07**	90.89±0.30	**77.44**	92.64
ASH	75.27±1.04	78.49±2.58	78.20±0.15	**80.58±0.66**	**82.38±0.19**	**93.90±0.27**	78.17	**95.74**
ASH (SHOT)	**77.99±2.90**	**80.71±4.38**	**78.39±0.23**	79.33±1.06	81.95±0.36	92.69±0.22	**78.22**	95.26
KNN	**90.64±0.20**	**92.96±0.14**	**80.18±0.15**	82.40±0.17	81.57±0.17	**93.16±0.22**	71.10	**90.18**
KNN (SHOT)	90.50±0.19	92.77±0.28	79.91±0.23	**82.44±0.99**	**81.59±0.15**	91.49±0.32	**71.48**	89.93
VIM	88.79±0.33	**93.59±0.42**	74.71±0.28	**81.74±1.88**	78.68±0.24	91.26±0.19	74.61	89.66
VIM (SHOT)	**88.82±0.30**	93.57±0.43	**74.82±0.22**	81.42±1.24	**79.18±0.26**	**92.60±0.06**	**76.83**	**92.16**
VRA	**88.91±0.68**	**92.08±0.67**	**80.63±0.41**	**81.39±0.83**	82.22±0.15	**93.53±0.23**	**77.75**	**94.89**
VRA (SHOT)	87.40±0.50	91.57±1.24	80.06±0.15	81.12±1.08	**82.32±0.06**	92.56±0.22	75.29	91.85
PLF	**89.74±0.53**	**92.66±0.72**	80.82±0.13	**80.69±0.94**	83.28±1.11	**93.82±0.85**	78.15	**96.24**
PLF (SHOT)	86.75±0.63	90.63±1.30	**80.93±0.09**	80.43± 0.74	**83.73±0.10**	92.65±0.16	**78.70**	92.90

Table 4: Performance comparison of SoA OOD detectors tuned with SHOT versus given test OOD data. We compare the performance of OOD detectors tuned with SHOT to the detectors tuned with given OOD data, specifically selected in OpenOOD benchmarks to have favorable properties for tuning. Detectors tuned by SHOT sometimes out-performs the same detectors tuned with given OOD data, and in the cases it doesn't, SHOT is generally comparable in performance. Near-OOD and Far-OOD correspond to average AUROC on near and far OOD datasets of OpenOOD.

Held-out classes M	**PLF**					**ReAct**				
	Number of random holdout splits N					Number of random holdout splits N				
	1	2	3	4	5	1	2	3	4	5
50	87.78	86.61	87.25	86.66	87.04	84.42	83.98	83.87	84.31	84.31
100	85.29	85.97	86.82	86.81	86.53	83.61	83.66	83.71	83.79	83.87
150	86.89	86.47	86.37	86.56	86.06	81.40	84.15	84.05	84.05	84.18
200	86.82	85.97	86.83	86.30	86.25	84.09	83.96	84.09	84.09	84.20
250	83.60	85.13	85.28	85.47	86.57	83.44	83.69	83.94	83.94	83.78
300	86.22	86.40	86.52	87.18	86.80	83.54	84.22	84.23	84.29	84.38

Table 5: Compute-performance sensitivity of SHOT on ImageNet-1K with ResNet-50. Rows vary the number of held-out classes M, while columns vary the number of independently sampled holdout splits N. Values report AUROC averaged across the test OOD datasets for PLF and ReAct.

where offline training resources are constrained, this cost can be reduced by using fewer holdout splits and a smaller set of candidate values of holdout categories, with only modest impact on performance.

6 Discussion and Limitations

Our approach introduces a non-negligible offline computational cost compared to training the original network. However, this cost should be compared against the alternative: constructing an OOD tuning dataset in practical settings, which can often be substantially more burdensome, requiring e.g., engineering/infrastructure cost, curation cost, quality assessment and iterative validation. Appendix G analyzes this cost based on published campaigns and compares it to the cost of SHOT. The cost of collecting OOD tuning data can be 1 to 2 orders of magnitudes greater than the AWS computing cost of SHOT. An alternative to this cost is using existing data that is e.g., scraped from the web or other human chosen sources. In Appendix Table 10, we simulate this approach and show that SHOT achieves better results than that expected performance, further illustrating SHOT's advantage.

Our approach currently relies on availability of the training set, which may not be available in some applications, e.g, large language models. SHOT currently applies to classification, further research may be needed to extend the approach to tasks such as object detection, segmentation and regression.

7 Conclusion

We showed that OOD detectors can exhibit large variance across OOD tuning sets, in particular those chosen in existing benchmarks. This makes OOD detectors sensitive to the predefined tuning set. Moreover, obtaining such a tuning set in practice may be difficult. We formulated the new problem of tuning OOD parameters without a given (real) OOD tuning data. Compared to strong baselines that OOD practitioners have considered for related problems and might consider to tune detectors without given OOD tuning data, SHOT achieves >2x as many wins across 180 evaluations spanning diverse detectors and datasets. Our method thus provides a path to deploying reliable OOD detectors in practical applications. This work opens up future research on the tuning problem.

REFERENCES

Amazon Web Services. Amazon EC2 capacity blocks for ML pricing. https://aws.amazon.com/ec2/capacityblocks/pricing/, 2026. Accessed September 2026.

Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. ObjectNet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In *Advances in Neural Information Processing Systems*, volume 32, pp. 9448–9458. Curran Associates, Inc., 2019.

Eric Brochu, Matthew W. Hoffman, and Nando de Freitas. Portfolio allocation for bayesian optimization. *CoRR*, abs/1009.5419, 2010. URL http://arxiv.org/abs/1009.5419.

Younjoon Chung, Hyoungseob Park, Patrick Rim, Xiaoran Zhang, Jihe He, Ziyao Zeng, Safa Cicek, Byung-Woo Hong, James S Duncan, and Alex Wong. Eta: Energy-based test-time adaptation for depth completion. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–12. IEEE, 2025.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3213–3223, June 2016.

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 Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, pp. 248–255, Los Alamitos, CA, USA, June 2009. IEEE Computer Society. doi: 10.1109/CVPR.2009.5206848. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2009.5206848.

Andrija Djurisic, Nebojsa Bozanic, Arjun Ashok, and Rosanne Liu. Extremely simple activation shaping for out-of-distribution detection. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ndYXTEL6cZz.

Peter I. Frazier. A tutorial on bayesian optimization, 2018. URL https://arxiv.org/abs/1807.02811.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1412.6572.

Charles Guille-Escuret, Pierre-André Noël, Ioannis Mitliagkas, David Vazquez, and Joao Monteiro. Expecting the unexpected: Towards broad out-of-distribution detection. *Advances in Neural Information Processing Systems*, 37:130953–130976, 2024.

Dongyoon Han, Junsuk Choe, Seonghyeok Chun, John Joon Young Chung, Minsuk Chang, Sangdoo Yun, Jean Y. Song, and Seong Joon Oh. Neglected free lunch – learning image classifiers using annotation byproducts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20200–20212, October 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision – ECCV 2016*, pp. 630–645, Cham, 2016a. Springer International Publishing.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016b. doi: 10.1109/CVPR.2016.90. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Hkg4TI9xl.

Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joseph Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8759–8773. PMLR, 17–23 Jul 2022.

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

Konstantin Kirchheim, Marco Filax, and Frank Ortmeier. Pytorch-ood: A library for out-of-distribution detection based on pytorch. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4351–4360, 2022.

Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. Why normalizing flows fail to detect out-of-distribution data. *Advances in neural information processing systems*, 33:20578–20589, 2020.

Lisa Jeanne Koetke, Dexter P. Hodder, and Chris J. Johnson. Using camera traps and N-mixture models to estimate population abundance: Model selection really matters. *Methods in Ecology and Evolution*, 15(5):900–915, 2024. doi: 10.1111/2041-210X.14320.

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

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 and cifar-100 datasets. `https://www.cs.toronto.edu/~kriz/cifar.html`, 2009b. Accessed: 2026-05-02.

Harold J Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1):97–106, 1964.

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.

Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=H1VGkIxRZ`.

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21464–21475, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/f5496252609c43eb8a3d147ab9b9c006-Abstract.html`.

Sudeepta Mondal, Zhuolin Jiang, and Ganesh Sundaramoorthi. A variational information theoretic approach to out-of-distribution detection. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. URL `https://arxiv.org/abs/2506.14194`.

NVIDIA. ResNet-50 v1.5 for PyTorch: Training and Performance Benchmark. `https://github.com/NVIDIA/DeepLearningExamples/tree/master/PyTorch/Classification/ConvNets/resnet50v1.5`, 2026. NVIDIA DeepLearningExamples; accessed September 2026.

Dim P Papadopoulos, Ethan Weber, and Antonio Torralba. Scaling up instance annotation via label propagation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15364–15373, 2021.

Jie Ren, Stanislav Fort, Jeremiah Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. A simple fix to mahalanobis distance for improving near-ood detection. *arXiv preprint arXiv:2106.09022*, 2021.

Chandramouli Shama Sastry and Sageev Oore. Detecting out-of-distribution examples with gram matrices. In *International conference on machine learning*, pp. 8491–8501. PMLR, 2020.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018.

Niranjan Srinivas, Andreas Krause, Matthias Seeger, and Sham M. Kakade. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In Johannes F"urnkranz and Thorsten Joachims (eds.), *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 1015–1022. Omnipress, 2010.

David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6976–6987, 2019.

Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. In *Advances in Neural Information Processing Systems*, volume 34, pp. 144–157, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/251d5fcd701a661b64f9745c8e4a26c3-Abstract.html.

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

Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *European Conference on Computer Vision*, pp. 560–574. Springer, 2018.

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*, pp. 4921–4930, 2022.

Mingyu Xu, Zheng Lian, Bin Liu, and Jianhua Tao. Vra: Variational rectified activation for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/5c20c00504e0c049ec2370d0cceaf3c4-Abstract-Conference.html.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 132:5635–5662, 2024a. doi: 10.1007/s11263-024-02117-4. URL https://doi.org/10.1007/s11263-024-02117-4.

William Yang, Byron Zhang, and Olga Russakovsky. Imagenet-ood: Deciphering modern out-of-distribution detection algorithms. In *International Conference on Learning Representations*, volume 2024, pp. 41877–41906, 2024b.

Jingyang Zhang, Jingkang Yang, Pengyun Wang, Haoqi Wang, Yueqian Lin, Haoran Zhang, Yiyou Sun, Xuefeng Du, Yixuan Li, Ziwei Liu, Yiran Chen, and Hai Li. Openood v1.5: Enhanced benchmark for out-of-distribution detection, 2024. URL https://arxiv.org/abs/2306.09301.

Xiaoran Zhang, Byung-Woo Hong, Hyoungseob Park, Daniel H Pak, Anne-Marie Rickmann, Lawrence H Staib, James S Duncan, and Alex Wong. Progressive test time energy adaptation for medical image segmentation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22338–22348. IEEE, 2025.

Jian Zhao, Wenxu Li, Bing Zhu, Peixing Zhang, Zhaozheng Hu, and Jie Meng. Corner case dataset for autonomous vehicle testing based on naturalistic driving data. *Smart Cities*, 8(4):129, 2025. doi: 10.3390/smartcities8040129.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Dae ki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018. URL https://api.semanticscholar.org/CorpusID:51805340.

APPENDIX

A DETECTOR DEFINITIONS AND PARAMETER SELECTION

ReAct. ReAct Sun et al. (2021) clips penultimate-layer feature activations element-wise:

$$\mathrm{ReAct}(z)_i = \min(z_i, \tau).$$

The single parameter τ is optimized by Bayesian optimization. The search interval is defined from the range of activations from the penultimate layer obtained with the ID data, similar to the setup of Sun et al. (2021).

ASH. ASH Djurisic et al. (2023) sparsifies intermediate activations by retaining the top-k elements and suppressing the rest, where k is determined by a percentile parameter p. The retained activations are set to a constant proportional to the activation energy. The only tunable parameter is p, which is optimized by Bayesian optimization over a valid percentile range on simulated ID/OOD splits. We use the ASH-B variant of ASH in our benchmarks.

VRA. VRA+ Xu et al. (2023) applies a piecewise shaping function:

$$\mathrm{VRA}(z) = \begin{cases} 0, & z < \alpha, \\ z + \gamma, & \alpha \le z \le \beta, \\ \beta, & z > \beta. \end{cases}$$

Rather than optimizing α and β directly, we parameterize them by quantiles of the empirical ID feature distribution:

$$\alpha = F_{\mathrm{ID}}^{-1}(\eta_\alpha), \quad \beta = F_{\mathrm{ID}}^{-1}(\eta_\beta).$$

The optimization variables are (η_α, u, γ) with

$$\eta_\alpha \in [0.1, 0.8], \quad u \in [0, 1], \quad \gamma \in [0, 5],$$

and $\eta_\beta = \eta_\alpha + \delta$, where

$$\delta = \delta_{\min} + u(\delta_{\max} - \delta_{\min}), \quad \delta_{\min} = 0.10, \quad \delta_{\max} = 0.99 - \eta_\alpha,$$

ensuring $\eta_\beta > \eta_\alpha$. Bayesian optimization selects $(\eta_\alpha, \eta_\beta, \gamma)$ on simulated ID/OOD splits.

PLF. PLF applies a seven-parameter piecewise linear shaping function to features Mondal et al. (2025), where $f(\cdot)$ consists of three linear segments defined by breakpoints x_1, x_2 and slopes m_1, m_2 with vertical offsets $(y_{\text{start}}, y_{\text{end}}, y_1)$. The breakpoints are defined by quantiles of the absolute ID features:

$$x_1 = F_{|Z_{\mathrm{ID}}|}^{-1}(q_1), \quad x_2 = F_{|Z_{\mathrm{ID}}|}^{-1}(q_2),$$

with $q_2 = q_1 + \delta$. The optimization variables are

$$(y_{\text{start}}, y_{\text{end}}, \Delta y, q_1, u, m_1, m_2),$$

with

$$y_{\text{start}} \in [-5, 0], \quad y_{\text{end}} \in [0, 5], \quad \Delta y \in [0, 5],$$
$$q_1 \in [0.1, 0.8], \quad u \in [0, 1], \quad m_1 \in [0, 5], \quad m_2 \in [-5, 5],$$

and $y_1 = y_{\text{end}} + \Delta y$. The upper quantile is defined as

$$\delta = \delta_{\min} + u(\delta_{\max} - \delta_{\min}), \quad \delta_{\min} = 0.10, \quad \delta_{\max} = 0.99 - q_1,$$

so that $q_2 = q_1 + \delta$ and $q_2 > q_1$. Parameters are selected by Bayesian optimization on simulated ID/OOD splits.

KNN. KNN Sun et al. (2022) computes OOD scores using the negative distance to the K-th nearest neighbor in normalized feature space. The only tunable parameter is K. We search over integer $K \in [1, K_{\max}]$ using Bayesian optimization, where distances are precomputed from a feature index built on ID training data. We choose $K_{max} = 500$.

VIM. VIM Wang et al. (2022) computes an OOD score by combining the classifier logits with a feature residual term. Let P_D denote the principal subspace spanned by the top D eigenvectors of the ID feature covariance, and let $\mathbf{u} = -(W^\top)^+ b$ denote the bias offset. The residual is computed as the norm of the component orthogonal to this subspace:

$$\text{Residual}(z) = \|(I - P_D P_D^\top)(z - \mathbf{u})\|.$$

The only tunable parameter is the subspace dimension D. We optimize D using Bayesian optimization over an integer range: $D \in [50, 400]$ for ResNet-18 (penultimate feature layer dimension 512), and $D \in [50, 1000]$ for larger scale networks like ResNet-50 (penultimate feature layer dimension 2048).

Optimization Protocol. For all detectors, parameters are selected using Bayesian optimization with a Gaussian process surrogate on simulated ID/OOD splits. The objective is to maximize AUROC computed from energy-based scores after applying the corresponding feature transformation.

B Bayesian Optimization Details

For all detectors and tuning methods, we use 100 Bayesian optimization evaluations, consisting of 80 random initializations followed by 20 guided Bayesian optimization steps. We use the *GP-Hedge* acquisition strategy (Brochu et al., 2010), which adaptively balances exploration and exploitation across multiple acquisition functions: Probability of Improvement (Kushner, 1964), Expected Improvement (Jones et al., 1998), and Lower Confidence Bound (Srinivas et al., 2010).

The same optimization procedure is used for all detectors and tuning methods. The detector-specific parameter search ranges are described in Appendix A. These ranges follow the corresponding detector definitions and, where applicable, the settings used in OpenOOD (Zhang et al., 2024).

C Visualization of Simulated ID/OOD Data and Near/Far OOD

We provide qualitative validation of intuition of Figure 2 using t-SNE plots. The figure shows that SHOT held-out (simulated OOD) data lies near the ID/OOD boundary, and thus separating it from held-in ID data approximates the true ID/OOD boundary.

D Additional Ablations on Holdout Size

We present additional ablations on the number of simulated OOD categories M for ImageNet-200 and ImageNet-1K in Tables 6 and 7, respectively. These experiments complement the compute-performance ablation in the main paper by evaluating the sensitivity of SHOT to the holdout size across all detector families. For most detectors, AUROC varies only modestly across the considered values of M, with some method-dependent fluctuations. Overall, the results indicate that SHOT is relatively robust to the choice of holdout size over a broad range of simulated OOD categories on both ImageNet benchmarks. We observe similar trends on the CIFAR benchmarks, and omit those results for brevity.

E Comparison of State of the Art Parametric and Parameter-free OOD Detectors

We report results for top-performing post-hoc OOD detectors in OpenOOD benchmark (Zhang et al., 2024) in Table 8 to justify our choice of the six chosen parametric OOD detectors to benchmark tuning approaches in the main paper. Also, the first three rows show performance of top performing parameter-free methods, showing that they largely do not compete with parametric methods with the exception of one on small scale benchmarks. This justifies our research into tuning parametric detectors when given OOD data is unavailable as parameter free detectors (that do not require given OOD data) are not competitive. In Table 9, we show that tuning OOD detectors with SHOT do not result in any performance degradation, and maintains the conclusion that parameter-free methods are uncompetitive with parametric approaches.

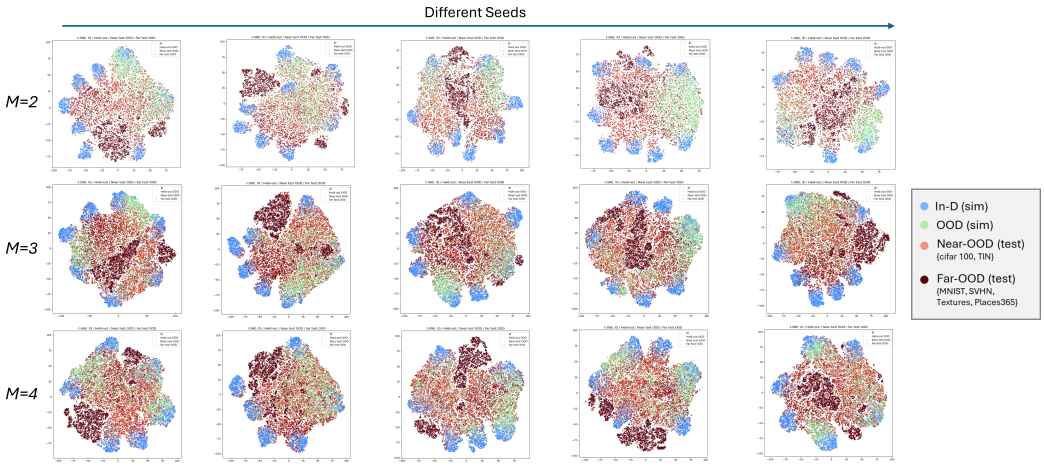

Figure 4: **Qualitative Evidence for the Intuition of Our Method in Figure 2.** We experiment with CIFAR-10 and ResNet-18, and make plots of penultimate layer features using the t-SNE visualization. The top row shows visualization of features for the network trained using $M = 2$ held-out classes, each column is a different random selection of 2 classes left out. The blue dots represent held-in (simulated ID) data features, green is left-out (simulated OOD) data features, in light red are near-OOD data and dark red is far-OOD data (as labeled in the OpenOOD benchmark). Notice the simulated OOD in green is typically near the true ID/OOD boundary; the union of all the green approximately captures the full ID/OOD boundary. The middle and last rows show the same observations for $M = 3$ and $M = 4$ left out classes.

		ImageNet-200 (ResNet-18)					
Method	Holdout	SSB-hard	NINCO	iNaturalist	Textures	OpenImage-O	AVG
ReAct	10	79.83 ± 0.02	85.17 ± 0.12	92.56 ± 0.50	90.87 ± 0.16	89.25 ± 0.25	87.536
	20	79.83 ± 0.02	85.17 ± 0.11	92.55 ± 0.50	90.79 ± 0.16	89.23 ± 0.26	87.514
	30	85.17 ± 0.11	82.50 ± 0.05	90.79 ± 0.16	89.23 ± 0.26	90.86 ± 0.21	87.710
	40	79.83 ± 0.02	85.17 ± 0.11	92.55 ± 0.50	90.79 ± 0.16	89.23 ± 0.26	87.514
ASH	10	79.38 ± 0.29	84.51 ± 0.43	94.52 ± 0.41	93.14 ± 0.17	90.42 ± 0.09	88.394
	20	79.74 ± 0.25	85.18 ± 0.51	94.91 ± 0.46	93.86 ± 0.10	91.21 ± 0.14	88.980
	30	79.86 ± 0.21	85.48 ± 0.58	95.04 ± 0.46	94.33 ± 0.08	91.59 ± 0.18	89.260
	40	79.86 ± 0.21	85.48 ± 0.58	95.04 ± 0.46	94.33 ± 0.08	91.59 ± 0.18	89.260
KNN	10	77.65 ± 0.16	85.52 ± 0.13	91.23 ± 0.51	94.78 ± 0.08	88.45 ± 0.38	87.526
	20	77.65 ± 0.16	85.52 ± 0.13	91.23 ± 0.51	94.78 ± 0.08	88.45 ± 0.38	87.526
	30	77.42 ± 0.20	86.17 ± 0.10	93.58 ± 0.37	95.17 ± 0.02	89.88 ± 0.31	88.444
	40	77.54 ± 0.19	86.13 ± 0.10	93.28 ± 0.38	95.11 ± 0.03	89.66 ± 0.32	88.344
VIM	10	71.32 ± 1.99	79.61 ± 2.84	88.90 ± 3.13	84.20 ± 7.67	83.78 ± 3.86	81.562
	20	71.58 ± 1.79	79.85 ± 2.65	89.15 ± 2.98	84.57 ± 7.40	84.07 ± 3.64	81.844
	30	74.27 ± 0.30	83.90 ± 0.22	93.24 ± 0.07	95.04 ± 0.03	89.43 ± 0.15	87.180
	40	74.47 ± 0.29	83.90 ± 0.22	93.31 ± 0.07	95.06 ± 0.04	89.42 ± 0.15	87.232
PLF	10	80.95 ± 0.04	86.52 ± 0.16	94.16 ± 0.46	92.25 ± 0.13	90.66 ± 0.20	88.908
	20	80.66 ± 0.04	86.28 ± 0.14	93.84 ± 0.46	92.09 ± 0.13	90.48 ± 0.19	88.670
	30	80.32 ± 0.05	86.03 ± 0.15	93.88 ± 0.38	92.37 ± 0.14	90.49 ± 0.17	88.618
	40	80.04 ± 0.04	86.30 ± 0.07	93.84 ± 0.43	93.30 ± 0.12	90.81 ± 0.14	88.858
VRA	10	79.25 ± 0.06	85.03 ± 0.16	92.84 ± 0.53	92.56 ± 0.15	90.06 ± 0.36	87.948
	20	79.26 ± 0.04	85.40 ± 0.13	93.40 ± 0.48	93.48 ± 0.14	90.67 ± 0.33	88.442
	30	79.22 ± 0.05	85.30 ± 0.14	93.27 ± 0.49	93.26 ± 0.13	90.51 ± 0.33	88.312
	40	79.28 ± 0.04	85.37 ± 0.13	93.37 ± 0.48	93.59 ± 0.14	90.73 ± 0.32	88.468

Table 6: AUROC on ImageNet-200 (ResNet-18) under Different Holdout-Class Settings

Method	Holdout	ImageNet-1K (ResNet-50) SSB-hard	NINCO	iNaturalist	Textures	OpenImage-O	AVG
ReAct	50	73.15	81.76	95.05	91.87	91.44	86.66
	100	72.48	81.39	94.23	91.14	90.92	86.14
	150	73.18	81.69	94.88	91.72	91.33	86.56
	200	73.61	81.65	94.78	91.63	91.27	86.50
	250	73.00	81.55	94.56	91.43	91.13	86.36
	300	73.27	81.76	95.04	91.87	91.43	86.66
ASH	50	73.15	83.30	96.73	96.48	93.04	88.54
	100	73.20	83.26	96.69	96.41	93.00	88.51
	150	73.05	83.37	96.83	96.63	93.17	88.61
	200	73.11	83.31	96.76	96.54	93.09	88.56
	250	73.27	83.17	96.60	96.29	92.90	88.45
	300	73.05	83.34	96.81	96.61	93.16	88.60
KNN	50	61.86	79.27	86.78	97.27	87.10	82.46
	100	61.74	79.20	86.82	97.30	87.10	82.43
	150	63.09	79.87	85.99	96.91	86.90	82.55
	200	63.09	79.87	85.99	96.91	86.90	82.55
	250	63.09	79.87	85.99	96.91	86.90	82.55
	300	63.09	79.87	85.99	96.91	86.90	82.55
VIM	50	71.88	81.78	94.11	91.40	90.97	86.03
	100	71.88	81.78	94.11	91.40	90.97	86.03
	150	71.88	81.78	94.11	91.40	90.97	86.03
	200	71.88	81.78	94.11	91.40	90.97	86.03
	250	70.04	81.17	93.46	89.99	89.48	84.83
	300	70.92	81.69	93.92	90.66	90.16	85.47
PLF	50	73.20	82.90	95.40	92.45	91.98	87.27
	100	73.98	82.65	94.87	91.19	91.45	86.83
	150	73.81	82.34	93.76	89.77	90.55	86.37
	200	73.80	81.47	93.03	87.55	89.59	85.09
	250	73.17	81.83	93.76	89.77	90.55	86.11
	300	74.44	82.95	95.41	91.55	91.72	87.21
VRA	50	70.33	80.28	92.98	94.00	91.23	85.76
	100	70.34	80.22	92.88	93.87	91.15	85.69
	150	70.74	79.85	92.16	92.82	90.57	85.23
	200	70.70	79.85	92.18	92.86	90.58	85.24
	250	71.05	79.91	92.14	92.62	90.53	85.25
	300	70.94	79.87	92.12	92.65	90.52	85.22

Table 7: AUROC on ImageNet-1K under different holdout-class settings.

		CIFAR-10 (ResNet-18)			**CIFAR-100** (ResNet-18)			**ImageNet-200** (ResNet-18)			**ImageNet-1K** (ResNet-50)		
	Method	Near-OOD	Far-OOD	ID Acc.	Near-OOD	Far-OOD	ID Acc.	Near-OOD	Far-OOD	ID Acc.	Near-OOD	Far-OOD	ID Acc.
Param. free	MSP	88.03±0.25	90.73±0.43	95.06±0.30	80.27±0.11	77.76±0.44	77.25±0.10	**83.34±0.06**	90.13±0.09	86.37±0.08	76.02	85.23	76.18
	Gram	58.66±4.83	71.73±3.20	95.06±0.30	51.66±0.77	73.36±1.08	77.25±0.10	67.67±1.07	71.19±0.24	86.37±0.08	61.70	79.71	76.18
	RMDS	89.80±0.28	92.20±0.21	95.06±0.30	80.15±0.11	**82.92±0.42**	77.25±0.10	82.57±0.25	88.06±0.34	86.37±0.08	76.99	86.38	76.18
Parametric ('Given'-OOD tuned)	ReAct	87.11±0.61	90.42±1.41	95.06±0.30	80.77±0.05	80.39±0.49	77.25±0.10	81.87±0.98	92.31±0.56	86.37±0.08	77.38	93.67	76.18
	ASH	75.27±1.04	78.49±2.58	95.06±0.30	78.20±0.15	80.58±0.66	77.25±0.10	82.38±0.19	**93.90±0.27**	86.37±0.08	**78.17**	95.74	76.18
	KNN	**90.64±0.20**	92.96±0.14	95.06±0.30	80.18±0.15	82.40±0.17	77.25±0.10	81.57±0.17	93.16±0.22	86.37±0.08	71.10	90.18	76.18
	VIM	88.79±0.33	**93.59±0.42**	95.06±0.30	74.71±0.28	81.74±1.88	77.25±0.10	78.68±0.24	91.26±0.19	86.37±0.08	74.61	89.66	76.18
	VRA	88.91±0.68	92.08±0.67	95.06±0.30	80.63±0.41	81.39±0.83	77.25±0.10	82.22±0.15	93.53±0.23	86.37±0.08	77.75	94.89	76.18
	PLF	89.74±0.53	92.66±0.72	95.06±0.30	**80.82±0.13**	80.69±0.94	77.25±0.10	83.28±1.11	93.82±0.85	86.37±0.08	78.15	**96.24**	76.18

Table 8: **Comparison of SoA Parameter- Free OOD Detectors to Parametric OOD Detectors Tuned on OpenOOD 'Given' Tuning Data** Performance comparison of OOD detectors on the OpenOOD benchmark across CIFAR-10, CIFAR-100, ImageNet-200, and ImageNet-1K using OpenOOD protocols ('given' OOD data for tuning). Detectors are grouped by the number of tunable parameters: top performing parameter-free (MSP, Gram, RMDS) methods, and top performing parametric methods - 1-parameter (ReAct, ASH, KNN, VIM), 3-parameter (VRA), and 7-parameter (PLF). Near-OOD and Far-OOD denote mean AUROC over the corresponding OOD datasets, and ID Acc. denotes in-distribution accuracy. Bold and underlined values indicate the first- and second-ranked results, respectively. These results show that parameter-free methods are largely uncompetitive with parameter-based approaches except sporadically in small scale benchmarks, consistent with observations in Hendrycks et al. (2022).

		CIFAR-10 (ResNet-18)			**CIFAR-100** (ResNet-18)			**ImageNet-200** (ResNet-18)			**ImageNet-1K** (ResNet-50)		
	Method	Near-OOD	Far-OOD	ID Acc.	Near-OOD	Far-OOD	ID Acc.	Near-OOD	Far-OOD	ID Acc.	Near-OOD	Far-OOD	ID Acc.
Param. free	MSP	88.03±0.25	90.73±0.43	95.06±0.30	80.27±0.11	77.76±0.44	77.25±0.10	83.34±0.06	90.13±0.09	86.37±0.08	76.02	85.23	76.18
	Gram	58.66±4.83	71.73±3.20	95.06±0.30	51.66±0.77	73.36±1.08	77.25±0.10	67.67±1.07	71.19±0.24	86.37±0.08	61.70	79.71	76.18
	RMDS	89.80±0.28	92.20±0.21	95.06±0.30	80.15±0.11	**82.92±0.42**	77.25±0.10	82.57±0.25	88.06±0.34	86.37±0.08	76.99	86.38	76.18
Parametric (SHOT-tuned)	ReAct	87.53±0.73	90.97±1.59	95.06±0.30	80.52±0.06	80.76±0.91	77.25±0.10	82.50±0.07	90.89±0.30	86.37±0.08	77.44	92.64	76.18
	ASH	77.99±2.90	80.71±4.38	95.06±0.30	78.39±0.23	79.33±1.06	77.25±0.10	81.95±0.36	**92.69±0.22**	86.37±0.08	78.22	**95.26**	76.18
	KNN	**90.50±0.19**	92.77±0.28	95.06±0.30	79.91±0.23	82.44±0.99	77.25±0.10	81.59±0.15	91.49±0.32	86.37±0.08	71.48	89.93	76.18
	VIM	88.82±0.30	**93.57±0.43**	95.06±0.30	74.82±0.22	81.42±1.24	77.25±0.10	79.18±0.26	92.60±0.06	86.37±0.08	76.83	92.16	76.18
	VRA	87.40±0.50	91.57±1.24	95.06±0.30	80.06±0.15	81.12±1.08	77.25±0.10	82.32±0.06	92.56±0.22	86.37±0.08	75.29	91.85	76.18
	PLF	86.75±0.63	90.63±1.30	95.06±0.30	**80.93±0.09**	80.43±0.74	77.25±0.10	**83.73±0.10**	92.65±0.16	86.37±0.08	**78.70**	92.90	76.18

Table 9: **Comparison of SoA Parameter- Free OOD Detectors to SHOT-tuned Parametric OOD Detectors** Performance comparison of OOD detectors on the OpenOOD benchmark across CIFAR-10, CIFAR-100, ImageNet-200, and ImageNet-1K. We compare SHOT-tuned parametric detectors to parameter-free methods. Detectors are grouped by the number of tunable parameters: top performing parameter-free (MSP, Gram, RMDS) methods, and top performing parametric methods - 1-parameter (ReAct, ASH, KNN, VIM), 3-parameter (VRA), and 7-parameter (PLF). Near-OOD and Far-OOD denote mean AUROC over the corresponding OOD datasets, and ID Acc. denotes in-distribution accuracy. Bold and underlined values indicate the first- and second-ranked results, respectively. These results show that SHOT-tuned detectors compare similarly to 'given' OOD tuning and do not change the conclusion that parameter-free methods are largely uncompetitive with parameter-based approaches except sporadically in small scale benchmarks.

Note OpenOOD specifies a given OOD tuning set from TinyImageNet for CIFAR 10/100, and OpenImage-O for ImageNet 200/1K on which the hyperparameters of the OOD detectors are tuned. These tuning sets are specifically chosen for favorable properties as tuning sets.

F Comparison of SHOT to Different 'Given' OOD Tuning Sets

In practice, to tune a detector, one could collect a new OOD dataset specifically for the application of interest or perhaps choose from existing data already collected (e.g., scraped from the web), which could avoid expensive data collects. The cost of the physical collection is expensive and described in the next section. We now analyze whether using existing collected data is a viable option for tuning, and how SHOT compares to that approach. In particular, we now show that if one were to select an already collected dataset, it would in expected value perform worse than SHOT, in most cases. This provides additional justification for SHOT over the scenario where one just uses existing data, avoiding collection costs.

On ImageNet-1K with ResNet-50, for each target OOD dataset in OpenOOD, we separately tune each detector using each of the other OOD datasets and evaluate on the target dataset. Across the 20 dataset/detector combinations considered, SHOT achieves higher AUROC than the mean performance

Method	Testing Dataset	SHOT	Mean	Min	Max	Std. Dev.
REACT	ssb_hard	**73.16**	73.06	72.85	73.20	0.13
	ninco	**81.84**	81.75	81.14	82.05	0.36
	inaturalist	94.86	**95.74**	94.78	96.24	0.58
	textures	91.60	**92.20**	91.53	92.65	0.42
	openimage_o	91.33	**91.60**	91.28	91.92	0.29
ASH	ssb_hard	73.04	**73.36**	73.01	74.26	0.52
	ninco	**83.45**	83.03	82.52	83.42	0.39
	inaturalist	96.82	**97.03**	96.72	97.33	0.27
	textures	**96.53**	96.40	96.26	96.51	0.09
	openimage_o	**93.15**	92.58	91.92	93.12	0.51
VRA	ssb_hard	**75.71**	74.60	73.56	75.43	0.67
	ninco	**83.30**	81.58	78.48	83.05	1.81
	inaturalist	96.97	**97.02**	96.75	97.23	0.19
	textures	**93.76**	93.37	90.75	94.28	1.51
	openimage_o	**92.40**	91.22	88.30	92.56	1.71
PLF	ssb_hard	**73.51**	71.67	70.77	73.84	1.26
	ninco	**83.11**	81.95	79.21	82.92	1.58
	inaturalist	**95.34**	94.31	87.98	96.91	3.69
	textures	**92.38**	90.02	75.40	96.38	8.63
	openimage_o	**91.95**	90.53	84.22	93.01	3.67

Table 10: Comparison of SHOT results to tuning with different given OOD tuning sets in the ImageNet-1K/ResNet50 benchmark of OpenOOD. For each testing set, we tune the detector on all other OpenOOD test sets and report the mean, min, max and standard deviation. The mean value approximates the expected value of choosing an existing dataset e.g., scraped from the web or other human selected dataset for the tuning set. As can be seen, SHOT mostly out-performs the expectation, showing on average SHOT would beat selecting from available datasets, which may not be specifically crafted for the test OOD.

obtained from alternative-OOD tuning in 14/20 cases. In the remaining cases, SHOT is still better than the worst alternative OOD tuning choice. These results highlight the risk of relying on an available but potentially mismatched OOD tuning set. Full results are provided in Table 10.

G Cost of OOD Data Collection Compared with SHOT

The following estimates are intended as illustrative examples grounded in published annotation and field-collection efforts rather than universal costs; actual costs can be substantially higher depending on the sensing modality, operating environment, rarity of the desired conditions, and logistical or regulatory constraints.

Cost of OOD tuning data. Annotation of already available images can be relatively inexpensive. Han et al. (2023), replicating the original ImageNet annotation procedure, report approximately $24,000 for a single annotation pass over the 1.28M-image ImageNet-1K training set and estimate approximately $240,000 to reproduce ImageNet's original ten-annotator protocol, corresponding to roughly $0.19 per training image. In contrast, detailed annotations can be considerably more expensive: Cityscapes reports more than 1.5 hours of annotation effort per image for fine pixel-level labels (Cordts et al., 2016). Thus, annotation cost depends strongly on the application, but for the image-level verification required for OOD tuning it is unlikely to dominate the cost of physically acquiring new observations.

Physical data acquisition can be substantially more expensive. Even in the relatively simple ObjectNet collection process, where workers physically located, posed, and photographed objects, approximately 95,824 images were acquired to retain 50,000 after validation, nearly two acquisitions for every usable image (Barbu et al., 2019). More representative field sensing can incur much greater costs:

Koetke et al. (2024) report \$24,476 and 85 person-days for a camera-based field campaign, including \$11,840 for field personnel, \$5,696 for transportation, \$3,240 for equipment, and \$3,700 for data processing.

OOD acquisition adds further burden because atypical conditions must often be deliberately sought, staged, or identified from substantially larger streams of routine observations. ObjectNet's approximately 52% retention rate already implies nearly a twofold acquisition overhead, while naturalistic-driving studies report that unusual corner cases constitute only a small fraction of collected scenarios (Zhao et al., 2025). Although corner cases are not synonymous with OOD samples, they illustrate the reduced yield associated with rare or atypical conditions. We therefore use a conservative $2\times$ multiplier on physical collection effort for OOD acquisition. Applying this factor to the \$24,476 field-campaign cost reported by Koetke et al. (2024) gives

$$2 \times \$24{,}476 \approx \$49{,}000,$$

or approximately \$50,000 for a standalone targeted OOD collection campaign. The additional cost of ImageNet-style image-level annotation is small relative to this physical acquisition cost.

We consider two practical cases. If the model relies on legacy ID data collected before OOD detection was considered, acquiring OOD data can require a new standalone field campaign; we therefore use approximately \$50,000 as an illustrative incremental cost. If ID and OOD data are collected within the same active program, equipment, transportation, mobilization, and portions of the processing infrastructure can be shared. Equipment and transportation account for approximately 36% of the Koetke et al. (2024) campaign cost; using a more conservative 30% shared-cost allowance gives

$$(1 - 0.30)(\$49{,}000) \approx \$34{,}000,$$

which we round to approximately \$35,000. We therefore use approximately \$35,000–\$50,000 as illustrative incremental costs.

Comparison with SHOT. SHOT replaces physical OOD-data acquisition with computation on the existing ID training set by training held-out task-network variants and using them to construct simulated OOD tuning data. For an ImageNet-scale ResNet-50 task, NVIDIA reports approximately three hours for a 90-epoch training run on eight A100 GPUs NVIDIA (2026). Using an AWS `p4d.24xlarge` at approximately \$11.80/hour Amazon Web Services (2026), the 30 training runs required for our ImageNet experiments cost approximately

$$30 \times 3 \times \$11.80 \approx \$1{,}062.$$

Allowing an additional 25% for detector optimization, evaluation, checkpointing, and storage, we conservatively estimate the total SHOT tuning cost at approximately \$1,300.

Thus, SHOT replaces roughly \$35,000–\$50,000 of incremental OOD-data collection and curation with approximately \$1,300 of cloud computation, corresponding to a reduction of approximately $27\times$–$38\times$, but could be substantially higher based on application. The savings can be even larger when OOD events are rare or costly to reproduce.