Few-shot Out-of-Distribution Detection

Kuan-Chieh Wang [1 2] **Paul Vicol** [1 2] **Eleni Triantafillou** [1 2] **Richard Zemel** [1 2]

Abstract

Out-of-distribution (OOD) detection is an important problem in real-world settings, and has inspired a wide range of methods, from simple ones based on the predicted probability of a classifier to more complicated ones based on likelihood ratios under deep generative models. We consider a variant of the OOD detection task appropriate to settings such as few-shot learning, in which classification involves a restricted set of novel classes. We establish baselines on the few-shot OOD detection tasks by adapting state-of-the-art OOD methods from the standard classification setting to the few-shot setting. Interestingly, strong baselines designed for the large data setting perform well in the few-shot setting after simple adaptation. Then we present a method for FS-OOD detection that specifically utilizes the structure of the few-shot problem, and show that it outperforms the previous methods Furthermore, we demonstrate that improvements in few-shot OOD detection can benefit downstream tasks, such as active learning and semi-supervised learning.

1. Introduction

A system for identifying out-of-distribution (OOD) datapoints is useful in various ways. From the perspective of AI safety, OOD detection is essential for preventing a recognition system from making mistakes on inputs not belonging to the task it was trained on (Amodei et al., 2016). In applications such as detecting malicious attackers of a computer system by recognizing unusual actions (Lane & Brodley, 1997), and discovering new species of bacteria (Ren et al., 2019), a good OOD data detector can discover these rare and potentially meaningful events.

In some applications, the task consists of learning about a small set of novel classes, each specified with only a few labeled examples. A speaker recognition system built to transcribe conversations from novel users could be given only a few registration utterances from the target speakers (Chung et al., 2020). When an unregistered speaker abruptly joins the conversation, a good system should flag them as unregistered, and not mistake their utterances as coming from one of the registered speakers. Another example is customized facial gesture recognition for previously unseen users (Wang et al., 2019). Such applications that enable customization to new scenarios can be naturally formulated as few-shot (FS) learning problems, the paradigm of learning where a model is asked to learn about new concepts from only a few examples (Lake et al., 2011). Motivated by these applications, the focus on the present study is this central task of OOD detection in the few-shot setting, which we will call *few-shot out-of-distribution (FS-OOD)* detection.

Recently, there has been a growing interest in the few-shot classification literature on the topic of uncertainty quantification (Ravi & Beatson, 2019; Finn et al., 2018; Yoon et al., 2018). Some of these studies (e.g., (Ravi & Beatson, 2019)) considered FS-OOD detection as an application demonstrating the utility of Bayesian methods. However, the lack of FS-OOD detection baselines makes interpreting their results, or FS-OOD detection results in general, difficult.

Contributions. In this paper, we contribute to the study of FS-OOD detection as follows:

- We establish competitive adapted OOD baselines by leveraging the progress in standard OOD detection, and our insights into differences between FS- and standard OOD detection.
- We propose a novel method, the *Out-of-Episode Classifier* (OEC), for FS-OOD detection that outperforms these adapted baselines in many relevant settings.
- We show the effect of FS-OOD detection on downstream classification, as a further demonstration of the utility of studying FS-OOD detection.

2. Background

This section lays down notation and terminology used in this study. We briefly describe the task of few-shot learning, and standard OOD detection. See Appendix A for a table of notation.

[1]Department of Computer Science, University of Toronto, Toronto, Canada [2]Vector Institute, Toronto, Canada. Correspondence to: Kuan-Chieh Wang <wangkua1@cs.toronto.edu>.

Presented at the ICML 2020 Workshop on Uncertainty and Robustness in Deep Learning.

2.1. Few-Shot Classification

In standard classification, models are trained and tested on the same set of classes. In few-shot classification (FSC), a model is trained on a set of training classes $\mathcal{C}^{train}$, and tested on a set of unseen test classes $\mathcal{C}^{test}$. In FSC, the aim is to learn about these new classes given only a few labeled examples. Each round of evaluation is referred to as an *episode*. In an episode, given a support set, $S = \{S_c = \{(\mathbf{x}_{i,c}, y_{i,c})\}_{i=1}^{N_S} | c \in \mathcal{C}^{episode}\}$ the model must classify each of the unlabeled queries $Q = \{\mathbf{x}_i\}_{i=1}^{N_Q}$ into one of the classes in $\mathcal{C}^{episode}$. S_c denotes the subset of the support set containing data from class c. $\mathbf{x}$ denotes an input vector/image, and y a class label in an episode. $\mathcal{C}^{episode}$ is the set of classes for this episode, a random subset of all test classes.

Following standard terminology, the number of classes $N_C = |\mathcal{C}^{episode}|$ is referred to as the *way* of the episode and the number of support examples per class N_S as the *shot* of the episode.

Recently, simple methods that fit a linear classifier on a (pre-)trained backbone/encoder have been shown to be as powerful as other more involved methods (Chen et al., 2019; Dhillon et al., 2019). We use the term 'backbone' to refer to the neural network used for the classification of the training classes, excluding the top-most output layer. We denote a baseline classifier as:

$$p(y|\mathbf{x}) = \text{softmax}(f_b(f_\phi(\mathbf{x}))) \tag{1}$$

where $f_\phi : \mathbb{R}^D \to \mathbb{R}^Z$ denotes the encoder , and $f_b : \mathbb{R}^Z \to \mathbb{R}^{|\mathcal{C}^{train}|}$ the linear layer that outputs logits for the standard classification problem on the training classes. To use a pretrained encoder for FSC evaluation, a linear classifier, $f_\psi(\cdot; S)$, is fitted given the support set of each episode, where ψ refers to the newly fitted parameters for this episode. The FS classifier is:

$$p(y|\mathbf{x}; S) = \text{softmax}(f_\psi(f_\phi(\mathbf{x}); S)) \tag{2}$$

where $f_\psi : \mathbb{R}^Z \to \mathbb{R}^{|\mathcal{C}^{episode}|}$. This general structure of having fixed parameters ϕ and episodic parameters ψ can describe the test-time algorithm of many FSC methods, including Prototypical Networks (Snell et al., 2017) and Baseline++ (Chen et al., 2019).

2.2. Out-of-distribution Detection

Out-of-distribution detection is a binary detection problem. The confidence estimator is required to produce a score, $s(\mathbf{x}) \in \mathbb{R}$. We desire $s(\mathbf{x}^{\text{in}}) > s(\mathbf{x}^{\text{out}})$, i.e, the scores for in-distribution examples $\mathbf{x}^{\text{in}}$ to be higher than those for out-of-distribution examples $\mathbf{x}^{\text{out}}$. When the confidence estimator has learnable parameters, it is denoted as $s_\theta(\cdot)$.

OOD Metrics. The Area Under the Receiver-Operating Curve (AUROC) is a standard metric used for evaluating a binary detection problem as it circumvents the need to set a threshold for the score. A scoring function that can completely separate $s(\mathbf{x}^{\text{in}})$ from $s(\mathbf{x}^{\text{out}})$ would achieve an AUROC score of 100%. In the case of having 50% OOD inputs in an evaluation batch, the base-rate (i.e., random guesses) is 50%. Other standard metrics for the OOD problems include area under the precision-recall curve, and false positive rate (FPR) at a given true positive rate.

3. Related Work

Few-shot Learning. To make the FSC task more applicable to real-world problems, it has been extended in ways such as being evaluated on test classes more distinct from the training classes (Triantafillou et al., 2019; Chen et al., 2019), learning in the presence of an unlabelled set (Ren et al., 2018; Sun et al., 2019), and being asked to solve ambiguous tasks (Finn et al., 2018). The focus of our work, FS-OOD detection, can be considered as another extension where the system needs to act in the presence of irrelevant inputs.

OOD Detection. The FS-OOD detection problem can be considered as a variant of the standard OOD detection problem, which has been studied extensively. Standard OOD detection methods can roughly be categorized into one of three families. The first family of approaches is based on fitting a density model to the inputs, $p(x)$ (Nalisnick et al., 2018; 2019; Ren et al., 2019; Serrà et al., 2019). The second family of approaches is based on the *predictive probability* of a classifier, $p(y|x)$. Many Bayesian approaches have been proposed and tested (see (Snoek et al., 2019) and references therein). The last family involves fitting a density model to *representations* of an encoder. (Lee et al., 2018) proposed a simple yet effective approach of fitting Gaussians at various layers of a trained deep classifier, which we will refer to as the deep Mahalanobis distance (DM). In Section 6.1 we discuss the effectiveness of these baselines for the FS-OOD problem.

4. Few-shot OOD Detection & Adapted Baselines

4.1. The Task: Few-shot OOD Detection

FS-OOD vs standard OOD Detection. In the standard OOD detection problem, the in-distribution examples $\mathbf{x}^{\text{in}}$ are the set of all examples belonging to the "training classes" $\mathcal{C}^{train}$. Every other input is considered as OOD, $\mathbf{x}^{\text{out}}$. The FS-OOD detection problem differs in two ways: 1) the in-distribution examples are ones belonging to the support set (i.e., a subset of the "test classes" $\mathcal{C}^{test}$), and 2) due

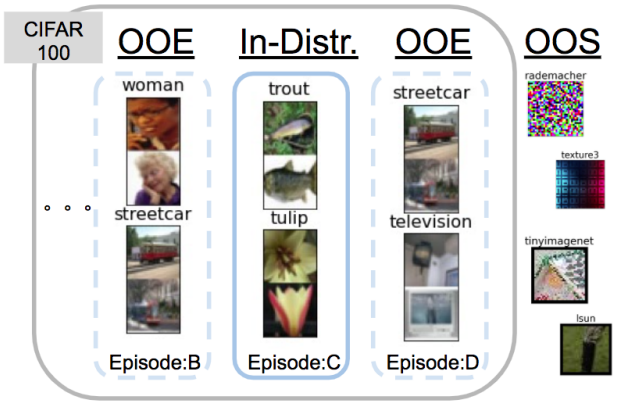

Figure 1: **Visual illustration of FS-OOD inputs.** Assume 'Episode:C' is the current episode. Examples not belonging to classes 'trout' or 'tulip' are considered OOE, such as examples in episodes B and D. OOS inputs are from other data sources.

to the first difference, the OOD set is every example not belonging to $\mathcal{C}^{episode}$. Consequently, what is considered in-distribution in one episode could be considered as OOD for another episode.

Below we describe the two distinct sources of OOD inputs (See Figure 1). For short, we refer to FS-OOD detection with OOE and OOS inputs as OOE and OOS detection respectively.

Out-of-Episode (OOE). OOE examples come from the same dataset, but from classes not in the current episode. In other words, if the current episode consists of classes in $\mathcal{C}^{episode}$, the OOE examples $\mathbf{x}^{\text{out}} \in R$ are from classes in $\mathcal{C}^{test} \setminus \mathcal{C}^{episode}$.

Out-of-Dataset (OOS). OOS examples come from a completely different dataset. For example, if the in-distribution set consists of CIFAR images, then the OOS examples can come from SVHN, ImageNet, etc. This is the type of OOD inputs typically considered in the standard OOD studies. Our results in Appendix H show that OOS and OOE detection present different challenges. Much of the progress made in standard OOD detection transfers to the OOS detection task. On the other hand, the OOE task presents challenges specific to FS and requires knowledge about the support set. Hence, it will be the focus of our experiments.

FS-OOD detection evaluation is done through some number of episodes. In each episode, the confidence estimator $s_\theta(\cdot)$ is evaluated on the set of in-distribution and OOD examples using standard metrics like AUROC and FPR. A complete description of the procedure can be found in Appendix F.

4.2. Adapted FS-OOD Detection Baselines

As discussed in Section 3, a variety of approaches have been studied in the standard OOD setting. Here we discuss two FS-adaptations of commonly used OOD baselines.

1. SPP: Softmax predictive Probability (Hendrycks & Gimpel, 2016)
2. DM: Deep Mahalanobis distance (Lee et al., 2018)

The few-shot adaptations are prefixed with "FS-". Please refer to Appendix C for description of the adaptation.

5. Out-of-Episode Classifier Network

We are interested in solving the FS-OOD detection task by learning a neural network, referred to as the Out-of-Episode Classifier (OEC) network. Given a support set S and a query $\mathbf{x}$, the network outputs whether $\mathbf{x}$ is in one of the classes in the support set, $s_\theta : (\mathbf{x}, S) \to \mathbb{R}$, where θ denotes the learnable parameters in OEC. OEC is trained episodically.

Learning Objective. Training is done over a sequence of training episodes. In essence, we form episodes with OOE inputs as outlined in Algorithm 1 in the Appendix with the training set instead of the test set. The set of in-distribution and out-of-distribution examples in an episode are denoted by $Q = \{\mathbf{x}_i^{\text{in}}\}_{i=1}^{N_Q}$ and $R = \{\mathbf{x}_i^{\text{out}}\}_{i=1}^{N_Q}$, respectively, where N_Q is the number of examples. Given an episode consisting of S, Q, R, the OEC is optimized with the binary cross-entropy objective.

Network Design. To avoid the input dimension from growing with the number of classes in S, the first design choice we make is to condition on the support of one class S_c instead of S , i.e. $s_\theta^c : (\mathbf{x}^{\text{in}}, S_c) \to \mathbb{R}$. Note that $s_\theta^c(\cdot)$ outputs a score for each class. To obtain a single score w.r.t. the full support set for a query, the class-conditional scores need to be aggregated. This is done by taking the maximum confidence over all the classes:

$$s_\theta(\mathbf{x}) = \max_{c \in C} s_\theta^c(\mathbf{x}, S_c) \tag{3}$$

Motivated by a baseline like DM, OEC capitalizes on a pretrained encoder, and takes as input embeddings from all layers $f_\phi(\cdot, l) \forall l$. For each layer, the OEC network takes in the concatenation $\mathbf{i}_l = [\boldsymbol{\mu}_{l,c}; f_\phi(\mathbf{x}, l)]$ where $\boldsymbol{\mu}_{l,c}$ is the averaged class embedding and $f_\phi(\mathbf{x}, l)$ is the query embedding at layer l. $[\cdot;\cdot]$ denotes concatenation.

We also include a residual path (with a learnable scale, and bias) whose input is $\|\boldsymbol{\mu}_{l,c} - f_\phi(\mathbf{x}, l)]\|^2$, and the output is added to the output of the OEC network. Hence, this residual path mimics the behavior of FS-DM, and is pre-trained for a few steps.

DataSet	SPP	DM	FS-SPP	FS-DM(-1)	FS-DM(iso)	FS-DM	OEC
				AUROC ↑			
CIFAR-FS	50.7±.1	48.9±.2	53.5±.1	66.3±.1	68.0±.1	67.8±.1	**72.2±.2**
miniImageNet	49.7±.0	50.3±.1	53.3±.0	67.6±.1	64.3±.1	65.8±.1	**71.0±.1**
				FPR90 ↓			
CIFAR-FS	90.3±.1	90.5±.1	87.9±.1	79.5±.2	77.7±.2	75.8±.2	**67.6±.3**
miniImageNet	90.8±.0	90.0±.1	87.4±.1	75.0±.1	80.2±.1	79.1±.1	**70.8±.1**

Table 1: **OOE detection results (AUROC, FPR90) at 5-way 5-shot.** 'FS-DM(-1), (iso)' denote FS-DM variants. '(-1)' uses only the last layer features, and '(iso)' uses isotropic covariance.

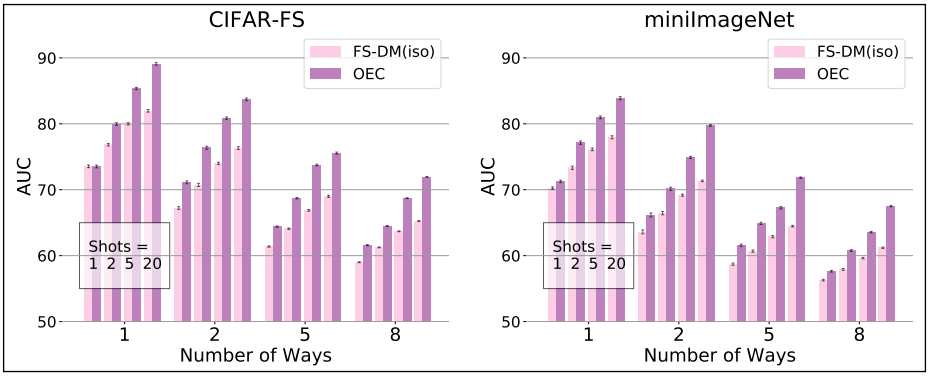

Figure 2: **OEC vs FS-DM at various FS configurations.**

6. Experiments

In this section, we evaluate OEC and our FS-adapted baselines on the OOE detection task. In Appendix J we show OEC's potential of being used with various FSC classifiers. In Appendix I we show that improvements in FS-OOD detection leads to improved accuracy in the FS-semi-supervised, and FS-active learning settings.

The architecture of encoder f_ϕ is the ResNet10 presented in (Chen et al., 2019). All results are evaluated with 15 of each of in and out of distribution test queries over 1000 test episodes if not otherwise specified. Descriptions of the in-distribution and OOS datasets are in Appendix E. OEC is early stopped by validating on 300 validation episodes on the validation classes. Error bars are 95% confidence interval. Code for reproducing these results can be found at `github.com/wangkua1/fs-ood`.

6.1. OOE Detection

Table 1 shows a comparison of OEC with various baselines on the FS-OOD task with OOE inputs for 5-way 5-shot episodes. Baselines from the standard OOD literature without adaptation do not work for this task, as indicated by the 50% AUC and 90% FPR90. Our adaptations FS-SPP, and FS-DM already improved over baselines. OEC outperformed the adapted baselines on both the CIFAR-FS and miniImageNet datasets. We were particularly careful about comparing OEC to non-episodic baselines as they were shown to be competitive in FSC (Dhillon et al., 2019). Hence we considered multiple variants of FS-DM, and OEC outperformed all. See Appendix D for a detailed analysis of all FS-DM hyperparameters.

Figure 2 includes a comparison of OEC and the strongest baseline FS-DM on the OOE detection task across a wide range of ways and shots. In general, as seen in the left subplot, smaller shots make the task more difficult, and so do larger ways. Across the board, OEC improves significantly over the baseline.

7. Conclusion

In this work, we studied the problem of few-shot out-of-distribution detection. Our proposed OEC significantly outperformed our adapted baselines across all considered ways and shots on both CIFAR-FS and miniImageNet on the FS-OOD detection task (Section 6.1). Additional results in Appendix I showed that improving FS-OOD detection can increase accuracy in few-shot semi-supervised, and active learning settings when OOD inputs are presented in the unlabeled set.

References

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C., and Huang, J.-B. A closer look at few-shot classification. In *International Conference on Learning Representations*, 2019.

Chung, J. S., Huh, J., Mun, S., Lee, M., Heo, H. S., Choe, S., Ham, C., Jung, S., Lee, B.-J., and Han, I. In defence of metric learning for speaker recognition. *arXiv preprint arXiv:2003.11982*, 2020.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *Conference on Computer Vision and Pattern Recognition*, pp. 3606–3613, 2014.

Dhillon, G. S., Chaudhari, P., Ravichandran, A., and Soatto, S. A baseline for few-shot image classification. *arXiv preprint arXiv:1909.02729*, 2019.

Finn, C., Xu, K., and Levine, S. Probabilistic model-agnostic meta-learning. *arXiv preprint arXiv:1806.02817*, 2018.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Hendrycks, D., Mazeika, M., and Dietterich, T. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyxCxhRcY7.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Lake, B., Salakhutdinov, R., Gross, J., and Tenenbaum, J. One shot learning of simple visual concepts. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 33, 2011.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 6402–6413, 2017.

Lane, T. and Brodley, C. E. Sequence matching and learning in anomaly detection for computer security. In *Conference on AI Approaches to Fraud Detection and Risk Management*, pp. 43–49, 1997.

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. *arXiv preprint arXiv:1706.02690*, 2017.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018.

Nalisnick, E., Matsukawa, A., Teh, Y. W., and Lakshminarayanan, B. Detecting out-of-distribution inputs to deep generative models using a test for typicality. *arXiv preprint arXiv:1906.02994*, 2019.

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.

Ravi, S. and Beatson, A. Amortized Bayesian meta-learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rkgpy3C5tX.

Ren, J., Liu, P. J., Fertig, E., Snoek, J., Poplin, R., DePristo, M. A., Dillon, J. V., and Lakshminarayanan, B. Likelihood ratios for out-of-distribution detection. *arXiv preprint arXiv:1906.02845*, 2019.

Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J. B., Larochelle, H., and Zemel, R. S. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

Serrà, J., Álvarez, D., Gómez, V., Slizovskaia, O., Núñez, J. F., and Luque, J. Input complexity and out-of-distribution detection with likelihood-based generative models. *arXiv preprint arXiv:1909.11480*, 2019.

Settles, B. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

Snoek, J., Ovadia, Y., Fertig, E., Lakshminarayanan, B., Nowozin, S., Sculley, D., Dillon, J., Ren, J., and Nado, Z. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pp. 13969–13980, 2019.

Sun, Q., Li, X., Liu, Y., Zheng, S., Chua, T.-S., and Schiele, B. Learning to self-train for semi-supervised few-shot classification. *arXiv preprint arXiv:1906.00562*, 2019.

Triantafillou, E., Zhu, T., Dumoulin, V., Lamblin, P., Xu, K., Goroshin, R., Gelada, C., Swersky, K., Manzagol, P.-A., and Larochelle, H. Meta-dataset: A dataset of datasets for learning to learn from few examples. *arXiv preprint arXiv:1903.03096*, 2019.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 2016.

Wang, K.-C., Wang, J., Truong, K., and Zemel, R. Customizable facial gesture recognition for improved assistive technology. 2019.

Yoon, J., Kim, T., Dia, O., Kim, S., Bengio, Y., and Ahn, S. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pp. 7332–7342, 2018.

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

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

Acknowledgement

Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute www.vectorinstitute.ai/#partners. K.C.W, E.T. and R.Z. are supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DoI/IBC) contract number D16PC00003. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/IBC, or the U.S. Government.

A. Notations

Symbol	Meaning
$\mathcal{C}^{test}, \mathcal{C}^{train}$	classes in the test, train set
$\mathcal{C}^{episode}$	classes in an episode
Q, S, R	query, support, distractors/out-of-distribution sets
S_c	subset of support that belongs to class c
N_E	number of episode
N_C, N_S	number of way/classes, number of shots per episode
N_Q	number of queries per episode
$\mathbf{x}$	generic image input
$\mathbf{x}^{\text{in}}, \mathbf{x}^{\text{out}}$	in-distribution query, and out-of-distribution examples
f_ϕ	encoder/backbone network
f_b	last layer linear classifier for standard classification
f_ψ	last layer linear classifier for FSC classification
$s(\cdot)$	confidence score

Table 2: Description of the notation used throughout this paper.

B. Alternative Objective for OEC

Instead of training the aggregated score, an alternative loss uses the following binary cross-entropy objective on the score before aggregation:

$$L_{\text{OEC}}(\theta; \{S, Q, R\}) = -\sum_{(c, \mathbf{x}^{\text{in}}) \in Q} \log \sigma(s^c_\theta(\mathbf{x}^{\text{in}}, S_c)) \\ -\sum_{\mathbf{x}^{\text{out}} \in R, c' \sim \text{unif}(\mathcal{C}^{episode})} \log(1 - \sigma(s^c_\theta(\mathbf{x}^{\text{out}}, S_{c'})))$$

For the OOE queries $\mathbf{x}^{\text{out}}$, we assigned them a label drawn from the uniform distribution of the in-distribution classes.

This alternative works reasonably well. The main reason the other version is presented in the main paper is to faciliate easier comparison with baselines and ablations.

C. Adapted Baselines

1. (FS-)SPP The SPP and FS-SPP scores are the largest values of the softmax from Equation 1, and Equation 2 respectively.

	$s(\mathbf{x}^{\text{in}})$ =
SPP	$\max_{c \in \mathcal{C}^{train}} \text{softmax}(f_b(f_\phi(\mathbf{x})))[c]$
FS-SPP	$\max_{c \in \mathcal{C}^{episode}} \text{softmax}(f_\psi(f_\phi(\mathbf{x}); S))[c]$

$f(\cdot)[c]$ denotes indexing into the c^{th} output of $f(\cdot)$. Notice that the $\max_c$ are over different sets of classes, and the linear classifiers are different.

2. (FS-)DM The DM method extracts features from each block of an encoder f_ϕ (e.g. ResNet18). The features are average pooled over the image dimensions. The resulting representation given a layer index l is denoted as $f_\phi(\mathbf{x}, l)$. A separate Gaussian distribution is fitted to every ResNet block for each class using all the examples in the training set. The covariance is shared across classes. This results in additional parameteres $\mu_{l,c}$ and Σ_l. The confidence is computed as:

$$s(\mathbf{x}) = \max_{c \in \mathcal{C}^{train}} \sum_l \log \mathcal{N}(f_\phi(\mathbf{x}, l); \mu_{l,c}, \Sigma_l) \tag{4}$$

See Appendix D for a more detailed comparison to the original implementation, and thorough ablation studies to justify our implementation. For FS-DM, the additional Gaussian parameters are fitted on the support set instead of the training set. Note that DM does not depend on the final linear classifier, making it agnostic to the last FS layer f_ψ in Equation 2.

The two settings (standard vs FS) have drastically different cardinality. For example, the training set of miniImageNet has 64 classes, and 600 examples per class. A 5-way 5-shot episode has 5 classes of 5 examples each. The dimensionality of $f_\phi(\mathbf{x}, l)$ is often much larger than 5. For ResNets, the last layer representation is 512-d in a single channel. Fitting a full precision/covariance with N=25 and d=512 is problematic as the covariance is singular. Surprisingly, in practice, using the pseudo-inverse, such precision matrices can be computed. We found the Shrunk Covariance estimator works well in practice [1].

D. DM ablation

2. DM Given embedding f_ϕ, the DM method extracts features from each block, and average pool over the image dimensions. The resulting representation given a layer index l is denoted as $f_\phi(\mathbf{x}, l)$. Given class supervision, one Gaussian distribution is fitted to each layer/block for each class. The covariance is shared across classes. This results in additional parameteres $\mu_{l,c}$ and Σ_l. In the original paper (Lee et al., 2018), the score is computed as

$$s_{l,\epsilon} = \max_c \log \mathcal{N}(f_\phi(\mathbf{x} + \epsilon g, l); \mu_{l,c}, \Sigma_l) \tag{5}$$

where g is the FGSM method and ϵ is the strength of the perturbation. The hyperparameters are validated for each of the OOD input types considered given 1000 target OOD inputs. For each type of OOD input, the final score is the one that had the best validation performance, i.e. $s = \max_{l,\epsilon} s_{l,\epsilon}$.

However, having to choose hyperparameters separataly given true OOD inputs is less desirable (Hendrycks et al., 2019). As shown in Figure 3, there does not exist one setting of l, ϵ that works best in general. However, for the OOE input problem, summing over all layers is better, hence this is the variant we use for the rest of our study.

E. Datasets

CIFAR-FS. CIFAR-FS uses the CIFAR100 dataset (Krizhevsky, 2009). It contains 32×32 color images. In total, it has 100 classes of 600 images each. CIFAR-FS refers to using 64 classes for training, 16 for validation, and 20 for test.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.covariance.ShrunkCovariance.html

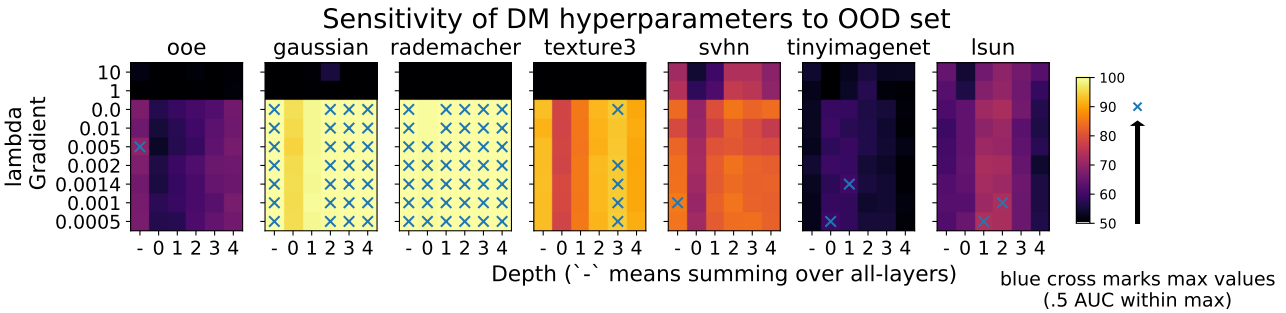

Figure 3: **Effect of DM hyperparameters.** There does not exist one setting of l, ϵ that works best (indicated by 'x') across different OOD inputs.

***mini*ImageNet.** The *mini*ImageNet dataset is another commonly used few-shot benchmark (Snell et al., 2017; Vinyals et al., 2016). It consists of 84×84 colored images. It also has 100 classes, and 600 examples each. Similarly, we used 64 classes for training, 16 for validation, and 20 for test.

Out-of-Dataset. The OOS datasets were adopted from previous studies including those by (Hendrycks et al., 2019; Liang et al., 2017). Since we experimented with in-distribution datasets of different image dimensions, the OOS inputs were scaled accordingly. All inputs are first scaled to the valid pixel range, then fed through the same data normalization scheme as the in-distribution inputs.

Synthetic OOS. We used uniform, Gaussian, and Rademacher noise.

Natural OOS.

- **LSUN** is a large-scale scene understanding dataset (Yu et al., 2015).
- **Texture** is a dataset with different real world patterns (Cimpoi et al., 2014).
- **Places** is another large scale scene understanding dataset (Zhou et al., 2017).
- **SVHN** refers to the Google Street View House Numbers dataset (Netzer et al., 2011).
- **TinyImagenet** consists of 64×64 color images from 200 ImageNet classes, with 600 examples of each class (Hendrycks et al., 2019).

F. FS-OOD evaluation details

Algorithm 1:
Episodic Evaluation for OOD Detection.

Require: config $= \{N_E, N_C, N_S, N_Q\}$
Require: s_θ a confidence score
Require: $\mathcal{D}_{\text{in}}$ In-distribution Dataset
Require: $\mathcal{D}_{\text{out}}$ OOD Source

M $\leftarrow \{\}$
for N_E **do**
  $S, Q \leftarrow$ GETEPISODE($\mathcal{D}_{\text{in}}$, config)
  $R \leftarrow$ GETOOD($S, \mathcal{D}_{\text{out}}$, config)
  $\text{S}_{\text{in}} \leftarrow \{\}$, $\text{S}_{\text{out}} \leftarrow \{\}$
  for $\mathbf{x}^{\text{in}}$ in Q **do**
    S_{in}.insert($s_\theta(\mathbf{x}^{\text{in}}; \phi, S)$)
  end for
  for $\mathbf{x}^{\text{out}}$ in R **do**
    S_{out}.insert($s_\theta(\mathbf{x}^{\text{out}}; \phi, S)$)
  end for
  M.insert(Metric(S_{in}, S_{out}))
end for
return Avg(M)

GETEPISODE() denotes the procedure of forming an episode based on the number of ways, shots, and queries. Based on the current episode and a given OOD source, we sample the OOD set R for the given episode. Then, confidence scores are computed for each of the in-distribution and OOD examples. These scores are summarized using standard metrics like AUROC, denoted by Metric$(\cdot, \cdot)$. Then the metric values are aggregated over N_E episodes.

G. Training details and Hyperparameters

G.1. FSC classifier/encoder

For all encoders, the network architecture is ResNet10 from (Chen et al., 2019). Input is preprocessed with data augmentation of random cropping and horizontal flips, and normalized using dataset statistics. All hyperparameters described below are validated to obtain the best FSC accuracy on 100 episodes consisting of the validation classes on varying learning rates.

Baseline++. The best run used SGD with Nestrov .9 momuntum, batch size 128, learning rate of 0.1. Learning rate is decreased by a factor of .2 at epochs 60, 120, 160.

ProtoNet. ProtoNet is trained episodically with 75 queries per episode (ways/shots is the same the test episode of a given experiment). The best run used Adam and a learning rate of 1e-3. The model is early-stopped using validation FSC accuracy, which is usually around 40000 episodes.

MAML. MAML is trained episodically with 75 queries per episode (ways/shots is the same the test episode of a given experiment). The best run used Adam and a learning rate of 1e-4. The model is early-stopped using validation FSC accuracy, which is usually between 50000 to 100000 episodes.

G.2. OEC

OEC is trained using Adam at a learning rate of 0.001. It is validated on AUROC, which resulted in a larger variance than FSC accuracy. Instead of 100 validation episodes, OEC is validated using 300 episodes to reduce variance of the best checkpoint.

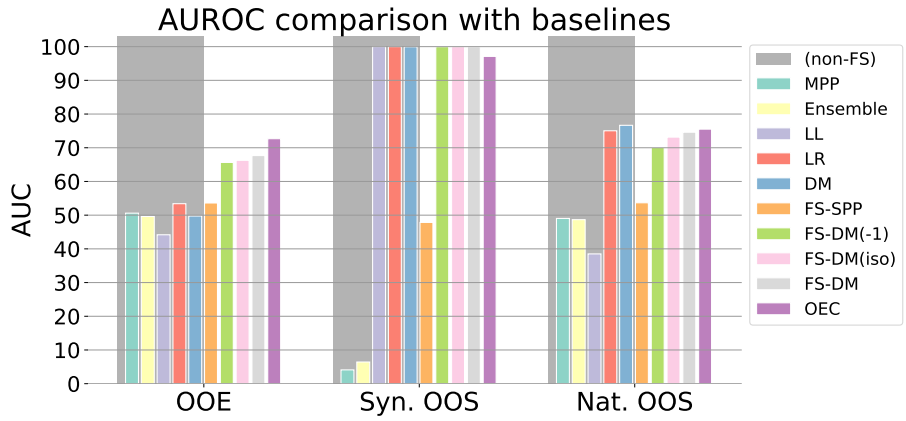

Figure 4: **(FS-)OOD methods on both OOE and OOS inputs**
(in-distribution: CIFAR-FS) 'OOE' numbers correspond to Table 1.

H. OOE vs OOS inputs

Here we discuss the different challenges presented by OOE and OOS inputs (see Figure 4). The methods we consider now include other non-FS baselines such as Ensemble (Lakshminarayanan et al., 2017), log-likelihood of Glow (LL) (Nalisnick et al., 2018), likelihood-ratio of Glow (LR) (Serrà et al., 2019). The types of OOD inputs considered are: OOE, synthetic OOS ("Syn. OOS"), and natural OOS ("Nat. OOS") (details in Appendix E). Again, non-FS baselines are inherently ineffective for the OOE task. The non-FS baselines, specifically LR and DM, perform well on the OOS inputs. There are two potential explanations: 1) compared to the OOS inputs, the support set is much closer to the training classes with high probability, and 2) the models are somewhat invariant to the shift from the training classes to test classes. The fact that DM does better than FS-DM on Nat. OOS is due to having a lot more examples to fit the extra MoG parameters. In general, OOD inputs can be of any type, hence OEC and the FS adapted methods are more broadly applicable.

I. FS-SSL and Active Learning

In this work, we are particularly interested in the realistic scenario where aside from the small support set for each new task, we are also given an *unlabeled set* $U = \{\mathbf{u}\}_i^{N_u}$. Much like the support set, these examples can be used to learn a classifier for the current classification task. However, an example of U differs from a support example in that: 1) it is unlabeled, and 2) it might be a *distractor*, i.e. might not belong to any of the classes of interest in the current episode. In our case, the OOE inputs are distractors.

Intuitively, there are two additional challenges in this setup: 1) filtering out the distractors, and 2) figuring out how to leverage the in-distribution (non-distractor) unlabeled examples (i.e., guessing a correct pseudo-label in the SSL case, and ranking which examples are more useful in the active case) for improving the classifier. We aim to improve upon the former: we show that the presence of distractors can lead to significantly worse performance if they are not accurately identified and discarded. Our focus is on methods that can effectively filter these out.

Few-shot Semi-Supervised Learning (FS-SSL). In FS-SSL, the model can use the unlabeled set $U = \{\mathbf{u}\}_i^{N_u}$ for improving its classification accuracy. (Ren et al., 2018) extended ProtoNets by predicting a soft label for each example in U, adding it to the support set, and computing 'refined' semi-supervised prototypes, $\tilde{\boldsymbol{\mu}}_c$, as:

$$\tilde{\boldsymbol{\mu}}_c = \frac{\sum_{\mathbf{x}_i \in S_c} f_\phi(\mathbf{x}_i) + \sum_{\mathbf{u}_i \in U} \tilde{p}_{i,c} f_\phi(\mathbf{u}_i)}{|S_c| + \sum_{\mathbf{u}_i \in U} \tilde{p}_{i,c}} \tag{6}$$

where $\tilde{p}_{i,c}$ is the predicted probability that $\mathbf{u}_i$ belongs to class c based on the labeled support set. Intuitively, unfiltered distractors in U can significantly affect the refined prototype, and hence degrade the performance of the classifier.

Few-shot Active Learning. Active learning (Settles, 2009) is a well-studied problem where the goal is to use as few labeled data as possible by requesting labels for only that subset of the overall available data that is deemed to be more

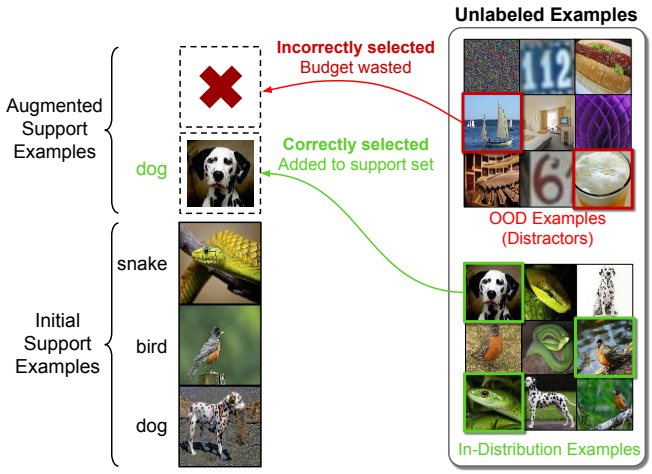

Figure 5: Few-shot active learning setup. A test episode consists of an *initial support set* and a set of unlabeled data. The unlabeled data contains both in-distribution and OOD data; here, the OOD data contains both out-of-episode (OOE) examples and out-of-dataset (OOS) examples. The model is given a budget that it can spend to obtain labels for a subset of unlabeled examples. In-distribution examples selected by the model are added to the *augmented support set* with their ground-truth labels. OOD examples selected by the model are not added to the support set; they simply waste the model's budget.

useful for learning.

Here, we consider an episodic variant of this formulation, where each episode contains a support set and a query set as usual, but with the possibility of augmenting the support set by requesting labels for some examples from an unlabeled set U. A common constraint in active learning is the budget. The model can only request a certain number of labels for examples in U. Upon requesting the label of an in-distribution example, that example is added to the support set with its ground-truth label. If a distractor label is requested, the oracle would give back the label "distractor". To simplify our analysis, these examples are discarded completely. This is a sensible thing to do, since it is not obvious how to extract useful information with this label. The penalty of requesting a label for a distractor decreases the budget by one. A model that is capable of OOD detection would request the labels of the relevant examples only, and would thus successfully increase the support set. In the worst case, the labeling budget is used up entirely on distractors, resulting to no additions to the support set (see Figure 5). In this study, the budgest is set to number of in-distribution unlabeled examples. In the best case, all of the in-distribution unlabeled examples will be included, and none of the distractors will be included.

		SSL				Active		
I:O	L (ref)	no OOD	FS-SPP	FS-DM	OEC	FS-SPP	FS-DM	OEC
1:5		-2.5	-2.2	-1.8	**0.0**	0.1	0.4	**0.6**
5:5	67.5	-0.2	-0.2	-0.2	**0.7**	**4.8**	3.8	4.1
5:50		-2.1	-1.9	-1.2	**-0.5**	-0.1	1.1	**1.6**
1:5		-2.8	-2.0	-1.5	**-0.4**	0.1	**0.3**	**0.3**
5:5	63.5	-1.0	-0.4	-0.8	**-0.3**	**3.9**	3.0	3.1
5:50		-3.6	-2.7	-1.9	**-1.1**	0.0	0.8	**0.9**

Table 3: **Effect of FS-OOD on FS-SSL/active learning.** 'I:O' is the number of the in-distribution unlabeled to the number of distractors in the unlabeled pool. Column 'L' is FSC accuracy using only the labeled set. Numbers in other columns are relative changes to 'L' column.

Results. The benefit of having a FS-OOD detector is more apparent when the ratio of distractors to in-distribution examples in the unlabeled pool increases from 1 to 5. In the small unlabeled set (1:5) case, good FS-OOD detection prevents

degradation of accuracy in the FS-SSL task. In the large unlabeled set (5:50) case, better FS-OOD detection leads to larger improvement in FSC accuracy in the FS-active learning task. Interestingly, in the (5:5) case, one of our adapted baseline, FS-SPP, is able to improve the accuracy in the FS-active task more than the strong FS-OOD methods. An interesting future direction will be to understand not only which examples should be filtered out, but also which ones should be included with higher priority.

J. Combining OEC with various FSC learners

	AUROC↑			FPR90↓		
	FS-SPP	**FS-DM**	**OEC**	**FS-SPP**	**FS-DM**	**OEC**
			CIFAR-FS			
Baseline++	48.8	69.0	**72.3**	92.9	75.2	**67.6**
ProtoNet	61.2	61.7	**70.5**	83.2	83.3	**69.2**
MAML	61.9	69.1	**69.9**	83.5	70.3	**67.2**
			miniImageNet			
Baseline++	49.5	64.2	**69.1**	91.6	80.2	**73.1**
ProtoNet	58.6	59.6	**65.7**	85.1	83.6	**77.4**
MAML	57.3	64.6	**67.5**	86.9	77.6	**73.7**

Table 4: **OOE detection when OEC is combined with different FS classifiers.**

Baseline++ learns the backbone using standard classification training, so this is the same method presented in Section 6.1. Both ProtoNet and MAML are learned episodically as described in Appendix G. The resulting backbone of each of the three methods is different.

Table 4 shows that for all backbones, OEC significantly outperformed other FS-OOD methods. This demonstrate the flexibility of OEC.