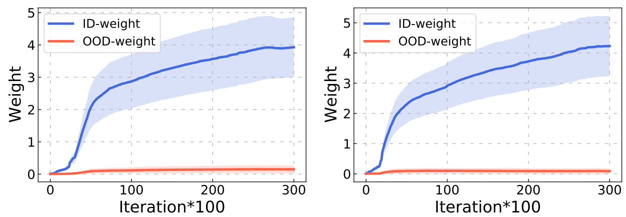
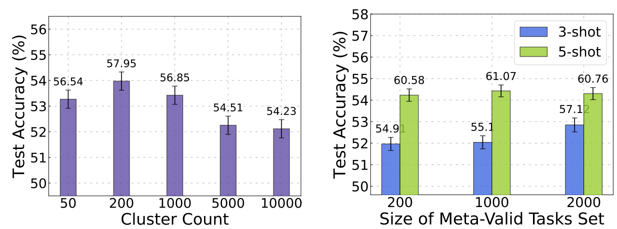
Noise Ratio	5-way 3-shot 20%	5-way 3-shot 30%	5-way 3-shot 50%	5-way 5-shot 20%	5-way 5-shot 30%	5-way 5-shot 50%
MAML-Noise-RM	60.2±0.02	59.35±0.01	58.21±0.71	61.2±0.21	60.3±0.32	59.1±0.68
MAML	54.8±0.64	53.9±1.10	51.8±0.12	59.2±0.28	57.6±0.36	53.5±0.48
NESTEDMAML (ours)	**55.24**±0.72	**54.7**±1.20	**53.68**±0.21	**59.6**±0.54	**58.16**±0.87	**55.61**±1.32

Table 3: Test accuracies on ***mini*-Imagenet** with 20%, 30%, and 50% flipped noisy labels during the meta-training phase.



(a) 5-way 3-shot (b) 5-way 5-shot

Figure 4: Weights trend as the iterations progress for 30% SVHN OOD experiment



(a) FashionMNIST (90%) (b) SVHN (30%)

Figure 5: (a) shows accuracies under 90% FashionMNIST OOD level with different cluster values, (b) shows accuracies under 30% SVHN OOD level with different sizes of meta-validation tasks set.

the effectiveness of the NESTEDMAML.

Instance-level Weighting For Noisy Labels

Similar to OOD experiments, we implement 5-way 3-shot (5-shot) experiments to evaluate the instance-level weighting scheme. We conduct experiments on noisy labels generated by randomly corrupting the original labels in ***mini*-ImageNet**. Specifically, different percentages (20%,30%, 50%) of training samples are selected randomly to flip their labels to simulate the noisy corrupted samples. Intuitively, a deep model robust to noise tries to ignore the data with noisy labels. Note that data containing noisy labels only exist in the meta-training stage. Hyper-parameters are shown in Appendix **??**.

Baselines. We compare our NESTEDMAML with the following baselines: (1) **MAML-Noise-RM** serves as a skyline. It is simply modified from MAML, and we manually fix zero weights to instances with noisy labels. (2) **MAML**.

Results. From the results shown in Table 3, we can conclude that NESTEDMAML performs better than MAML with high accuracies. Furthermore, to circumvent overfitting and reduce computational complexity due to the weight matrix's high dimension, we group instance weights with 200 clusters by K-means, where instances in each cluster share the same weight initialized at 0.005.

Sensitivity Analysis

We perform an ablation study to determine how the number of hyper-parameters and meta-validation sets' size can affect the NESTEDMAML algorithm's performance. To that extent, we evaluate the NESTEDMAML algorithm's performance using a different number of clusters in a 5-way 5-shot 90% FashionMNIST OOD setting. Figure 5a shows test accuracies versus different numbers of clusters. We observed the best performance when the cluster count is 200. From Figure (5a), it is evident that the test accuracy decreases with an increase in the number of clusters that need to be determined. Contrarily, using a tiny number of clusters will also decrease the performance due to decreased clustering efficiency. We used 200 clusters for all our experiments. We also evaluate NESTEDMAML algorithm's performance using different sizes of the meta-validation set in 5-way 3-shot (5-shot) 30% SVHN OOD setting. Figure (5b) shows that NESTEDMAML algorithm performs well even when the meta-validation set size is tiny(i.e., 1% of meta-training set).

Conclusion

We propose a novel robust meta-learning algorithm for reweighting tasks/instances of corrupted data in the meta-training phase. Our method is model-agnostic, can be directly applied to any deep learning architecture in an end-to-end manner. To the best of our knowledge, NESTEDMAML is the first algorithm to solve a *nested bi-level* optimization problem in an online manner with a convergence result. Finally, empirical evaluation results in OOD task and noisy label scenarios show that NESTEDMAML outperforms state-of-the-art meta-learning methods by efficiently mitigating the effects of unwanted instances or tasks.

Repellat voluptates unde placeat id provident, magnam adipisci repellat modi quaerat earum, voluptates adipisci placeat tempora odit, est in ab, cupiditate tempore magni odit commodi excepturi repellendus natus perspiciatis quasi voluptatum?Velit nisi placeat laudantium ipsam, hic eius libero fugiat iusto tenetur beatae sint at, amet quos ipsa atque veniam.Quae odio eum earum praesentium cum voluptatum voluptas, in omnis dolores at iure eos aspernatur cupiditate, commodi eveniet odit deleniti ipsam possimus consectetur laudantium sit quasi molestiae suscipit, est minus consectetur dolore eaque odit quidem aspernatur asperiores?Impedit quo ex dolorum sunt ab placeat dolore nam rerum deleniti provident, minima molestiae eaque aut provident impedit necessitatibus magnam quis pariatur repudiandae odit?Veritatis corporis quis a at exercitationem fugiat quasi, alias vero maiores quibusdam tempora delectus assumenda tenetur voluptatibus magnam reiciendis, sit sint voluptatibus facere delectus minima temporibus, a quisquam dolorum harum consectetur cupiditate accusamus similique, tempora nesciunt pariatur soluta.Consequuntur velit labore pariatur vero distinctio, amet labore praesentium dicta laboriosam quae quibusdam deleniti, consequatur blanditiis dolorum reiciendis repudiandae incidunt minima illo ducimus.Rerum illum laudantium doloremque totam, quae non nisi quo, vitae sequi ea in obcaecati, repudiandae suscipit