On the Objective Landscapes and Generalization in Gradient-Based Meta-Learning

Anonymous Authors[1]

Abstract

In this work we empirically study generalization of neural networks in gradient-based meta-learning, by analyzing various properties of its objective landscapes, focusing on few-shot image classification. We experimentally demonstrate that coherence of meta-test gradients evaluated at meta-train solutions, measured by their average inner product, is correlated with generalization. Furthermore, we provide empirical evidence that generalization to new tasks is also correlated with the coherence between their adaptation trajectories in parameter space, measured by the average cosine similarity between adaptation trajectory directions, starting from a same meta-train solution. Lastly, we show that as meta-training progresses, the meta-test solutions obtained by adapting the meta-train solution of the model to new tasks via few steps of gradient-based fine-tuning, become flatter. However, we observe that the meta-test solution keeps getting flatter even after meta-overfitting, and can't affirm that flatness of minima might be well correlated to generalization in gradient-based meta-learning.

1. Introduction

To address the problem of the few-shot learning, many meta-learning approaches have been proposed recently (Finn et al., 2017), (Ravi & Larochelle, 2017), (Rothfuss et al., 2018), (Oreshkin et al., 2018) and (Snell et al., 2017) among others. In this work, we take steps towards understanding the characteristics of the landscapes of the loss functions, and their relation to generalization, in the context of gradient-based few-shot meta-learning. While we are interested in understanding the properties of optimization landscapes that are linked to generalization in gradient-based meta-learning in general, we focus our experimental work here within a setup that follows the recently proposed Model Agnostic Meta-Learning (MAML) algorithm (Finn et al., 2017). The MAML algorithm is a good candidate for studying gradient-based meta-learning because of its independence from the underlying network architecture, and because of its reasonable success as a few-shot image classification algorithm.

Our main insights and contributions can be summarized as follows:

1. In Section 5.3 we empirically observe that generalization to new tasks is correlated with the coherence between meta-test gradients, measured by the average inner product between meta-test gradient vectors evaluated at meta-train solutions. We show that this metric is also correlated to generalization in few-shot regression tasks.

2. In an attempt to provide an intuitive explanation for the correlation between generalization to new tasks and the similarity of meta-test gradients in inner product, we suggest that MAML, in the standard case of few-shot image classification, might be learning a representation space based on the inner product. We provide empirical evidence showing the correlation between the average inner product between the representation vectors, produced by the model at meta-train solution, for the meta-test data taken as input, and the ability of the model to generalize to the meta-test tasks.

3. We analyze the flatness of the minima after the model has adapted to new tasks, and observe that, as gradient-based meta-training progresses, the adapted meta-test solutions become flatter on average.

Furthermore, based on these observations, we take initial steps to propose a regularizer for MAML based training and provide experimental evidence for its effectiveness, primarily intended as an additional experiment in support of our empirical observations on the objective landscapes.

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2. Related work

There has been extensive research efforts on studying the optimization landscapes of neural networks in the standard supervised learning setup. Such work has focused on the presence of saddle points versus local minima in high dimensional landscapes (Pascanu et al., 2014),(Dauphin et al., 2014), the role of overparametrization in generalization (Freeman & Bruna, 2016), loss barriers between minima and their connectivity along low loss paths (Garipov et al., 2018; Draxler et al., 2018), to name a few examples. One hypothesis that has gained popularity is that the flatness of minima of the loss function found by stochastic gradient-based methods results in good generalization, (Hochreiter & Schmidhuber, 1997; Keskar et al., 2016). (Xing et al., 2018) and (Li et al., 2017) measure the flatness by the spectral norm of the hessian of the loss, with respect to the parameters, at a given point in the parameter space. Both (Smith & Le, 2017) and (Jastrzebski et al., 2017) consider the determinant of the hessian of the loss, with respect to the parameters, for the measure of flatness. For all of the work on flatness of minima cited above, authors have found that flatter minima correlate with better generalization. Nevertheless, some authors have argued against this correlation. (Dinh et al., 2017) theoretically demonstrated that because of the symmetries within the architectures, neural networks can be reparametrized to equivalent models corresponding to arbitrarily sharper minima.

More recently, some works have started to analyze theoretical aspects of gradient-based meta-learning. (Finn et al., 2019) introduced the Online Meta-Learning setting, where in online learning the agent faces a sequence of tasks, and provided a theoretical upper bound for the regret of MAML. (Denevi et al., 2019) study meta-learning through the perspective of biased regularization, where the model adapts to new tasks by starting from a biased parameter vector, which we refer in this work as the meta-training solution. For simple tasks such as linear regression and binary classification, they prove the advantage of starting from the meta-training solution, when learning new tasks via SGD. They use an assumption on the task similarity where the weight vectors parameterizing the tasks are assumed to be close to each other. Working in the framework for Online Convex Optimization where the model learns from a stream of tasks, (Khodak et al., 2019) make an assumption that the optimal solution for each task lies in a small subset of the parameter space and use this assumption to design an algorithm such that the "Task-averaged-regret (TAR)" scales with the diameter of this small subset of the parameter space, when using Reptile (Nichol et al., 2018), a first-order meta-learning algorithm. Unlike their work, in section 5.2 and 5.3, we analyse how, during the course of the meta-training, the similarity of the adaptation trajectories for unseen tasks changes and correlates to generalization to these new tasks. We then use these observations to design a regularizer for MAML.

3. Gradient-based meta-learning

We consider the meta-learning scenario where we have a distribution over tasks $p(\mathcal{T})$, and a model f parametrized by θ, that must learn to adapt to tasks $\mathcal{T}_i$ sampled from $p(\mathcal{T})$. The model is trained on a set of training tasks $\{\mathcal{T}_i\}^{train}$ and evaluated on a set of testing tasks $\{\mathcal{T}_i\}^{test}$, all drawn from $p(\mathcal{T})$. In this work we only consider classification tasks, with $\{\mathcal{T}_i\}^{train}$ and $\{\mathcal{T}_i\}^{test}$ using disjoint sets of classes to constitute their tasks. Here we consider the setting of k-shot learning, that is, when f adapts to a task $\mathcal{T}_i^{test}$, it only has access to a set of few support samples $\mathcal{D}_i = \{(\mathbf{x}_i^{(1)}, \mathbf{y}_i^{(1)}), ..., (\mathbf{x}_i^{(k)}, \mathbf{y}_i^{(k)})\}$ drawn from $\mathcal{T}_i^{test}$. We then evaluate the model's performance on $\mathcal{T}_i^{test}$ using a new set of target samples $\mathcal{D}'_i$. By gradient-based meta-learning, we imply that f is trained using information about the gradient of a certain loss function $\mathcal{L}(f(\mathcal{D}_i;\theta))$ on the tasks. Throughout this work the loss function is the cross-entropy between the predicted and true class.

3.1. Model-Agnostic Meta-Learning (MAML)

MAML learns an initial set of parameters θ such that on average, given a new task $\mathcal{T}_i^{test}$, only a few samples are required for f to learn and generalize well to the new task. During a meta-training iteration s, where the current parametrization of f is θ^s, a batch of n training tasks is sampled from $p(\mathcal{T})$. For each task $\mathcal{T}_i$, a set of support samples $\mathcal{D}_i$ is drawn and f adapts to $\mathcal{T}_i$ by performing T steps of full batch gradient descent on $\mathcal{L}(f(\mathcal{D}_i;\theta))$ w.r.t. θ, obtaining the adapted solution $\tilde{\theta}_i$:

$$\tilde{\theta}_i = \theta^s - \alpha \sum_{t=0}^{T-1} \nabla_\theta \mathcal{L}(f(\mathcal{D}_i;\theta_i^{(t)})) \qquad (1)$$

where $\theta_i^{(t)} = \theta_i^{(t-1)} - \alpha \nabla_\theta \mathcal{L}(f(\mathcal{D}_i;\theta_i^{(t-1)}))$ and adaptation trajectories for all $\mathcal{T}_i$ are independent and start from θ^s, i.e. $\theta_i^{(0)} = \theta^s, \forall i$. Then from each $\mathcal{T}_i$, a set of target samples $\mathcal{D}'_i$ is drawn, and the adapted meta-training solution θ^{s+1} is obtained by minimizing the loss on the target samples $\mathcal{D}'_i$, across all task $\mathcal{T}_i$ as follows:

$$\theta^{s+1} = \theta^s - \beta \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta \mathcal{L}(f(\mathcal{D}'_i;\tilde{\theta}_i)) \qquad (2)$$

As one can see in Eq.1 and Eq.2, deriving the meta-gradients implies computing second-order derivatives, which can come at a significant computational expense. The authors introduced a first-order approximation of MAML, where these second-order derivatives are ommited, and we refer to that other algorithm as First-Order MAML.

3.2. Finetuning baseline

For the finetuning baseline, the model is trained in a standard supervised learning setup: the model is trained to classify all the classes from the training split using a stochastic gradient-based optimization algorithm, its output layer size being equal to the number of meta-train classes. During evaluation on meta-test tasks, the model's final layer (fully-connected) is replaced by a layer with the appropriate size for the given meta-test task (e.g. if 5-way classification, the output layer has five logits), with its parameter values initialized to random values or with another initialization algorithm, then all the model parameters are optimized to the meta-test task, just like for the other meta-learning algorithms.

4. Analyzing the objective landscapes

In the context of gradient-based meta-learning, we define generalization as the model's ability to reach a high accuracy on a testing task $\mathcal{T}_i^{test}$, evaluated with a set of target samples $\mathcal{D}'_i$, for several testing tasks. This accuracy is computed after f, starting from a given meta-training parametrization θ^s, has optimized its parameters to the task $\mathcal{T}_i^{test}$ using only a small set of support samples $\mathcal{D}_i$, resulting in the adapted solution $\tilde{\theta}_i^{test}$ (minima). We thus care about the expected accuracy $\mathbb{E}_{\mathcal{T}_i^{test}\sim p(\mathcal{T})}[Acc(f(\mathcal{D}'_i;\tilde{\theta}_i^{test}))]$. With these definitions in mind, for many meta-test tasks $\mathcal{T}_i^{test}$, we consider the optimization landscapes $\mathcal{L}(f(\mathcal{D}_i;\theta))$, and 1) the properties of these loss landscapes evaluated at the solutions $\tilde{\theta}_i^{test}$; 2) the adaptation trajectories when f, starting from θ^s, adapts to those solutions; as well as 3) the properties of those landscapes evaluated at the meta-train solutions θ^s. See Figure 1 for a visualization of our different metrics. We follow the evolution of the metrics as meta-training progresses: after each epoch, which results in a different parametrization θ^s, we adapt f to several meta-test tasks, compute the metrics averaged over those tasks, and compare with $\mathbb{E}[Acc(f(\mathcal{D}'_i;\tilde{\theta}_i^{test}))]$. We do not deal with the objective landscapes involved during meta-training, as this is beyond the scope of this work. From here on, we drop the superscript $test$ from our notation, as we exclusively deal with objective landscapes involving meta-test tasks $\mathcal{T}_i$, unless specified otherwise.

4.1. Flatness of minima

We start our analysis of the objective loss landscapes by measuring properties of the landscapes at the adapted meta-test solutions $\tilde{\theta}_i$. More concretely, we measure the curvature of the loss at those minima, and whether flatter minima are indicative of better generalization for the meta-test tasks.

After s meta-training iterations, we have a model f parametrized by θ^s. During the meta-test, f must adapt to several meta-test tasks $\mathcal{T}_i$ independently. For a given $\mathcal{T}_i$, f adapts by performing a few steps of full-batch gradient descent on the objective landscape $\mathcal{L}(f(\mathcal{D}_i;\theta))$, using the set of support samples $\mathcal{D}_i$, and reaches an adapted solution $\tilde{\theta}_i$. Here we are interested in the curvature of $\mathcal{L}(f(\mathcal{D}_i;\tilde{\theta}_i))$, that is, the objective landscape when evaluated at such solution, and whether on average, flatter solutions favour better generalization. Considering the hessian matrix of this loss w.r.t the model parameters, defined as $H_\theta(\mathcal{D}_i;\tilde{\theta}_i) \doteq \nabla^2_\theta \mathcal{L}(f(\mathcal{D}_i;\tilde{\theta}_i))$, we measure the curvature of the loss surface around $\tilde{\theta}_i$ using the spectral norm $\|\cdot\|_\sigma$ of this hessian matrix:

$$\begin{aligned}\left\|H_\theta(\mathcal{D}_i;\tilde{\theta}_i)\right\|_\sigma &= \sqrt{\lambda_{max}\left(H_\theta(\mathcal{D}_i;\tilde{\theta}_i)^{\mathrm{H}} H_\theta(\mathcal{D}_i;\tilde{\theta}_i)\right)} \\ &= \lambda_{max}(H_\theta(\mathcal{D}_i;\tilde{\theta}_i)) \qquad (3)\end{aligned}$$

as illustrated in Figure 1 (1). (We get $\|H_\theta(\mathcal{D}_i;\tilde{\theta}_i)\|_\sigma = \lambda_{max}(H_\theta(\mathcal{D}_i;\tilde{\theta}_i))$ since $H_\theta(\mathcal{D}_i;\tilde{\theta}_i)$ is real and symmetric.)

We define the average loss curvature for meta-test solutions $\tilde{\theta}_i$, obtained from a meta-train solution θ^s, as:

$$\mathbb{E}_{\mathcal{T}_i\sim p(\mathcal{T})}[\|H_\theta(\mathcal{D}_i;\tilde{\theta}_i)\|_\sigma] \qquad (4)$$

Note that we do not measure curvature of the loss at θ^s, since θ^s is not a point of convergence of f for the meta-test tasks. In fact, at θ^s, since the model has not been adapted to the unseen meta-test classes, the target accuracy for the meta-test tasks is random chance on average. Thus, measuring the curvature of the meta-test support loss at θ^s does not relate to the notion of flatness of minima. Instead, in this work we characterize the meta-train solution θ^s by measuring the average inner product between the meta-test gradients, as explained later in Section 4.3.

4.2. Coherence of adaptation trajectories

Other than analyzing the objective landscapes at the different minima reached when f adapts to new tasks, we also analyze the adaptation trajectories to those new tasks, and whether some similarity between them can be indicative of good generalization. Let's consider a model f adapting to a task $\mathcal{T}_i$ by starting from θ^s, moving in parameter space by performing T steps of full-batch gradient descent with $\nabla_\theta \mathcal{L}(f(\mathcal{D}_i;\theta))$ until reaching $\tilde{\theta}_i$. We define the adaptation trajectory to a task $\mathcal{T}_i$ starting from θ^s as the sequence of iterates $(\theta^s, \theta_i^{(1)}, \theta_i^{(2)}, ..., \tilde{\theta}_i)$. To simplify the analyses and alleviate some of the challenges in dealing with trajectories of multiple steps in a parameter space of very high dimension, we define the trajectory displacement vector $(\tilde{\theta}_i - \theta^s)$. We define a trajectory direction vector $\vec{\theta}_i$ as the unit vector: $\vec{\theta}_i \doteq (\tilde{\theta}_i - \theta^s)/\|\tilde{\theta}_i - \theta^s\|_2$.

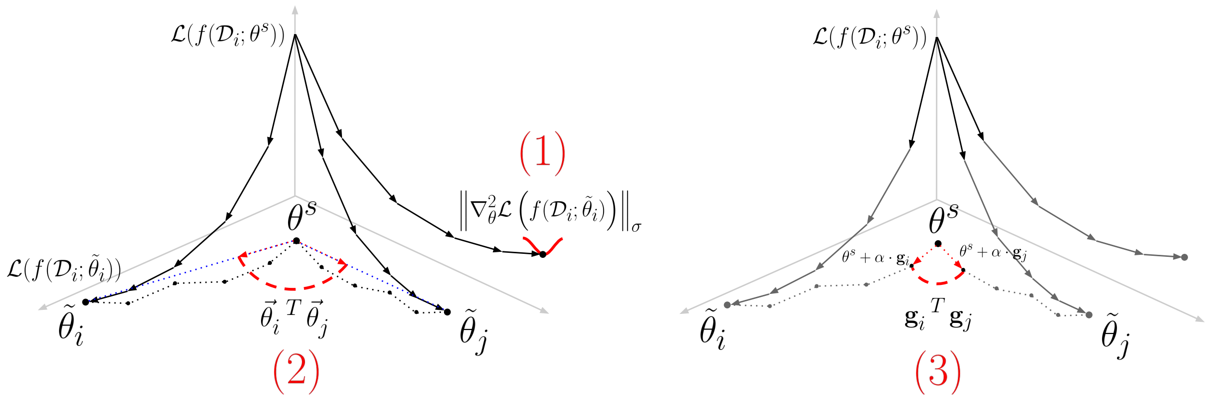

$\mathcal{L}(f(\mathcal{D}_i;\theta^s))$
(1)
$\left\|\nabla^2_\theta \mathcal{L}\left(f(\mathcal{D}_i;\tilde{\theta}_i)\right)\right\|_\sigma$
θ^s
$\mathcal{L}(f(\mathcal{D}_i;\tilde{\theta}_i))$
$\tilde{\theta}_i$
$\vec{\theta}_i^{\,T}\vec{\theta}_j$
$\tilde{\theta}_j$
(2)
$\mathcal{L}(f(\mathcal{D}_i;\theta^s))$
θ^s
$\theta^s+\alpha\cdot\mathbf{g}_i$
$\theta^s+\alpha\cdot\mathbf{g}_j$
$\mathbf{g}_i^{\,T}\mathbf{g}_j$
$\tilde{\theta}_i$
$\tilde{\theta}_j$
(3)

Figure 1. Visualizations of metrics measuring properties of objective loss landscapes. The black arrows represent the descent on the support loss and the dotted lines represent the corresponding displacement in the parameter space. (1): Curvature of the loss for an adapted meta-test solution $\tilde{\theta}_i$ (for a task $\mathcal{T}_i$), is measured as the spectral norm of the hessian matrix of the loss. (2): Coherence of adaptation trajectories to different meta-test tasks is measured as the average cosine similarity for pairs of trajectory directions. A direction vector is obtained by dividing a trajectory displacement vector (from meta-train solution θ^s to meta-test solution $\tilde{\theta}_i$) by its Euclidean norm, i.e. $\vec{\theta}_i = (\tilde{\theta}_i - \theta^s)/\|\tilde{\theta}_i - \theta^s\|_2$. (3): Characterizing a meta-train solution by the coherence of the meta-test gradients, measured by the average inner product for pairs of meta-test gradient vectors $\mathbf{g}_i = -\nabla_\theta \mathcal{L}(f(\mathcal{D}_i;\theta^s))$.

We define a metric for the coherence of adaptation trajectories to meta-test tasks $\mathcal{T}_i$, starting from a meta-train solution θ^s, as the average inner product between their direction vectors:

$$\mathbb{E}_{\mathcal{T}_i,\mathcal{T}_j\sim p(\mathcal{T})}[\vec{\theta}_i^{\,T}\vec{\theta}_j] \tag{5}$$

The inner product between two meta-test trajectory direction vectors is illustrated in Figure 1 (2).

4.3. Characterizing meta-train solutions by the average inner product between meta-test gradients

In connection to characterizing the adaptation trajectories at meta-test time, we characterize the objective landscapes at the meta-train solutions θ^s, the starting point of those trajectories. More concretely, we measure the coherence of the meta-test gradients $\nabla_\theta \mathcal{L}(f(\mathcal{D}_i;\theta^s))$ evaluated at θ^s.

The coherence between the meta-test gradients can be viewed in relation to the metric for coherence of adaptation trajectories of Eq. 5 from Section 4.2. Even after simplifying an adaptation trajectory by its displacement vector, measuring distances between trajectories of multiple steps in the parameter space can be problematic: because of the symmetries within the architectures of neural networks, where neurons can be permuted, different parameterizations θ can represent identically the same function f that maps inputs to outputs. This problem is even more prevalent for networks with higher number of parameters. Since here we ultimately care about the functional differences that f undergoes in the adaptation trajectories, measuring distances between functions in the parameter space, either using Euclidean norm or cosine similarity between direction vectors, can be problematic (Benjamin et al., 2018). Thus to further simplify the analyses on adaptation trajectories, we can measure coherence between trajectories of only one step ($T = 1$). Since we are interested in the relation between such trajectories and the generalization performance of the models, we measure the target accuracy at those meta-test solutions obtained after only one step of gradient descent. We define those solutions as: $\theta^s + \alpha \cdot \mathbf{g}_i$, with meta-test gradient $\mathbf{g}_i = -\nabla_\theta \mathcal{L}(f(\mathcal{D}_i;\theta^s))$. To make meta-training consistent with meta-testing, for the meta-learning algorithms we also use $T = 1$ for the inner loop updates of Eq. 1.

We thus measure coherence between the meta-test gradient vectors $\mathbf{g}_i$ that lead to those solutions. Note that the learning rate α is constant and is the same for all experiments on a same dataset. In contrast to Section 4.2, here we observed in practice that the average inner product between meta-test gradient vectors, and not just their direction vectors, is more correlated to the average target accuracy. The resulting metric is thus the average inner product between meta-test gradients evaluated at θ^s.

We define the average inner product between meta-test gradient vectors $\mathbf{g}_i$, evaluated at a meta-train solution θ^s, as:

$$\mathbb{E}_{\mathcal{T}_i,\mathcal{T}_j\sim p(\mathcal{T})}[\ \mathbf{g}_i^{\,T}\mathbf{g}_j\ ] \tag{6}$$

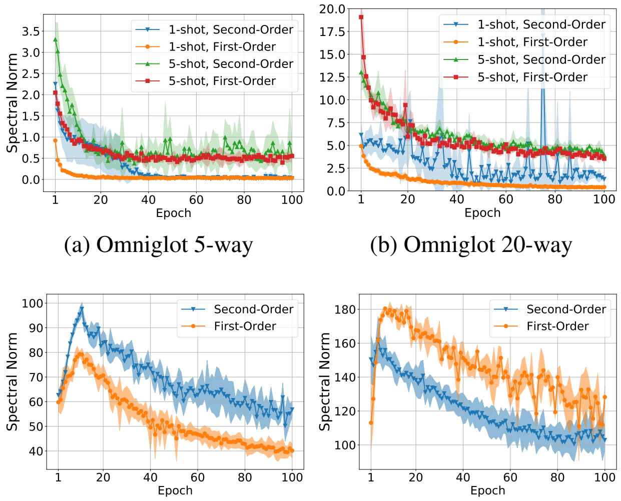
The inner product between two meta-test gradients, evaluated at θ^s, is illustrated in Figure 1 (3). We show in the experimental results in Section 5.2 and 5.3 that the coherence of the adaptation trajectories, as well as of the meta-test gradients, correlate with generalization on the meta-test tasks.

5. Experiments

We apply our analyses to the two most widely used benchmark datasets for few-shot classification problems: Omniglot and MiniImagenet datasets. We use the standardized CNN architecture used by (Vinyals et al., 2016) and (Finn et al., 2017). See A.1 for more details on the architecture. We perform our experiments using three different gradient-based meta-learning algorithms: MAML, First-Order MAML and a Finetuning baseline. For more details on the meta-learning datasets, architecture and meta-learning hyperparameters, see Section A.

For our reproduction results on the meta-train and meta-test accuracy, see Figure 12a and 12b in B.1.

5.1. Flatness of meta-test solutions



(a) Omniglot 5-way

(b) Omniglot 20-way

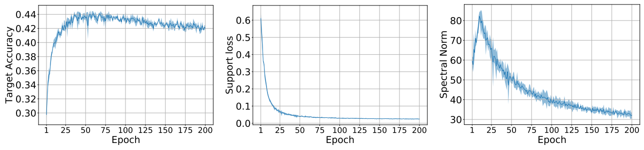
(c) MiniImagenet 5-way, 1-shot

(d) MiniImagenet 5-way, 5-shot

Figure 2. Flatness of meta-test solutions for MAML and First-Order MAML, on Omniglot and MiniImagenet

After each training epoch, we compute $\mathbb{E}[\|H_\theta(\mathcal{D}_i;\tilde{\theta}_i)\|_\sigma]$ using a fixed set of 60 randomly sampled meta-test tasks $\mathcal{T}_i$. Across all settings, we observe that MAML first finds sharper solutions $\tilde{\theta}_i$ until reaching a peak, then as the number of epoch grows, those solutions become flatter, as seen in Figure 2.

However, we were not able to find an indication that the flatness of those minima reflects their ability to generalize well to new tasks. If we train models beyond the point of meta-overfitting, where $\mathbb{E}[Acc(f(\mathcal{D}'_i;\tilde{\theta}_i))]$ starts to decrease (see Figure 3a), we observe that the average loss curvature $\mathbb{E}[\|H_\theta(\mathcal{D}_i;\tilde{\theta}_i)\|_\sigma]$ keeps decreasing (see Figure 3c). We perform the same analysis for our finetuning baseline (Figures 4a, 4c), with results suggesting that flatness of solutions might be more linked with $\mathbb{E}[\mathcal{L}(f(\mathcal{D}_i;\tilde{\theta}_i))]$, the average level of support loss attained by the solutions $\tilde{\theta}_i$ (see Figures 4b and 3b), which is not a good indicator for generalization. We also noted that across all settings involving MAML and First-Order MAML, this average meta-test support loss $\mathbb{E}[\mathcal{L}(f(\mathcal{D}_i;\tilde{\theta}_i))]$ decreases monotonically as meta-training progresses. Future empirical or theoretical work might indicate a correlation between flatness of those minima and their generalization, which would be beyond the scope of our analysis. Here our observations led us to consider the adaptation trajectories that lead to those minima, and whether some properties of those trajectories could be linked to generalization.

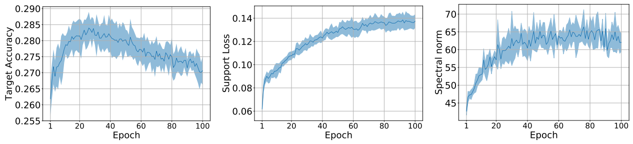


(a) Target Accuracy

(b) Support loss

(c) Curvature of solutions

Figure 3. MAML: Characterization of meta-test solutions



(a) Target accuracy

(b) Support loss

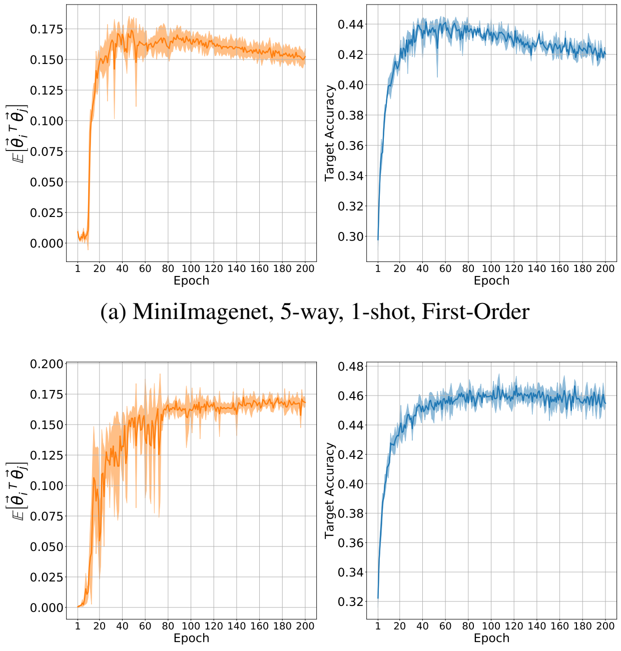
(c) Curvature of solutions

Figure 4. Finetune baseline : Characterization of meta-test solutions

5.2. Coherence of adaptation trajectories

In this section, we use the same experimental setup as in Section 5.1, except here we measure $\mathbb{E}[\vec{\theta_i}^T\vec{\theta_j}]$. To reduce the variance on our results, we sample 500 tasks after each meta-training epoch. Also for experiments on Omniglot, we drop the analyses with First-Order MAML, since it yields performance very similar to that of the Second-Order MAML. We start our analyses with the setting of "MiniImagenet, First-Order MAML, 5-way 1-shot", as it allowed us to test and invalidate the correlation between flatness of solutions and generalization, earlier in Section 5.1.

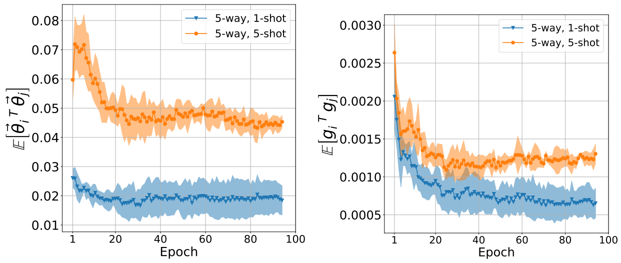
We clearly observe a correlation between the coherence of adaptation trajectories and generalization to new tasks, with higher average inner product between trajectory directions, thus smaller angles, being linked to higher average target accuracy on those new tasks, as shown in Figure 5a. We then performed the analysis on the other settings, with the same observations (see Figure 5b and Figure 13 in Appendix B.2 for full set of experiments). We also perform the analysis



(a) MiniImagenet, 5-way, 1-shot, First-Order

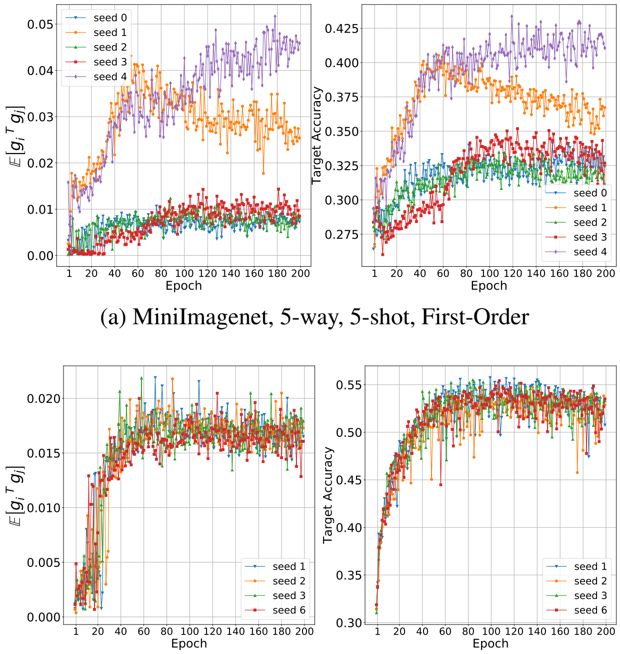
(b) MiniImagenet, 5-way, 1-shot, Second-Order

Figure 5. Comparison between average cosine similarity among meta-test trajectory direction (orange), and average target accuracy on meta-test tasks (blue), MAML First-Order and Second-Order, MiniImagenet 5-way 1-shot. See Figure 13 in Appendix B.2 for full set of experiments.



(a) Trajectories coherence (b) Gradients coherence

Figure 6. (a): Average inner product between meta-test adaptation direction vectors, for Finetuning baseline on MiniImagenet. (b): Average inner product between meta-test gradients, for Finetuning baseline on MiniImagenet.



(a) MiniImagenet, 5-way, 5-shot, First-Order

(b) MiniImagenet, 5-way, 5-shot, Second-Order

Figure 7. Comparison between average inner product between meta-test gradient vectors, evaluated at meta-train solution, and average target accuracy on meta-test tasks, with higher average inner product being linked to better generalization. See Figure 14 in Appendix B.3 for full set of experiments.

on the Finetuning baselines, which reach much lower target accuracies, and where we see that $\mathbb{E}[\vec{\theta}_i^{\,T}\vec{\theta}_j]$ remains much closer to zero, meaning that trajectory directions are roughly orthogonal to each other, akin to random vectors in high dimension (see Figure 6a). As an added observation, here we include our experimental results on the average meta-test trajectory norm $\mathbb{E}[\|\tilde{\theta}_i - \theta^s\|_2]$, in Figure 15a, 15b of Appendix B.4, where $\mathbb{E}[\|\tilde{\theta}_i - \theta^s\|_2]$ grows as meta-training progresses when f is meta-trained with MAML, as opposed to the Finetune baseline, and note that this norm does not reflect generalization.

5.3. Characterizing meta-train solutions by the average inner product between meta-test gradients

Despite the clear correlation between $\mathbb{E}[\vec{\theta}_i^{\,T}\vec{\theta}_j]$ and generalization for the settings that we show in Figure 5 and 13, we observed that for some other settings, this relationship appears less linear. We conjecture that such behavior might arise from the difficulties of measuring distances between networks in the parameter space, as explained in Section 4.3. Nevertheless, those observations and their indications motivated us to analyze the starting point of the adaptation trajectories. Here we present our results on the characterization of the objective landscapes at the meta-train solutions θ^s, by measuring the average inner product between meta-test gradient vectors $\mathbf{g}_i$. We observe that coherence between meta-test gradients is correlated to generalization, which is consistent with the observations on the coherence of adaptation trajectories from Section 5.2. In Figure 7, we compare $\mathbb{E}[\ \mathbf{g}_i^{\,T}\mathbf{g}_j\ ]$ to the target accuracy (here we show results for individual model runs rather than the averages over the runs). See Figure 14 in Appendix B.3 for the full set of experiments. This metric consistently correlates with generalization across the different settings. Similarly as in Section 5.2, for our fine-tuning baselines we observe very low coherence between meta-test gradients (see Figure 6b). Based on the observations we make in Section 5.2 and 5.3, we propose to regularize gradient-based meta-learning as described in Section 7.

5.3.1. Few-shot regression: Average inner product between meta-test gradients

Here we extend our analysis by presenting experimental results on $\mathbb{E}[\ \mathbf{g}_i{}^T \mathbf{g}_j\ ]$ for few-shot regression. Specifically we use a leaning problem which is composed of training task and test tasks, where each of these tasks are sine functions parameterized as $y = a\sin(bx + c)$. We train a two-layer MLP which learns to fit meta-training sine functions using only few support samples, and generalization implies reaching a low Mean Squared Error (MSE) averaged over the target set of many meta-test sine functions. For this setting, which is also present in the work of (Finn et al., 2017), we use a fully-connected architecture of two hidden layers, 40 neurons wide. We use the Mean Square Error as the loss function. Tasks consists of fitting one dimensional sine functions evaluated on the domain $[-5, 5]$, Here sine functions vary in amplitude and phase, and meta-train and meta-test sine functions are generated with disjoint ranges of amplitude and phase. Results are presented in Figure 8. Similar to our analysis of the Few-shot classification setting, we observe in the case of Few-shot regression, generalization (negative average target MSE on Meta-test Task) strongly correlates with $\mathbb{E}[\ \mathbf{g}_i{}^T \mathbf{g}_j\ ]$.

6. The learned representations and their relations to task gradients

As they appear, our observations on the coherence of meta-test gradients, and their relation to generalization to new tasks, are surprising yet hard to interpret them intuitively. In this section, we attempt to provide such interpretation, which is an informal hypothesis rather than a theoretical claim, but which we further verify empirically. Our intuition is that MAML, in order to represent the data and classify from few examples, might be learning a metric space based on the inner product. The model $f(x)$ can be expressed as $f_{lin}(f_{feat}(x))$ where f_{lin} is the linear classifier, with of a weight matrix W and a bias vector b followed by a softmax, and f_{feat} is a feature network, that outputs a representation vector h, such that $h = f_{feat}(x)$. Recently, (Raghu et al., 2019) showed for MAML that at meta-test time, the vectors h representing the new task inputs barely change during adaptation, as opposed to the outputs of f_{lin}. Moreover, they show that while achieving the same generalization, the adaptation to those new tasks can be performed by freezing the layers of f_{feat}, thus only adapting f_{lin}, with the vectors h remaining unchanged. This suggests that generalization to new tasks might be linked to how the learned representation space will embed the vectors h for the unseen data. The Prototypical Network (Snell et al., 2017) learns a metric space in which an example x is classified based on a softmax on the Euclidean distances between its vector h and the learned cluster mean vectors. In MAML, x is classified according to a score from the logits of f_{lin} followed by a softmax, the scores being proportional to the inner products $h^T W_{i,:}$ between h and the rows of W. Our intuition is that vectors h that are closer in inner products could lead to gradients vectors closer in inner product, and that the representation space learned by MAML could be based on the inner product. To empirically verify this, we computed the average inner product between the vectors h_i from all images of the meta-test data, produced by f_{feat} at θ^s, after each meta-train epoch. In Figure 9, we observe that meta-overfitting reflects $\mathbb{E}[h_i^T h_j]$, and in Figure 10 we show the correlation between $\mathbb{E}[h_i^T h_j]$ and generalization. These results suggest that this interpretation is plausible, while further theoretical work is required to further validate it. Thus according to this interpretation, for MAML, meta-training would learn a representation space in which the embeddings for the new, previously unseen data will gradually appear more similar to each other according to their inner product. The model would gradually learn general features, which are able to represent new data, more closely in an inner product space, but s meta-overfitting occurs, they become too specific to the training classes less general with respect to new data.

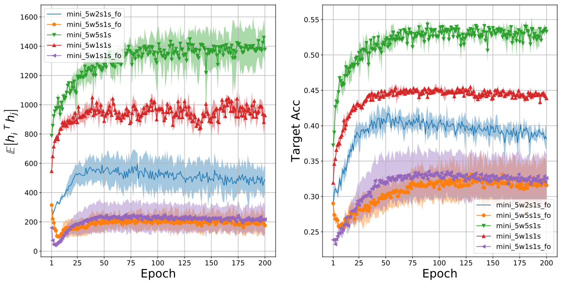


Figure 9. Comparison between average inner product between representation vectors, generated by the feature network at meta-train solution, and average target accuracy on meta-test tasks

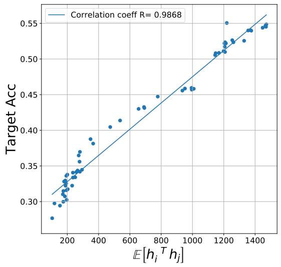
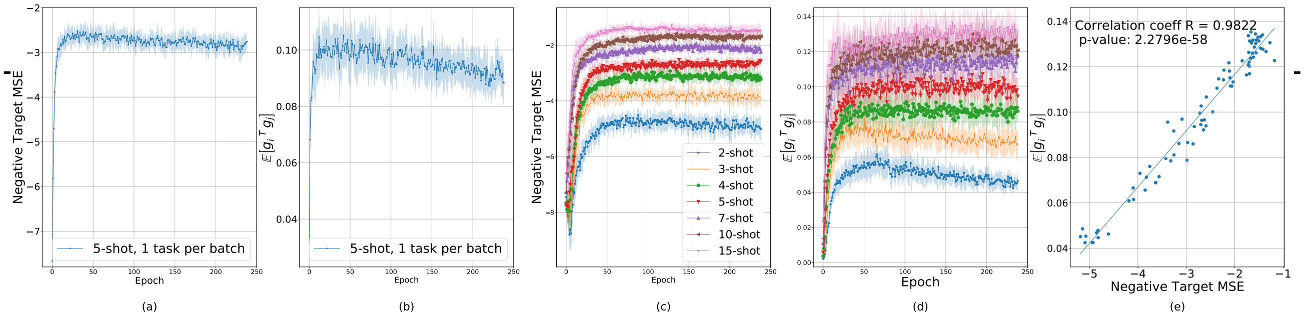


Figure 10. Correlation between $\mathbb{E}[h_i^T h_j]$ and generalization for MAML and First-Order MAML, with k varying between 1 and 5



(a) (b) (c) (d) (e)

Figure 8. Analysis for Few-shot regression. Comparison between $\mathbb{E}[\ \mathbf{g}_i{}^T \mathbf{g}_j\ ]$ and average negative target Mean Squared Error on meta-test tasks(generalization performance). (a) and (b) show generalization performance correlates with $\mathbb{E}[\ \mathbf{g}_i{}^T \mathbf{g}_j\ ]$ through-out the meta-training (c) and (d) show the correlation across many values of k (number of shots), while (e) shows the correlation coefficient R between $\mathbb{E}[\ \mathbf{g}_i{}^T \mathbf{g}_j\ ]$ and final generalization performance, for models with k varying between 2 and 15.

7. Towards regularizing MAML

Based on our observations on the coherence of adaptation trajectories, we take *first steps* in this direction by adding a regularization term based on $\mathbb{E}[\vec{\theta_i}{}^T\vec{\theta_j}]$. Within a meta-training iteration, we first let f adapt to the n training tasks $\mathcal{T}_i$ following Eq 1. We then compute the average direction vector $\vec{\theta}_\mu = \frac{1}{n}\sum_{i=1}^{n}\vec{\theta}_i$. For each task, we want to reduce the angle defined by $\vec{\theta_i}{}^T\vec{\theta}_\mu$, and thus introduce the penalty on $\Omega(\theta) = -\vec{\theta_i}{}^T\vec{\theta}_\mu$, obtaining the regularized solutions $\hat{\theta}_i$. The outer loop gradients are then computed, just like in MAML following Eq 2, but using these regularized solutions $\hat{\theta}_i$ instead of $\tilde{\theta}_i$. We obtain the variant of MAML with regularized inner loop updates, as detailed in Algorithm 1. We used this regularizer with MAML (Second-Order), for "Omniglot 20-way 1-shot", thereby tackling the most challenging few-shot classification setting for Omniglot. As shown in Figure 11, we observed an increase in meta-test target accuracy: the performance increases from 94.05% to 95.38% (average over five trials, 600 test tasks each), providing $\sim 23\%$ relative reduction in meta-test target error.

Algorithm 1 Regularized MAML: Added penalty on angles between inner loop updates

```
Sample a batch of $n$ tasks $\mathcal{T}_i \sim p(\mathcal{T})$
for all $\mathcal{T}_i$ do
    Perform inner loop adaptation as in Eq. 1: $\tilde{\theta}_i = \theta^s - \alpha\sum_{t=0}^{T-1}\nabla_\theta\mathcal{L}(f(\mathcal{D}_i;\theta_i^{(t)}))$
end for
Compute the average direction vector:
$\vec{\theta}_\mu = \frac{1}{n}\sum_{i=1}^{n}\vec{\theta}_i$
Compute the corrected inner loop updates:
for all $\mathcal{T}_i$ do
    $\hat{\theta}_i = \tilde{\theta}_i - \gamma\nabla_\theta\Omega(\theta)$ where $\Omega(\theta) = -\vec{\theta_i}{}^T\vec{\theta}_\mu$
end for
Perform the meta-update as in Eq. 2, but using the corrected solutions:
$\theta^{s+1} = \theta^s - \beta\frac{1}{n}\sum_{i=1}^{n}\nabla_\theta\mathcal{L}(f(\mathcal{D}'_i;\hat{\theta}_i))$
```

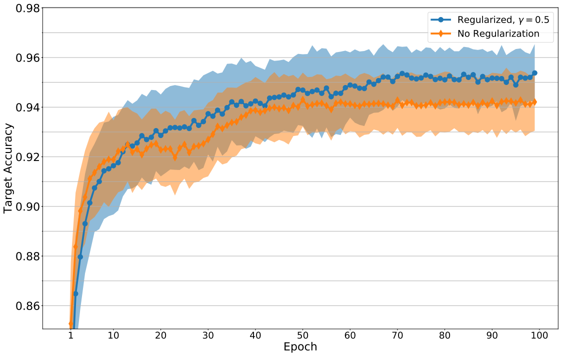


Figure 11. Average target accuracy on meta-test tasks using our proposed regularizer on MAML, for Omniglot 20-way 1-shot learning, with regularization coefficient $\gamma = 0.5$

8. Conclusion

Focusing on few-shot image classification, we experimentally demonstrate that when using gradient-based meta-learning algorithms such as MAML, generalization to new tasks appears correlated with the average inner product between gradients of new tasks. We also observed that this might be linked with the kind of metric space that MAML learns for its representation, possibly based on the inner product, for the case of few-shot classification with a linear layer followed by a softmax. We also show this correlation for few-shot regression tasks. In addition, we observe that meta-test solutions, obtained after adapting neural networks to new tasks via few-shot learning, become flatter as meta-training progresses. Based on these observations, we take first steps towards regularizing MAML based meta-training. As a future work, we plan to test the effectiveness of this regularizer on various datasets and meta-learning problem settings, architectures and gradient-based meta-learning algorithms.

References

Benjamin, A. S., Rolnick, D., and Körding, K. P. Measuring and regularizing networks in function space. *CoRR*, abs/1805.08289, 2018. URL http://arxiv.org/abs/1805.08289.

Dauphin, Y., Pascanu, R., Gülçehre, Ç., Cho, K., Ganguli, S., and Bengio, Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *CoRR*, abs/1406.2572, 2014. URL http://arxiv.org/abs/1406.2572.

Denevi, G., Ciliberto, C., Grazzi, R., and Pontil, M. Learning-to-learn stochastic gradient descent with biased regularization. *CoRR*, abs/1903.10399, 2019. URL http://arxiv.org/abs/1903.10399.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. *CoRR*, abs/1703.04933, 2017. URL http://arxiv.org/abs/1703.04933.

Draxler, F., Veschgini, K., Salmhofer, M., and Hamprecht, F. A. Essentially No Barriers in Neural Network Energy Landscape. *ArXiv e-prints*, March 2018.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, abs/1703.03400, 2017. URL http://arxiv.org/abs/1703.03400.

Finn, C., Rajeswaran, A., Kakade, S. M., and Levine, S. Online meta-learning. *CoRR*, abs/1902.08438, 2019. URL http://arxiv.org/abs/1902.08438.

Freeman, C. D. and Bruna, J. Topology and Geometry of Half-Rectified Network Optimization. *ArXiv e-prints*, November 2016.

Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D., and Wilson, A. G. Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs. *ArXiv e-prints*, February 2018.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M. (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL http://proceedings.mlr.press/v9/glorot10a.html.

Hochreiter, S. and Schmidhuber, J. Flat minima. *Neural Comput.*, 9(1):1–42, January 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.1.1. URL http://dx.doi.org/10.1162/neco.1997.9.1.1.

Jastrzebski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. J. Three factors influencing minima in SGD. *CoRR*, abs/1711.04623, 2017. URL http://arxiv.org/abs/1711.04623.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. *CoRR*, abs/1609.04836, 2016. URL http://arxiv.org/abs/1609.04836.

Khodak, M., Balcan, M., and Talwalkar, A. Provable guarantees for gradient-based meta-learning. *CoRR*, abs/1902.10644, 2019. URL http://arxiv.org/abs/1902.10644.

Li, H., Xu, Z., Taylor, G., and Goldstein, T. Visualizing the loss landscape of neural nets. *CoRR*, abs/1712.09913, 2017. URL http://arxiv.org/abs/1712.09913.

Nichol, A., Achiam, J., and Schulman, J. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018. URL http://arxiv.org/abs/1803.02999.

Oreshkin, B. N., López, P. R., and Lacoste, A. TADAM: task dependent adaptive metric for improved few-shot learning. *CoRR*, abs/1805.10123, 2018. URL http://arxiv.org/abs/1805.10123.

Pascanu, R., Dauphin, Y. N., Ganguli, S., and Bengio, Y. On the saddle point problem for non-convex optimization. *CoRR*, abs/1405.4604, 2014. URL http://arxiv.org/abs/1405.4604.

Raghu, A., Raghu, M., Bengio, S., and Vinyals, O. Rapid learning or feature reuse? towards understanding the effectiveness of maml, 09 2019.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. URL https://openreview.net/forum?id=rJY0-Kcll.

Rothfuss, J., Lee, D., Clavera, I., Asfour, T., and Abbeel, P. Promp: Proximal meta-policy search. *CoRR*, abs/1810.06784, 2018. URL http://arxiv.org/abs/1810.06784.

Smith, S. L. and Le, Q. V. A bayesian perspective on generalization and stochastic gradient descent. *CoRR*, abs/1710.06451, 2017. URL http://arxiv.org/abs/1710.06451.

Snell, J., Swersky, K., and Zemel, R. S. Prototypical networks for few-shot learning. *CoRR*, abs/1703.05175, 2017. URL http://arxiv.org/abs/1703.05175.

Vinyals, O., Blundell, C., Lillicrap, T. P., Kavukcuoglu, K., and Wierstra, D. Matching networks for one shot learning. *CoRR*, abs/1606.04080, 2016. URL http://arxiv.org/abs/1606.04080.

Xing, C., Arpit, D., Tsirigotis, C., and Bengio, Y. A Walk with SGD. *ArXiv e-prints*, February 2018.

A. Additional Experimental Details

A.1. Model Architectures

We use the architecture proposed by (Vinyals et al., 2016) which is used by (Finn et al., 2017), consisting of 4 modules stacked on each other, each being composed of 64 filters of of 3 × 3 convolution, followed by a batch normalization layer, a ReLU activation layer, and a 2 × 2 max-pooling layer. With Omniglot, strided convolution is used instead of max-pooling, and images are downsampled to 28 × 28. With MiniImagenet, we used fewer filters to reduce overfitting, but used 48 while MAML used 32. As a loss function to minimize, we use cross-entropy between the predicted classes and the target classes.

A.2. Meta-Learning datasets

The Omniglot dataset consists of a total of 1623 classes, each comprising 20 instances. The classes correspond to distinct characters, taken from 50 different datasets, but the taxonomy among characters isn't used. The MiniImagenet dataset comprises 64 training classes, 12 validation classes and 24 test classes. Each of those classes was randomly sampled from the original Imagenet dataset, and each contains 600 instances with a reduced size of 84 × 84.

A.3. Hyperparameters used in meta-training and meta-testing for few-shot classification

We follow the same experimental setup as (Finn et al., 2017) for training and testing the models using MAML and First-Order MAML. During meta-training, the inner loop updates are performed via five steps of full batch gradient descent (except for Section 5.3 where $T = 1$), with a fixed learning rate α of 0.1 for Omniglot and 0.01 for MiniImagenet, while ADAM is used as the optimizer for the meta-update, without any learning rate scheduling, using a meta-learning rate β of 0.001. At meta-test time, adaptation to meta-test task is always performed by performing the same number of steps as for the meta-training inner loop updates. We use a mini-batch of 16 and 8 tasks for the 1-shot and 5-shot settings respectively, while for the MiniImagenet experiments, we use batches of 4 and 2 tasks for the 1-shot and 5-shots settings respectively. Let's also precise that, in *k-shot* learning for an *m-way* classification task $\mathcal{T}_i$, the set of support samples $\mathcal{D}_i$ comprises $k \times m$ samples. Each meta-training epoch comprises 500 meta-training iterations.

For the finetuning baseline, we kept the same hyperparameters for the ADAM optimizer during meta-training, and for the adaptation during meta-test. We searched the training hyperparameter values for the mini-batch size and the number of iterations per epoch. Experiments are run for a 100 epochs each. In order to limit meta-overfitting and maximize the highest average meta-test target accuracy, the finetuning models see roughly 100 times less training data per epoch compared to a MAML training epoch. In order to evaluate the baseline on the 1-shot and 5-shot meta-test tasks, during training we used mini-batches of 64 images with 25 iterations per epoch for 1-shot learning, and mini-batches of 128 images with 12 iterations per epoch, for 5-shot learning. At meta-test time, we use Xavier initialization (Glorot & Bengio, 2010) to initialize the weights of the final layer.

B. Additional Experimental Results

B.1. Performance of models trained with MAML and First-Order MAML, on the few-shot learning settings

The performance of the models trained with MAML and First-Order MAML, for the few-shot learning settings of Omniglot and MiniImagenet, are presented in Figure 12. They include the target accuracies on meta-train tasks and on meta-test tasks (generalization), as meta-training progresses.

B.2. Coherence of adaptation trajectories

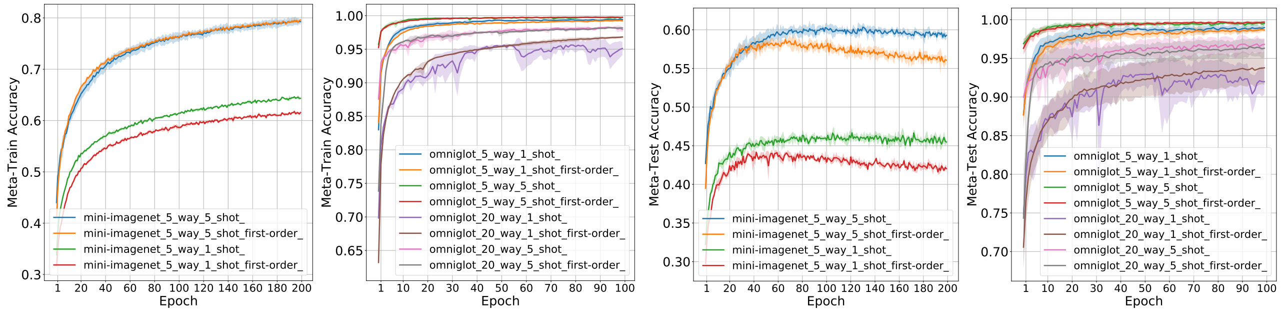
The relation between target accuracy on meta-test tasks, and angles between trajectory directions is presented in Figure 13.

B.3. Average inner product between meta-test gradients

The relation between target accuracy on meta-test tasks, and average inner product between meta-test gradients evaluated at meta-train solution, is presented in Figure 14.

B.4. Average l_2 norm of adaptation trajectories

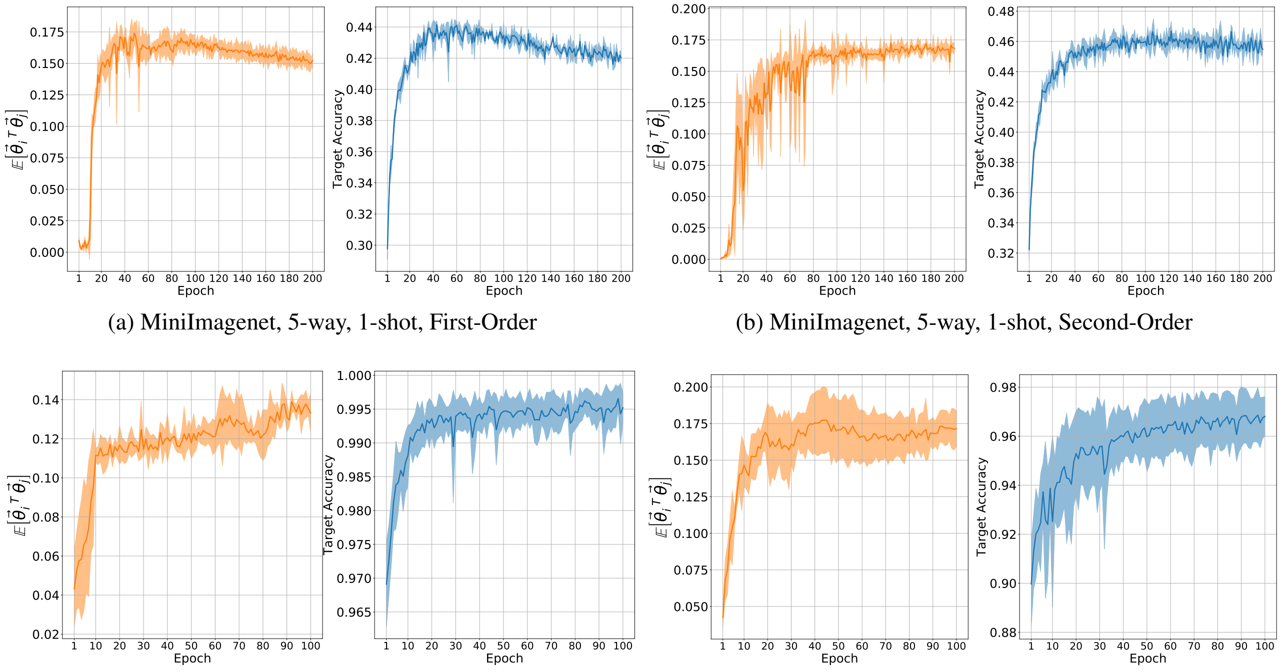
The experimental results on the average l_2 norm of the adaptation trajectories, are presented in Figure 15.



(a) Meta-Train Accuracy

(b) Meta-Test Accuracy

Figure 12. MAML: Accuracies on training and testing tasks



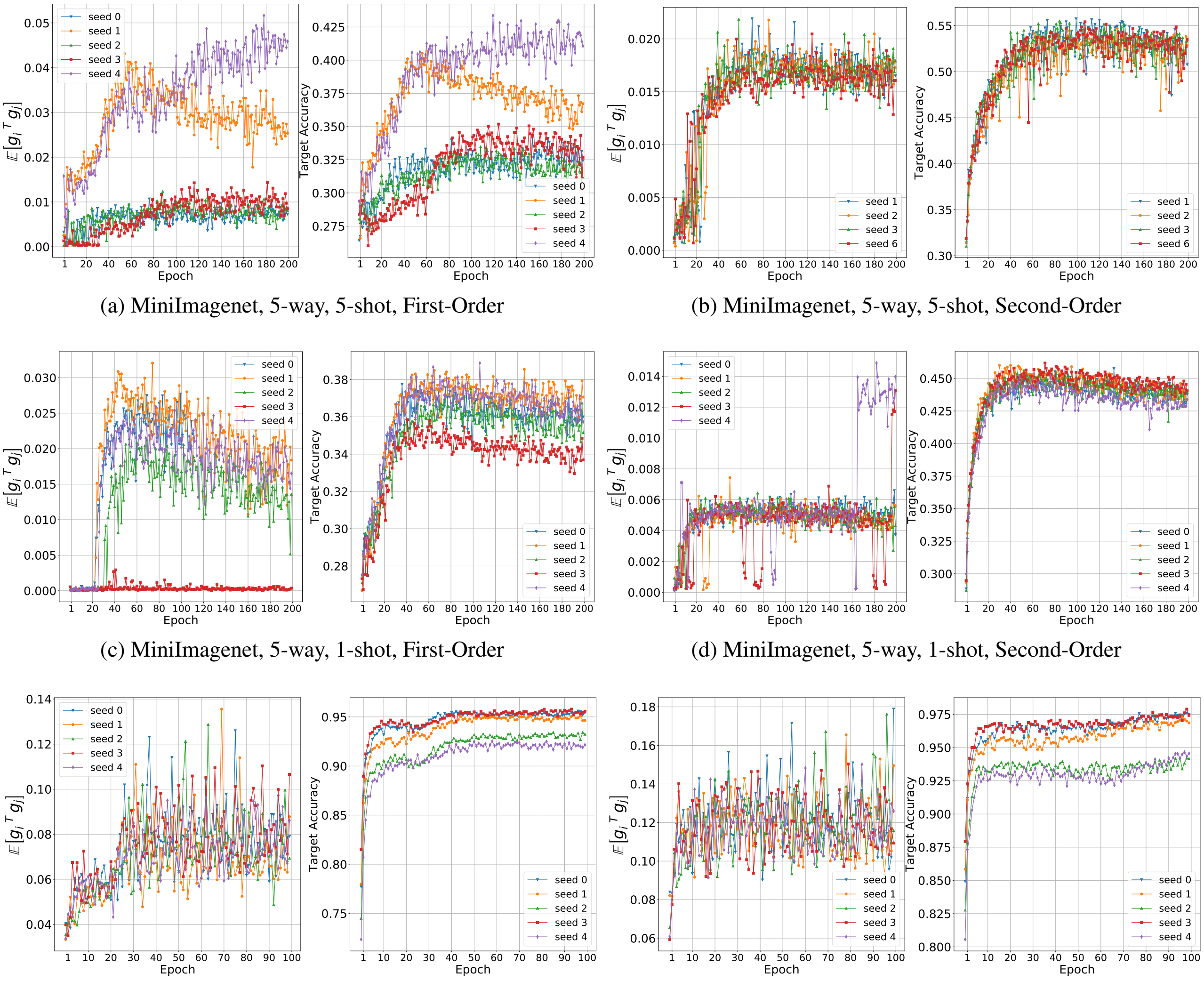
(a) MiniImagenet, 5-way, 1-shot, First-Order

(b) MiniImagenet, 5-way, 1-shot, Second-Order

(c) Omniglot, 5-way, 5-shot, Second-Order

(d) Omniglot, 20-way, 5-shot, Second-Order

Figure 13. Comparison between average inner product between trajectory directions and average target accuracy on meta-test tasks. Full set of experiments.



(a) MiniImagenet, 5-way, 5-shot, First-Order

(b) MiniImagenet, 5-way, 5-shot, Second-Order

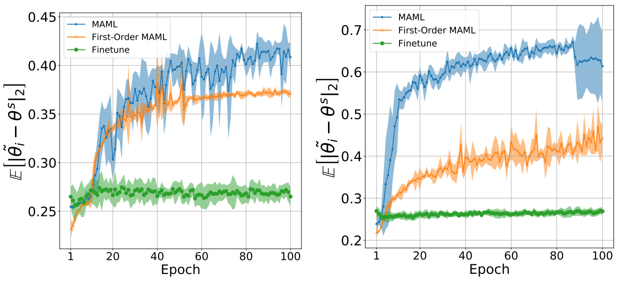
(c) MiniImagenet, 5-way, 1-shot, First-Order

(d) MiniImagenet, 5-way, 1-shot, Second-Order

(e) Omniglot, 20-way, 1-shot, Second-Order

(f) Omniglot, 20-way, 5-shot, Second-Order

Figure 14. Comparison between average inner product between trajectory displacement vectors, and average target accuracy on meta-test tasks. Full set of experiments.



(a) l_2 norm of trajectories (1-shot)

(b) l_2 norm of trajectories (5-shot)

Figure 15. Average l_2 norm of meta-test adaptation trajectories, all algorithms on MiniImagenet, (a): 1-shot learning, (b): 5-shot learning.