On the Convergence of Model-Agnostic Meta-Learning

Noah Golmant
UC Berkeley
noah.golmant@berkeley.edu

Abstract

We analyze the behavior of a popular gradient-based meta-learning algorithm from an optimization perspective. We show that for a large class of functions, Model-Agnostic Meta-Learning implicitly optimizes for sample efficiency in a few-shot learning context by converging to an initialization that prioritizes tasks with more difficult optimization landscapes. Additionally, we show that MAML generalizes to problems whose solutions and optimization landscapes are sufficiently similar to those of the constituent training objectives.

1 Introduction

The ability to learn quickly is a hallmark of intelligence. We often hope that good prior solutions to problems will enable us to learn how to solve new tasks more quickly. In a sense, we would like to optimize our solutions for adaptability to new tasks. Gradient-based meta-learning techniques attempt to formalize this intuition through stochastic optimization. The most popular among these techniques is model-agnostic meta-learning (MAML), which is able to learn good initialization points for gradient descent on multiple objective functions. This enables gradient descent to arrive at a good solution for a particular task with only a few additional training points.

The goal of this work is to understand how particular notions of "task similarity" can help us understand MAML's behavior. To do so, we examine convergence properties of MAML on a family of objectives for which task similarity can be reduced to a few easily analyzed quantities. These assumptions provide a suitable venue for exploring the role of loss surface curvature in the MAML optimization procedure, since they delineate reasonable expectations about the landscape.

We first study the optimization landscape of the MAML objective in its own right. As a distinct loss function, the MAML objective has received relatively little attention in terms of understanding its convergence properties and behavior under gradient descent dynamics. The shape of this "meta-objective" may give insights into how MAML implicitly compares training objectives throughout the training process.

Motivated by the MAML objective analysis, we propose an "optimization landscape" perspective on task similarity to better understand the MAML solution. This perspective enables us to derive a bound on the location of the MAML "meta-model" in terms of the condition numbers of the training tasks and the locations of task optima in parameter space. We show that we can interpret the meta-model as a sort of "curvature-weighted average" of the training task optima, since the weight on a given task depends on the spectrum of its Hessian. We justify this interpretation by taking a look at a family of quadratic objectives for which we can derive an analytic expression for the MAML solution. We conclude by discussing implications for many-task generalization, as well as how MAML optimizes for sample efficiency in a few-shot learning setup.

2 Related Work

2.1 Model-Agnostic Meta-Learning

In MAML [FAL17], the goal of the optimization procedure is to learn a set of parameters from which one can quickly adapt to a new task using gradient descent. Let $\mathcal{F} = \{f_i : \mathbb{R}^d \to \mathbb{R}\}_{i=1}^n$ be a collection of objective functions. For a fixed step size $\eta > 0$, let $G_i : \mathbb{R}^d \to \mathbb{R}^d$ denote the gradient descent update for f_i:

$$G_i(x) = x - \eta \nabla f_i(x) \tag{1}$$

Consider a probability distribution over $\mathcal{F}$ such that $\mathbb{P}(f_i) = \theta_i, \theta_i \geq 0, \sum_{i=1}^n \theta_i = 1$. The goal of MAML is to minimize the expected loss when you first perform one step of gradient descent on a task. Concretely, the objective is

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \;\; L(x) := \sum_{i=1}^n \theta_i (f_i \circ G_i)(x) \tag{2}$$

Where $\circ$ denotes function composition. After producing a "meta-model" x^* that minimizes L, we observe some training data for a new task. By using x^* as the initialization for gradient descent on this task, we hope to find the task-specific minimum x_i^* in only a few steps. Although the MAML objective is only a "one-shot" expectation over the original objectives, it manages to perform well in few-shot learning environments. In such problems, the model is constrained to learn how to solve many tasks with only a few training points for each task. MAML's success has enabled the use of deep learning techniques in regimes with low amounts of per-task training data.

2.2 Prior work and analysis

MAML is one of several gradient-based meta-learning techniques. Prominent examples of algorithms in this family include Reptile [NAS18], Meta-SGD [Li+17], and Snail [Mis+17]. In most gradient-based approaches, the objective is formulated based on some expectations about how the primary optimization procedure will behave once the meta-learning algorithm has reached some solution. These inductive biases play an essential role in determining the meta-optimization procedure and its stability. Moreover, they imply a certain set of judgements about how two tasks should be related in order for the algorithm to perform well.

In [FL17], the authors show that gradient-based meta-learning approaches can approximate any learning algorithm, and that such learning strategies tend can generalize well to a variety of tasks. More recent work recasts MAML in a Bayesian hierarchical modeling framework to formalize meta-learning as inference of parameters that are shared across tasks [Gra+18]. MAML has also achieved significant successes in domains like robotics and reinforcement learning [Yu+18]. These works indicate that MAML provides a simple framework for meta-learning that makes reasonable assumptions about task relatedness through parameter similarity, and that the inductive biases of MAML have a strong basis for solving real-world tasks.

Some other work has analyzed the bias introduced by short time-horizon bias in gradient-based meta-learning [Wu+18]. For non-convex objectives with many spurious local minima and saddle points, the local gradient information utilized by gradient-based meta-learning techniques might be insufficient to estimate long-term optimization trajectories. This significantly complicates any analysis of the gradient-based meta-learning objectives. However, for more well-behaved objectives, we might expect short time-horizon gradient information to be sufficient. We focus on such objectives here to provide an initial analysis of MAML behavior from an optimization perspective.

3 Analysis of the MAML Objective

In this section, we will consider a class of functions for which we have much more knowledge about the behavior of gradient descent on any particular task. Although several of the subsequent theorems can be demonstrated with weaker conditions, we focus on this case to derive more explicit guarantees on the behavior of MAML on these objectives in the aggregate.

Assumption 1. Each objective $f : \mathbb{R}^d \to \mathbb{R}$ is α_f-strongly convex. For all $x, y \in \mathbb{R}^d$,

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \alpha_f \|x - y\|^2 \tag{3}$$

Assumption 2. Each objective $f : \mathbb{R}^d \to \mathbb{R}$ is β_f-smooth. For all $x, y \in \mathbb{R}^d$,

$$\|\nabla g(x) - \nabla g(y)\| \leq \beta_f \|x - y\| \tag{4}$$

Assumption 3. Each objective is C_f-Hessian Bi-Lipschitz. For all $x, y \in \mathbb{R}^d$,

$$\frac{1}{C_f}\|x - y\| \leq \|\nabla^2 f(x) - \nabla^2 f(y)\|_*^2 \leq C_f \|x - y\| \tag{5}$$

Where $\|\cdot\|_*$ denotes the spectral norm.

The last assumption is a statement about variability and degeneracy in the loss surface. In particular, the mapping $x \mapsto \nabla^2 f(x)$ is a homeomorphism, i.e. we can continuously identify a point x with the second-order data provided to us by f.This assumption also constrains our Hessian to lie in an n-dimensional subspace of $\mathbb{R}^{n \times n}$.

The MAML objective (2) is an expectation over "one-shot" versions of the constituent functions in $\mathscr{F}$. So in order to understand (2) and its gradient descent dynamics, we must understand the loss landscape of these one-shot updates. Let $f : \mathbb{R}^d \to \mathbb{R}$ be an objective function, and let $G : \mathbb{R}^d \to \mathbb{R}^d$ be the gradient descent update for f:

$$G(x) = x - \eta \nabla f(x)$$

Define $F(x) = f \circ G$ to be the one-shot version of f. The next theorem essentially states that for a nicely behaved choice of f, the one-shot objective is "almost" as well-behaved. The convexity and smoothness parameters of F depends on the bi-Lipschitz parameter C, which represents our higher-order regularity assumptions about change in the loss landscape.

Theorem 1. *Suppose $f : \mathbb{R}^d \to \mathbb{R}$ is an α-strongly convex, β-smooth, and C-Hessian bi-Lipschitz function. Then F is α'-strongly convex and β'-smooth, where $\alpha' = \frac{\eta}{C}(1 - \eta\beta), \beta' = \eta C(1 - \eta\alpha)$. Call $\kappa = \beta/\alpha$ the condition number of f. If $\eta \leq \frac{2}{\alpha+\beta}$, then $\alpha' \geq \frac{\eta}{C}\left(\frac{\kappa-1}{\kappa+1}\right)$.*

Proof. Let $E(x) = DG(x) = I - \nabla^2 f(x)$ be the derivative of the gradient descent update at x. $E(x)$ represents a linearization of the dynamics of gradient descent around the point x. Let $x' = G(x), y' = G(y)$. Then we have that

$$\|\nabla F(x) - \nabla F(y)\| = \|E(x)\nabla f(x') - E(y)\nabla f(y')\| \tag{6}$$

Since $E(x) - E(y) = \eta\left(\nabla^2 f(x) - \nabla^2 f(y)\right)$, we get the following short lemma.

Lemma 1. *Suppose $\nabla^2 f(x)$ is C-Hessian Bi-Lipschitz and our step size for gradient descent is $\eta > 0$. Then $\frac{1}{\eta}E(x)$ is C-Hessian Bi-Lipschitz, i.e.*

$$\frac{\eta}{C}\|x - y\| \leq \|E(x) - E(y)\| \leq \eta C \|x - y\| \tag{7}$$

To finish the claim, we use the following lemma, which provides bounds on the contractive behavior of the gradient descent update G.

Lemma 2. *Let $G(x) = x - \eta\nabla f(x)$ be the gradient descent update for f with step size $\eta > 0$, where f is α-strongly convex and β-smooth. Then G satisfies*

$$(1 - \eta\beta)\|x - y\| \leq \|G(x) - G(y)\| \leq (1 - \eta\alpha)\|x - y\| \tag{8}$$

Proof. It suffices to show that $DG(x) = I - \eta\nabla^2 f(x)$ has bounded spectral norm. Indeed, for any $v \in \mathbb{R}^d$ with $\|v\| = 1$, we have

$$v^T\left(I - \eta\nabla^2 f(x)\right)v = 1 - \eta v^T \nabla^2 f(x) v \tag{9}$$

Strong convexity and smoothness then imply that $\alpha I \preceq \nabla^2 f(x) \preceq \beta I$, and the result follows. □

We can finish the theorem now. We start with smoothness. Starting from (6), we have

$$\|\nabla F(x) - \nabla F(y)\| \leq \eta C \|G(x) - G(y)\| \tag{10}$$

$$\leq \eta C(1 - \eta\alpha)\|x - y\| \tag{11}$$

Similarly, for strong convexity,

$$\|\nabla F(x) - \nabla F(y)\| \geq \frac{\eta}{C}\|G(x) - G(y)\| \tag{12}$$

$$\geq \frac{\eta}{C}(1 - \eta\beta)\|x - y\| \tag{13}$$

□

Next, it's worthwhile to check that the one-shot objective has the same minimizer as the original objective.

Theorem 2. *Let f be as in Theorem 1. Then $F = f \circ G$ has the same unique minimizer x_f^* as f.*

Proof. The gradient of F is given by

$$\nabla F(x_0) = (I - \eta\nabla^2 f(x_0))\nabla f(x') \tag{14}$$

Where $x' = G(x_0)$. Since $\eta \leq \frac{2}{\beta+\alpha}$, $I - \eta\nabla^2 f(x_0)$ is nonsingular for all x_0. Hence, $\nabla F(x_0) = 0$ if and only if $\nabla f(x') = 0$. Since we chose our step size to be small enough, by Lemma 2 $\nabla f(x') = 0$ if and only if $x_0 = x_f^*$. □

With the bulk of our work done, we can finally state the main result of the section, which follows simply from looking at the Rayleigh quotient of $\nabla^2 L(x)$.

Theorem 3. *Suppose each $f_i \in \mathscr{F}$ satisfies the conditions of Theorem 1 with constants α_i, β_i and C_i. Call α_i', β_i' the resulting constants for $F_i = f_i \circ G_i$. Then the MAML objective L is A-strongly convex and B-smooth, for $A = \sum_{i=1}^n \theta_i\alpha'$ and $B = \sum_{i=1}^n \theta_i\beta_i'$.*

This theorem has a couple consequences.

- These constants represent a weighted sum of the one-shot convexity and smoothness constants. In this sense, the convergence rate of gradient descent on L depends strongly on the convergence rates of the objective functions $\mathscr{F}$. The learning rate η plays a role in determining this "difficulty" of minimizing L, since by scaling the learning rate down, I "flatten out" the resulting loss landscape for the one-shot objectives.
- The strong convexity of L implies that the MAML objective has a unique minimizer x^*. This minimizer in some sense manages to balance the competing trajectories of the one-shot objectives, giving more weight to those tasks whose loss functions are less "well-behaved" in the one-shot case. For a particular objective f_i, as α_i diverges from β_i and the condition number increases, we observe a weaker bound on the strong convexity of F_i. This, in turn, attenuates the weight of α_i in the average, A.

4 What is the MAML optimum?

Now that we have managed to show the uniqueness of the MAML solution, it is worthwhile to take a closer look at it to better understand potentially useful notions of task similarity. We first first consider simple quadratic objectives where the Hessian is readily available. Then, we make a preliminary attempt to extend this result to objectives satisfying our previously stated assumptions.

4.1 Building Intuition

To build intuition for the form of the MAML solution, we will consider a simple case in which we can derive an analytic expression for the minimizer of L. In particular, we will look at quadratic objectives with positive definite matrices. Let $f(x) = \frac{1}{2}(x-a)^T J(x-a), h(x) = \frac{1}{2}(x-b)^T H(x-b)$ for task-specific minima $a, b \in \mathbb{R}^d$, with J, H positive definite matrices.

We first calculate the one-shot versions of f and h. We deal with $F = f \circ G$ since the second case is treated identically. We first calculate

$$\begin{aligned} G(x) &= x - \eta \nabla f(x) \\ &= x - \eta J(x - a) \end{aligned}$$

The resulting one-shot loss is then

$$\begin{aligned} F(x) &= (G(x) - a)^T J(G(x) - a) \\ &= (x - a)^T (I - \eta J) J (I - \eta J)(x - a) \end{aligned}$$

This looks like another quadratic objective whose parameters are given by an "atennuated" version of the original matrix J. The eigenvalues of this new objective matrix depend on the condition number κ of J as well as the original eigenvalues. For example, when we set $\eta = 2/(\lambda_{min}(J) + \lambda_{max}(J))$, the largest eigenvalue of this new matrix is $\left(\frac{\kappa-1}{\kappa+1}\right)^2 \lambda_{max}$. We can see that for diagonal matrices, the ith diagonal element of the new objective is simply $(1 - \eta J_{ii})^2 J_{ii}$, and the minimizer for this objective is still a.

We will weight each objective equally, so the resulting MAML loss is $L(x) = \frac{1}{2}(F(x) + G(x))$. Let's call $D_J = (I - \eta J)J(I - \eta J)$, and $D_H = (I - \eta H)H(I - \eta H)$. Since we know L has a unique minimizer, we set $\nabla L(x) = 0$ and solve:

$$\nabla L(x) = D_J(x - a) + D_H(x - b) = 0 \tag{15}$$

As a result, our minimizer is $x^* = (D_J + D_H)^{-1}(D_J a + D_H b)$. This is a sort of matrix-weighted average of the two minima. In fact, when F and H are diagonal, x^* is a coordinate-wise weighted average of the optima, with coordinate weights given by the eigenvalues of D_J and D_H. As a result, the solution in the quadratic case is a curvature-weighted average of the per-task solutions.

4.2 The General Case

We can extend the above analysis to a more general case by using the same assumptions as Theorem 2. From our previous work, we know that the MAML objective L has a unique minimizer x^* where $\nabla L(x^*) = 0$. By using A-strong convexity of L, we have that

$$\begin{aligned} \|x - x^*\| &\leq \frac{1}{A}\|\nabla L(x) - \nabla L(x^*)\| = \frac{1}{A}\|\nabla L(x)\| \quad (16) \\ &\leq \frac{1}{A}\sum_{i=1}^{n} \theta_i \left(\frac{\kappa_i - 1}{\kappa_i + 1}\right) \|\nabla G_i(x)\| \quad (17) \\ &\leq \frac{1}{A}\sum_{i=1}^{n} \theta_i \left(\frac{\kappa_i - 1}{\kappa_i + 1}\right)^2 \|x - x_i^*\| \quad (18) \end{aligned}$$

This bound is minimized at $x = \sum_{i=1}^{n} \theta_i \left(\frac{\kappa_i-1}{\kappa_i+1}\right)^2 x_i^*$, which is a weighted average of the per-task optima. There are a couple interesting features to note:

- This bound is tight, in the sense that there is a family of objectives where the MAML solution x^* is equal to this weighted average. Consider our previous quadratic example in two dimensions, where $J = H = \text{diag}(\alpha, \beta)$. We can exactly achieve this weighted average by selecting a learning rate $\eta = \frac{2}{\alpha+\beta}$.
- The closeness of the solution to this weighted average depends on the strong convexity parameter of L and the learning rate η in a natural way. As we increase η to the maximum stable value, this weighted average becomes a better representative for the MAML solution. We can see this phenomenon in our optimal selection of η in the quadratic case. The location of the MAML optimum relative to this average depends on the overall convexity of L.

More significantly, this weighted average is biased towards optima with higher condition numbers. This implies a weak preference for tasks that have lower convergence rates for gradient descent.

From a sample efficiency perspective, this means that MAML uses the large number of samples aggregated over *all* tasks to move closer to the optima for difficult tasks.

Normally, it would be difficult to achieve a good solution for such tasks using gradient descent – in a few-shot learning environment, we only have a small number of examples per task, and each sample moves us marginally closer to the task optimum. But if we have enough tasks, MAML can overcome this issue. It can use training data for other tasks to "nudge" towards the optima for these more difficult ones. Because of its bias towards these harder tasks, the "fine-tuning" step in MAML will produce a set of task-specific parameters with relatively balanced error, despite the initial differences in underlying task difficulty. To see this, suppose all tasks are weighted equally. Then the MAML solution will be closer to the optima for tasks with higher condition numbers. Hence, after one iteration of gradient descent on each task, the per-task solutions will all be about the same distance away from their respective optima.

5 First-order approximations

In the original MAML paper, the authors proposed a method to solve an approximation to the original MAML objective that doesn't rely on computing the Hessian at each step. This method is computationally feasible for models with a large number of parameters. In the one-shot objective F, the gradient is given by

$$\Delta(x) = (I - \eta\nabla^2 f(x))\nabla f(x')|_{x'=G_f(x)}$$

In a first-order approximation of the MAML update, we assume that $I - \eta\nabla^2 f(x) \approx I$, and we end up with an approximation to the gradient that looks like

$$\hat{\Delta}(x) = \nabla f(x')|_{x'=G_f(x)}$$

This approximation works well in practice, and significantly speeds up the MAML training process. This is a decent approximation because the Hessian for heavily over-parameterized models like neural networks is highly degenerate, with only a few non-zero eigenvalues. Even without a degeneracy assumption, this helps when the learning rate is very small relative to the smoothness parameter β.

By making this approximation, we can make more statements about convergence of MAML for a wider class of objective functions. In particular, we no longer need the bi-Lipschitz condition on the Hessian to talk about convergence. The "attenuation" by a factor of C is replaced by the convexity and smoothness constants for the original objective.

Theorem 4. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a α-strongly convex and β-smooth, with step size $0 < \eta < 2/(\alpha+\beta)$. Let $\hat{F}$ denote the approximation to the one-shot objective so that $\nabla\hat{F}(x) = \hat{\Delta}(x)$. Then $\hat{F}$ is α'-strongly convex and β'-smooth for $\alpha' = \left(\frac{\kappa-1}{\kappa+1}\right)\alpha$ and $\beta' = (1-\eta\alpha)\beta$.*

Proof. We demonstrate the same Lipschitz conditions as before. Let $x, y \in \mathbb{R}^d$. Then by smoothness of f and Lemma 2, we have

$$\begin{aligned}\|\hat{\Delta}(x) - \hat{\Delta}(y)\| &\leq \beta\|G_f(x) - G_f(y)\| \\ &\leq (1-\eta\alpha)\beta\|x-y\|\end{aligned}$$

Similarly, by strong convexity and the same lemma, we get

$$\begin{aligned}\|\hat{\Delta}(x) - \hat{\Delta}(y)\| &\geq \alpha\|G_f(x) - G_f(y)\| \\ &\geq (1-\eta\beta)\alpha\|x-y\| \\ &\geq \left(\frac{\kappa-1}{\kappa+1}\right)\alpha\|x-y\|\end{aligned}$$

□

This means that the first-order approximation allows us to apply MAML to a larger class of objectives by assuming that the Hessian is relatively small. Interestingly, this new assumption removes the linear dependence on η that we previously observed for the one-shot strong convexity parameter α'. In this sense, the behavior of the approximation depends more on the convexity and smoothness parameters than tunable hyperparameters like the learning rate.

6 Discussion

In this work, we analyzed the behavior of MAML on a somewhat reasonable class of objective functions. We found that the MAML solution is "close to" a weighted average of the task optima, with weights based on the condition numbers of the tasks. This results in a sort of heuristic sample efficiency argument, but an important next step is to extend this to derive finite-sample error guarantees for tasks starting from a MAML initialization. This would allow us to identify some sort of balance between sample size and task similarity to determine whether or not MAML is useful in a particular setup. For example, if we have a huge number of samples for each task, and the task optima are very far apart, then it would be more effective to independently train several models using plain gradient descent. However, if the task optima are all close to each other and the number of samples is small, MAML should be more effective than a random initialization.

The most useful way to characterize this tradeoff would be through a probabilistic argument based on random initializations. We can consider two possible initialization schemes:

- Select a starting point for per-task gradient descent in a ball of some fixed radius about the task minimum.
- First, select a MAML starting point in a ball centered around all the task optima. Then, descend the MAML objective for some number of steps. Finally, use this as the starting point for per-task gradient descent.

There is a balance between these two strategies that depends on the contractive properties of the gradient descent update map $G(x)$, and hence the approximated dynamics of gradient descent on that task.

It is also not clear how essential the bi-Lipschitz assumption is for the exact objective case, although some control over the continuity of the Hessian is probably necessary to bound the behavior of the second-order term in the gradient. Relaxing these assumptions would be useful – it would allow us to think about more functions, since the Hessian would no longer have to lie in a very low-dimensional submanifold.

Another very important direction would be to explore the behavior of MAML when we receive the gradient through a stochastic oracle. In that case, we are working with a noisy "inner gradient" in the MAML update. This noise may make the algorithm less stable. However, for non-convex problems, it could induce exploratory behavior like the anisotropic noise of mini-batch SGD in non-convex optimization [Xin+18].

The most interesting direction we would like to explore is MAML analysis for non-convex objectives. In Reptile [NAS18], the authors motivate the algorithm by appealing to the notion of a solution manifold. This brings to mind recent work in analyzing the behavior of gradient descent from a dynamical systems theory perspective, by applying tools like the Stable Manifold Theorem to analyze the behavior of gradient descent near saddle points [Lee+16]. This analysis could yield additional insight into the behavior of real-world applications of MAML that rely on neural networks.

Finally, one last avenue would be to analyze more recent extensions of MAML, as well as alternatives like Reptile [NAS18], which promise simpler algorithms with competitive results on few-shot classification tasks. It is not clear whether other gradient-based meta-learning techniques exhibit similar convergence behavior, since they rely on different inductive biases about the behavior of the optimization procedure.

References

[Lee+16] Jason D. Lee et al. *Gradient Descent Converges to Minimizers*. 2016. eprint: arXiv:1602.04915.

[FAL17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks”. In: *CoRR* abs/1703.03400 (2017). arXiv: 1703.03400. URL: http://arxiv.org/abs/1703.03400.

[FL17] Chelsea Finn and Sergey Levine. “Meta-Learning and Universality: Deep Representations and Gradient Descent can Approximate any Learning Algorithm”. In: *CoRR* abs/1710.11622 (2017). arXiv: 1710.11622. URL: http://arxiv.org/abs/1710.11622.

[Li+17] Zhenguo Li et al. “Meta-SGD: Learning to Learn Quickly for Few Shot Learning”. In: *CoRR* abs/1707.09835 (2017). arXiv: 1707.09835. URL: http://arxiv.org/abs/1707.09835.

[Mis+17] Nikhil Mishra et al. “Meta-Learning with Temporal Convolutions”. In: *CoRR* abs/1707.03141 (2017). arXiv: 1707.03141. URL: http://arxiv.org/abs/1707.03141.

[Gra+18] Erin Grant et al. *Recasting Gradient-Based Meta-Learning as Hierarchical Bayes*. 2018. eprint: arXiv:1801.08930.

[NAS18] Alex Nichol, Joshua Achiam, and John Schulman. *On First-Order Meta-Learning Algorithms*. 2018. eprint: arXiv:1803.02999.

[Wu+18] Yuhuai Wu et al. *Understanding Short-Horizon Bias in Stochastic Meta-Optimization*. 2018. eprint: arXiv:1803.02021.

[Xin+18] Chen Xing et al. *A Walk with SGD*. 2018. eprint: arXiv:1802.08770.

[Yu+18] Tianhe Yu et al. “One-Shot Imitation from Observing Humans via Domain-Adaptive Meta-Learning”. In: *CoRR* abs/1802.01557 (2018). arXiv: 1802.01557. URL: http://arxiv.org/abs/1802.01557.