# BatMan-CLR: Making Few-shots Meta-Learners Resilient Against Label Noise

Jeroen M. Galjaard[1] ✉, Robert Birke[2], Juan F. Pérez[3], and Lydia Y. Chen[1,4]

[1] Delft University of Technology, Mekelweg 5, 2628 CC Delft, The Netherlands
{J.M.Galjaard,Y.Chen-10}@tudelft.nl
[2] University of Turin, Via Pessinetto 12, 10149 Torino, Italy robert.birke@unito.it
[3] Universidad de los Andes, Cra 1 #18A-12, Bogotá, Colombia
jf.perez33@uniandes.edu.co
[4] Université de Neuchâtel, Rue Emile-Argand 11, 2000 Neuchâtel, Switzerland
lydiaychen@ieee.org

**Abstract.** The negative impact of label noise is well studied in classical supervised learning yet remains an open research question in meta-learning. Meta-learners aim to adapt to unseen tasks by learning a good initial model in meta-training and fine-tuning it to new tasks during meta-testing. In this paper, we present an extensive analysis of the impact of label noise on the performance of meta-learners, specifically gradient-based $N$-way $K$-shot learners. We show that the accuracy of Reptile, iMAML, and foMAML drops by up to 34% when meta-training is affected by label noise on the three representative datasets: Omniglot, CifarFS, and MiniImageNet. To strengthen the resilience against label noise, we propose two sampling techniques, namely manifold (Man) and batch manifold (BatMan), which transforms the noisy supervised learners into semi-supervised learners to increase the utility of noisy labels. We construct $N$-way 2-contrastive-shot tasks through augmentation, learn the embedding via a contrastive loss in meta-training, and perform classification through zeroing on the embeddings in meta-testing. We show that our approach can effectively mitigate the impact of meta-training label noise. Even with 60% wrong labels BatMan and Man can limit the meta-testing accuracy drop to 2.5, 9.4, 1.1 percent points with existing meta-learners across Omniglot, CifarFS, and MiniImageNet, respectively. We provide our code online: `https://gitlab.ewi.tudelft.nl/dmls/publications/batman-clr-noisy-meta-learning`.

## 1 Introduction

Few-Shot Learning (FSL) poses the problem where learners need to quickly adapt to new unseen tasks by using a low number of samples. Meta-learning [21,6] emerged as a promising solution to this problem. Like humans, meta-learners learn the information at a higher abstraction or meta-level, providing the inductive bias to adapt to new tasks quickly. Among existing meta-learners, gradient-based few-shot learners, e.g., iMAML [20] and foMAML(+ZO) [6,8], have been shown effective to solve $N$-way $K$-shot $(N, K)$ problems, that need to learn $N$

classes for each way given only $K$ samples each. Such few-shot learners are composed of two stages, each with their own labeled support and query sets. The meta-training stage learns a meta-model using two nested optimization loops. The inner-loop tunes the model to a specific task via supervised learning on the support set. The outer-loop uses the task-specific model and query set to update the meta-model. The meta-testing stage verifies how well the meta-model performs on new tasks. Using a similar structure, it uses supervised learning to adapt the meta-model to an unseen task via a test support set. Then, it computes the learner's accuracy on the test query set comparing given and predicted labels. Each way's class labels are thus crucial in both meta-training and meta-testing.

Label noise is more the norm than a rarity and can significantly degrade the performance of supervised learners [23]. Prior studies address label noise mainly in classical supervised learners. In this context, samples hold labels different from the underlying ground truth. In the context of FSL, label noise means that a shot (example) may not correspond to the way (class) it was provided with. This yields a degenerate $N$-way $K$-shot problem where ways become indistinguishable since they contain shots of the same ground truth. Such noise may appear in the support sets of meta-training and meta-testing and the query set of meta-training.

Despite the importance of labels in meta-training and meta-testing, only a few studies [16,14,15] address the challenge of noisy labels in FSL and only at the meta-testing support set level. However, label noise can appear in all support and query sets, affecting both meta-testing and meta-training, and little is known on its impact and resolution. As the number of samples per way is very limited, e.g., five to ten shots, the task adaptation step can be easily over-parameterized by label noise, leading to significant degradation. Moreover, existing meta-learners that account for label noise still require clean data to learn a meta-objective [10,26].

In this paper, we first empirically show that gradient-based FSL methods, i.e., Reptile [19], Eigen-Reptile [4], iMAML [20], and foMAML+ZO [8], are significantly affected by label noise in both query and support sets during meta-training. To counter the effect of label noise, we propose Man and BatMan, which turn any supervised few-shot learner into a semi-supervised learner by a novel (batch) manifold sampling and contrastive learning. Specifically, we turn a noisy $N$-way $K$-shot problem into a self-cleansed $N$-way 2-*contrastive* shot problem. We first augment the original shots and construct contrastive pairs, ensuring the shots are from the same class. We then sample such pairs from the $N$ ways, termed manifold (Man) samples. To lower the probability of an update seeing only noisy $N$-ways, i.e., overlapping classes for different ways, we draw a batch of such Man samples, termed BatMan sampling. Combining this approach with a self-supervised contrastive loss, we can effectively learn the embedding of the initial model, which can then be adapted in meta-testing to a new $N$-way $K$-shot task. The specific contributions of this paper are:

1. An extensive study on the impact of label noise in meta-training for representative gradient-based meta-learners.
2. A generic, self-cleansing framework, BatMan-CLR, which turns meta-learners into semi-supervised ones by manifold sampling $N$-way 2-contrastive shots.
3. Extensive evaluation on four meta-learners, Reptile, Eigen-Reptile, iMAML, and foMAML, shows nearly no performance degradation under the presence of up to 60% label noise.

Table 1: Meta-learning algorithm's meta-test accuracies (mean with 95% confidence for 3 runs) on clean $\mathcal{D}_{\text{test}}$ under varying levels of $\mathcal{D}_{\text{train}}$ label noise. Evaluated on increasing strenghts of label noise $\epsilon \in \{0.0, 0.3, 0.6\}$, corresponding to clean (0%), 30% and 60% *dataset-level* noise.

(a) Few-shot results with increasing noise on CifarFS [2], and Omniglot [12].

| | CifarFS | | | Omniglot | | |
|---|---|---|---|---|---|---|
| Algorithm | $\epsilon = 0.0$ | $\epsilon = 0.3$ | $\epsilon = 0.6$ | $\epsilon = 0.0$ | $\epsilon = 0.3$ | $\epsilon = 0.6$ |
| foMAML | 69.5±0.26 | 65.2±0.27 | 40.3±0.21 | 99.3±0.04 | 97.7±0.07 | 90.3±0.15 |
| iMAML | 64.0±0.24 | 55.9±0.26 | 46.3±0.24 | 96.9±0.11 | 91.0±0.18 | 82.6±0.19 |
| Reptile | 65.5±0.24 | 58.6±0.26 | 51.8±0.25 | 92.5±0.13 | 79.7±0.21 | 71.5±0.24 |
| Eigen-Reptile | 65.3±0.24 | 58.1±0.27 | 52.7±0.24 | 93.6±0.12 | 83.6±0.19 | 73.4±0.24 |

(b) Few-shot results under increasing noise on MiniImageNet [24].

| Algorithm | $\epsilon = 0.0$ | $\epsilon = 0.3$ | $\epsilon = 0.6$ |
|---|---|---|---|
| foMAML | 52.2±0.22 | 37.1±0.18 | 28.2±0.15 |
| iMAML | 53.9±0.22 | 45.5±0.21 | 20.0±0.08 |
| Reptile | 54.2±0.21 | 27.8±0.14 | 24.6±0.14 |
| Eigen-Reptile | 58.7±$^{0.25}$ | 44.8±0.20 | 24.8±0.14 |

## 2 Preliminary and Related work

**Preliminary on FSL**. FSL considers the setting where a learned model must adapt to new tasks leveraging only few samples. We consider the $N$-way $K$-shot classification problem, which consists of a family of tasks, each comprised of $N$ classes with $K$ samples—termed $(N, K)$-FSL tasks. We focus on gradient-based meta-learners, which use an iterative two-step meta-training algorithm to find a meta-model parameterized by $\theta$ capable of quickly adapting towards a task-specific parameterization $\phi$. Each meta-training epoch $t$, starts by randomly drawing a task $\mathcal{T}$ from a collection of meta-training tasks and samples two disjoint sets, support $\mathcal{D}^{\mathcal{T}}_{\text{support}}$ and query $\mathcal{D}^{\mathcal{T}}_{\text{query}}$, from the training data $\mathcal{D}^{\mathcal{T}}_{\text{train}}$ associated with $\mathcal{T}$. Next, the inner loop of meta-training transforms $\theta_t$ in task-specific parameters $\phi$ by minimizing a supervised loss function $\mathcal{L}_{sup}$ over the input-output pairs $(X_s, Y_s) \in \mathcal{D}^{\mathcal{T}}_{\text{support}}$. Then, the meta-training outer loop uses $\phi$, $\mathcal{L}_{sup}$ and the data $(X_q, Y_q) \in \mathcal{D}^{\mathcal{T}}_{\text{query}}$ to obtain $\theta_{t+1}$ for the next iteration. Note that while the support set $\mathcal{D}^{\mathcal{T}}_{\text{support}}$ is used during an inner-loop to train for a specific task, the query set $\mathcal{D}^{\mathcal{T}}_{\text{query}}$ is used in an outer-loop to learn initialization of the meta-model. Meta-testing evaluates the ability of a trained meta-model with parameters $\theta^*$ to adapt to new tasks. Analogous to meta-training, meta-testing first randomly selects a task $\mathcal{T}_{test}$ for testing and samples $\mathcal{D}^{\mathcal{T}_{test}}_{\text{support}}$ and $\mathcal{D}^{\mathcal{T}_{test}}_{\text{query}}$ from the associated test data $\mathcal{D}_{\text{test}}$. Next, the adaptation step uses $\mathcal{D}^{\mathcal{T}_{test}}_{\text{support}}$ to fine tune $\theta^*$ into task-specific parameters $\phi^*$. Finally, $\phi^*$ is tested on the query set $(X_q, Y_q) \in \mathcal{D}^{\mathcal{T}_{test}}_{\text{query}}$ by comparing its predictions $\hat{Y}_q$ against known labels $Y_q$.

**Gradient-based meta-learners**. MAML [6], the father of gradient-based meta-learners, updates the meta-model via gradient descent through gradient descent. As this operation is very resource intensive and sensitive to hyperparameters [6,20,18], many works propose approximations outperforming the original MAML. iMAML and foMAML+ZO approximate the meta-gradient with respect to $\theta_t$ by leveraging the first-order gradient on $\mathcal{D}_{\text{query}}$ with respect to $\phi_u$. This drops the need for gradient descent through gradient descent. iMAML considers that it can compute the meta-gradient using more adaptation steps and weight regularization. foMAML+ZO assumes that the higher-order meta-gradient components can be ignored altogether. Reptile drops $\mathcal{D}_{\text{query}}$ during meta-training by estimating the meta-gradient by stepping towards $\phi_u$ to find $\theta_{t+1}$. Eigen-Reptile further decomposes parameterization of the inner-learners optimization path from $\phi_0 = \theta_t$ to $\phi_u$ following $u$ optimization steps, i.e., $[\phi_0, \phi_1, \ldots, \phi_u]$, and steps towards the direction with the largest variance.

**Label noise in FSL**. Label noise poses a major challenge to meta-learners, especially in the absence of clean data that can be used as ground truth during meta-training. Although a large collection of work on robust supervised learning exists [25,13,28], these are not directly applicable to meta-learners due to the limited number of samples. Related studies [26,15,10,14,16,14] mainly focus on distilling the label noise in *meta-testing* by explicitly studying the noise patterns [26,14,15], using soft-relabeling [16] through clustering or re-weighting suspicious samples [10] based on additional *clean* data. To our best knowledge, Eigen-Reptile [4] is the only study that addresses noisy *training data* by updating only along the highest variance direction. Such an approach lacks generalization to other meta-learners. Alternatively, UMTRA [9] and CACTUS [7] assume labels are unavailable during meta-training and propose to syntheticly generate support and query sets from $N \times K$ images from pre-generated clusters of training data. Although these methods show promising results, they have considerably larger compute and memory requirements for training on considerably larger batch sizes (256+) and are unable to exploit (noisy) label information during (meta-)training.

**The impact of label noise**. We motivate the need for noise resilience in FSL via an empirical study on three representative datasets, CifarFS [2], Omniglot [11], and MiniImageNet [24], in a (5, 5)-FSL setting with a query set of 15 samples per class. We consider symmetric label noise and train four gradient-based meta-learners, i.e., Reptile (Rp) [19], Eigen-Reptile (ER) [4], fo-MAML+ZO (fM) [8], and iMAML (iM) [20], using hyperparameters comparable to the ones in the corresponding paper (details in Section 4). We remark that we assume that meta-learners have no access to clean data—including validation data, while original works report accuracies of meta-validated models, which are expected to score higher. Table 1 shows the meta-test accuracy across 2048 (5, 5)-FSL tasks of 15 samples obtained by each meta-learner under 0% (no), 30%, and 60% corrupted training labels. Across datasets, meta-learners show significant performance degradation as the noise ratio increases. With CifarFS, accuracy drops across all meta-learners on average by 10.1% and 27.4% under 30% and 60% corrupted labels, respectively. Omniglot shows similar trends but more limited in amplitude, with 8.1% and 17.0% average degradation. This is due to the fact that the Omniglot dataset is easier to learn (without noise all meta-learners achieve above 90% accuracy). Lastly, MiniImageNet shows a lower initial accuracy without label noise due to its more challenging nature, with a mean deterioration of 15.6% at 30% noise, and close to random guessing (20%) at 60% noise. Although ER is the sole meta-learner that explicitly aims to counter noise, it is not always the most robust one. Under moderate noise, i.e., 30%, foMAML+ZO is the least affected. Only on CifarFS and 60% noise, ER is the least affected with only a 0.9% margin. More in general, Reptile, ER and iMAML show a higher but almost linear impact of noise, while foMAML+ZO degrades less with 30% noise but gets much worse under 60% noise. Overall, the results underline the need for better noise resilience across all considered meta-learners.

## 3 Proposed method

The core challenge of dealing with noise is that labels lose meaning and misguide meta-learners. This challenge is amplified in the FSL setting as the limited number of samples (shots) in each class (way) makes it harder to isolate noise from the signal in each class. In other words, the few clean samples —those corresponding to the original uncorrupted class— may not be enough to appropriately guide the gradient descent algorithm. Thus, our approach aims to build clean ways and shots—so that each way is more likely to have samples with valid ground truths for FSL. Specifically, the core of the proposed Man sampling is to use data augmentation to create replacement shots for each way rather than leveraging *other* shots of the same way. This guarantees that the underlying ground truth label for the shots of way is effectively the same. BatMan sampling further introduces batches to increase the likelihood of observing all $N$ classes in a single inner-loop

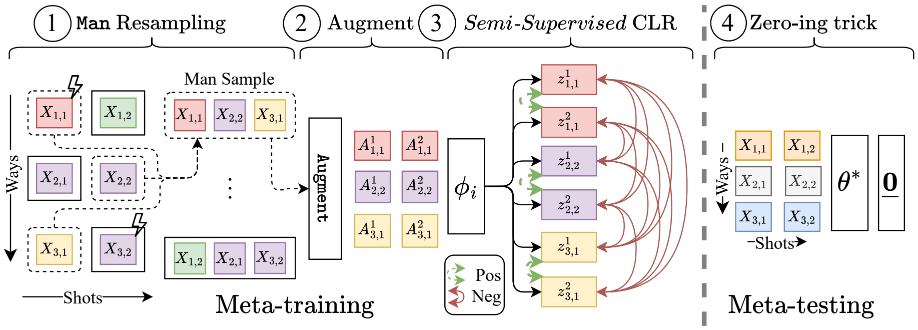


Fig. 1: Illustration of meta-learning with BatMan on a noisy 3-way 2-shot few-shot learning task, i.e., a $(3, 2)$-FSL. The task is provided to the learner as containing three classes (ways): red, purple, and yellow, with two supporting samples (shots) each. Colors indicate the *true* underlying class (ground-truth), while lightning bolts indicate shots provided with a mismatched label (i.e., noisy) (way 1 shot 1, and way 3 shot 2). Steps: ① Man sampling (Algorithm. 2) creates a batch of (3, 1)-FSL manifold samples; ② `Augment` creates semi-supervised (3, 2)-FSL sub-tasks via independent random augmentations; ③ a contrastive loss jointly optimizes all sub-tasks via (Pos)itive and (Neg)ative pairs; ④ trained meta-model $\theta^*$ is meta-tested in a supervised way by appending a zero-ed out linear layer.

step. Figure 1 shows the four main steps of our proposed method: ① resampling, either with Man or BatMan, ② shot augmentation with random independent transformations, ③ 'semi'-supervised meta-training with a contrastive loss, and ④ classification of new tasks (meta-testing) leveraging a zeroing trick. One major benefit of our approach is that these steps can be incorporated into *existing meta-learning algorithms* to achieve label noise robustness.

***Step ①–②: (Batched) Manifold sampling and data augmentation.*** We start with an $(N, K)$-FSL problem with noisy labels and observe that we can frame it as a set of $(N, 2)$-FSL problems. For each $(N, K)$-FSL, we methodically select and augment one shot from each $N$ ways, constructing a new $N$-way 2-shot problem. More formally, each adapted FSL sample is created by independently augmenting sub-sampled shots,

$$M_i = \bigcup_{j=1}^{N} \left\{ \left(\texttt{Augment}(X_j^y), \texttt{Augment}(X_j^y)\right) \mid X_j^y \sim D_{(\cdot)}^{\mathcal{T}}, y = j) \right\},$$

where $X_j^y$ represents a random shot the $j^{\text{th}}$ way of a task's set $\mathcal{D}_{(\cdot)}^{\mathcal{T}}$, and `Aug` a random augmentation function. As a result, we end up with two samples with the same label for each way in each constructed Manifold FSL problem—i.e., the augmentation for each sampled way's shot. Since we sample $N$ observations from all $N \times K$ samples in $\mathcal{D}_{\text{query}}$, we end up with *up to $N$ actual classes*, as selected shots from different ways may share their underlying ground-truth class due to label noise. We coin this sampling approach Manifold (Man) sampling.

**Algorithm 1** BatMan-CLR inner-loop style for gradient-based meta-learners.

**Require:** Inner-learner parameters $\phi$, support set $\mathcal{D}_{\text{support}}$, query set $\mathcal{D}_{\text{query}}$, inner-learning rate $\alpha$, BatMan sample hyper-parameter $N$, inner-loop steps $u$.

**function** BatMan-CLR
  **for** $B_i \in [\texttt{BatMan}(\mathcal{D}_{sup}, N)]_{i=1}^{u}$ **do**
    ▷ ③ *Joint* ***inner-loop*** *Manifold optimization.*
    $\phi \leftarrow \phi - \frac{\alpha}{|B_i|} \nabla_\phi \sum_{i=1}^{|B_i|} \mathcal{L}_{con}(\phi, M_i)$
  $B_{\text{query}} \leftarrow \texttt{BatMan}(\mathcal{D}_{\text{query}}, N)$
  ▷ ③ *Joint* ***outer-loop*** *Manifold optimization.*
  **return** $\phi, \sum_{M_j \in B_{\text{query}}} \mathcal{L}_{con}(\phi, M_j)$

**Algorithm 2** Pseudocode for BatMan Sampling.

**Require:** `Augment` input augmentation function, meta train/test set $\mathcal{D}$, number of manifold batches $B$.

**function** BatMan
  $M = \{\}$
  **for** $j \in [1, \ldots, B]$ **do**
    ▷ ① *Sample single shot each way.*
    $X \leftarrow rand\left((X, y) \in \mathcal{D} \mid y = j\right)$
    ▷ ② *Augment selected samples.*
    $M \leftarrow M \cup \{(\texttt{Augment}(X), j)\}_1^2$
  **return** $M$

We propose a Batched Manifold, as shown in Figure 2, sampling as an extension of Man sampling, where multiple Man samples are grouped to employ more samples in a single step. Multiple 'sub-problems' can be jointly optimized by leveraging a batch of individually created Man samples. More formally, this allows for the joint optimization of,

$$\frac{1}{N} \sum_{M_j \in \texttt{BatMan}(\mathcal{D}_*, B)} \mathcal{L}_{con}(\phi_i, M_j),$$

for both the support and query samples, using a BatMan batch size of $B$, as label information is only leveraged to construct the manifold samples. Herein, $\mathcal{L}_{con}$ is the contrastive loss used to optimize a learner $\phi$. on the sampled manifold samples $M_j$ jointly, and `BatMan` is a function that returns a batched manifold sample for a given support/query set and batch size, as seen in Figure 2). Additionally, this increases the likelihood of considering *all* the provided queries' shots together while calculating meta-gradients.

We illustrate this process in Figure 1 on a 3-way 2-shot FSL task with label noise. The samples $X_{q,r}$ are ordered such that each row index $q$ represents a 'way' and a column index $r$ a corresponding shot. The underlying classes—i.e., ground-truth labels—in the few-shot task are illustrated using different colors. The corrupted two shots are highlighted with lightning bolts: a red one $X_{1,1}$ is provided as green, and a purple $X_{3,2}$ one appears yellow. The (3, 2)-FSL task in this example is thus provided to the learner to discern between 'green', 'purple', and 'yellow' samples. A Man sample is generated by sampling 1 shot from each of the 3 ways and augmenting it twice. To speed up the process, the Man sampler can sample augmentations from a set of pre-fetched augmentations. For BatMan, these steps are repeated to create a batch of such manifold sub-tasks.

In the presence of noise, class clean-up is necessary as shots from two separate ways may share the same ground-truth class. Suppose in the simplest case with 2 classes and a probability $p = 1-\epsilon$ that a sample has a ground truth label, the classes resulting from the Man sampling process belong to different classes with probability $p^2 + (1-p)^2$, as either both lack noise or both are noisy. On the contrary, the 2 samples belong to the same class with probability $2p(1-p)$. In general, with $N$ classes, there are $N!$ combinations in which the Man samples actually correspond to the $N$ *different* classes while there are $N^N$ combinations to select the $N$ samples, with replacement. Let us consider the case of symmetric noise, where a sample is mislabeled with probability $(1-p)/(N-1)$. The probability of obtaining a clean selection of classes can then be posed as the probability of obtaining one of the $N!$ combinations in which this occurs. To represent each possible selection we employ permutation matrices. Let $P_N^i$ be the $N \times N$ $i$th permutation matrix out of the $N!$ such matrices. For instance, in the $N = 3$ case, we have 6 different permutation matrices, i.e.,

$$\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}, \begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}, \begin{bmatrix}0&1&0\\1&0&0\\0&0&1\end{bmatrix}, \dots, \begin{bmatrix}0&0&1\\0&1&0\\1&0&0\end{bmatrix}.$$

Let us also define the $N \times N$ matrix $Q$ with entries $q_{ij}$ such that $q_{ii} = p$ and $q_{ij} = (1-p)/(N-1)$ for $i \neq j$. The probability of obtaining one of these valid permutations under Man sampling can thus be obtained as the trace of the matrix $P_N^i Q$. As a result, the probability of obtaining a clean selection of ways is given by: $\sum_{i=1}^{N!} \operatorname{trace}(P_N^i Q)$, considering all possible valid sample selections that lead to a set with $N$ different classes. As this probability becomes smaller with increasing label noise, BatMan sampling helps by introducing additional samples that increase the likelihood of observing all $N$ classes in a single meta-epoch.

***Step ③: Semi-supervised meta-training with contrastive loss.*** Although re-sampling allows for *likely valid* $(N, 2)$-FSL sub-problems, *which exact classes* they contain remains unknown. These sub-tasks can be considered as *semi-supervised*, as there are now *at most* $N$ classes in each manifold. To allow for (meta-)learning with the semi-supervised sub-problems, we incorporate a contrastive loss to allow for joint optimization of potentially semantically misaligned sub-tasks. Note that we artificially build positive and negative pairs from the sampled augmentations obtained via steps ①–②. These positive and negative pairs can be optimized under a contrastive learning strategy. We use the Decoupled Contrastive Loss (DCL) [27] as contrastive loss function, an adaptation of the infoNCE loss [5],

$$\mathcal{L}_{\mathrm{DCL}} = -\frac{1}{N}\sum_{i=1}^{N} \log\left(\exp(\langle z_i^{(1)} | z_i^{(2)} \rangle)\right) - \log\left(\sum_{j=1}^{2N} \mathbb{1}\left[z_j \notin \{z_i^{(1)}, z_i^{(2)}\}\right] \exp(\langle z_i^{(1)} | z_j \rangle)\right), \quad (1)$$

where $\langle z|z'\rangle$ is the (normalized) cosine-similarity between embedding vectors $z, z'$, and $\mathbb{1}\left[\,\cdot\,\right]$ the indicator function returning 1 when its clause $[\,\cdot\,]$ holds else 0. By design, the decoupled contrastive loss is particularly well-suited for smaller batch sizes, although alternative contrastive losses can replace it. Once samples are augmented and BatMan sampling applied, the batches are used in the meta-training step where the embeddings $z$ are computed and contrasted using a contrastive loss.

***Step ④: Classification of new tasks.*** The meta-model $\theta^*$ trained using the contrastive loss produces embeddings instead of class logits as output. To solve this, we append the meta-model with a fully connected layer $C_{\mathbf{0}} = (\boldsymbol{W}, \underline{\boldsymbol{b}})$ with $\boldsymbol{W} = \mathbf{0}$ and $\underline{\boldsymbol{b}} = \underline{\mathbf{0}}$. This approach decouples the embedding learning from the classification task. Similar to [8], this allows us to treat the model as a semi-supervised meta-learned backbone. The resulting meta-model $\theta^{*\prime} = C_{\mathbf{0}} \circ \theta^*$ can then be treated as a supervised learner utilizing the cross-entropy (CE) classification loss. We found that applying the Zeroing Out trick on the *classification* layer significantly impacts the learner's performance because it allows leveraging the optimized embedding from the pre-trained meta-model $\theta^*$. This is because the stochastic gradient descent will directly use the embeddings as activations without noise introduced by randomly initialized weights.

For conceptual clarity, the algorithm above differs slightly in two points from our implementation of BatMan-CLR, prioritizing efficiency during dataloading while training. First, we change the order of operations from constructing manifolds and then augmenting them to creating augmentations and sampling (batches of) manifolds. This approach enables parallel pre-fetching of FSL tasks with $A$ augmentations for each way's shots. By creating a fixed set of augmentations, the BatMan sub-sampler can randomly sample $N \times K \frac{A(A-1)}{2}!$ Man-samples for each FSL problem. Second, all samples of a BatMan sample are forwarded as a single batch in our BatMan-CLR implementation for each inner-loop and meta-adaptation step—line 3–6 and line 7–8 in Figure 1, using masking and vectorization. Through these design considerations, the BatMan-CLR meta-learners can efficiently perform each (meta-)adaptation step.

## 4 Evaluation Results

We present the effectiveness of BatMan-CLR in enhancing the noise resilience for a total of four gradient-based meta-learners, Eigen-Retipe, Reptile, iMAML,

Table 2: Meta-test accuracy with 95th Confidence Interval ($\overline{acc}\pm_{CI95}$) of Meta-Pretrained models (5, 5)-FSL with varying degrees of label noise $\epsilon \in \{0.0, 0.3, 0.6\}$ during training.

(a) Few-shot results with increasing noise on CifarFS [2], and Omniglot [12].

| | | CifarFS | | | Omniglot | | |
|---|---|---|---|---|---|---|---|
| Algorithm | Sampler | $\epsilon$=0.0 | $\epsilon$=0.3 | $\epsilon$=0.6 | $\epsilon$=0.0 | $\epsilon$=0.3 | $\epsilon$=0.6 |
| foMAML | BatMan | 66.6±0.167 | 65.2±0.17 | 64.8±0.16 | 98.2±0.07 | 98.2±0.06 | 98.0±0.07 |
| | Man | 66.2±0.16 | 64.8±0.17 | 64.0±0.17 | 98.1±0.06 | 98.1±0.06 | 98.1±0.06 |
| iMAML | BatMan | 64.2±0.185 | 62.7±0.25 | 62.9±0.20 | 97.5±0.08 | 98.1±0.07 | 98.3±0.06 |
| | Man | 62.8±0.17 | 62.6±0.17 | 61.7±0.17 | 97.8±0.08 | 98.2±0.07 | 98.2±0.06 |
| Reptile | BatMan | 66.5±0.17 | 65.0±0.17 | 64.1±0.17 | 97.9±0.07 | 97.3±0.08 | 96.2±0.10 |
| | Man | 61.8±0.18 | 62.0±0.18 | 61.4±0.17 | 97.8±0.07 | 97.8±0.07 | 97.7±0.07 |
| Eigen-Reptile | BatMan | 66.3±0.17 | 64.4±0.17 | 63.8±0.18 | 92.6±0.13 | 93.0±0.12 | 93.2±0.12 |
| | Man | 58.0±0.25 | 57.9±0.24 | 58.1±0.24 | 93.7±0.12 | 93.9±0.11 | 94.0±0.11 |

(b) Few-shot results under increasing noise on MiniImageNet [24].

| | | MiniImageNet | | |
|---|---|---|---|---|
| Algorithm | Sampler | $\epsilon$=0.0 | $\epsilon$=0.3 | $\epsilon$=0.6 |
| foMAML | BatMan | 50.4±0.22 | 50.5±0.22 | 50.4±0.21 |
| | Man | 51.6±0.22 | 51.4±0.22 | 51.2±0.22 |
| iMAML | BatMan | 50.9±0.15 | 50.5±0.15 | 50.5±0.14 |
| | Man | 50.1±0.21 | 50.3±0.22 | 50.2±0.22 |
| Reptile | BatMan | 53.2±0.16 | 52.8±0.15 | 52.1±0.15 |
| | Man | 50.8±0.20 | 51.6±0.21 | 51.9±0.20 |
| Eigen-Reptile | BatMan | 51.5±0.22 | 51.2±0.22 | 51.5±0.22 |
| | Man | 48.5±0.20 | 48.7±0.20 | 49.2±0.20 |

and foMAML, under varying levels of label noise. We include Eigen-Reptile specifically as a noise-aware FSL [4].

**Setup**. We consider three data sets in a (5, 5)-FSL setting: Omniglot, CifarFS, and MiniImageNet. We emulate training label noise by adding symmetric uniform random noise to $\mathcal{D}_{\text{train}}$ with 30% and 60% corrupted labels on the underlying datasets, i.e., *prior* to the few-shot tasks are randomly sampled. For instance, for experiments with 60% noise, we randomly select 60% samples of each class in $\mathcal{D}_{\text{train}}$ and assign them to a different class *within the same split* with equal probability. Reported meta-test results are on *clean* data, to evaluate the learners under a base-case scenario. The support set size is 5 (15), and query set size is 15 (Reptile learners), following [6,19]. All experiments ran on machines with 128 GB RAM, dual 16-Core AMD CPUs, and an Nvidia A4000 16GB GPU. We used cross-entropy loss for supervised meta-training and meta-testing. Each learner uses a ConvNet-4 architecture with 64 filters and a linear layer with output dimension $\mathbb{R}^{128}$. On Omniglot, the number of filters was increased to 128. iMAML uses weight decay centered around $\theta_t$ [20] and foMAML+ZO resets its final layer to zero at the beginning of each inner-loop. Learners were trained with the original papers' hyper-parameters, except for the following changes. Eigen-Reptile and Reptile run with 7 inner-loop steps, iMAML with 12 (16 for Omniglot), and foMAML with 5. iMAML's proximal decay was set to 0.5 (2.0 for Omniglot). Each learner was meta-tested after 5K, 15K (10K), 15K (10K), training outer-loop steps (iMAML), respectively, for Omniglot, CifarFS, and MiniImageNet. To augment the samples before Manifold sampling, we use a set of independently random augmentations to each image. On CifarFS and MiniImageNet, we use the augmentations proposed in [1], consisting of: cropping, random horizontal flip, random color jitter, random grayscale, random blurring, and normalization. For Omniglot, we follow the augmentation scheme in [3], applying one of: random crop, affine transform, or perspective transform to the source image. During meta-testing, the task-specific model is fine-tuned for 10 steps (Omniglot and CifarFS) and 20 steps (MiniImageNet). When applying Man-CLR or BatMan-CLR, we keep the same model sizes with the addition of a larger output dimension: $\mathbb{R}^{128}$ rather than $\mathbb{R}^{N}$. The batch size of BatMan is 5 for all inner-loop adaptations, resulting in mini-batching [19] for (Eigen) Reptile, and 15 for the meta-gradient calculation of iMAML and foMAML. For each task's support set, we create five random augmentations for each shot. Each query sample is augmented twice, allowing the inner loop to sample more varied tasks. Results are averaged over 3 runs, using 2048 tasks sampled from $\mathcal{D}_{\text{test}}$.

**Meta-test accuracy**. Table 2 summarizes the BatMan-CLR results on noisy CifarFS, Omniglot, and MiniImageNet with both Man and BatMan under 0% (no), 30% and 60% noisy training labels. One can easily observe that BatMan-CLR clearly strengthens the resilience of all learners to noise across all three datasets. Only marginal decreases in testing accuracy occur under increasing label noise. On Omniglot when encountering the label noise in meta-training, all learners can still learn effective initial models for task adaptation, which for most is around 97% and slightly worse for Eigen-Reptile. All learners display an accuracy under label noise similar to that without label noise. On CifarFS, we observe similar results, where most learners reach a test accuracy between 62–64%, which remains almost constant under increasing noise levels. Finally, even on MiniImageNet, the most difficult dataset of the three, consisting of more diverse classes and larger inputs, the testing accuracy of all meta-learners is limited to drops in the range between 0.5–1 percent points under BatMan-CLR. These results strongly validate the effectiveness of BatMan, which self-cleanses the shots by creating contrastive pairs and ways in batched Man samples. The only exception is Man sampling on Eigen-Reptile on Omniglot and MiniIMageNet, where we observe a minor increase of 0.3–0.7 percent points under 60% noise. We speculate that this is because Eigen-Reptile's meta-gradient approximation is performed by selecting the optimization direction with the highest variance. However, a high noise level introduces a high variance in the optimization di-

Table 3: Meta-test accuracy with 95th confidence intervals ($\overline{acc}\pm_{CI95}$), meta-trained on (5, 5)-FSL tasks with label noise $\epsilon \in \{0.0, 0.3, 0.6\}$ on Omniglot and CifarFS, and different Inner/Outer-loop samplers: Random Manifold (R), and BatMan (B). SSL represents a self-supervised meta-trained model trained without information. Results from Table 2 are highlighted.

(a) Omniglot MAML-like.

| Alg. | $\epsilon$ | B/B | B/R | R/B | R/R |
|---|---|---|---|---|---|
| foMAML | SSL | 94.7±0.11 | | | |
| | 0.0 | 98.2±0.07 | 94.1±0.12 | 98.3±0.05 | 94.5±0.12 |
| | 0.3 | 98.2±0.06 | 96.9±0.09 | 98.4±0.05 | 96.7±0.09 |
| | 0.6 | 98.0±0.07 | 97.9±0.07 | 98.4±0.05 | 98.0±0.07 |
| iMAML | SSL | 97.0±0.09 | | | |
| | 0.0 | 97.5±0.08 | 94.9±0.11 | 98.1±0.07 | 94.5±0.12 |
| | 0.3 | 98.1±0.07 | 96.5±0.10 | 98.3±0.07 | 96.7±0.09 |
| | 0.6 | 98.3±0.06 | 97.8±0.08 | 98.2±0.07 | 97.9±0.07 |

(b) Omniglot Reptile-like.

| Alg. | $\epsilon$ | B/- | R/- |
|---|---|---|---|
| Reptile | SSL | 96.0±0.10 | |
| | 0.0 | 97.9±0.07 | 65.3±0.29 |
| | 0.3 | 97.3±0.08 | 69.5±0.27 |
| | 0.6 | 96.2±0.10 | 73.1±0.27 |
| Eigen-Reptile | SSL | 95.1±0.13 | |
| | 0.0 | 92.6±0.13 | 82.5±0.21 |
| | 0.3 | 93.0±0.12 | 71.9±0.27 |
| | 0.6 | 93.2±0.12 | 73.5±0.26 |

(c) CifarFS MAML-like.

| Alg. | $\epsilon$ | B/B | B/R | R/B | R/R |
|---|---|---|---|---|---|
| foMAML | SSL | 52.8±0.16 | | | |
| | 0.0 | 66.6±0.17 | 58.4±0.24 | 62.0±0.23 | 58.0±0.24 |
| | 0.3 | 65.2±0.17 | 59.7±0.23 | 61.8±0.24 | 59.7±0.23 |
| | 0.6 | 64.8±0.16 | 60.5±0.24 | 61.1±0.24 | 60.5±0.24 |
| iMAML | SSL | 54.5±0.24 | | | |
| | 0.0 | 64.2±0.19 | 58.0±0.24 | 60.9±0.30 | 56.9±0.29 |
| | 0.3 | 62.7±0.25 | 59.2±0.24 | 60.3±0.23 | 58.3±0.29 |
| | 0.6 | 62.9±0.20 | 59.7±0.24 | 60.3±0.29 | 60.0±0.29 |

(d) CifarFS Reptile-like.

| Alg. | $\epsilon$ | B/- | R/- |
|---|---|---|---|
| Reptile | SSL | 55.0±0.17 | |
| | 0.0 | 66.5±0.17 | 53.9±0.22 |
| | 0.3 | 65.0±0.17 | 55.3±0.22 |
| | 0.6 | 64.1±0.17 | 56.4±0.22 |
| Eigen-Reptile | SSL | 54.5±0.16 | |
| | 0.0 | 66.3±0.17 | 57.2±0.27 |
| | 0.3 | 64.4±0.17 | 57.8±0.23 |
| | 0.6 | 63.8±0.18 | 58.8±0.23 |

rections, making it harder to select an appropriate direction even with the use of Man. By employing the less noisy BatMan sampling strategy, the learner has higher chances to see more diverse shots and can better select an optimization direction, achieving performance closer to Reptile with Man sampling.

In terms of comparison between BatMan and Man, there is a visible advantage in using BatMan, especially on the more challenging MiniImageNet and CifarFS. This suggests that taking steps with more information, as in BatMan, provides greater benefits than taking a larger number of simpler steps, as in Man. Zooming into the performance of different learners on CifarFS and MiniImageNet, the difference in testing accuracy between Man and BatMan is smaller with foMAML and iMAML, compared to Reptile and Eigen-Reptile. This can be explained by the fact that in our experiments, the MAML style learners use BatMan to calculate the meta-gradient, resulting in more informative meta-updates.

Reptile-style learners do not calculate their meta-gradients using query data but directly using the inner-optimization *direction*.

**Ablations**. We consider three types of ablation studies using the Omniglot and CifarFS (and MiniImageNet) datasets. First, we consider the impact of supervised task generation by training meta-learners in a self-supervised learning (SSL) setting. Similar to UMTRA [9] and CACTUS [7], we construct $(5, 5/15)$-FSL tasks (MAML/Reptile) by drawing 5 random images from $\mathcal{D}_{\text{train}}$. To construct the required shots, $K + Q$ augmentations are created and divided into support and query sets, with $|\mathcal{D}_{\text{support}}| = K$ and $|\mathcal{D}_{\text{query}}| = Q$. Table 3 provides the results in the rows indicated with SSL. We use the same hyper-parameters and loss function as for BatMan-CLR, with the meta-batch size increased to 25 (from 5), so that all learners see a comparable number of expected unique ways per meta-update. Although this approach shows similar performance to BatMan-CLR on Omniglot (see Table 3a and b), on CifarFS (see Table 3c and d) there is a considerable gap of 11–13.8 percent points compared to BatMan-CLR (gray columns). Indicating that BatMan-CLR benefits from performing a joint optimization of more ways and shots in the inner-loop.

Second, we replace BatMan with a random manifold batch sampler (Rand) to investigate the impact of the BatMan sampling strategy. This Rand-sampler differs from BatMan by *uniformly sampling from* the FSL task, i.e., allowing multiple instances from the same way to be selected in a single random manifold. We pair Rand (R) with BatMan (B) sampling in different configurations for the inner and outer-loop, marked as (inner)/(outer) in the column names in Table 3a and c. Reptile and Eigen-Reptile only use a support set during meta-training, so we only replace their inner-sampling strategy (see Table 3b and d). We keep the same hyper-parameters as used in the corresponding BatMan-CLR setting. In general, the learners trained with random sampling in the outer-loop show an increased accuracy as the noise level increases. Learners show an increase in accuracy of around 2–8% and 2–3% on Omniglot and CifarFS, respectively, comparing the $\epsilon = 0$ (no) and $\epsilon = 0.6$ noise levels, whereas BatMan sees a slight drop. Even so, it stays ahead of Rand across the board. This shows that BatMan has the capability to self-clean. An interesting exception is Omniglot combined with Eigen-Reptile when increasing the noise level from 0 to 0.3 (Table 3b). This is expected as higher noise levels increase the *expected* number of unique ground-truth classes in a task, yielding fewer false negatives in each Rand manifold during contrastive learning. Replacing only the inner or outer-loop sampler for iMAML and foMAML with Rand, we see that the contribution of the inner-loop is less significant than the outer-loop. This shows that BatMan-CLR is also an effective strategy when replacing only the outer-loop (R/B).

Lastly, we consider the impact of the meta-testing label noise on meta-learned supervised and BatMan-CLR models. While meta-train label noise exists at the dataset level, meta-testing lies at the task level. Task-level noise is added by corrupting an original $(5, 5)$-FSL problem, where a fraction ($\epsilon_{test}$) of shots from each way within the FSL problem is remapped to a different class. As such, the meta-test error models class confusion—e.g., for an $(3, *)$-FSL problem a fraction

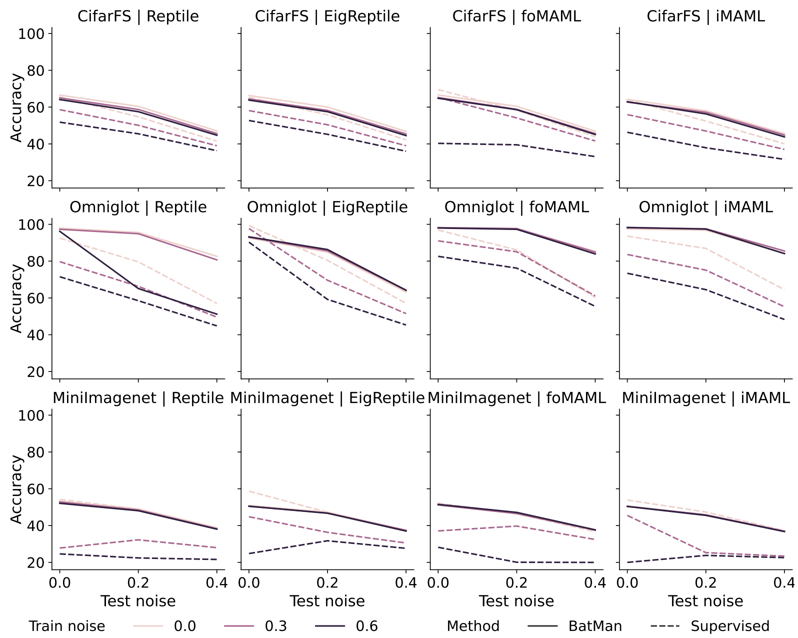


Fig. 2: Meta-test accuracy of supervised and BatMan-CLR meta-trained models with different meta-learners, training, and testing noise levels. Results shown meta-trained with dataset-level label noise ($\epsilon \in \{0.0, 0.3, 0.6\}$) and meta-testing label noise ($\epsilon \in \{0.0, 0.2, 0.4\}$) at task-level.

($\epsilon_t est$) of class 'car' gets re-labeled as 'truck', 'truck' to 'plane', and 'plane' to 'car'. We consider different noise levels during both meta-training ($\epsilon_{train} \in \{0.0, 0.3, 0.6\}$), and meta-testing ($\epsilon_{test} \in \{0.0, 0.2, 0.4\}$). Here, we evaluate the impact of meta test-noise under supervised and BatMan-CLR. In Figure 2, we show the accuracy curves of the learners under varying noise configurations.

Generally, in Figure 2, we observe a negative trend across all configurations as both meta-train and meta-test noise increase. Additionally, the impact of meta-training noise is more pronounced in the baseline learners, compared against the BatMan-CLR learned results—consistent with the results from Table 1 and Table 2. Moreover, this performance difference remains stable as testing noise increases, resulting in a larger vertical spread for the baseline results than for the BatMan-CLR results, corresponding with improved robustness against training label noise. An exception to this is the performance of Reptile paired with BatMan-CLR with an $\epsilon_{train} = 0.6$, where the learner's meta-testing accuracy degrades around the level of its baseline counterpart with training noise of $\epsilon = 0.3$.

On the CifarFS and (Omniglot) datasets—top and (middle) row in Figure 2—the BatMan-CLR meta-learned models showcase a reduced sensitivity to testing noise compared to their supervised counterparts. We see that the BatMan-CLR meta-learned models at $\epsilon_{test} = 0.2$ have an accuracy loss reduction of 2.1–5.7% (4.5–16.0%) and at $\epsilon_{test}$ of 4.6–5.5% (6.1–25.6%) compared to the supervised learned models. Nonetheless, the BatMan-CLR learners see a degradation as the meta-testing noise increases, of 5.5–6.6% (0.5–31.1%) at $\epsilon = 0.2$ and 18.6–19.7% (12.6–45.0%) with $\epsilon = 0.4$. MiniImagenet results (bottom row in Figure 2) show that the BatMan-CLR meta-learned models under test noise perform similarly to the cleanly pre-trained meta-learner. Whereas the supervised baseline models with $\epsilon_{train} > 0$ on MiniImagenet show a drastic deterioration in performance, the BatMan-CLR trained models perform on-par with the baseline trained without meta-training noise.

## 5 Conclusion

Motivated by the ubiquitous presence of label noise, we empirically unveil the impact of label noise on existing few-shot meta-learners, with a particular focus on noise in meta-training. As the number of shots per way is low, the label noise can be exceedingly detrimental to meta-learners and highly challenging to address. To enhance the resilience against label noise for few-shot learners, we propose BatMan—a generic approach that turns supervised few-shot tasks into semi-supervised ones. BatMan is capable of self-cleansing noisy $N$-way $K$-shots instances by (i) batch manifold sampling that re-constructs $N$-way 2-contrastive-shots via augmentation and (ii) introducing the DCL [27] contrastive loss. Our results on three datasets, Omniglot, CifarFS, and MiniImageNet, show that BatMan can maintain the effectiveness of few-shot learners independent of noise levels, i.e., recouping up to 30% accuracy degradation (20% on average under 60% noise).

As future work, we aim to explore further (label noisy) meta-testing paired with BatMan-CLR and adding class awareness [9,22]. Herein, the impact of alternative augmentation strategies can also be explored. Further exploring the utility of BatMan-CLR under the meta-testing setting would be valuable, as well as considering the incorporation of other loss functions, such as ProtoCLR [17]. Lastly, we leave the consideration of other types of noise for future work, such as out-of-domain noise, asymmetric noise, or task-level corruption [26].

## Acknowledgments

This research is part of the Priv-GSyn project, 200021E_229204 of Swiss National Science Foundation and the DEPMAT project, P20-22 / N21022, of the research programme Perspectief, which is partly financed by the Dutch Research Council (NWO). This research was partly funded by the Spoke "FutureHPC & BigData" of the ICSC-Centro Nazionale di Ricerca in "High Performance Computing, Big Data and Quantum Computing", funded by the European Union - NextGenerationEU, and by the DYMAN project funded by the European Union - European Innovation Council under G.A. n. 101161930.

## References


1. Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning Representations by Maximizing Mutual Information Across Views. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., D'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. pp. 15509–15519 (2019)
2. Bertinetto, L., Torr, P.H., Henriques, J., Vedaldi, A.: Meta-learning with differentiable closed-form solvers. In: 7th International Conference on Learning Representations, ICLR 2019 (2019)
3. Boutin, V., Singhal, L., Thomas, X., Serre, T.: Diversity vs. Recognizability: Human-like generalization in one-shot generative models. ArXiv (2022)
4. Chen, D., Wu, L., Tang, S., Yun, X., Long, B., Zhuang, Y.: Robust Meta-learning with Sampling Noise and Label Noise via Eigen-Reptile. In: Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S. (eds.) Proceedings of the 39th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 162, pp. 3662–3678. PMLR (2022)
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: 37th International Conference on Machine Learning, ICML 2020. vol. PartF16814, pp. 1575–1585 (2020)
6. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. 34th International Conference on Machine Learning, ICML 2017 **3**, 1856–1868 (2017)
7. Hsu, K., Levine, S., Finn, C.: Unsupervised Learning via Meta-Learning. In: 7th International Conference on Learning Representations, {ICLR} 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net (1 2019)
8. Kao, C.H., Chiu, W.C., Chen, P.Y.: MAML is a Noisy Contrastive Learner in Classification. The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022 (2021)
9. Khodadadeh, S., Bölöni, L., Shah, M.: Unsupervised Meta-Learning for Few-Shot Image Classification. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., D'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. vol. 32, pp. 10132–10142 (2019)
10. Killamsetty, K., Li, C., Zhao, C., Chen, F., Iyer, R.: A Nested Bi-level Optimization Framework for Robust Few Shot Learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 7176–7184. No. 7, OpenReview.net (2022)
11. Lake, B.M., Salakhutdinov, R., Gross, J., Tenenbaum, J.B.: One shot learning of simple visual concepts. Proceedings of the Annual Meeting of the Cognitive Science Society **33**(33) (2011)
12. Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B.: Human-level concept learning through probabilistic program induction. Science **350**(6266), 1332–1338 (12 2015)
13. Li, M., Soltanolkotabi, M., Oymak, S.: Gradient Descent with Early Stopping is Provably Robust to Label Noise for Overparameterized Neural Networks. In: Chiappa, S., Calandra, R. (eds.) The 23rd International Conference on Artificial Intelligence and Statistics, {AISTATS} 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]. Proceedings of Machine Learning Research, vol. 108, pp. 4313–4324. PMLR (2020)

14. Liang, K.J., Rangrej, S.B., Petrovic, V., Hassner, T.: Few-shot Learning with Noisy Labels. CVPR (2022)
15. Lu, J., Jin, S., Liang, J., Zhang, C.: Robust few-shot learning for user-provided data. IEEE Transactions on Neural Networks and Learning Systems **32**(4), 1433–1447 (4 2021)
16. Mazumder, P., Singh, P., Namboodiri, V.P.: RNNP: A Robust Few-Shot Learning Approach. In: IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021. pp. 2663–2672. IEEE (2021)
17. Medina, C., Devos, A., Grossglauser, M.: Self-Supervised Prototypical Transfer Learning for Few-Shot Classification. CoRR (2020)
18. Muhammad Abdullah Jamal, Wang, L., Gong, B.: A Lazy Approach to Long-Horizon Gradient-Based Meta-Learning. International Conference on Computer Vision pp. 6577–6586 (2021)
19. Nichol, A., Achiam, J., Schulman, J.: On First-Order Meta-Learning Algorithms. CoRR pp. 1–15 (2018)
20. Rajeswaran, A., Finn, C., Kakade, S.M., Levine, S.: Meta-Learning with Implicit Gradients. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., D'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. pp. 113–124. No. NeurIPS (2019)
21. Schmidhuber, J.: Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook. Diploma thesis, Inst. f. Inf., Tech. Univ. Munich (1987)
22. Shirekar, O.K., Singh, A., Rad, H.J.: Self-Attention Message Passing for Contrastive Few-Shot Learning. In: {IEEE/CVF} Winter Conference on Applications of Computer Vision, {WACV} 2023, Waikoloa, HI, USA, January 2-7, 2023. pp. 5414–5425. IEEE (10 2023)
23. Song, H., Kim, M., Park, D., Shin, Y., Lee, J.G.: Learning from noisy labels with deep neural networks: A survey. IEEE Transactions on Neural Networks and Learning Systems pp. 1–19 (2022)
24. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching networks for one shot learning. In: Advances in Neural Information Processing Systems. vol. 29, pp. 3637–3645 (2016)
25. Wang, Z., Hu, G., Hu, Q.: Training Noise-Robust Deep Neural Networks via Meta-Learning. In: 2020 {IEEE/CVF} Conference on Computer Vision and Pattern Recognition, {CVPR} 2020, Seattle, WA, USA, June 13-19, 2020. pp. 4523–4532. Computer Vision Foundation / {IEEE} (2020)
26. Yao, H., Wang, Y., Wei, Y., Zhao, P., Mahdavi, M., Lian, D., Finn, C.: Meta-learning with an Adaptive Task Scheduler. In: Advances in Neural Information Processing Systems. vol. 9, pp. 7497–7509 (2021)
27. Yeh, C.H., Hong, C.Y., Hsu, Y.C., Liu, T.L., Chen, Y., LeCun, Y.: Decoupled Contrastive Learning. In: Avidan, S., Brostow, G.J., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision - {ECCV} 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part {XXVI}. Lecture Notes in Computer Science, vol. 13686, pp. 668–684. Springer (2022)
28. Yu, X., Liu, T., Gong, M., Zhang, K., Batmanghelich, K., Tao, D.: Label-Noise Robust Domain Adaptation. In: Proceedings of the 37th International Conference on Machine Learning, {ICML} 2020, 13-18 July 2020, Virtual Event. Proceedings of Machine Learning Research, vol. 119, pp. 10913–10924. PMLR (2020)