Few-Shot One-Class Classification via Meta-Learning

Anonymous authors
Paper under double-blind review

Abstract

Although few-shot learning (Wang & Yao, 2019) and one-class classification (Khan & Madden, 2014) have been separately well studied, their intersection remains rather unexplored. Our work addresses the few-shot one-class classification problem and presents a meta-learning approach that requires only few data examples from only one class to adapt to unseen tasks. The proposed method builds upon the model-agnostic meta-learning (MAML) algorithm (Finn et al., 2017) and explicitly trains for few-shot class-imbalance learning, aiming to learn a model initialization that is particularly suited for learning one-class classification tasks after observing only a few examples of one class. Experimental results on datasets from the image domain and the time-series domain show that our method substantially outperforms the baselines, including MAML, and demonstrate the ability to learn new tasks from only few majority class samples. Moreover, we successfully learn anomaly detectors for a real world application involving sensor readings recorded during industrial manufacturing of workpieces with a CNC milling machine using only few examples from the normal class.

1 Introduction

The anomaly detection (AD) task (Chandola et al., 2009; Aggarwal, 2015) consists in differentiating between normal and abnormal data samples. AD applications are common in various domains that involve different data types, including medical diagnosis (Prastawa et al., 2004), cybersecurity (Garcia-Teodoro et al., 2009) and quality control in industrial manufacturing (Scime & Beuth, 2018). Due to the rarity of anomalies, the data underlying AD problems exhibits high class-imbalance. Therefore, AD problems are usually formulated as one-class classification (OCC) problems (Moya et al., 1993), where either only few or no anomalous data samples are available for training the model (Khan & Madden, 2014). While most of the developed approaches (Khan & Madden, 2014) require a substantial amount of normal data to yield good generalization, in many real-world applications, e.g. in industrial manufacturing, only small datasets are available. Data scarcity can have many reasons: data collection itself might be expensive, e.g. in healthcare, or happen only gradually, such as in a cold-start situation. To enable learning from few examples, various viable meta-learning approaches (Lake et al., 2011; Ravi & Larochelle, 2016; Finn et al., 2017) have been developed. However, they rely on having examples from each of the classification task's classes, which prevents their application to OCC tasks. To the best of our knowledge, the few-shot one-class classification (FS-OCC) problem has only been addressed by Kozerawski & Turk (2018) in the image data domain (see Section 3).

In this work, we introduce One-class model-agnostic meta-learning (OC-MAML), an algorithm that quickly learns FS-OCC tasks. It builds upon Model-agnostic meta-learning (MAML) (Finn et al., 2017), which is a meta-learning method that explicitly optimizes for few-shot learning and yields a model initialization that enables quick adaptation to a new task using only few of its datapoints. Like MAML, OC-MAML yields model parameters that are easily adaptable to unseen tasks. The difference is that the model initialization delivered by OC-MAML is particularly suited for adaptation to OCC tasks and hence only requires examples from one class of the target task for good adaptation. To find such model parameters, we modify the class-imbalance rate of the *inner loop* data to match the one of the target task, as we detail in Section 2.3. While recent meta-learning approaches focused on the few-shot learning problem, i.e. learning to learn with few examples, we extend their use to the OCC problem, i.e. learning to learn with examples from only one class.

We find that OC-MAML yields significantly higher generalization performance and faster convergence than the baselines in learning FS-OCC tasks, including MAML (Finn et al., 2017). We evaluate our approach on four datasets from the image and the time-series domains, and demonstrate its robustness and maturity for real-world application by successfully validating it on a dataset of sensor readings recorded during manufacturing of metal workpieces with a CNC milling machine.

2 Approach

2.1 Problem statement

Our goal is to learn a one-class classification (OCC) task using only few examples from its majority class. For this, we formulate a meta-learning problem. We assume access to data from classification tasks T_i^{train} sampled from a task distribution $p(T)$ related to the one of our target OCC task(s). In the few-shot classification context, N-way K-shot learning tasks are usually used to test the learning procedure yielded by the meta-learning algorithm. An N-way K-shot classification task includes K examples from each of the N classes available for learning this task, after which a disjoint set of data is used to test the trained classifier (Vinyals et al., 2016). When the test task is an OCC task, only examples from one class are available for learning, which can be viewed as a 1-way K-shot classification task. In order to align with the anomaly detection problem, the available examples have to belong to the normal (majority) class, which usually has a lower variance than the anomalous (minority) class.

As a result, the meta-learned model needs to acquire a sufficiently general prior that is able to approximate a *generalized* decision boundary for a particular class, in our case the normal class, given only few of its examples. Learning such a class decision boundary in the few-shot one-class regime can be especially challenging. On the one hand, if the model overfits to the few available datapoints, the class decision boundary would be too tight, and some normal samples would be predicted as anomalies. On the other hand, if the model overfits to the majority class, e.g. predicting almost everything as normal, the class decision boundary would be too big, and out-of-distribution (anomalous) samples would not be detected. This problem formulation is our prototype for a practical use case in which an application-specific anomaly detector is needed and only few data samples from the normal class are available.

2.2 Model Agnostic Meta-Learning

Model-agnostic meta-learning (MAML) (Finn et al., 2017) is an optimization-based meta-learning algorithm upon which we build in our present work. MAML learns a model initialization that enables quick adaptation to unseen tasks using only few data samples. For that, MAML trains a model explicitly for few-shot learning on tasks T_i coming from the same task distribution $p(T)$ as the unseen target task T_{test}.

To assess the model's adaptation ability to *unseen* tasks, the available tasks are divided into mutually disjoint task sets: one for meta-training S^{tr}, one for meta-validation S^{val} and one for meta-testing S^{test}. Each task T_i is divided into two disjoint sets of data, each of which is used for a particular MAML operation: D^{tr} is used for adaptation and D^{val} is used for validation, i.e. evaluating the adaptation. The adaptation procedure of a model f_θ to a particular task T_i consists in taking one (or more) gradient descent step(s) using *few* datapoints sampled from D^{tr}, and can be expressed as:

$$\theta_i^{'} = \theta - \alpha \nabla_\theta L_{T_i}^{tr}(f_\theta), \tag{1}$$

where α denotes the learning rate. Usually, multiple gradient descent updates are applied for adaptation. We also refer to the adaptation updates as *inner loop updates*.

A good measure for the suitability of the initialization parameters θ for few-shot adaptation to a considered task T_i is the loss $L_{T_i}^{val}(f_{\theta_i^{'}})$, which is computed on the validation set D^{val} using the task-specific adapted model $f_{\theta_i^{'}}$. In order to optimize for few-shot learning, the model parameters θ are updated by minimizing the aforementioned loss across all meta-training tasks. This update, called the *outer loop update*, can be expressed as:

$$\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{T_i \sim p(T)} L_{T_i}^{val}(f_{\theta_i^{'}}), \tag{2}$$

where β is the learning rate used for the outer loop.

To avoid meta-overfitting, i.e. overfitting to the meta-training tasks, model selection can be done via conducting validation episodes using tasks from S^{val} throughout meta-training. At meta-test time, the few-shot adaptation to unseen tasks from S^{test} is evaluated. We note that, in the case of few-shot classification, K datapoints from *each* class are sampled from D^{tr} for the adaptation, during meta-training, meta-validation and meta-testing.

2.3 One-Class Model Agnostic Meta-Learning

The primary contribution of our work is to show that gradient-based meta-learning is a viable approach to the underexplored few-shot one-class classification (FS-OCC) problem. We achieve this by adequately modifying the objective of the adaptation step, i.e. the inner loop updates, of the MAML algorithm. We choose to build upon gradient-based meta-learning algorithms, because these were shown to be universal learning algorithm approximators (Finn & Levine, 2017), which means that they could approximate a learning algorithm tailored for FS-OCC. Moreover, MAML outperforms RNN-based meta-learning algorithms when faced with out-of-domain tasks (Finn & Levine, 2017). This characteristic is important for applications on industrial sensor data due to the high domain diversity induced by the high diversity of the sensors, machines and manufacturing processes.

As explained in Section 2.2, MAML optimizes explicitly for few-shot adaptation by creating and using auxiliary tasks that have the same characteristic as the target tasks, in this case tasks that include only few datapoints for training. Analogously, OC-MAML trains explicitly for quick adaptation to OCC tasks by creating OCC auxiliary tasks for meta-training. Concretely, this is done by modifying the class-imbalance rate (CIR) of the inner loop data batches to match the one of the test task. The meta-training procedure of OC-MAML is described by the pseudo-code in Algorithm 1.

Algorithm 1 Few-shot one-class classification with OC-MAML

```
Require: $S^{tr}$: Set of meta-training tasks
Require: $\alpha, \beta$: Learning rates
Require: $K, Q$: Batch size for the inner and outer updates
Require: $c$: CIR for the inner-updates
Randomly initialize $\theta$
while not done do
    Sample batch of tasks $T_i$ from $S^{tr}$ Let $\{D^{tr}, D^{val}\} = T_i$
    for all sampled $T_i$ do
        Sample $K$ datapoints $B = \{\mathbf{x}^{(l)}, \mathbf{y}^{(l)}\}$ from $D^{tr}$ such that CIR= $c$
        Initialize $\theta_i^{'} = \theta$
        for number of adaptation steps do
            Compute adaptation loss $L_{T_i}^{tr}(f_{\theta_i^{'}})$ using $B$
            Compute adapted parameters with gradient descent: $\theta_i^{'} = \theta_i^{'} - \alpha \nabla_{\theta_i^{'}} L_{T_i}^{tr}(f_{\theta_i^{'}})$
        end for
        Sample $Q$ datapoints $B^{'} = \{\mathbf{x}^{'(l)}, \mathbf{y}^{'(l)}\}$ from $D^{val}$
        Compute outer loop loss $L_{T_i}^{val}(f_{\theta_i^{'}})$ using $B^{'}$
    end for
    Update $\theta$: $\theta \leftarrow \theta - \beta \nabla_{\theta} \sum_{T_i} L_{T_i}^{val}(f_{\theta_i^{'}})$
end while
return meta-learned parameters $\theta$
```

As described in Section 1, OCC problems are binary classification scenarios where only few or no minority class samples are available. In order to address both of theses cases, we introduce the hyperparameter c which sets the CIR of the batch sampled for the inner updates. Hereby, c gives the percentage of the samples belonging to the minority (anomalous) class w.r.t. the total number of samples, e.g. setting $c = 0\%$ means only majority class samples are contained in the data batch.

The key difference between MAML and OC-MAML is in the sampling operation of the inner loop batch (operation 5 in algorithm 1). By reducing the size of the batch used for the adaptation (via the hyperparameter K), MAML trains for few-shot adaptation. OC-MAML extends this approach to train for few-shot one-class adaptation by reducing the CIR of the batch used for adaptation (via the hyperparameter c). In order to evaluate the performance of the adapted model on both classes, we use a class-balanced validation batch $B^{'}$ for the outer loop updates. This way, we maximize the performance of the model in recognizing both classes after having *seen* examples from only one class during adaptation. Using OCC tasks for adaptation during meta-training favors model initializations that enable a quick adaptation to OCC tasks over those that require class-balanced tasks. From a representation learning standpoint, OC-MAML learns representations that are not only broadly suitable for the data underlying $p(T)$, but also particularly suited for OCC tasks.

During meta-training, meta-validation episodes are conducted to perform model selection. In order to mimic the adaptation to unseen FS-OCC tasks with CIR $c = c_{target}$ at test time, the CIR of the batches used for adaptation during meta-validation episodes is also set to $c = c_{target}$. We note that the hyperparameter K denotes the total number of datapoints, i.e. batch size, used to perform the adaptation updates, and not the number of datapoints *per class* as done by Finn et al. (2017). Hence, a task with size $K = 10$ and CIR $c = 50\%$ is equivalent to a 2-way 5-shot classification task.

3 Related works

The method we propose aims to address the few-shot one-class classification (FS-OCC) problem, i.e. solving binary classification problems using only *few* datapoints from only *one* class. To the best of our knowledge, this problem was only addressed by Kozerawski & Turk (2018), and exclusively in the image data domain. Kozerawski & Turk (2018) train a feed-forward neural network (FFNN) to learn a transformation from feature vectors, extracted by a CNN pre-trained on ILSVRC 2014 (Russakovsky et al., 2015), to SVM decision boundaries. Hereby, the FFNN is trained on ILSVRC 2012. At test time, a SVM boundary is inferred by using one image of one class from the test task, and then used to classify the query examples. This approach is specific to the image domain since it relies on the availability of very large, well annotated datasets and uses data augmentation techniques specific to the image domain, e.g. mirroring. OC-MAML offers a more general approach to FS-OCC since it is data-domain-agnostic. In fact, it does not require a pre-trained feature extraction model, which might not be available for some data domains, e.g. sensor readings. Instead, we learn a parameter initialization of a single model such that gradient-based finetuning using only few datapoints from unseen OCC tasks leads to good generalization. We successfully validate our approach on datasets from the image and the time-series domains. In the following, we present works related to few-shot classification and one-class classification separately, since the intersection of these two fields remains rather unexplored.

3.1 Few-shot classification

The most recent approaches for few-shot classification may be broadly categorized in optimization-based methods (Ravi & Larochelle, 2016; Finn et al., 2017; Li et al., 2017; Nichol & Schulman, 2018) and metric-based methods (Koch, 2015; Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018). The optimization-based approaches aim to learn an optimization algorithm (Ravi & Larochelle, 2016) and/or a parameter initialization (Finn et al., 2017; Li et al., 2017; Nichol & Schulman, 2018), that is tailored for few-shot learning. Metric-based techniques learn a metric space where samples belonging to the same class are close together, which facilitates few-shot classification (Koch, 2015; Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018). Rusu et al. (2018) develops a hybrid method that combines the advantages of both categories.

Prior meta-learning approaches to few-shot classification addressed the N-way K-shot classification problem described in Section 2.1. Consequently, they only consider neatly class-balanced test classification tasks. Optimization-based techniques require these samples for finetuning during the adaptation phase (Ravi & Larochelle, 2016; Finn et al., 2017; Li et al., 2017; Nichol & Schulman, 2018). As for the metric-based methods, these samples are necessary to compute the class prototypes (Snell et al., 2017), the embeddings needed for verification (Koch, 2015) or the relation scores (Sung et al., 2018). Our approach, however, requires only samples from one of the test task's classes for learning. Moreover, while the evaluation of the previous approaches in the classification context was limited to the image domain, we validate OC-MAML on datasets from both, the image and time-series domains. Some metric-based meta-learning methods can perform zero-shot classification, i.e. they do not require any samples from the test task to learn it (Snell et al., 2017; Sung et al., 2018). To do so, however, they rely on having meta-data from *each* of the test task's classes, which is available only for few datasets and requires domain knowledge.

3.2 One-class classification

Classical OCC approaches rely on SVMs (Schölkopf et al., 2001; Tax & Duin, 2004) to distinguish between normal and abnormal samples. However, a series of classification analyses conducted by Pal & Foody (2010) shows that the classification accuracy of SVMs decreases with an increasing number of input features, particularly when small datasets are available for training. Hybrid approaches combining SVM-based techniques with feature extractors were developed to compress the input samples in lower dimensional representations (Xu et al., 2015; Erfani et al., 2016; Andrews et al., 2016). Fully deep methods that jointly perform the feature extraction step and the OCC step have also been developed (Ruff et al., 2018). Another category of approaches to OCC use the reconstruction error of antoencoders (Hinton & Salakhutdinov, 2006) trained with only normal class examples as an anomaly score (Hawkins et al., 2002; An & Cho, 2015; Chen et al., 2017). Yet, determining a decision threshold for such an anomaly score requires labeled data from both classes. Further more recent techniques rely on GANs (Goodfellow et al., 2014) to perform OCC (Schlegl et al., 2017; Ravanbakhsh et al., 2017; Sabokrou et al., 2018).

The aforementioned hybrid and fully deep approaches require a considerable amount of data from the target OCC task to train the typically highly parametrized models and, by these means, learn features specific to the normal class. By leveraging auxiliary OCC tasks and explicitly optimizing for few-shot learning, OC-MAML learns a representation that can be easily adapted to the target OCC task, requiring only few of its datapoints.

4 Experimental evaluation

We evaluate our approach on four different datasets. The conducted experiments [1] aim to address the following key questions: (a) How does OC-MAML perform for different class-imbalance rates (CIRs) and adaptation set sizes (K), and particularly how good is the adaptation to OCC tasks in the few-shot regime? (b) Does using OCC tasks for meta-training improve adaptation to such tasks, as it is the case for few-shot tasks (Finn et al., 2017)? (c) How does OC-MAML perform in few-shot one-class classification (FS-OCC) problems from the time-series data domain, and how mature is it for industrial real-world applications?

4.1 Baselines

MAML: In order to address the second question and investigate the benefit of our modification to the MAML's adaptation objective, we compare to MAML (Finn et al., 2017) which optimizes for class-balanced adaptation, independently of the CIR of the target task.

Multi-task learning (MTL): This baseline uses the tasks available for meta-training S^{tr} to train a feature extractor in a multi-task setting and then learns a classifier for the target task on top of it, via transfer learning.

Single task learning (STL): This baseline is trained from scratch using only the target task's data.

In all baseline experiments, we balance the classes of the target task(s) by randomly oversampling the minority class. We use the same convolutional neural network architecture for OC-MAML, MAML and STL. For MTL, an additional fully connected layer is used for each task. Experimental details including hyperparameters and model architectures are provided in Appendix A.

[1] Our implementation of OC-MAML and the experimental evaluation will be made public upon paper acceptance.

4.2 Datasets

In this Section we provide information about the datasets used and the task creation procedures.

Multi-task MNIST (MT-MNIST): We derive 10 binary classification tasks from the MNIST dataset (LeCun et al., 2010), where every task consists in recognizing one of the digits. This is a classical one-class classification benchmark dataset. For a particular task T_i, images of the digit i are labeled as normal samples, while out-of-distribution samples, i.e. the other digits, are labeled as anomalous samples. We use 8 tasks for meta-training, 1 for meta-validation and 1 for meta-testing. Hereby, images of digits to be recognized in the validation and test tasks are not used as anomalies in the meta-training tasks. This ensures that the model is not exposed to normal samples from the test task during meta-training. Moreover, the sets of anomalous samples of the meta-training, meta-validation and meta-testing tasks are mutually disjoint. We conduct experiments on 9 MT-MNIST datasets, each of which involves a different target task ($T_0 - T_8$). The task T_9 is used as a meta-validation task across all experiments.

MiniImageNet: This dataset was proposed by Ravi & Larochelle (2016) and includes 64 classes for training, 16 for validation and 20 for testing, and is a classical challenging benchmark dataset for few-shot learning. To adapt it to the few-shot *one-class* classification setting, we create 64 binary classification tasks for meta-training, each of which consists in differentiating one of the training classes from the others, i.e. the anomalous examples of a task T_i are randomly sampled from the 63 classes with labels different from i. We do the same to create 16 meta-validation and 20 meta-testing tasks using the corresponding classes.

Synthetic time-series (STS): In order to investigate the applicability of OC-MAML to time-series (question (c)), we created two datasets, each including 30 synthetically generated time-series that underlie 30 different anomaly detection tasks. The time-series underlying the datasets are sawtooth waveforms (STS-Sawtooth) and sine functions (STS-Sine). Each time-series is generated with random frequencies, amplitudes, noise boundaries, as well as anomaly width and height boundaries. Additionally, the width of the rising ramp as a proportion of the total cycle is sampled randomly for the sawtooth dataset, which results in tasks having rising and falling ramps with different steepness values. The data samples of a particular task are generated by randomly cropping windows of length 128 from the corresponding time-series. We generate 200 normal and 200 anomalous data examples for each task. Details about the data generation procedure as well as figures of exemplary signals are presented in Appendix B.1. For each dataset, we randomly choose 20 tasks for meta-training, 5 for meta-validation and 5 for meta-testing. We propose the STS-datasets as benchmark datasets for the few-shot one-class classification problem in the time-series domain, and will make them public upon paper acceptance.

CNC Milling Machine Data (CNC-MMD): This dataset consists of ca. 100 aluminum workpieces on which various consecutive roughing and finishing operations (pockets, edges, holes, surface finish) are performed. The sensor readings which were recorded at a rate of 500Hz measure various quantities that are important for the process monitoring including the torques of the various axes. Each run of machining a single workpiece can be seen as a multivariate time-series. We segmented the data of each run in the various operations performed on the workpieces. E.g. one segment would describe the milling of a pocket where another describes a surface finish operation on the workpiece. Since most manufacturing processes are highly efficient, anomalies are quite rare but can be very costly if undetected. For this reason, anomalies were provoked for 6 operations during manufacturing to provide a better basis for the analysis. The data was labeled by domain experts from Siemens Digital Industries. It should be noted that this dataset more realistically reflects the data situation in many real application scenarios from industry where anomalies are rare and data is scarce and for this reason training models on huge class-balanced datasets is not an option.

For our experiments, we created 30 tasks per operation by randomly cropping windows of length 2048 from the corresponding time-series of each operation. As a result, the data samples of a particular task T_i cropped from a milling operation O_j correspond to the same trajectory part of O_j, but to different workpieces. The task creation procedure ensures that at least two anomalous data samples are available for each task. The resulting tasks include between 15 and 55 normal samples, and between 2 and 4 (9 and 22) anomalous samples for finishing (roughing) operations. We validate our approach on all 6 milling operations in the case where only 10 samples belonging to the normal class ($K = 10$, $c = 0\%$) are available. Given the type of the target milling operation,e.g. finishing,

Table 1: Test F1-scores on MT-MNIST with $T_{test} = T_0$ (top), MiniImageNet (middle) and STS-Sawtooth (bottom). For complete results including confidence intervals see Appendix C

MT-MNIST (T_0)	$K = 2$		$K = 10$		$K = 100$		
Model \ c	50%	0%	50%	0%	50%	1%	0%
STL	63.0%	1.5%	86.3%	0.4%	88.9%	21.5%	0.0%
MTL	45.0%	36.4%	85.0%	37.4%	**96.2%**	27.1%	34.0%
MAML	**83.0%**	78.2%	**93.2%**	80.1%	95.9%	27.7%	0.1%
OC-MAML (ours)	—	**87.5%**	—	**92.6%**	—	**94.6%**	**94.6%**

MiniImageNet	$K = 2$		$K = 10$		$K = 100$		
Model \ c	50%	0%	50%	0%	50%	1%	0%
STL	47.6%	8.2%	56.0%	0.8%	68.9%	5.1%	0.7%
MTL	52.3%	40.3%	62.7%	32.9%	72.6%	45.2%	41.1%
MAML	**58.4%**	55.5%	**70.7%**	55.8%	**78.6%**	51.8%	27.5%
OC-MAML (ours)	—	**60.1%**	—	**72.8%**	—	**73.5%**	**74.5%**

STS (Sawtooth)	$K = 2$		$K = 10$		$K = 100$		
Model \ c	50%	0%	50%	0%	50%	1%	0%
STL	38.9%	0.0%	46.5%	0.0%	65.9%	0.6%	0.0%
MTL	38.6%	0.0%	61.3%	0.0%	**93.1%**	0.5%	0.0%
MAML	**90.4%**	78.0%	**96.3%**	79.3%	87.3%	81.0%	74.6%
OC-MAML (ours)	—	**96.9%**	—	**97.0%**	—	**97.9%**	**93.9%**

we use the tasks from the other operations of the same type for meta-training. We note that the model is not exposed to any sample belonging to any task of the target operation during training.

4.3 Results

Our results on MT-MNIST with $T_{test} = T_0$, on MiniImageNet and on STS-Sawtooth are summarized in Table 1. The results on the MT-MNIST datasets of the other digits and on the STS-Sine (see Appendix C) are consistent with the results in Table 1. We note that in the class-balanced case ($c_{target} = 50\%$), OC-MAML is equivalent to MAML. We choose the F1-score metric as evaluation metric, since it focuses on the model performance on the minority class, i.e. in detecting anomalies, which is more important in the anomaly detection applications.

OC-MAML consistently outperforms all the baseline by a significant margin in all OCC experiments, i.e. $c_{target} \in \{0\%, 1\%\}$, across all target task sizes K, on all datasets. The performance gap between the MAML and OC-MAML models shows that explicitly optimizing for high class-imbalance learning, i.e. learning with few or no anomalous samples, yields an initialization that enables a more successful adaptation to unseen OCC tasks. While MAML learns relatively good model initializations when the target OCC task size is small ($K \in \{2, 10\}$), it completely fails at doing so for higher task sizes ($K = 100$), particularly in the absence of anomalous samples ($c = 0\%$). This holds for all 9 MT-MNIST, MiniImageNet and STS-Sine datasets. We note that, when the target task is class-balanced, MAML's performance increases with an increasing task size K. Two questions can arise from this observation: (1) How can MAML learn a good initialization for adapting to OCC tasks, despite optimizing for few-shot *class-balanced* classification? and (2) Why does this only happen in the few-shot regime ($K \in \{2, 10\}$)? In the following, we give a potential explanation for these questions.

In a gradient descent update, using fewer datapoints to compute the loss gradient leads to a higher stochasticity in the direction of this latter. This is due to the noise introduced by the sampling operation of these datapoints from all datapoints defining the task considered (Goodfellow et al., 2016). During meta-training, MAML optimizes for good generalization on a given task, independently of which K of its datapoints are used for adaptation. Hence, lower values of K lead to learning initializations that tolerate higher loss gradient stochasticities at adaptation time. Since datapoints from different classes yield different loss gradient directions, we can say that changing the CIR of a batch of data samples also induces gradient stochasticity. We explain MAML's ability to adapt to OCC tasks, despite being trained for class-balanced learning, by its ability to tolerate gradient stochasticity acquired during *few-shot* meta-training. In other words, the resistance to gradient stochasticity

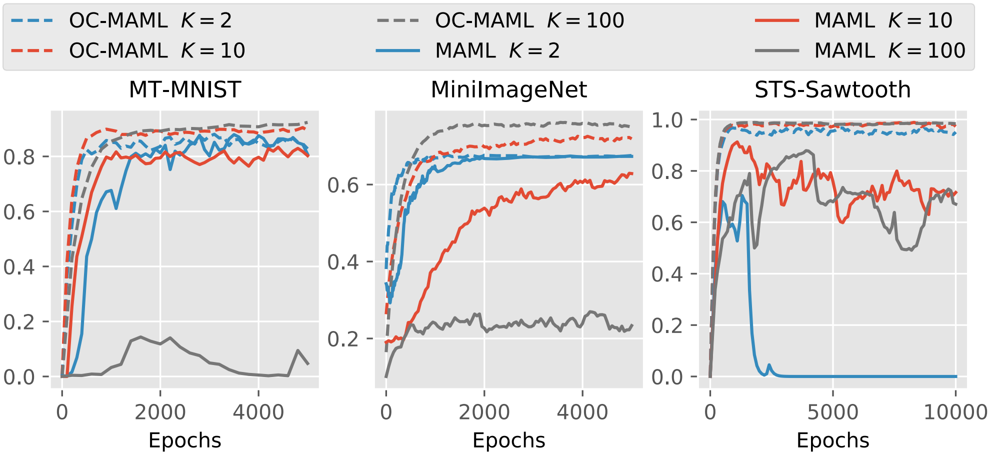


Figure 1: F1-scores computed on the meta-validation tasks during meta-training of MAML and OC-MAML in experiments with CIR $c_{target} = 0\%$ on the MT-MNIST dataset with $T_{test} = T_0$ (left), on the STS-Sawtooth dataset (middle) and the MiniImagenet dataset (right).

Table 2: Test F1-scores of OC-MAML on finishing (F_i) and roughing (R_j) operations of the CNC-MMD dataset, when only $K = 10$ normal examples are available ($c_{target} = 0\%$). The ± shows 95% confidence intervals over 150 tasks sampled from the test operations.

F_1	F_2	F_3	F_4	R_1	R_2
$80.0 \pm 2.3\%$	$89.6 \pm 2.1\%$	$95.9 \pm 1.1\%$	$93.6 \pm 3.4\%$	$85.3 \pm 1.4\%$	$82.6 \pm 1.4\%$

acquired by few-shot learning is used to enable class-imbalance learning. Since this ability is only acquired when optimizing for few-shot learning ($K \in \{2, 10\}$), MAML fails at adapting to OCC tasks when $K = 100$. The model initializations learned by OC-MAML, however, enable quick adaptation to OCC tasks in both, the few-shot and the many-shot data regimes. The superior performance of OC-MAML shows the benefit of explicitly optimizing for OCC. Besides outperforming MAML in terms of adapted model's performance, OC-MAML enables a faster and more stable meta-training, in the absence of anomalous examples, in the few-shot and many-shot data regimes. The faster learning is indicated by the higher steepness of the validation F1-score curves at the beginning of meta-training in Figure 4.3.

The results of OC-MAML experiments on the CNC-MMD dataset are presented in Table 2. OC-MAML consistently achieves high F1-scores between, ca. 80% and ca. 96%, across the 6 different milling processes. This high model performance on the minority class, i.e. in detecting anomalous data samples, is reached by using only $K = 10$ non-anomalous data samples ($c = 0\%$). These results show that OC-MAML yielded a parameter initialization suitable for learning OCC tasks in the time-series data domain. Moreover, the high performance reached show the maturity of this method for industrial real-world applications.

5 Conclusion

This work addressed the novel and challenging problem of few-shot one-class classification (FS-OCC). We introduced OC-MAML, a meta-learning approach to FS-OCC problems that learns model parameters which are easily adaptable to unseen tasks using few datapoints from only one class. We demonstrated the viability of our method on four datasets from the image and time-series domain, including a real-world dataset of industrial sensor readings. Our approach is simple, fast and data domain agnostic. Future works could focus on designing an unsupervised approach to the FS-OCC problem, as done by Hsu et al. (2018) for the few-shot class-balanced classification problem.

REFERENCES

Charu C Aggarwal. Outlier analysis. In *Data mining*, pp. 237–263. Springer, 2015.

Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2:1–18, 2015.

Jerone TA Andrews, Thomas Tanay, Edward J Morton, and Lewis D Griffin. Transfer representation-learning for anomaly detection. ICML, 2016.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

Jinghui Chen, Saket Sathe, Charu Aggarwal, and Deepak Turaga. Outlier detection with autoencoder ensembles. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pp. 90–98. SIAM, 2017.

Sarah M Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, and Christopher Leckie. High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition*, 58:121–134, 2016.

Chelsea Finn and Sergey Levine. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm, 2017.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135. JMLR. org, 2017.

Pedro Garcia-Teodoro, Jesus Diaz-Verdejo, Gabriel Maciá-Fernández, and Enrique Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, 28(1-2):18–28, 2009.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

Simon Hawkins, Hongxing He, Graham Williams, and Rohan Baxter. Outlier detection using replicator neural networks. In *International Conference on Data Warehousing and Knowledge Discovery*, pp. 170–180. Springer, 2002.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

Kyle Hsu, Sergey Levine, and Chelsea Finn. Unsupervised learning via meta-learning, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL `http://www.scipy.org/`. [Online; accessed ¡today¿].

Shehroz S Khan and Michael G Madden. One-class classification: taxonomy of study and review of techniques. *The Knowledge Engineering Review*, 29(3):345–374, 2014.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Gregory R. Koch. Siamese neural networks for one-shot image recognition. 2015.

Jedrzej Kozerawski and Matthew Turk. Clear: Cumulative learning for one-shot one-class image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3446–3455, 2018.

Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The mnist database of handwritten digits, 2010. `http://yann.lecun.com/exdb/mnist/`.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning, 2017.

Mary M Moya, Mark W Koch, and Larry D Hostetler. One-class classifier networks for target recognition applications. *NASA STI/Recon Technical Report N*, 93, 1993.

Alex Nichol and John Schulman. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2018.

Mahesh Pal and Giles M Foody. Feature selection for classification of hyperspectral data by svm. *IEEE Transactions on Geoscience and Remote Sensing*, 48(5):2297–2307, 2010.

Marcel Prastawa, Elizabeth Bullitt, Sean Ho, and Guido Gerig. A brain tumor segmentation framework based on outlier detection. *Medical image analysis*, 8(3):275–283, 2004.

Mahdyar Ravanbakhsh, Moin Nabi, Enver Sangineto, Lucio Marcenaro, Carlo Regazzoni, and Nicu Sebe. Abnormal event detection in videos using generative adversarial nets. *2017 IEEE International Conference on Image Processing (ICIP)*, Sep 2017. doi: 10.1109/icip.2017.8296547. URL `http://dx.doi.org/10.1109/icip.2017.8296547`.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International Conference on Machine Learning*, pp. 4393–4402, 2018.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization, 2018.

Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. doi: 10.1109/cvpr.2018.00356. URL `http://dx.doi.org/10.1109/cvpr.2018.00356`.

Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, pp. 146–157. Springer, 2017.

Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

Luke Scime and Jack Beuth. Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm. *Additive Manufacturing*, 19: 114–126, 2018.

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

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1): 45–66, 2004.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning, 2016.

Yaqing Wang and Quanming Yao. Few-shot learning: A survey. *arXiv preprint arXiv:1904.05046*, 2019.

Dan Xu, Elisa Ricci, Yan Yan, Jingkuan Song, and Nicu Sebe. Learning deep representations of appearance and motion for anomalous event detection. *Procedings of the British Machine Vision Conference 2015*, 2015. doi: 10.5244/c.29.8. URL `http://dx.doi.org/10.5244/C.29.8`.

A Experiment Details

For MT-MNIST, we use the same 4-block convolutional architecture as used by Hsu et al. (2018) for their multi-class MNIST experiments, where each block includes a 3 x 3 convolutional layer with 32 filters, a 2 x 2 pooling, a batch normalization layer (Ioffe & Szegedy, 2015) and a ReLU non-linearity. The same model architecture is used for the MiniImageNet experiments as done by Ravi & Larochelle (2016). On the STS datasets, the model architecture used is composed of 3 modules, each including a 5 x 5 convolutional layer with 32 filters, a 2 x 2 pooling and a ReLU non-linearity. The model architecture used for the CNC-MMD experiments is composed of 4 of these aforementioned modules, except that the convolutional layers in the last two modules include 64 filters. The last layer of all architectures is a linear layer followed by softmax. We note that in the experiments on the time-series datasets (STS and CNC-MMD) 1-D convolutional filters are used.

Table 3: Hyperparameters overview

Hyperparameter	MT-MNIST	MiniImageNet	STS	CNC-MMD
Input size	28 x 28	84 x 84 x 3	128	2048 x 3
OC-MAML and MAML				
Outer learning rate (β)	0.001	0.001	0.001	0.001
Inner learning rate (α)	0.05	0.01	0.01	0.0001
Task batch size	8	8	8	16
Adaptation steps	5	5	10	5
Outer loop size (Q)	40	60	50	4 - 16
MTL				
Batch size	32	32	32	—
Learning rate	0.05	0.01	0.01	—
STL				
Batch size	32	32	32	—
Learning rate	0.05	0.01	0.01	—

Table 3 shows the hyperparameters used in the experiments of each model on the different datasets. We note that we did not fix the outer loop size Q in the experiments on the CNC-MMD dataset, because the sizes and CIRS of the validation sets D^{val} differ across the different tasks. In the MAML and OC-MAML experiments, we used vanilla SGD in the inner loop and the Adam optimizer (Kingma & Ba, 2014) in the outer loop, as done by Finn et al. (2017). The STL and MTL baselines are also trained with the Adam optimizer.

In the following, we provide details about the meta-training procedure adopted in the MAML and OC-MAML experiments. We use disjoint sets of data for adaptation (D^{tr}) and validation (D^{val}) on the meta-training tasks, as it was empirically found to yield better final performance (Nichol & Schulman, 2018). Hereby, the same sets of data are used in the MAML and OC-MAML experiments. In the MT-MNIST, MiniImageNet and STS experiments, the aforementioned sets of data are class-balanced. The sampling of the batch used for adaptation B ensures that this latter has the appropriate CIR ($c = 50\%$ for MAML and $c = c_{target}$ for OC-MAML). In the OC-MAML experiments, where $c_{target} = 0\%$, i.e. no anomalous samples of the target task are available, only normal examples are sampled from D^{tr} during meta-training. In order to ensure that MAML and OC-MAML are exposed to the same data during meta-training, we move the anomalous examples from the adaptation set of data (D^{tr}) to the validation set of data (D^{val}). We note that this is only done in the OC-MAML experiments, where $c_{target} = 0\%$.

In the following, we provide details about the adaptation to the target task(s) and the subsequent evaluation. In the MT-MNIST and MiniImageNet experiments, we randomly sample 20 adaptation sets from the target task(s)' data, each including K examples with the CIR corresponding to the experiment considered. After each adaptation episode conducted using one of these sets, the adapted model is evaluated on a disjoint class-balanced test set that includes 4,000 images for MT-MNIST and 600 for MiniImageNet. We note that the samples included in the test sets of the test tasks are not used nor for meta-training neither for meta-validation. This results in 20 and 400 (20 adaptation sets created from each of the 20 test classes) different test tasks for MT-MNIST and MiniImageNet, respectively. The results presented in Tables 4 and 5, i.e. mean and 95% confidence intervals, are computed over all adaptation episodes. Likewise, in the STS experiments, we evaluate the model on 10 different adaptation sets from each of the 5 test tasks. In the CNC-MMD experiments, the 30 tasks created from the target operation are used for adaptation and subsequent evaluation. For each of these target tasks, we randomly sample K datapoints belonging to the normal class that we use for adaptation, and use the rest of the datapoints for testing. We do this 5 times for each target task, which results in 150 testing tasks.

In the MT-MNIST and MiniImageNet experiments, where batch normalization (Ioffe & Szegedy, 2015) is used, we ensure that no information is shared between the test samples via batch normalization statistics. We do so, by computing the batch normalization statistics using only the K datapoints available for adaptation. These statistics are then used to evaluate the adapted model on the test set. Nichol & Schulman (2018) refer to this experiment setting as non-transductive.

In our MTL baseline experiments, we use the meta-validation task(s) for model choice, like in the MAML and OC-MAML experiments. Hereby, for each validation task, we finetune a fully connected layer on top of the shared multi-task learned layers, as it is done at test time.

B Details about the time-series datasets STS and CNC-MMD

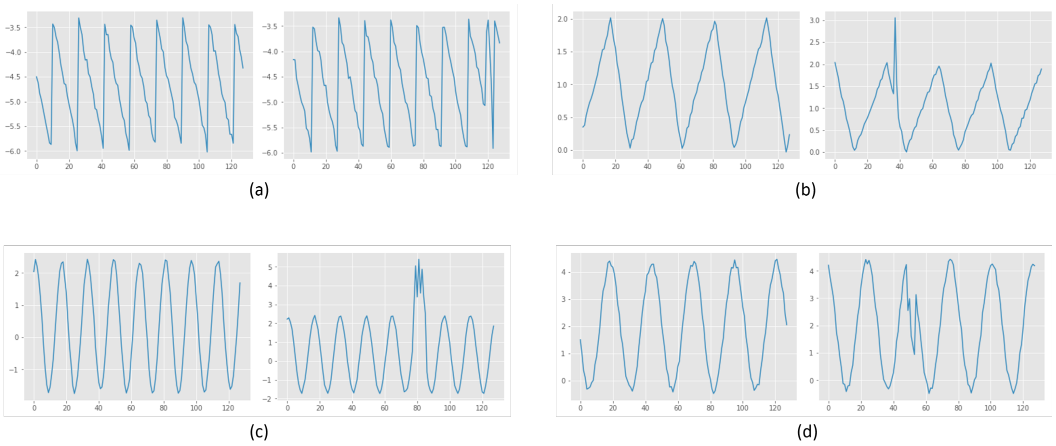
B.1 STS

In this Section, we give details about the generation procedure adopted to create the STS-Sawtooth dataset. The same steps were conducted to generate the STS-Sine dataset. First, we generate the sawtooth waveforms underlying the different tasks by using the Signal package of the Scipy library (Jones et al., 2001–). Thereafter, a randomly generated noise is applied to each signal. Subsequently, signal segments with window length $l = 128$ are randomly sampled from each noisy signal. These represent the normal, i.e. non-anomalous, examples of the corresponding task. Then, some of the normal examples are randomly chosen, and anomalies are added to them to produce the anomalous examples.

Figure 2 shows exemplary normal and anomalous samples from the STS-Sawtooth and STS-Sine datasets. In order to increase the variance between the aforementioned synthetic signals underlying the different tasks, we randomly sample the frequency, i.e. the number of periods within the window length l, with which each waveform is generated, as well as the amplitude and the vertical position (see Figure 2). For sawtooth waveforms, we also randomly sample the width of the rising ramp as a proportion of the total cycle between 0% and 100%, for each task. Setting this value to 100% and to 0% produces sawtooth waveforms with rising and falling ramps, respectively. Setting it to 50% corresponds to triangle waveforms.

We note that the noise applied to the tasks are randomly sampled from *task-specific* intervals, the boundaries of which are also randomly sampled. Likewise, the width and height of each anomaly is sampled from a random task specific-interval. Moreover, we generate the anomalies of each task, such that half of them have a height between the signal's minimum and maximum (e.g. anomalies (a) and (d) in Figure 2), while the other half can surpass these boundaries, i.e. the anomaly is

Figure 2: Exemplary normal (left) and anomalous (right) samples belonging to different tasks from the STS-Sawtooth (a and b) and the STS-Sine (c and d) datasets



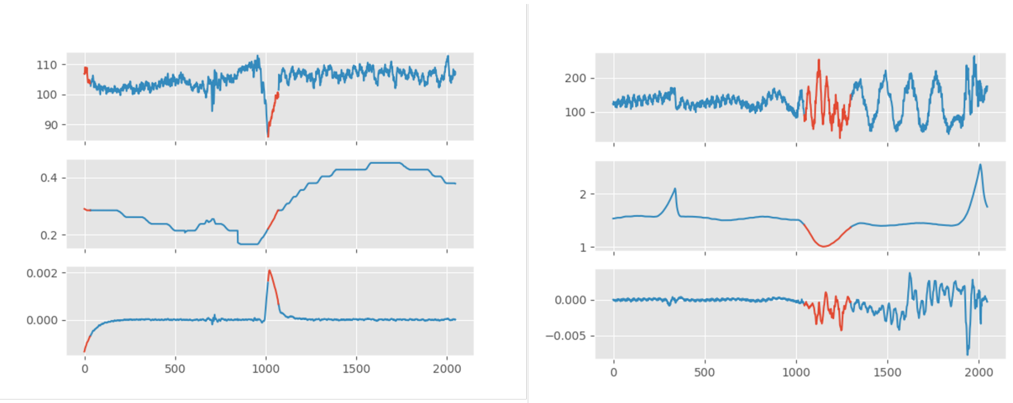
higher than the normal signal's maximum or lower than its minimum at least at one time step (e.g. anomalies (b) and (c) in Figure 2). We note that an anomalous sample can have more than one anomaly.

We preprocess the data by removing the mean and scaling to unit variance. Hereby, only the available *normal* examples are used for the computation of the mean and the variance. This means that in the experiments, where the target task's size $K = 2$ and only normal samples are available $c = 0\%$, only two examples are used for the mean and variance computation. We note that the time-series in Figure 2 are not preprocessed.

B.2 CNC-MMD

We preprocess each of the three signals separately by removing the mean and scaling to unit variance, as done for the STS datasets. Likewise, only the available *normal* examples are used for the computation of the mean and the variance.

Figure 3: Exemplary anomalous samples from a finishing (left) and a roughing (right) operations, where the anomalous time-steps are depicted in red.



Exemplary anomalous signals recorded from a finishing and a roughing operations are shown in Figure 3. These signals are not mean centered and scaled to unit variance. We note that we do not use the labels per time-step, but rather the label ”anomalous” is assigned to each time-series that contains at least an anomalous time-step.

C Experimental Results

In this Section, we present the complete results of the experiments on the two STS datasets and the 9 MT-MNIST datasets, including confidence intervals 95%.

Table 4: Complete results on the MT-MNIST datasets ($T_{target} = T_0 - T_4$). The ± shows 95% confidence interval computed over 20 test tasks.

MT-MNIST (T_0)	$K = 2$		$K = 10$		$K = 100$		
Model \ c	50%	0%	50%	0%	50%	1%	0%
STL	62.9 ± 9.8%	1.5 ± 1.5%	86.3 ± 1.6%	0.4 ± 0.6%	88.9 ± 9.0%	21.5 ± 7.8%	0.0 ± 0.0%
MTL	45.0 ± 8.5%	36.4 ± 8.5%	85.0 ± 2.9%	37.4 ± 8.0%	**96.1 ± 0.8%**	27.1 ± 8.4%	34.0 ± 0.0%
MAML	**83.0 ± 2.2%**	78.2 ± 2.5%	**93.2 ± 0.9%**	80.1 ± 2.1%	95.9 ± 0.3%	27.7 ± 4.9%	0.1 ± 0.0%
OC-MAML (ours)	—	**87.5 ± 2.7%**	—	**92.6 ± 0.6%**	—	**94.6 ± 0.3%**	**94.6 ± 0.2%**

MT-MNIST (T_1)	$K = 2$		$K = 10$		$K = 100$		
Model \ c	50%	0%	50%	0%	50%	1%	0%
STL	45.1 ± 12.2%	0.9 ± 1.1%	89.0 ± 3.0%	0.3 ± 0.3%	96.4 ± 1.1%	57.1 ± 7.8%	0.0 ± 0.0%
MTL	45.5 ± 10.2%	44.6 ± 10.1%	90.1 ± 2.0%	9.3 ± 6.7%	97.0 ± 1.3%	26.4 ± 7.1%	14.3 ± 0.0%
MAML	**88.9 ± 2.5%**	86.7 ± 2.6%	**95.7 ± 0.7%**	70.0 ± 3.1%	**98.4 ± 0.1%**	40.5 ± 8.6%	0.6 ± 0.2%
OC-MAML (ours)	—	**92.5 ± 1.4%**	—	**95.2 ± 0.8%**	—	**92.8 ± 1.0%**	**91.0 ± 2.3%**

MT-MNIST (T_2)	$K = 2$		$K = 10$		$K = 100$		
Model \ c	50%	0%	50%	0%	50%	1%	0%
STL	58.1 ± 8.1%	1.6 ± 1.5%	76.6 ± 4.1%	0.4 ± 0.5%	92.5 ± 1.1%	16.7 ± 7.9%	0.0 ± 0.0%
MTL	51.3 ± 7.6%	34.2 ± 6.9%	83.2 ± 2.6%	10.8 ± 7.5%	**96.3 ± 0.9%**	10.3 ± 4.4%	49.3 ± 12.4%
MAML	**83.7 ± 2.2%**	75.0 ± 4.2%	**89.3 ± 1.1%**	60.4 ± 4.5%	95.9 ± 0.3%	13.6 ± 6.5%	0.2 ± 0.1%
OC-MAML (ours)	—	**78.8 ± 2.2%**	—	**92.4 ± 0.6%**	—	**95.3 ± 0.3%**	**94.4 ± 0.3%**

MT-MNIST (T_3)	$K = 2$		$K = 10$		$K = 100$		
Model \ c	50%	0%	50%	0%	50%	1%	0%
STL	49.0 ± 8.1%	2.9 ± 4.9%	74.7 ± 3.0%	0.2 ± 0.3%	88.7 ± 3.2%	8.1 ± 3.2%	0.0 ± 0.0%
MTL	43.4 ± 6.3%	39.4 ± 5.2%	82.2 ± 3.0%	25.0 ± 5.1%	95.2 ± 1.3%	20.6 ± 5.5%	33.8 ± 5.7%
MAML	**83.0 ± 2.7%**	80.8 ± 2.8%	**89.0 ± 1.0%**	79.0 ± 3.3%	**95.4 ± 0.4%**	23.1 ± 7.6%	0.0 ± 0.0%
OC-MAML (ours)	—	**82.1 ± 2.9%**	—	**92.7 ± 0.7%**	—	**91.7 ± 0.5%**	**89.8 ± 0.8%**

MT-MNIST (T_4)	$K = 2$		$K = 10$		$K = 100$		
Model \ c	50%	0%	50%	0%	50%	1%	0%
STL	36.3 ± 10.4%	0.3 ± 0.4%	77.9 ± 2.4%	0.0 ± 0.0%	86.6 ± 5.6%	12.0 ± 5.1%	0.0 ± 0.0%
MTL	35.1 ± 9.1%	29.5 ± 12.4%	81.0 ± 2.3%	16.8 ± 8.1%	**95.0 ± 0.9%**	16.2 ± 2.2%	22.3 ± 11.6%
MAML	**77.0 ± 2.7%**	60.2 ± 3.5%	**91.5 ± 1.2%**	83.3 ± 1.7%	93.9 ± 0.7%	20.0 ± 6.8%	0.2 ± 0.1%
OC-MAML (ours)	—	**82.1 ± 2.2%**	—	**87.0 ± 1.1%**	—	**90.8 ± 0.3%**	**89.5 ± 0.4%**

Table 5: Complete results on the MT-MNIST datasets ($T_{target} = T_5 - T_8$). The ± shows 95% confidence interval computed over 20 test tasks.

MT-MNIST (T_5)	$K = 2$		$K = 10$		$K = 100$		
Model $\backslash c$	50%	0%	50%	0%	50%	1%	0%
STL	$59.0 \pm 6.7\%$	$1.1 \pm 1.7\%$	$73.0 \pm 5.5\%$	$0.0 \pm 0.0\%$	$91.9 \pm 1.6\%$	$11.1 \pm 4.2\%$	$0.0 \pm 0.0\%$
MTL	$37.2 \pm 10.9\%$	$15.2 \pm 10.4\%$	$88.2 \pm 1.1\%$	$22.0 \pm 11.7\%$	$\mathbf{95.8 \pm 1.2\%}$	$13.9 \pm 5.1\%$	$12.3 \pm 10.3\%$
MAML	$\mathbf{84.6 \pm 2.7\%}$	$79.2 \pm 1.7\%$	$\mathbf{89.2 \pm 1.2\%}$	$83.7 \pm 1.4\%$	$94.5 \pm 0.6\%$	$32.4 \pm 6.5\%$	$16.0 \pm 1.2\%$
OC-MAML (ours)	—	$\mathbf{86.6 \pm 2.4\%}$	—	$\mathbf{91.7 \pm 0.7\%}$	—	$\mathbf{93.0 \pm 0.4\%}$	$\mathbf{92.8 \pm 0.6\%}$

MT-MNIST (T_6)	$K = 2$		$K = 10$		$K = 100$		
Model $\backslash c$	50%	0%	50%	0%	50%	1%	0%
STL	$55.5 \pm 9.0\%$	$3.3 \pm 2.7\%$	$83.8 \pm 3.4\%$	$0.4 \pm 0.6\%$	$94.8 \pm 1.3\%$	$24.6 \pm 8.3\%$	$0.0 \pm 0.0\%$
MTL	$61.6 \pm 5.4\%$	$48.7 \pm 7.1\%$	$90.6 \pm 2.7\%$	$41.3 \pm 7.7\%$	$\mathbf{97.9 \pm 0.2\%}$	$43.9 \pm 4.0\%$	$42.7 \pm 5.3\%$
MAML	$\mathbf{86.3 \pm 1.5\%}$	$76.8 \pm 3.1\%$	$\mathbf{96.7 \pm 0.6\%}$	$76.7 \pm 2.5\%$	$\mathbf{97.9 \pm 0.2\%}$	$47.3 \pm 9.6\%$	$6.2 \pm 0.8\%$
OC-MAML (ours)	—	$\mathbf{92.7 \pm 1.4\%}$	—	$\mathbf{97.4 \pm 0.3\%}$	—	$\mathbf{98.0 \pm 0.1\%}$	$\mathbf{97.5 \pm 0.2\%}$

MT-MNIST (T_7)	$K = 2$		$K = 10$		$K = 100$		
Model $\backslash c$	50%	0%	50%	0%	50%	1%	0%
STL	$44.5 \pm 11.5\%$	$0.7 \pm 0.9\%$	$75.0 \pm 6.6\%$	$0.6 \pm 0.6\%$	$92.5 \pm 1.2\%$	$10.7 \pm 4.5\%$	$0.0 \pm 0.0\%$
MTL	$46.7 \pm 13.2\%$	$27.9 \pm 11.2\%$	$84.2 \pm 3.4\%$	$18.2 \pm 10.9\%$	$95.5 \pm 1.9\%$	$19.3 \pm 7.1\%$	$48.8 \pm 9.1\%$
MAML	$\mathbf{85.2 \pm 2.6\%}$	$82.1 \pm 1.8\%$	$\mathbf{91.5 \pm 1.0\%}$	$72.0 \pm 3.9\%$	$\mathbf{95.7 \pm 0.3\%}$	$35.9 \pm 8.2\%$	$10.6 \pm 2.0\%$
OC-MAML (ours)	—	$\mathbf{87.3 \pm 2.8\%}$	—	$\mathbf{93.7 \pm 0.3\%}$	—	$\mathbf{89.3 \pm 0.7\%}$	$\mathbf{90.7 \pm 0.5\%}$

MT-MNIST (T_8)	$K = 2$		$K = 10$		$K = 100$		
Model $\backslash c$	50%	0%	50%	0%	50%	1%	0%
STL	$52.9 \pm 8.6\%$	$0.9 \pm 1.5\%$	$70.5 \pm 6.4\%$	$0.0 \pm 0.0\%$	$74.6 \pm 12.0\%$	$11.1 \pm 3.9\%$	$0.0 \pm 0.0\%$
MTL	$43.6 \pm 4.6\%$	$48.6 \pm 4.6\%$	$75.6 \pm 4.4\%$	$35.6 \pm 6.0\%$	$\mathbf{95.8 \pm 1.0\%}$	$15.7 \pm 3.2\%$	$39.5 \pm 2.6\%$
MAML	$\mathbf{80.3 \pm 2.9\%}$	$\mathbf{77.4 \pm 2.8\%}$	$\mathbf{87.8 \pm 1.7\%}$	$49.0 \pm 6.0\%$	$93.8 \pm 0.5\%$	$23.9 \pm 5.8\%$	$0.0 \pm 0.0\%$
OC-MAML (ours)	—	$76.5 \pm 2.1\%$	—	$\mathbf{90.0 \pm 0.8\%}$	—	$\mathbf{88.8 \pm 0.9\%}$	$\mathbf{92.5 \pm 0.6\%}$

Table 6: Complete results on the MiniImageNet dataset. The ± shows 95% confidence interval computed over 400 test tasks.

MiniImageNet	$K = 2$		$K = 10$		$K = 100$		
Model $\backslash c$	50%	0%	50%	0%	50%	1%	0%
STL	$47.6 \pm 1.7\%$	$8.2 \pm 1.1\%$	$56.0 \pm 1.2\%$	$0.8 \pm 0.3\%$	$68.9 \pm 1.6\%$	$5.1 \pm 0.8\%$	$0.7 \pm 2.7\%$
MTL	$52.3 \pm 1.3\%$	$40.3 \pm 1.1\%$	$62.7 \pm 1.1\%$	$32.9 \pm 1.2\%$	$72.6 \pm 0.9\%$	$45.2 \pm 0.7\%$	$41.1 \pm 0.4\%$
MAML	$\mathbf{58.4 \pm 1.0\%}$	$55.5 \pm 1.2\%$	$\mathbf{70.7 \pm 0.7\%}$	$55.8 \pm 1.4\%$	$\mathbf{78.6 \pm 0.5\%}$	$51.8 \pm 1.6\%$	$27.5 \pm 1.4\%$
OC-MAML (ours)	—	$\mathbf{60.1 \pm 0.7\%}$	—	$\mathbf{72.8 \pm 0.5\%}$	—	$\mathbf{73.5 \pm 0.5\%}$	$\mathbf{74.5 \pm 0.4\%}$

Table 7: Complete results on the STS-Sawtooth (top) and STS-Sine (bottom) datasets. The ± shows 95% confidence interval computed over 50 test tasks.

STS (Sawtooth)	$K = 2$		$K = 10$		$K = 100$		
Model \ c	50%	0%	50%	0%	50%	1%	0%
STL	38.9 ± 5.3%	0.0 ± 0.0%	46.5 ± 4.5%	0.0 ± 0.0%	65.9 ± 2.6%	0.6 ± 0.6%	0.0 ± 0.0%
MTL	38.6 ± 8.3%	0.0 ± 0.0%	61.3 ± 5.1%	0.0 ± 0.0%	**93.1 ± 2.2%**	0.5 ± 0.5%	0.0 ± 0.0%
MAML	**90.4 ± 2.7%**	78.0 ± 6.3%	**96.3 ± 1.4%**	79.3 ± 6.7%	87.3 ± 3.1%	81.0 ± 5.3%	74.6 ± 5.0%
OC-MAML (ours)	—	**96.9 ± 1.1%**	—	**97.0 ± 1.3%**	—	**97.9 ± 0.4%**	**93.9 ± 1.5%**

STS (Sine)	$K = 2$		$K = 10$		$K = 100$		
Model \ c	50%	0%	50%	0%	50%	1%	0%
STL	33.0 ± 6.5%	0.0 ± 0.0%	35.9 ± 3.2%	0.0 ± 0.0%	75.7 ± 3.1%	0.8 ± 0.5%	0.0 ± 0.0%
MTL	54.9 ± 4.4%	0.0 ± 0.0%	90.3 ± 1.3%	0.0 ± 0.0%	95.2 ± 1.1%	47.4 ± 2.8%	13.8 ± 0.7%
MAML	**99.9 ± 0.1%**	98.9 ± 0.2%	**99.6 ± 0.1%**	95.9 ± 0.7%	**99.4 ± 0.2%**	99.3 ± 0.2%	0.4 ± 0.2%
OC-MAML (ours)	—	**99.9 ± 0.1%**	—	**99.8 ± 0.1%**	—	**99.8 ± 0.1%**	**98.0 ± 1.0%**