TOWARDS UNDERSTANDING GENERALIZATION IN GRADIENT-BASED META-LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In this work we study generalization of neural networks in gradient-based meta-learning by analyzing various properties of the objective landscapes. We experimentally demonstrate 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, lower in loss, and further away from the meta-train solution. We also show that those meta-test solutions become flatter even as generalization starts to degrade, thus providing an experimental evidence against the correlation between generalization and flat minima in the paradigm of gradient-based meta-leaning. Furthermore, we provide empirical evidence that generalization to new tasks is correlated with the coherence between their adaptation trajectories in parameter space, measured by the average cosine similarity between task-specific trajectory directions, starting from a same meta-train solution. We also show that coherence of meta-test gradients, measured by the average inner product between the task-specific gradient vectors evaluated at meta-train solution, is also correlated with generalization.

1 INTRODUCTION

To address the problem of the few-shot learning, many meta-learning approaches have been proposed recently (Finn et al., 2017), (Ravi and 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.

Our main insights and contributions can be summarized as follows:

1. As gradient-based meta-training progresses:
   - the adapted meta-test solutions become flatter on average, while the opposite occurs when using a finetuning baseline.
   - the adapted final solutions reach lower average support loss values, which never increases, while the opposite occurs when using a finetuning baseline.
2. When generalization starts to degrade due to overtraining, meta-test solutions keep getting flatter, implying that, in the context of gradient-based meta-learning, flatness of minima is not correlated with generalization to new tasks.
3. We empirically show that generalization to new tasks is correlated with the coherence between their adaptation trajectories, measured by the average cosine similarity between trajectory directions. Also correlated with generalization is the coherence between meta-test gradients, measured by the average inner product between meta-test gradient vectors evaluated at meta-train solution. We also show that this metric is correlated to generalization for few-shot regression tasks where the model must learn to fit sine function curves.

Furthermore, based on these observations, we take initial steps to propose a regularizer for MAML based training and provide experimental evidence for its effectiveness.

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 and 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 and 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 and 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.

In contrast to previous work on understanding the objective landscapes of neural networks in the classical supervised learning paradigm, in our work, we explore the properties of objective landscapes in the setting of gradient-based meta-learning.

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 $\mathcal{D}_i^0$. 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 that 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 $\tilde{\theta}_i$:

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

where $\theta_i^{(t)} = \theta_i^{(t-1)} - \nabla \mathcal{L}(f(\mathcal{D}_i; \theta_i^{(t-1)}))$ and all adaptations 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 $\mathcal{D}_i^0$ is drawn, and the adapted meta-training solution θ^{s+1} is obtained by averaging the target gradients, such that:

$$\theta^{s+1} = \theta^s - \frac{1}{n} \sum_{i=1}^{n} \nabla \mathcal{L}(f(\mathcal{D}_i^0; \tilde{\theta}_i)) \tag{2}$$

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

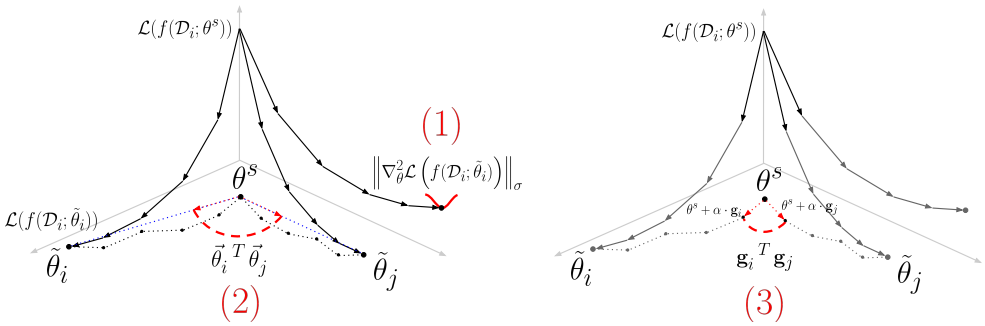


Figure 1: Visualizations of metrics measuring properties of objective loss landscapes. The black arrows represent the descent on the support loss and the dotted lines represent the corresponding displacement in the parameter space. (1): Curvature of the loss for an adapted meta-test solution $\tilde{\ }_i$ (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 ^{s} to meta-test solution $\tilde{\ }_i$) by its Euclidean norm, i.e. $\tilde{\ }_i = (\tilde{\ }_i - \ ^s)/\|\tilde{\ }_i - \ ^s\|_2$. (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 $\mathbf{g}_i = -\nabla_\theta \mathcal{L}(f(\mathcal{D}_i;\ ^s))$.

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 $\mathcal{D}_i^0$, for several testing tasks. This accuracy is computed after f, starting from a given meta-training parametrization ^{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 $\tilde{\ }_i^{test}$ (minima). We thus care about the average accuracy $\mathbb{E}_{\mathcal{T}_i^{test} \quad p(\mathcal{T})}[Acc(f(\mathcal{D}_i^0;\tilde{\ }_i^{test}))]$. With these definitions in mind, for many meta-test tasks $\mathcal{T}_i^{test}$, we consider the optimization landscapes $\mathcal{L}(f(\mathcal{D}_i;\ ))$, and 1) the properties of these loss landscapes evaluated at the solutions $\tilde{\ }_i^{test}$; 2) the adaptation trajectories when f, starting from ^{s}, adapts to those solutions; as well as 3) the properties of those landscapes evaluated at the meta-train solutions ^{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 ^{s}, we adapt f to several meta-test tasks, compute the metrics averaged over those tasks, and compare with $\mathbb{E}[Acc(f(\mathcal{D}_i^0;\tilde{\ }_i^{test}))]$. 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 $\tilde{}_i$. 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 ^{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 $\mathcal{L}(f(\mathcal{D}_i; \ ))$, using the set of support samples $\mathcal{D}_i$, and reaches an adapted solution $\tilde{}_i$. Here we are interested in the curvature of $\mathcal{L}(f(\mathcal{D}_i; \tilde{}_i))$, 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 $H(\mathcal{D}_i; \tilde{}_i) \doteq \nabla^2 \mathcal{L}(f(\mathcal{D}_i; \tilde{}_i))$, we measure the curvature of the loss surface around $\tilde{}_i$ using the spectral norm $\| \cdot \|$ of this hessian matrix:

$$H(\mathcal{D}_i; \tilde{}_i) = \sqrt{\max H(\mathcal{D}_i; \tilde{}_i)^{\mathsf{H}} H(\mathcal{D}_i; \tilde{}_i)} = \max(H(\mathcal{D}_i; \tilde{}_i)) \tag{3}$$

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

We define the average loss curvature for meta-test solutions obtained from a meta-train solution ^{s}, as:

$$\mathbb{E}_{\mathcal{T}_i \ p(\mathcal{T})}[\|H(\mathcal{D}_i; \tilde{}_i)\|] \tag{4}$$

Note that we do not measure curvature of the loss at ^{s}, since ^{s} is not a point of convergence of f for the meta-test tasks. In fact, at ^{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 ^{s} does not relate to the notion of flatness of minima. Instead, in this work we characterize the meta-train solution ^{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 ^{s}, moving in parameter space by performing T steps of full-batch gradient descent with $\nabla \mathcal{L}(f(\mathcal{D}_i; \ ))$ until reaching $\tilde{}_i$. We define the adaptation trajectory to a task $\mathcal{T}_i$ starting from ^{s} as the sequence of iterates $(^s; \ ^{(1)}_i; \ ^{(2)}_i; \ldots; \tilde{}_i)$. 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 $(\tilde{}_i - \ ^s)$. We define a trajectory direction vector $\tilde{}_i$ as the unit vector: $\tilde{}_i \doteq (\tilde{}_i - \ ^s)/\|\tilde{}_i - \ ^s\|_2$.

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

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

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 addition to characterizing the adaptation trajectories at meta-test time, we characterize the objective landscapes at the meta-train solutions ^{s}. More concretely, we measure the coherence of the meta-test gradients $\nabla \mathcal{L}(f(\mathcal{D}_i; \ ^s))$ evaluated at ^{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 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: $^{s} + \cdot \mathbf{g}_i$, with meta-test gradient $\mathbf{g}_i = -\nabla \mathcal{L}(f(\mathcal{D}_i;\ ^{s}))$. 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 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 ^{s}.

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

$$\mathbb{E}_{T_i;T_j \quad p(T)}[\ \mathbf{g}_i{}^{T}\mathbf{g}_j\ ] \tag{6}$$

The inner product between two meta-test gradients, evaluated at ^{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.

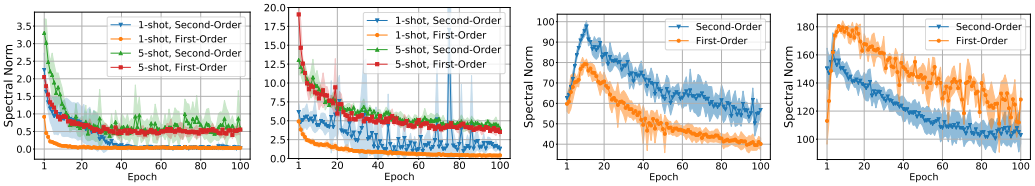
5 Experiments

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). 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 Appendix A

We closely follow the experimental setup of (Finn et al., 2017). Except for the Finetune baseline, the meta-learning algorithms use during meta-training the same number of ways and shots as during meta-testing. For our experiments, we follow the setting of (Vinyals et al., 2016): for MiniImagenet, training and testing our models on 5-way classification 1-shot learning, as well as 5-way 5-shot, and for Omniglot, 5-way 1-shot; 5-way 5-shot; 20-way 1-shot; 20-way 5-shot. Each experiment was repeated for five independent runs. For the meta-learning algorithms, the choice of hyperparameters closely follows (Finn et al., 2017). For our finetuning baseline, most of the original MAML hyperparameters were left unchanged, as we want to compare the effect of the pre-training procedure, thus are kept fixed the architecture and meta-test procedures. We kept the same optimizer as for the meta-update of MAML (ADAM), and performed hyperparameter search on the mini-batch size to use, for each setting that we present. (For our reproduction results on the meta-train and meta-test accuracy, see Figure 10a and 10b in B.1.)

5.1 Flatness of meta-test solutions

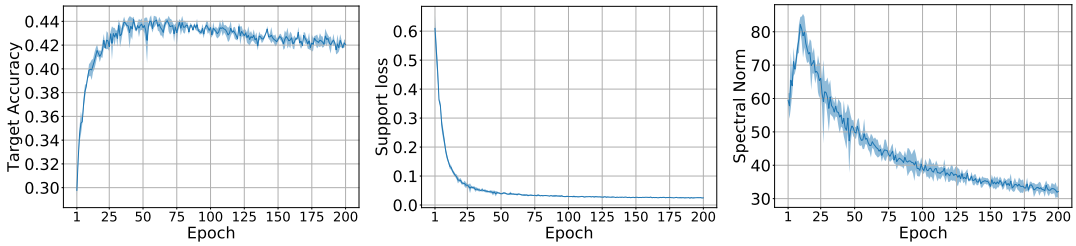
After each training epoch, we compute $\mathbb{E}[\|H\ (\mathcal{D}_i;\ \tilde{}_i)\|\ ]$ using a fixed set of 60 randomly sampled meta-test tasks $\mathcal{T}_i$. Across all settings, we observe that MAML first finds sharper solutions $\tilde{}_i$ until reaching a peak, then as the number of epoch grows, those solutions become flatter, as seen in Figure 2. To verify the correlation between $\mathbb{E}[\|H\ (\mathcal{D}_i;\ \tilde{}_i)\|\ ]$ and $\mathbb{E}[Acc(f(\mathcal{D}_i^{0};\ \tilde{}_i))]$, we train models for an extra number of epochs until clearly observing a decrease in the generalization performance $\mathbb{E}[Acc(f(\mathcal{D}_i^{0};\ \tilde{}_i))]$, using First-Order MAML with 5-way 1-shot learning on MiniImagenet, and we



(a) Omniglot 5-way (b) Omniglot 20-way (c) MiniImagenet 5-way, 1-shot (d) MiniImagenet 5-way, 5-shot

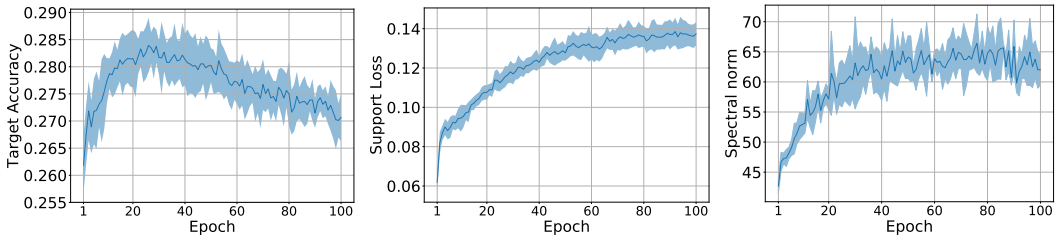
Figure 2: Flatness of meta-test solutions for MAML and First-Order MAML, on Omniglot and MiniImagenet

verify if it is reflected by an increase in $\mathbb{E}[\|H(\mathcal{D}_i; \tilde{\theta}_i)\|]$. On the contrary, and remarkably, even as f starts to show poorer generalization (see Figure 3a), the solutions keep getting flatter, as shown in Figure 3c. Thus for the case of gradient-based meta-learning, flatter minima don't appear to favour better generalization. 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 an 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.



(a) Target Accuracy (b) Support loss (c) Curvature of solutions

Figure 3: MAML: Characterization of meta-test solutions

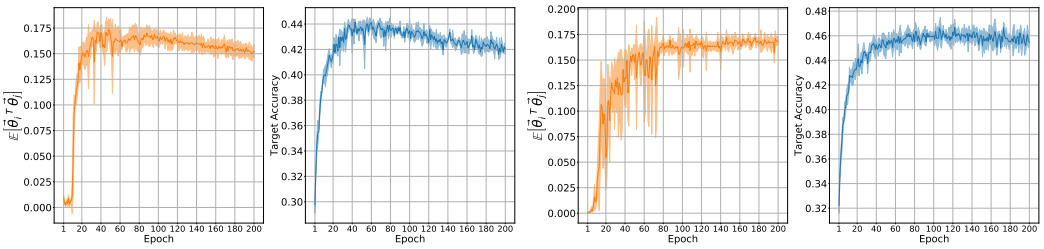


(a) Target accuracy (b) Support loss (c) Curvature of solutions

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

5.2 Coherence of adaptation trajectories

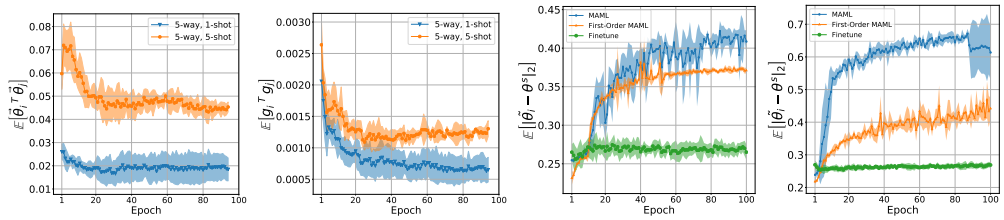
In this section, we use the same experimental setup as in Section 5.1, except here we measure $\mathbb{E}[\tilde{\theta}_i^T \tilde{\theta}_j]$. To reduce the variance on our results, we sample 500 tasks after each meta-training epoch. Also for experiments on Omniglot, we drop the analyses with First-Order 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.



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

Figure 5: Comparison between average inner product between meta-test trajectory direction vectors (orange), and average target accuracy on meta-test tasks (blue), MAML First-Order and Second-Order, MiniImagenet 5-way 1-shot. See Figure 11 in Appendix B.2 for full set of experiments.

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 11 in Appendix B.2 for full set of experiments). We also perform the analysis on the Finetuning baselines, which reach much lower target accuracies, and where we see that $\mathbb{E}[\tilde{}_i{}^T \tilde{}_j]$ 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 $\mathbb{E}[\|\tilde{}_i - {}^s\|_2]$, in Figure 6c and 6d, where $\mathbb{E}[\|\tilde{}_i - {}^s\|_2]$ 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.



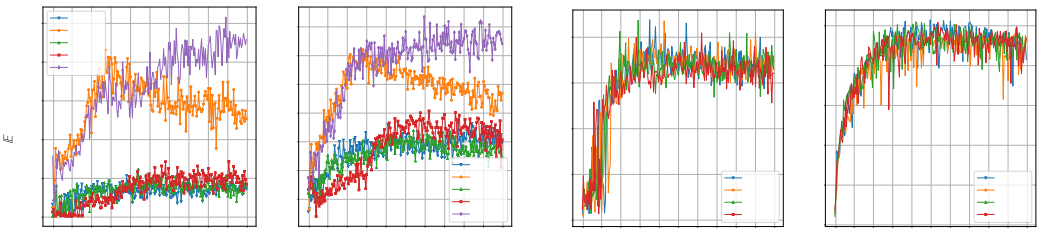
(a) Trajectories coherence (b) Gradients coherence (c) l_2 norm of trajectories (1-shot) (d) l_2 norm of trajectories (5-shot)

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. Average l_2 norm of meta-test adaptation trajectories, all algorithms on MiniImagenet, (c): 1-shot learning, (d): 5-shot learning.

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

Despite the clear correlation between $\mathbb{E}[\tilde{}_i{}^T \tilde{}_j]$ and generalization for the settings that we show in Figure 5 and 11, 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. Here we present our results on the characterization of the objective landscapes at the meta-train solutions ^{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 12 in Appendix B.3 for the full set of



(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 12 in Appendix B.3 for full set of experiments.

experiments. This metric consistently correlates with generalization across the different settings. Similarly as in Section 5.2, for our finetuning 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 6.

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. Results are presented in Figure 8. Similar to our analysis of 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 $\mathbb{E}[\,\mathbf{g}_i^T \mathbf{g}_j\,]$. See Appendix A.4 for the experimental details.

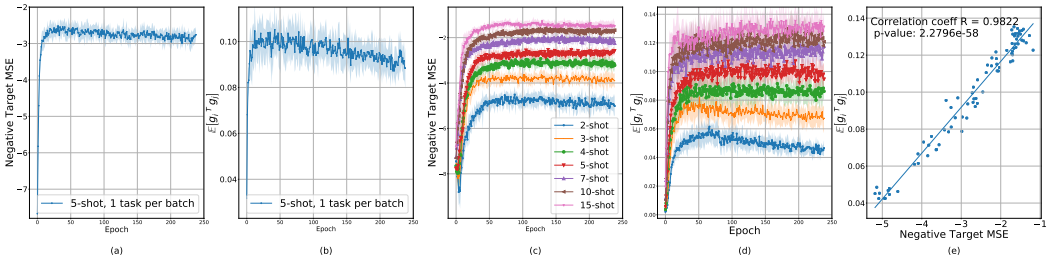


Figure 8: Analysis for Few-shot regression. Comparison between $\mathbb{E}[\,\mathbf{g}_i^T \mathbf{g}_j\,]$ and average negative target Mean Squared Error on meta-test tasks(generalization performance). (a) and (b) show generalization performance correlates with $\mathbb{E}[\,\mathbf{g}_i^T \mathbf{g}_j\,]$ 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.

6 First steps towards regularizing MAML

Although, MAML has become a popular method for meta-training, there exist a significant generalization gap between its performance on target set of the meta-train tasks and the target set of the meta-test task, and regularizing MAML has not received much research attention yet. Based on our observations on the coherence of adaptation trajectories, we take *first steps* in this direction by adding a regularization term based on $\mathbb{E}[\tilde{}_i^T \tilde{}_j]$. 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 $\tilde{} = \frac{1}{n}\sum_{i=1}^{n} \tilde{}_i$. For each task, we want to reduce the angle defined by $\tilde{}_i^T \tilde{}$, and thus introduce the penalty on

$(\quad) = -\tilde{}_i^T \tilde{}$, obtaining the regularized solutions $\hat{}_i$. The outer loop gradients are then computed, just like in MAML following Eq 2, but using these regularized solutions $\hat{}_i$ instead of $\tilde{}_i$. 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 9, 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

1: Sample a batch of n tasks $\mathcal{T}_i \sim p(\mathcal{T})$
2: **for all** $\mathcal{T}_i$ **do**
3: Perform inner loop adaptation as in Eq. 1: $\tilde{}_i = {}^s - \sum_{t=0}^{T-1} \nabla \mathcal{L}(f(\mathcal{D}_i; {}_i^{(t)}))$
4: **end for**
5: Compute the average direction vector: $\tilde{} = \frac{1}{n} \sum_{i=1}^{n} \tilde{}_i$
6: Compute the corrected inner loop updates:
7: **for all** $\mathcal{T}_i$ **do**
8: $\hat{}_i = \tilde{}_i - \nabla (\quad)$ where $(\quad) = -\tilde{}_i^T \tilde{}$
9: **end for**
10: Perform the meta-update as in Eq. 2, but using the corrected solutions: $^{s+1} = {}^s - \frac{1}{n} \sum_{i=1}^{n} \nabla \mathcal{L}(f(\mathcal{D}_i^0; \hat{}_i))$

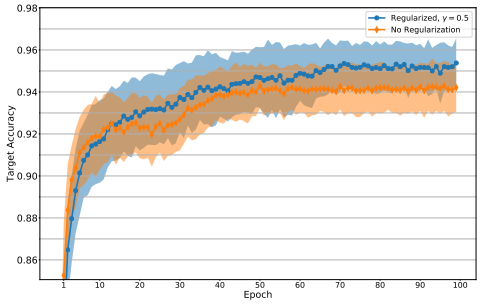


Figure 9: Average target accuracy on meta-test tasks using our proposed regularizer on MAML, for Omniglot 20-way 1-shot learning, with regularization coefficient $= 0:5$

7 Conclusion

We experimentally demonstrate that when using gradient-based meta-learning algorithms such as MAML, meta-test solutions, obtained after adapting neural networks to new tasks via few-shot learning, become flatter, lower in loss, and further away from the meta-train solution, as meta-training progresses. We also show that those meta-test solutions keep getting flatter even when generalization starts to degrade, thus providing an experimental argument against the correlation between generalization and flat minima. More importantly, we empirically show that generalization to new tasks is correlated with the coherence between their adaptation trajectories, measured by the average cosine similarity between the adaptation trajectory directions, but also correlated with the coherence between the meta-test gradients, measured by the average inner product between meta-test gradient vectors evaluated at meta-train solution. We also show this correlation for few-shot regression tasks. 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. (2018). Measuring and regularizing networks in function space. *CoRR*, abs/1805.08289.

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

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

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *CoRR*, abs/1703.03400.

Freeman, C. D. and Bruna, J. (2016). Topology and Geometry of Half-Recti ed Network Optimization. *ArXiv e-prints*

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

Glorot, X. and Bengio, Y. (2010). Understanding the dif culty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Arti cial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Hochreiter, S. and Schmidhuber, J. (1997). Flat minima. *Neural Comput.*, 9(1):1–42.

Jastrzebski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. J. (2017). Three factors in uencing minima in SGD. *CoRR*, abs/1711.04623.

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

Li, H., Xu, Z., Taylor, G., and Goldstein, T. (2017). Visualizing the loss landscape of neural nets. *CoRR*, abs/1712.09913.

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

Pascanu, R., Dauphin, Y. N., Ganguli, S., and Bengio, Y. (2014). On the saddle point problem for non-convex optimization. *CoRR*, abs/1405.4604.

Ravi, S. and Larochelle, H. (2017). 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*

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

Smith, S. L. and Le, Q. V. (2017). A bayesian perspective on generalization and stochastic gradient descent. *CoRR*, abs/1710.06451.

Snell, J., Swersky, K., and Zemel, R. S. (2017). Prototypical networks for few-shot learning. *CoRR*, abs/1703.05175.

Vinyals, O., Blundell, C., Lillicrap, T. P., Kavukcuoglu, K., and Wierstra, D. (2016). Matching networks for one shot learning. *CoRR*, abs/1606.04080.

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

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 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 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 the k-shot, m-way 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 fine-tuning 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 fine-tuning 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 and Bengio, 2010) to initialize the weights of the final layer.

A.4 EXPERIMENTAL DETAILS FOR FEW-SHOT REGRESSION

For the few-shot regression problems (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.

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 10. They include the target accuracies on meta-train tasks and on meta-test tasks (generalization), as meta-training progresses.

(a) Meta-Train Accuracy (b) Meta-Test Accuracy

Figure 10: MAML: Accuracies on training and testing tasks

B.2 COHERENCE OF ADAPTATION TRAJECTORIES

The relation between target accuracy on meta-test tasks, and angles between trajectory directions is presented in Figure 11.

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 12.

(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 11: 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 12: Comparison between average inner product between trajectory displacement vectors, and average target accuracy on meta-test tasks. Full set of experiments.