# Understanding deep meta-learning

Huisman, M.

# Chapter 7

# Conclusions

**Chapter overview**

In this dissertation, we have investigated the field of deep meta-learning and focused on addressing several research questions related to the performance and understanding of deep meta-learning algorithms with the aim of extracting knowledge about deep meta-learning algorithms beyond simple performance comparisons (method A outperforms method B). In addition, we have investigated how theoretical principles can be used to improve the performance of deep meta-learning algorithms. In this chapter, we revisit and answer the research questions listed in Chapter 1, discuss the implications and limitations of our work, and propose several fruitful directions for future research.

## 7.1 Answers to research questions

**RQ1:** What causes the performance gap between MAML and the meta-learner LSTM?

For the investigation of this research question, we have proposed a simplified version of the meta-learner LSTM called TURTLE, by using a feedforward neural network instead of an LSTM and raw inputs rather than preprocessed inputs. When using a first-order approximation of the meta-gradients in a similar fashion to the original meta-learner LSTM, TURTLE performs on par with the meta-learner LSTM. When using second-order gradients, however, TURTLE systematically yields better performance. Moreover, we have shown that by enhancing the meta-learner LSTM with the same inputs as TURTLE and second-order gradients, the observed performance gap to MAML was closed. This shows that the answer to this research question is the fact that the original meta-learner LSTM used processed inputs and a first-order approximation of the meta-gradients to perform meta-updates rather than the exact meta-gradients, which contain second-order effects.

Interestingly, the effect of making first-order approximations is different for MAML compared with the meta-learner LSTM and TURTLE. For MAML, it was shown by Finn et al. (2017) that using first-order approximations of the meta-gradients yields comparable performance compared with the exact second-order gradients. For the other two methods, however, this is not the case, suggesting that learning a weight update procedure (as done by TURTLE and the meta-learner LSTM) introduces more curvature into the meta-loss landscape. Nonetheless, learning a weight update procedure can be beneficial for performance, as also noted by Andrychowicz et al. (2016).

Lastly, we note that whilst we have managed to close the observed performance gap by using second-order gradients, this comes at the cost of increased computational cost as there are $\mathrm{O}(N^2)$ second-order gradients to be computed, where $N$ is the number of parameters of the neural network. This can be intractable for larger neural networks with millions of parameters.

**RQ2:** How do the learning behaviors of finetuning, MAML, and Reptile differ from each other and how does this influence their ability to quickly learn to perform new tasks?

These techniques have distinct optimization objectives and exhibit varying behaviors, which in turn affect their ability to learn new tasks quickly.

In the research conducted in Chapter 4, the optimization objectives of these techniques were interpreted as maximizing different performance aspects. Finetuning aims to maximize direct performance with the current set of parameters. It focuses on improving performance without explicitly considering future adaptation steps. On the other hand, MAML aims to maximize performance after a few adaptation steps, making it a look-ahead objective. It searches for an initialization that may not yield the best immediate performance but leads to promising results after a few gradient update steps. Reptile combines both direct performance and performance after each update step, striking a balance between the two extremes. As a result of these differing objectives, finetuning tends to favor an initialization that jointly minimizes the loss function, emphasizing immediate performance. In contrast, MAML may settle for an inferior initialization if it shows potential for improvement after adaptation steps. Reptile finds a middle ground, considering both initial performance and subsequent adaptation.

Furthermore, our research has demonstrated that MAML and Reptile exhibit a specialization for adaptation in low-data scenarios within the distribution of training tasks. This specialization is facilitated by two key factors: the weights of the output layer and the scarcity of data in the training tasks. These factors allow MAML and Reptile to explicitly specialize for few-shot learning, giving them an advantage over finetuning when the test tasks are similar to those seen at training time. It is worth noting finetuning cannot specialize for few-shot learning using these two factors, as it begins with a random output layer when learning to perform a new task, and the finetuning method is not pre-trained on simulated low-data tasks.

Our results further show that finetuning learns a broad and diverse set of features that allows it to discriminate between many different classes. MAML and Reptile, in contrast, optimize a look-ahead objective and settle for a less diverse and broad feature space as long as it facilitates robust adaptation in low-data regimes. This can explain findings by Chen et al. (2019), who show that finetuning can yield superior few-shot learning performance when tasks become distant from the training tasks, which is further supported by the fact that there are statistically significant relationships between the broadness of the learned features and the few-shot learning ability for finetuning.

Another result is that MAML yields the best few-shot learning performance when using a shallow convolutional backbone. Interestingly, the features learned by MAML become less discriminative as the depth of the backbone increases. This may indicate an over-specialization. We did not observe such a phenomenon for the finetuning method.

**RQ3:** Are LSTMs good few-shot learners when evaluated on modern benchmarks?

In Chapter 5, we revisited the plain LSTM approach–originally proposed by Hochreiter et al. (2001) and Younger et al. (2001)–and analyzed it from a few-shot learning perspective. The main idea of this approach is that given a new task, we can enter the entire training dataset into the LSTM, and afterward enter query inputs for which we want to obtain predictions. This plain LSTM approach thus ingests training data for a specific task and conditions predictions for new query inputs on the resulting hidden state. However, as we have argued, there are two potential issues associated with this approach:

1. The hidden embeddings of the support set are not permutation invariant.
2. The learning algorithm and the input embedding mechanism are intertwined, leading to a challenging optimization problem and an increased risk of overfitting.

To address the first issue, mean pooling is proposed to make the embeddings permutation invariant. This method significantly improves the performance of the plain LSTM on few-shot sine wave regression and image classification tasks. It even outperforms the popular meta-learning method MAML on the sine wave regression problem. However, it still struggles to achieve good performance on few-shot image classification tasks, highlighting the optimization difficulties of the plain LSTM approach.

To overcome these difficulties, a new technique called Outer Product LSTM (OP-LSTM) is proposed. OP-LSTM uses an LSTM to update the weights of a base-learner network, effectively decoupling the learning algorithm from the input representation mechanism. This resolves the second issue and allows for more effective optimization. The theoretical analysis shows that OP-LSTM can perform an approximate form of gradient descent (similar to MAML) and a nearest prototype-based approach (similar to Prototypical Networks), demonstrating its flexibility and expressiveness. Empirically, we demonstrate that OP-LSTM overcomes the optimization issues associated with the plain LSTM approach in few-shot image classification benchmarks while using fewer parameters. It achieves competitive or superior performance compared to MAML and Prototypical Networks, which it can approximate.

In summary, while the plain LSTM approach initially had limitations, improvements such as mean pooling and the development of OP-LSTM have made LSTMs more effective as few-shot learners, yielding competitive performance in various few-shot learning tasks.

**RQ4:** Can the few-shot learning ability of deep neural networks be improved by learning which subsets of parameters to adjust?

In Chapter 6, we proposed a new deep meta-learning algorithm called *Subspace Adaptation Prior* (SAP). SAP jointly learns a good neural network initialization and good parameter subspaces (or subsets of operations) in which new tasks can be learned within a few gradient descent updates from a few data. That is, instead of adapting all parameters when learning a new task, which may be suboptimal and may lead to overfitting during few-shot learning, SAP learns which parameters to adjust when adapting to new tasks.

Our experiments show that SAP outperforms similar existing gradient-based meta-learners in few-shot sine wave regression, yields better performance in single-domain few-shot image classification settings, and yields competitive or superior performance in cross-domain few-shot image classification, where the test tasks are more distant to the training tasks. This highlights the advantage of learning suitable subspaces in which to perform gradient descent when learning new tasks.

This could be due to the regularization effect of not having to adjust all parameters as well as due to the ability to match structures inherently present in task families. Our experiments in Section 6.5.3 on synthetic task families demonstrate that the SAP is able to learn operations that match the task structure in simple settings in 75% of the cases. In other cases, it may compensate by using other operations that are not inherently present in the task structure.

Inspection of the subspace activation strengths in few-shot image classification reveals that simple and low-dimensional operations, such as shifting features by a single scalar or element-wise by a vector, are important. This is in line with recent work and findings (Triantafillou et al., 2021; Requeima et al., 2019; Bateni et al., 2020) which show that adapting pre-trained embeddings by means of such low-dimensional transformations, such as FiLM layers (Perez et al., 2018), can yield excellent performance.

One limitation of SAP is that it requires the computation of second-order gradients by default during meta-training in order to update the initialization parameters, in a similar fashion as other gradient-based meta-learners such as MAML (Finn et al., 2017), (M)T-Net (Lee and Choi, 2018), and Warp-MAML (Flennerhag et al., 2020). This limitation can be bypassed by using a first-order approximation, which comes at the cost of a performance penalty (between 0.2% and 7.3% accuracy in our experiments).

Gradient-based meta-learning methods struggle to scale well to deep networks as recent work suggests that simple pre-training and fine-tuning of the output layer (Tian et al., 2020; Chen et al., 2021; Huisman et al., 2021) can yield superior performance on common few-shot image classification benchmarks. This is also the reason, besides searching for energy-efficient few-shot learners, that in our experiments we focus on relatively shallow backbones that adapt all layers when learning new tasks, instead of only the output layer. Other limitations are that SAP introduces more parameters and that the candidate pools of operations are selected by hand, despite the fact that these operations are general.

## 7.2 Future work

This dissertation represents significant progress in advancing our understanding of deep meta-learning algorithms beyond mere performance evaluations. However, with each step forward, we uncover a multitude of unanswered questions. Consequently, our research opens up numerous promising avenues for future exploration, aimed at delving even deeper into the inner workings of deep meta-learning algorithms, which, in turn, can lead to enhanced performance. In addition to the future work directions mentioned in the individual research chapters, below we describe future research directions that we think are most fruitful based on a high-level view of this entire dissertation.

**Quantification of task distances** A popular criticism of deep meta-learning methods is that their few-shot learning performance is often evaluated on tasks that are quite similar to the tasks seen a training time (within-distribution) (Chen et al., 2019; Triantafillou et al., 2020; Ullah et al., 2022), thereby giving an overly optimistic picture of their ability to generalize from limited data. The reason for this is that the test tasks are often sampled from the same dataset from which the training tasks were sampled. This critique has sparked interest in evaluating deep meta-learning techniques on test tasks sampled from different datasets than those used for training (out-of-distribution), which is also what we have done throughout Chapter 3, Chapter 4, Chapter 5, and Chapter 6. While this is an important step forward, we argue that it would be even more beneficial to use a measure that quantifies how similar a given test task is compared to the training task distribution, which is something that has also been studied in the field of algorithm selection (Brazdil et al., 2022; Peng et al., 2002). This would allow us to perform a deeper investigation into the inner workings of deep meta-learning algorithms and evaluate their performance as a function of the novelty of tasks. This increased understanding could translate itself into new algorithms or training objectives and improved performance in more challenging scenarios, where the test tasks deviate from the observed training tasks.

**Deep meta-learning on different data modalities** One of the limitations of the present dissertation lies in the fact that our results were solely obtained on few-shot image classification benchmarks. Recently, the field has also moved towards testing meta-learning algorithms on different data modalities such as text (Lee et al., 2022; Sun et al., 2021), audio (Heggan et al., 2022; Shi et al., 2020), and video (Alet et al., 2021; Wang et al., 2020a). We think that it is an important direction for future work to explore the impact of the data modality, such as images, text, video, and speech, on the performance and behavior of deep meta-learning algorithms. Investigating this promising direction of future work will enable a more comprehensive understanding of how different types of data influence the effectiveness and adaptability of meta-learning techniques. By delving into these unexplored territories, we can uncover potential variations in the capabilities and limitations of meta-learning algorithms across various modalities, paving the way for more informed and tailored approaches in future research.

**Towards more general deep meta-learning** In Chapter 5 we have revisited the idea of using an LSTM as a general meta-learning algorithm (Hochreiter et al., 2001; Younger et al., 2001) that is fed the training data and can make predictions on new inputs conditioned on the resulting hidden state. A problem with this approach is that the learning algorithm and the input representations are intertwined, which can render the optimization difficult and make the approach susceptible to overfitting. That is, when applying this approach to image data, the LSTM module (the learning algorithm) is applied after a convolutional feature extractor (the input representation). Normally, however, the input representation would be directly fed into a linear layer. By feeding the input representation into an LSTM module, which can consist of multiple stacked LSTMs, the input representation is passed through multiple nonlinearities that could lead to overfitting. We solved this issue by decoupling the input representation from the learning algorithm by having the LSTM module parameterize an outer-product weight update function. However, one could also argue that the main problem is not the intertwinement of the learning algorithm and input representation, but rather the fact that the learning algorithm is applied after the input representation has been computed.

An alternative would be to intertwine the learning algorithm with the input representation from the lowest level of abstraction to the computation of the prediction. This could be implemented through a convolutional LSTM (convLSTM) (Shi et al., 2015) that maintains a state also in the convolutional layers. This can facilitate learning at lower levels of input representation. It is also possible to investigate other architectures for this purpose such as general transformer architectures. Given the success of other prompt-based models, we think that this approach, whilst computationally expensive, could lead to novel learning algorithms and improved few-shot learning performance.

**Understanding black-box learning algorithms** While the general-purpose meta-learning algorithms belonging to the black-box/model-based category of deep meta-learning techniques have the potential to improve the few-shot learning capabilities of deep neural networks, it is difficult to interpret their inner workings (how they adapt to new tasks). Future work can aim to address this interpretability challenge, and allow us to gain valuable insights that, in turn, can enable us to develop new hand-crafted learning algorithms. That is, by studying the behavior of these black-box/meta-based meta-learning models, we can uncover novel learning strategies, principles, or heuristics that could be employed in the development of even more efficient interpretable learning algorithms. Increasing our understanding of black-box models can also help bridge the gap between cutting-edge deep learning techniques and traditional, interpretable approaches. Additionally, by unraveling the inner workings of these powerful meta-learning algorithms, we can enhance their trustworthiness and facilitate the integration of these models into real-world applications where explainability and interpretability are essential. Overall, addressing the challenges associated with interpretability not only contributes to the advancement of deep meta-learning techniques but also opens up new avenues to assisting human researchers in designing new learning algorithms.