META-LEARNING IN REPRODUCING KERNEL HILBERT SPACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Model Agnostic Meta-Learning (MAML) has emerged as a standard framework for meta-learning, where a *meta-model* is learned with the ability of fast adapting to new tasks. However, as a double-looped optimization problem, MAML needs to differentiate through the whole inner-loop optimization path for every outer-loop training step, which may lead to both computational inefficiency and sub-optimal solutions. In this paper, we generalize MAML to allow meta-learning to be defined in function spaces, and propose the first meta-learning paradigm in the Reproducing Kernel Hilbert Space (RKHS) induced by the meta-model's Neural Tangent Kernel (NTK). Within this paradigm, we introduce two meta-learning algorithms in RKHS, which no longer need an explicit inner-loop adaptation as in the MAML framework. We achieve this goal by 1) replacing the adaptation with a fast-adaptive regularizer in the RKHS; and 2) solving the adaptation analytically based on the NTK theory. Extensive experimental studies demonstrate the superiority of our paradigm in both efficiency and quality of solutions compared to related meta-learning algorithms. Another interesting feature of our proposed methods is that they are much more robust to adversarial attacks and out-of-distribution adaptation than existing approaches, as demonstrated by our experiments.

1 INTRODUCTION

Meta-learning (Schmidhuber, 1987) has made tremendous progresses in the last few years. It aims to learn abstract knowledge from many related tasks so that fast adaption to new and unseen tasks becomes possible. For example, in few-shot learning, meta-learning corresponds to learning a meta-model or meta-parameters so that they can fast adapt to new tasks with a limited number of data samples. Among all existing meta-learning methods, Model Agnostic Meta-Learning (MAML) (Finn et al., 2017) is perhaps one of the most popular and flexible ones, with a number of follow-up works such as (Nichol et al., 2018; Finn et al., 2018; Yao et al., 2019; Khodak et al., 2019a;b; Denevi et al., 2019; Fallah et al., 2020; Lee et al., 2020; Tripuraneni et al., 2020). MAML adopts a double-looped optimization framework, where adaptation is performed at one or several gradient-descent steps in the inner-loop optimization. Such a framework could lead to some undesirable issues related to computational inefficiency and sub-optimal solutions. The main reasons are that 1) it is computationally expensive to back-propagate through a stochastic-gradient-descent chain, and 2) it is hard to tune the number of adaptation steps in the inner-loop as it can be different for both training and testing. Several previous works tried to address these issues, but they can only alleviate them to certain extents. For example, first order MAML (FOMAML) (Finn et al., 2017) ignores the high-order terms of the standard MAML, which can speed up the training but may lead to worse performance; MAML with Implicit Gradient (iMAML) (Rajeswaran et al., 2019) directly minimizes the objective of the outer-loop without performing the inner-loop optimization. But it still needs an iterative solver to estimate the meta-gradient.

To better address these issues, we propose two algorithms that generalize meta-learning to the Reproducing Kernel Hilbert Space (RKHS) induced by the meta-model's Neural Tangent Kernel (NTK) (Jacot et al., 2018). In this RKHS, instead of using parameter adaptation, we propose to perform an implicit function adaptation. To this end, we introduce two algorithms to avoid explicit function adaptation: one replaces the function adaptation step in the inner-loop with a new meta-objective for the outer-loop with a fast-adaptive regularizer inspired by MAML; the other solves the adaptation problem analytically based on tools from NTK so that the meta-objective can be

directly evaluated on samples in a closed-form. When restricting the function space to be RKHS, the solutions to the proposed two algorithms become conveniently solvable. In addition, we provide theoretical analysis on our proposed algorithms in the cases of using fully-connected neural networks and convolutional neural networks as the meta-model. Our analysis shows close connections between our methods and the existing ones. Particularly, we prove that one of our algorithms is closely related to MAML with some high-order terms ignored in the meta-objective function, thus endowing effective optimization. In summary, our main contributions are:

- We re-analyze the meta-learning problem and introduce two new algorithms for meta-learning in RKHS. Different from all existing meta-learning algorithms, our proposed methods can be solved conveniently in a single-looped optimization procedure.
- We conduct theoretically analysis on the proposed algorithms, which suggests that our proposed algorithms are closely related to the existing MAML methods when fully-connected neural networks and convolutional neural networks are used as the meta-model.
- We conduct extensive experiments to validate our algorithms. Experimental results indicate the effectiveness of our proposed methods, through standard few-shot learning, robustness to adversarial attacks and out-of-distribution adaptation.

2 PRELIMINARIES

2.1 META-LEARNING

Meta-learning can be roughly categorized as black-box adaptation methods (Andrychowicz et al., 2016; Graves et al., 2014; Mishra et al., 2018), optimization-based methods (Finn et al., 2017), non-parametric methods (Vinyals et al., 2016; Snell et al., 2017) and Bayesian meta-learning methods (Finn et al., 2018; Yoon et al., 2018; Ravi & Beatson, 2019). In this paper, we focus on the framework of Model Agnostic Meta-Learning (MAML) (Finn et al., 2017), which has two key components, meta initialization and fast adaptation. Specifically, MAML solves the meta-learning problem through a double-looped optimization procedure. In the inner-loop, MAML runs a task-specific adaptation procedure to transform a meta-parameter, $\boldsymbol{\theta}$, to a task-specific parameter, $\{\boldsymbol{\phi}_m\}_{m=1}^B$, for a total of B different tasks. In the outer-loop, MAML minimizes a total loss of $\sum_{m=1}^B \mathcal{L}(f_{\boldsymbol{\phi}_m})$ with respect to meta-parameter $\boldsymbol{\theta}$, where $f_{\boldsymbol{\phi}_m}$ is the model adapted on task m that is typically represented by a deep neural network. It is worth noting that in MAML, one potential problem occurs when computing the meta-gradient $\nabla_{\boldsymbol{\theta}} \sum_{m=1}^B \mathcal{L}(f_{\boldsymbol{\phi}_m})$. This gradient computation requires one to differentiate through the whole inner-loop optimization path, which could be very inefficient.

2.2 GRADIENT FLOW

Our proposed method relies on the concept of gradient flow. Generally speaking, gradient flow is a continuous-time version of gradient descent. In the finite-dimensional parameter space, a gradient flow is defined by an ordinary differential equation (ODE), $\mathrm{d}\boldsymbol{\theta}^t/\mathrm{d}t = -\nabla_{\boldsymbol{\theta}^t} F(\boldsymbol{\theta}^t)$, with a starting point $\boldsymbol{\theta}^0$ and function $F : R^d \to R$. Gradient flow is also known as steepest descent curve.

One can generalize gradient flows to infinite-dimensional function spaces. Specifically, given a function space $\mathcal{H}$, a functional $\mathcal{F} : \mathcal{H} \to R$, and a starting point $f^0 \in \mathcal{H}$, a gradient flow is similarly defined as the solution of $\mathrm{d}f^t/\mathrm{d}t = -\nabla_{f^t}\mathcal{F}(f^t)$. This is a curve in the function space $\mathcal{H}$. In this paper, we use notation $\nabla_{f^t}\mathcal{F}(f^t)$, instead of $\nabla_{\mathcal{H}}\mathcal{F}(f^t)$, to denote the general function derivative of the energy functional $\mathcal{F}$ with respect to function f^t (Villani, 2008).

2.3 THE NEURAL TANGENT KERNEL

Neural Tangent Kernel (NTK) is a recently proposed technique for characterizing the dynamics of a neural network under gradient descent (Jacot et al., 2018; Arora et al., 2019; Lee et al., 2019). NTK allows one to analyze deep neural networks (DNNs) in RKHS induced by NTK. One immediate benefit of this is that the loss functional in the function space is often convex, even when it is highly non-convex in the parameter space. This property allows one to better understand the property of DNNs. Specifically, let $f_{\boldsymbol{\theta}}$ be a DNN parameterized by $\boldsymbol{\theta}$. The corresponding NTK $\boldsymbol{\Theta}$ is defined as: $\boldsymbol{\Theta}(\mathbf{x}_1, \mathbf{x}_2) = \dfrac{\partial f_{\boldsymbol{\theta}}(\mathbf{x}_1)}{\partial \boldsymbol{\theta}} \dfrac{\partial f_{\boldsymbol{\theta}}(\mathbf{x}_2)}{\partial \boldsymbol{\theta}}^{\intercal}$, where $\mathbf{x}_1, \mathbf{x}_2$ are two data points. In our paper, we will define meta-learning on an RKHS induced by such a kernel.

3 META-LEARNING IN RKHS

We first define the meta-learning problem in a general function space, and then restrict the function space to be an RKHS, where two frameworks will be proposed to make meta-learning feasible in RKHS, along with some theoretical analysis. For simplicity, in the following we will hide the superscript time t unless necessary, *e.g.*, when the analysis involves time-changing.

3.1 META-LEARNING IN FUNCTION SPACE

Given a function space $\mathcal{H}$, a distribution of tasks $P(\mathcal{T})$, and a loss function $\mathcal{L}$, the goal of meta-learning is to find a *meta function* $f^* \in \mathcal{H}$, so that it performs well after simple adaptation on a specific task. Let $\mathcal{D}_m^{tr}$ and $\mathcal{D}_m^{test}$ be the training and testing sets, respectively, sampled from a data distribution of task $\mathcal{T}_m$. The meta-learning problem on function space $\mathcal{H}$ is defined as:

$$f^* = \arg\min_{f \in \mathcal{H}} \mathcal{E}(f), \text{ with } \mathcal{E}(f) = \mathbb{E}_{\mathcal{T}_m}\left[\mathcal{L}\Big(\mathsf{Adapt}(f, \mathcal{D}_m^{tr}), \mathcal{D}_m^{test}\Big)\right] \tag{1}$$

where Adapt denotes some adaptation algorithms, *e.g.*, several steps of gradient descent; $\mathcal{E} : \mathcal{H} \to R$ is called energy functional, which is used to evaluate the model represented by the function f.

In theory, solving equation 1 is equivalent to solving the gradient flow equation $\mathrm{d}f^t/\mathrm{d}t = -\nabla_{f^t}\mathcal{E}(f^t)$. However, solving the gradient flow equation is generally infeasible, since *i*) it is hard to directly apply optimization methods in function space and *ii*) the energy functional $\mathcal{E}$ contains an adaptation algorithm Adapt, making the functional gradient infeasible. Thus, a better way is to design a special energy functional so that it can be directly optimized without running the specific adaptation algorithm. In the following, we first specify the functional meta-learning problem in RKHS, and then propose two methods to derive efficient solutions for the problem.

3.2 META-LEARNING IN RKHS

We consider a function f that is parameterized by $\boldsymbol{\theta} \in \mathbb{R}^P$, denoted as $f_{\boldsymbol{\theta}}$, with P being the number of parameters. Define a realization function $F : \mathbb{R}^P \to \mathcal{H}$ that maps parameters to a function. With these, we can then define an energy function in the parameter space as $E \triangleq \mathcal{E} \circ F : R^P \to R$ with $\circ$ being the composition operator. Consequently, with an initialized $\boldsymbol{\theta}^0$, we can define the gradient flow of $E(\boldsymbol{\theta}^t)$ in parameter space as: $\mathrm{d}\boldsymbol{\theta}^t/\mathrm{d}t = -\nabla_{\boldsymbol{\theta}^t}E(\boldsymbol{\theta}^t)$. In the following, we first establish an equivalence between the gradient flow in RKHS and the gradient flow in the parameter space. We then propose two algorithms for meta-learning in the RKHS induced by NTK.

Theorem 1 *Let $\mathcal{H}$ be an RKHS induced by the NTK $\boldsymbol{\Theta}$ of $f_{\boldsymbol{\theta}}$. With $f^0 = f_{\boldsymbol{\theta}^0}$, the gradient flow of $\mathcal{E}(f^t)$ coincides with the function evolution of $f_{\boldsymbol{\theta}^t}$ driven by the gradient flow of $E(\boldsymbol{\theta}^t)$.*

The proof of Theorem 1 relies on the property of NTK (Jacot et al., 2018), and is provided in the Appendix. Theorem 1 serves as a foundation of our proposed methods, which indicates that solving the meta-learning problem in RKHS can be done by some appropriate manipulations. In the following, we describe two different approaches termed Meta-RKHS-I and Meta-RKHS-II, respectively.

3.3 META-RKHS-I: META-LEARNING IN RKHS WITHOUT ADAPTATION

Our goal is to design an energy functional that has no adaptation component, but is capable of achieving fast adaptation. For this purpose, we first introduce two definitions: empirical loss function $\mathcal{L}(f_{\boldsymbol{\theta}}, \mathcal{D}_m)$ and expected loss function $\mathcal{L}(f_{\boldsymbol{\theta}})$. Let $\mathcal{D}_m = \{\mathbf{x}_{m,i}, \mathbf{y}_{m,i}\}_{i=1}^n$ be a set containing the data of a regression task $\mathcal{T}_m$. The empirical loss function $\mathcal{L}(f_{\boldsymbol{\theta}}, \mathcal{D}_m)$ and the expected loss function $\mathcal{L}_m(f_{\boldsymbol{\theta}})$ are defined as:

$$\mathcal{L}(f_{\boldsymbol{\theta}}, \mathcal{D}_m) = \frac{1}{2n}\sum_{i=1}^{n}\left\|f(\mathbf{x}_{m,i}) - \mathbf{y}_{m,i}\right\|^2, \quad \mathcal{L}_m(f_{\boldsymbol{\theta}}) = \mathbb{E}_{\mathbf{x}_m, \mathbf{y}_m}\left[\frac{1}{2}\left\|f(\mathbf{x}_m) - \mathbf{y}_m\right\|^2\right].$$

Our idea is to define a regularized functional such that it endows the ability of fast adaptation in RKHS. Our solution is based on some property of the standard MAML. We start from analyzing the

meta-objective of MAML with a k-step gradient-descent adaptation, *i.e.*, applying k gradient-descent steps in the inner-loop. The objective can be formulated as

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{\mathcal{T}_m}\left[\mathcal{L}(f_{\boldsymbol{\phi}}, \mathcal{D}_m^{test})\right], \text{ with } \boldsymbol{\phi} = \boldsymbol{\theta} - \alpha \sum_{i=0}^{k-1} \nabla_{\boldsymbol{\theta}_i} \mathcal{L}(f_{\boldsymbol{\theta}_i}, \mathcal{D}_m^{tr}),$$

where α is the learning rate of the inner-loop, $\boldsymbol{\theta}_0 = \boldsymbol{\theta}$, and $\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \alpha \nabla_{\boldsymbol{\theta}_i} \mathcal{L}(f_{\boldsymbol{\theta}_i}, \mathcal{D}_m^{tr})$. By Taylor expansion, we have

$$\mathbb{E}_{\mathcal{T}_m}\left[\mathcal{L}(f_{\boldsymbol{\phi}}, \mathcal{D}_m^{test})\right] \approx \mathbb{E}_{\mathcal{T}_m}\left[\mathcal{L}(f_{\boldsymbol{\theta}}, \mathcal{D}_m^{test}) - \alpha \sum_{i=0}^{k-1} \nabla_{\boldsymbol{\theta}_i} \mathcal{L}(f_{\boldsymbol{\theta}_i}, \mathcal{D}_m^{tr}) \nabla_{\boldsymbol{\theta}} \mathcal{L}(f_{\boldsymbol{\theta}}, \mathcal{D}_m^{test})^{\intercal}\right]. \quad (2)$$

Since $\mathcal{D}_m^{tr}$ and $\mathcal{D}_m^{test}$ come from the same distribution, equation 2 is an unbiased estimator of

$$\mathcal{M}_k = \mathbb{E}_{\mathcal{T}_m}[\mathcal{L}_m(f_{\boldsymbol{\theta}}) - \sum_{i=0}^{k-1} \beta_i], \text{ where } \beta_i = \alpha \nabla_{\boldsymbol{\theta}_i} \mathcal{L}_m(f_{\boldsymbol{\theta}_i}) \nabla_{\boldsymbol{\theta}} \mathcal{L}_m(f_{\boldsymbol{\theta}})^{\intercal}. \quad (3)$$

We focus on the case of $k = 1$, which is $\mathcal{M}_1 = \mathbb{E}_{\mathcal{T}_m}\left[\mathcal{L}_m(f_{\boldsymbol{\theta}})\right] - \alpha \mathbb{E}_{\mathcal{T}_m}\left[\|\nabla_{\boldsymbol{\theta}} \mathcal{L}_m(f_{\boldsymbol{\theta}})\|^2\right]$. The first term on the RHS is the traditional multi-task loss evaluated at $\boldsymbol{\theta}$ for all tasks. The second term corresponds to the negative gradient norm; minimizing it means choosing a $\boldsymbol{\theta}$ with the maximum gradient norm. Intuitively, when $\boldsymbol{\theta}$ is not a stationary point of a task, one should choose the steepest descent direction to reduce the loss maximally for a specific task, thus leading to *fast adaptation*.

The above understanding suggests us to propose the following regularized energy functional, $\widetilde{\mathcal{E}}_\alpha$, for meta-learning in RKHS with *fast function adaptation*:

$$\widetilde{\mathcal{E}}(\alpha, f_{\boldsymbol{\theta}}) = \mathbb{E}_{\mathcal{T}_m}\left[\mathcal{L}_m(f_{\boldsymbol{\theta}}) - \alpha \|\nabla_{f_{\boldsymbol{\theta}}} \mathcal{L}_m(f_{\boldsymbol{\theta}})\|_{\mathcal{H}}^2\right], \quad (4)$$

where $\|\cdot\|_{\mathcal{H}}$ denotes the functional norm in $\mathcal{H}$, and α is a hyper-parameter.

Solving the Function Optimization Problem To solve equation 4, we first derive Theorem 2 to reduce the function optimization problem to a parameter optimization problem.

Theorem 2 *Let $f_{\boldsymbol{\theta}}$ be a neural network with parameter $\boldsymbol{\theta}$ and $\mathcal{H}$ be the RKHS induced by the NTK $\boldsymbol{\Theta}$ of $f_{\boldsymbol{\theta}}$. Then, the following are equivalent*

$$\widetilde{\mathcal{E}}(\alpha, f_{\boldsymbol{\theta}}) = \mathcal{M}_1, \text{ and } \|\nabla_{f_{\boldsymbol{\theta}}} \mathcal{L}_m(f_{\boldsymbol{\theta}})\|_{\mathcal{H}}^2 = \|\nabla_{\boldsymbol{\theta}} \mathcal{L}_m(f_{\boldsymbol{\theta}})\|^2.$$

Theorem 2 is crucial to our approach as it indicates that solving problem equation 4 is no more difficult than the original parameter-based MAML, although it only considers one-step adaptation case. Next, we will show that multi-step adaptation in the parameter space can also be well-approximated by our objective equation 4 but with a scaled regularized parameter α. In the following, we consider the squared loss $\mathcal{L}$. The case with the cross-entropy loss is discussed in the Appendix. We assume that $f_{\boldsymbol{\theta}}$ is parameterized by either fully-connected or convolutional neural networks, and only consider the impact of number of hidden layers L in our theoretical results.

Theorem 3 *Let $f_{\boldsymbol{\theta}}$ be a fully-connected neural network with L hidden layers and ReLU activation function, $s_1, ..., s_{L+1}$ be the spectral norm of the weight matrices, $s = \max_h s_h$, and α be the learning rate of gradient descent. If $\alpha \leq O(qr)$ with $q = \min(1/(Ls^L), L^{-1/(L+1)})$ and $r = \min(s^{-L}, s)$, then the following holds*

$$|\mathcal{M}_k - \widetilde{\mathcal{E}}(k\alpha, f_{\boldsymbol{\theta}})| \leq O\Big(\frac{1}{L}\Big).$$

Theorem 4 *Let $f_{\boldsymbol{\theta}}$ be a convolutional neural network with $L - l$ convolutional layers and l fully-connected layers and with ReLU activation function, and d_x be the input dimension. Denote by W^h the parameter **vector** of the convolutional layer for $h \leq L - l$, and the weight **matrices** of the fully connected layers for $L - l + 1 \leq h \leq L + 1$. $\|\cdot\|_2$ means both the spectral norm of a matrix and the Euclidean norm of a vector. Define $s_h = \sqrt{d_x}\|W^h\|_2$ if $h = 1, ..., L - l$, and $\|W^h\|_2$ if $L - l + 1 \leq h \leq L + 1$. Let $s = \max_h s_h$ and α be the learning rate of gradient descent. If $\alpha \leq O(qr)$ with $q = \min(1/(Ls^L), L^{-1/(L+1)})$ and $r = \min(s^{-L}, s)$, the following holds*

$$|\mathcal{M}_k - \widetilde{\mathcal{E}}(k\alpha, f_{\boldsymbol{\theta}})| \leq O\Big(\frac{1}{L}\Big).$$

Comparisons with Reptile and MAML By Theorem 1, we known that gradient flow of an energy functional can be approximated by gradient descent in a parameter space. Reptile with 1-step adaptation (Nichol et al., 2018) is equivalent to the approximation of the gradient flow of $\widetilde{\mathcal{E}}(\alpha, f_{\boldsymbol{\theta}})$ with $\alpha = 0$, which does not include the fast-adaptation regularization as in our method.

From the equivalent parameter-optimization form indicated in Theorem 2, we know that our energy functional $\widetilde{\mathcal{E}}(\alpha, f_{\boldsymbol{\theta}})$ is closely related to MAML. However, with this form, our method does not need the explicit adaptation steps in training (*i.e.*, the inner-loop of MAML), leading to a single-looped optimization problem. We will show that our proposed method leads to better results.

3.4 META-RKHS-II: META-LEARNING IN RKHS WITH A CLOSED-FORM ADAPTATION

In this section, we present our second solution for meta-learning in RKHS by deriving a closed-form adaptation function, *i.e.*, we focus on a case where $\mathsf{Adapt}(f, \mathcal{D}_m^{tr})$ is analytically solvable using the theory of NTK. Specifically, we are given a loss function $\mathcal{L}$, tasks $\mathcal{T}_m$ with randomly split training set $\mathcal{D}_m^{tr} = \{\mathbf{x}_{m,i}^{tr}, \mathbf{y}_{m,i}^{tr}\}_{i=1}^n$, and testing set $\mathcal{D}_m^{test}$. Let $\boldsymbol{\theta}_m^t$ and $f_{m,\boldsymbol{\theta}}^t$ denote the parameters and the corresponding function at time t adapted by task $\mathcal{T}_m$ from the meta parameter $\boldsymbol{\theta}$ and meta function $f_{\boldsymbol{\theta}}$, respectively. From the NTK theory (Jacot et al., 2018; Arora et al., 2019; Lee et al., 2019), we can write the function/parameter evolution as:

$$\frac{\mathrm{d}\boldsymbol{\theta}_m^t}{\mathrm{d}t} = -\nabla_{\boldsymbol{\theta}_m^t} \mathcal{L}(f_{m,\boldsymbol{\theta}}^t, \mathcal{D}_m^{tr}), \quad \text{and} \quad \frac{\mathrm{d}f_{m,\boldsymbol{\theta}}^t}{\mathrm{d}t} = \frac{\mathrm{d}\boldsymbol{\theta}_m^t}{\mathrm{d}t} \frac{\partial f_{m,\boldsymbol{\theta}}^t}{\partial \boldsymbol{\theta}_m^t}^{\intercal} = \sum_{i=1}^n \frac{\partial \mathcal{L}(f_{m,\boldsymbol{\theta}}^t, \mathcal{D}_m^{tr})}{\partial f_{m,\boldsymbol{\theta}}^t(\mathbf{x}_{m,i}^{tr})} \boldsymbol{\Theta}(\mathbf{x}_{m,i}^{tr}, \cdot).$$

The above differential equation corresponds to the adaptation step, *i.e.*, how to adapt the meta parameter/function for task m. By the NTK theory, we can show that this admits closed-form solutions. In our meta-learning settings, this indicates that no explicit adaptation steps are necessary.

To see why this is the case, we first investigate the regression case, where the loss function $\mathcal{L}$ is the squared loss. Let $\mathbf{x} \in \mathcal{D}_m^{test}$ be a test data point. As shown in Arora et al. (2019); Lee et al. (2019), with a large enough neural network we can safely assume that NTK will not change too much during the training. In this case, we can have a closed-form solution for $f_{m,\boldsymbol{\theta}}^t$ as

$$f_{m,\boldsymbol{\theta}}^t(\mathbf{x}) = f_{\boldsymbol{\theta}}(\mathbf{x}) + H(\mathbf{x}, \mathbf{X}_m^{tr}) H^{-1}(\mathbf{X}_m^{tr}, \mathbf{X}_m^{tr}) \left(e^{-tH(\mathbf{X}_m^{tr}, \mathbf{X}_m^{tr})} - \mathbf{I} \right) \left(f_{\boldsymbol{\theta}}(\mathbf{X}_m^{tr}) - Y^{tr} \right) , \quad (5)$$

where e is the matrix exponential map, which can be approximated by $Padé$ approximation (M.Arioli et al., 1996). $H(\mathbf{X}_m^{tr}, \mathbf{X}_m^{tr})$ is an $n \times n$ kernel matrix with its (i, j) element being $\boldsymbol{\Theta}(\mathbf{x}_{m,i}, \mathbf{x}_{m,j})$, $H(\mathbf{x}, \mathbf{X}_m^{tr})$ is a $1 \times n$ vector with its i-th element being $\boldsymbol{\Theta}(\mathbf{x}, \mathbf{x}_{m,i})$, $f_{\boldsymbol{\theta}}(\mathbf{X}_m^{tr}) \in R^n$ is the predictions of all training data at the initialization, and $Y^{tr} \in R^n$ is the target value of the training data. Specifically, at time $t = \infty$, we have

$$f_{m,\boldsymbol{\theta}}^{\infty}(\mathbf{x}) = f_{\boldsymbol{\theta}}(\mathbf{x}) + H(x, \mathbf{X}_m^{tr}) H^{-1}(\mathbf{X}_m^{tr}, \mathbf{X}_m^{tr}) \left(Y^{tr} - f_{\boldsymbol{\theta}}(\mathbf{X}_m^{tr}) \right) . \quad (6)$$

The above results allow us to directly define an energy functional by substituting $\mathsf{Adapt}(f, \mathcal{D}_m^{tr})$ in equation 1 with its closed-form solution $f_{m,\boldsymbol{\theta}}^t$. In other words, our new energy functional is

$$\overline{\mathcal{E}}(t, f_{\boldsymbol{\theta}}) = \mathbb{E}_{\mathcal{T}_m} \left[\mathcal{L}_m(f_{m,\boldsymbol{\theta}}^t) \right] , \quad (7)$$

where $f_{m,\boldsymbol{\theta}}^t$ is defined in equation 5, and $\mathcal{L}_m(f_{m,\boldsymbol{\theta}}^t)$ is the expectation of $\mathcal{L}\Big(f_{m,\boldsymbol{\theta}}^t, \mathcal{D}_m^{test}\Big)$. For classification problems, we follow the same strategy as in Arora et al. (2019) to extend regression to classification. Mores details can be found in the Appendix, including the algorithm in Appendix A.

On Potential Robustness of Meta-RKHS-II It is known that the NTK evolution on the NTK regime is essentially a linear model (Jacot et al., 2018; Chizat et al., 2019). In our setting, this indicates that conditioned on the meta parameter, each adaptive task model is a linear model. Intuitively, linear models should have fewer adversarial samples and thus can be deem more robust. Combining with the expressive meta-model, our Meta-RKHS-II is thus expected to perform much better than a simple linear model while endowing better robustness than standard neural networks. A theoretical understanding of the robustness would be much involved and complicated, which we leave for interesting future research. However, we will demonstrate through extensive empirical verification, and show that Meta-RKHS-II is indeed a more robust model.

Table 1: Running time comparison per iteration with $C_1 = d_x p + Lp^2$ and $C_2 = d_x p + Ld_x p^2$.

	FOMAML	Reptile	Meta-RKHS-I	Meta-RKHS-II
Fully-connected	$O(n(k+1)C_1)$	$O(nkC_1)$	$O(nC_1)$	$O(nC_1 + n^3)$
Convolutional	$O(n(k+1)C_2)$	$O(nkC_2)$	$O(nC_2)$	$O(nC_2 + n^3)$

Connection with Meta-RKHS-I The proposed two methods choose different strategies to avoid explicit adaptation in meta-learning, which seem to be two very different algorithms. We prove below theorem, which indicates that the difference of the underlying gradient flows of the two algorithms indeed increases w.r.t. both T and the depth L of a DNN (we only consider impacts of T and L). In our experiments, we observe that Meta-RKHS-I is as fast as FOMAML, which means that it is more computationally efficient than the standard MAML. Meanwhile Meta-RKHS-II is the most robust model in tasks of adversarial attack and out-of-distribution adaptation.

Theorem 5 *Let $f_{\boldsymbol{\theta}}$ be a neural network with L hidden layers, with each layer being either fully-connected or convolutional. Assume that $\|\mathcal{L}\|_\infty < \infty$. Then, $error(T) = |\widetilde{\mathcal{E}}(T, f_{\boldsymbol{\theta}}) - \overline{\mathcal{E}}(T, f_{\boldsymbol{\theta}})|$ is a non-decreasing function of T. Furthermore, for arbitrary $T > 0$ we have $error(T) \leq O\big(T^{2L+3}\big)$.*

Connection with iMAML Our proposed method is similar to the iMAML algorithm (Finn & Levine, 2019) in the sense that both methods try to solve meta-learning without executing the optimization path in the inner-loop. Different from iMAML, which still relies on an iterative solver for the inner-loop, our method only needs to solve a single-looped optimization problem.

3.5 Time Complexity Analysis

We compare the time complexity of our proposed methods with other first-order meta-learning methods. Without loss of generality, we analyze the complexity in the case a L-layer MLP or L-layer convolutional neural networks. Recall that d_x is the input dimension. Assume each layer has width (filter number) $O(p)$. Let n be the data batch size, k the adaptation steps of inner-loop optimization. We summarize the time complexity in Table 1, where we simply assume the complexity of multiplying matrices with sizes $a \times b$ and $b \times c$ to be $O(abc)$. Note in the meta-learning setting, n is typically small, indicating the efficiency of our proposed methods.

4 Experiments

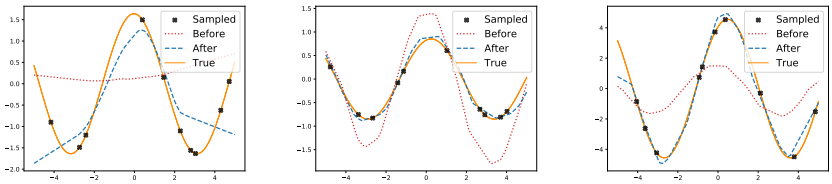
We conduct a set of experiments to evaluate the effectiveness of our proposed methods, including a sine wave regression toy experiment, few-shot classification, robustness to adversarial attacks, out-of-distribution generalization and ablation study. Due to space limit, more results are provided in the Appendix. We compare our models with related baselines including MAML (Finn et al., 2017), the first order MAML (FOMAML) (Finn et al., 2017), Reptile (Nichol et al., 2018) and iMAML (Rajeswaran et al., 2019). Results are reported as mean and variance over three independent runs.

4.1 Regression

Following Finn et al. (2017); Nichol et al. (2018), we first test our proposed methods on the 1-dimensional sine wave regression problem. This problem is instructive, where a model is trained on many different sine waves with different amplitudes and phases, and tested by adapting the trained model to new sine waves with only a few data points using a fixed number of gradient-descent steps. Following Finn et al. (2017); Nichol et al. (2018), we use a fully-connected neural network with 2 hidden layers and the ReLU activation function. The results are shown in Figure 1.

4.2 Few-shot Image Classification

For this experiment, we choose two popular datasets adopted for meta-learning: Mini-ImageNet and FC-100 (Oreshkin et al., 2018). The cross-entropy loss is adopted for Meta-RKHS-I; while the squared loss is used for Meta-RKHS-II following Arora et al. (2019); Novak et al. (2019). Similar to Finn et al. (2017), the model architecture is set to be a four-layer convolutional neural network with ReLU activation. The filter number is set to be 32. The Adam optimizer (Kingma & Ba, 2015) is



(a) Random Initialized (b) Meta-RKHS-I (c) Meta-RKHS-II

Figure 1: Performance of random initialized network and our methods. The models before/after adaptation are shown in dotted/dashed lines, samples used for adaptation are also shown in the figure.

used to minimize the energy functional. Meta batch size is set to be 16 and learning rates are set to be 0.01 for Meta-RKHS-II.

The results are shown in Table 2. Note the results of Reptile is different from those in Nichol et al. (2018), because we re-evaluate it under the same setting as Finn et al. (2017), *i.e.*, 10 steps of adaptation is applied during testing. Our results of iMAML is based on the implementation of Spigler (2019). It is observed that our proposed methods achieve better accuracy than different baselines. Interestingly, our Meta-RKHS-I performs better than FOMAML (this is also the case in other experiments), although they share a similar objective. We conjecture the reason is because our Meta-RKHS-I restricts the function to be in an RKHS, making the functional space smaller thus easier to optimize compared to the unrestricted version of FOMAML. In terms of our two algorithms, there is not always a winner on all the tasks. We note that Meta-RKHS-I is more efficient in training. However, we show below that Meta-RKHS-II is better in terms of robustness to adversarial attacks and out-of-distribution generalization.

Table 2: Few-shot classification results on Mini-ImageNet and FC-100.

	Mini-ImageNet		FC-100	
Algorithm	5 Way 1 Shot	5 Way 5 Shots	5 Way 1 Shot	5 Way 5 Shots
MAML	$48.70 \pm 1.84\%$	$63.11 \pm 0.93\%$	$38.00 \pm 1.95\%$	$49.34 \pm 0.97\%$
FOMAML	$48.07 \pm 1.75\%$	$63.15 \pm 0.91\%$	$37.73 \pm 1.93\%$	$49.05 \pm 0.99\%$
iMAML	$49.30 \pm 1.88\%$	$64.89 \pm 0.95\%$	$38.38 \pm 1.70\%$	$49.41 \pm 0.80\%$
Reptile	$49.70 \pm 1.83\%$	$65.91 \pm 0.84\%$	$38.40 \pm 1.94\%$	$50.50 \pm 0.87\%$
Meta-RKHS-I	$\mathbf{51.10 \pm 1.82}\%$	$\mathbf{66.19 \pm 0.80}\%$	$38.90 \pm 1.90\%$	$\mathbf{51.47 \pm 0.86}\%$
Meta-RKHS-II	$50.53 \pm 2.09\%$	$65.40 \pm 0.91\%$	$\mathbf{41.20 \pm 2.17}\%$	51.36 ± 0.96

4.3 Robustness to Adversarial Attack

We now compare the adversarial robustness of our methods and other baselines. We adopt both white-box and black-box attacks in this experiment. For the white-box PGD Attack (Madry et al., 2017), we use ℓ_∞ norm and compare the results on Mini-imagenet and FC-100. We compare the robust accuracy with different magnitude with 20 steps attack and step size of $2/255$. For Black-box attack, we adopt the strong query efficient attack method (Guo et al., 2019). Follow the setting of Guo et al. (2019), we use a fixed step size of 0.2.

Due to space limitation, we show some results in Figure 2 and 3, leaving some other results in the Appendix. We consider both finite-time and infinite-time adaptation in this experiment. For finite-time adaptation, the Padé approximation with $P = Q = 1$ and $P = Q = 2$ to approximate the matrix exponential are considered (Butcher & Chipman, 1992). We use Meta-RKHS-II_t100_PQ1 and Meta-RKHS-II_t100_PQ2 to denote methods using finite time $t = 100$, $P = Q = 1$ or $P = Q = 2$, respectively. We observe other finite time t makes similar predictions, thus we only consider $t = 100$. The results from the black-box attack in Figure 2 indicate the robustness of our Meta-RKHS-II. In fact, the gaps are significantly large, making it the only useful robust model in the adversarial-attack setting. Our Meta-RKHS-I is not as robust as Meta-RKHS-II, but still slightly outperforms other baselines. Regarding the white-box attack, results in Figure 3 again show that our proposed Meta-RKHS-II is significantly more robust than other methods. It is also interesting to see that our Meta-RKHS-I performs slightly better than Meta-RKHS-II in some cases, *e.g.*, in the Mini-ImageNet 5-way 1-shot case when the attack magnitude is not too small.

4.4 Out-of-distribution Generalization

We adopt similar strategy in Lee et al. (2020); Triantafillou et al. (2020) to test a model's ability of generalizing to out-of-distribution datasets. The CUB (Wah et al., 2011) and VGG Flower Nilsback & Zisserman (2008) are fine-grained datasets used in this experiment, where all images are resized to

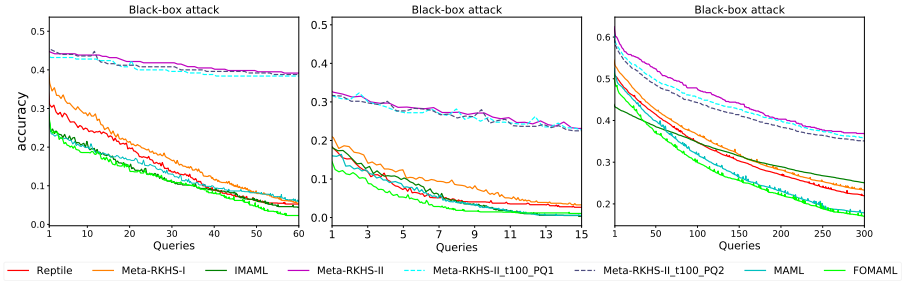


Figure 2: Black-box attack on Mini-ImageNet and FC-100. Mini-ImageNet 5-way 1-shot (left), FC-100 5-way 1-shot (middle) and Mini-ImageNet 5-way 5-shot (right).

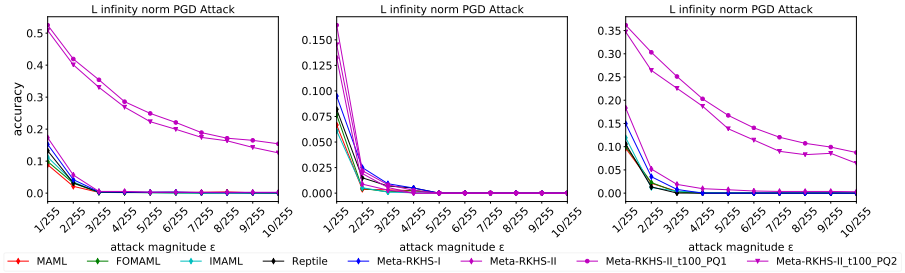


Figure 3: ℓ_∞ norm PGD attack on Mini-ImageNet and FC-100. Mini-ImageNet 5-way 5-shot (left), Mini-ImageNet 5-way 1-shot (middle) and FC-100 5-way 5-shot (right).

Table 4: Meta-RKHS-II with different time t.

	TIME t	$t = 0.1$	$t = 1$	$t = 10$	$t = 100$	$t = \infty$
MINI-IMAGENET	5 WAY 1 SHOT	$49.67 \pm 2.23\%$	$48.27 \pm 2.23\%$	$\mathbf{50.53 \pm 2.09}\%$	$49.13 \pm 2.19\%$	$48.70 \pm 2.28\%$
	5 WAY 5 SHOTS	$64.51 \pm 0.93\%$	$64.28 \pm 0.98\%$	$\mathbf{65.40 \pm 0.91}\%$	$64.24 \pm 1.06\%$	$64.95 \pm 0.96\%$
FC-100	5 WAY 1 SHOT	$36.50 \pm 2.10\%$	$38.80 \pm 2.32\%$	$\mathbf{41.20 \pm 2.17}\%$	$38.80 \pm 2.21\%$	$37.60 \pm 2.13\%$
	5 WAY 5 SHOTS	$48.35 \pm 1.02\%$	$49.79 \pm 1.04\%$	$\mathbf{51.36 \pm 0.96}\%$	$48.59 \pm 1.09\%$	$49.48 \pm 0.98\%$

84×84. We follow Lee et al. (2020) to split these datasets into meta training/validation/testing sets. We first train all the methods on Mini-ImageNet or FC-100 datasets, then conduct meta-testing on CUB and VGG Flower datasets. The results are shown in Table 3. Again, our methods achieve the best results, especially for the Meta-RKHS-II, indicating the robustness of our proposed methods. More results are presented in the Appendix.

4.5 ABLATION STUDY

Table 3: Meta testing on different out-of-distribution datasets with model trained on Mini-ImageNet.

	5 WAY 1 SHOT		5 WAY 5 SHOT	
ALGORITHM	CUB	VGG FLOWER	CUB	VGG FLOWER
MAML	$34.23 \pm 1.52\%$	$52.98 \pm 1.76\%$	$52.36 \pm 0.94\%$	$67.52 \pm 1.30\%$
FOMAML	$35.32 \pm 1.69\%$	$53.86 \pm 1.64\%$	$52.02 \pm 0.71\%$	$68.83 \pm 1.16\%$
REPTILE	$35.61 \pm 1.38\%$	$53.57 \pm 1.58\%$	$51.93 \pm 0.89\%$	$71.62 \pm 1.25\%$
IMAML	$40.55 \pm 0.61\%$	$54.97 \pm 0.80\%$	$46.31 \pm 2.03\%$	$60.67 \pm 1.91\%$
META-RKHS-I	$36.73 \pm 1.26\%$	$54.79 \pm 1.61\%$	$54.19 \pm 0.73\%$	$72.76 \pm 1.08\%$
META-RKHS-II	$\mathbf{45.36 \pm 0.87}\%$	$\mathbf{60.80 \pm 1.02}\%$	$\mathbf{65.21 \pm 0.64}\%$	$\mathbf{78.25 \pm 0.49}\%$

We conduct several ablation studies, including: comparing Reptile with Meta-RKHS-I under different adaptation steps (results shown in the Appendix), testing the impact of choosing different time t in Meta-RKHS-II (results shown in Table 4) and the impact of network architecture with different number of CNN feature channels (results shown in the Appendix). It is interesting to see that a finite-time (around $t = 10$) achieves the best accuracy, although the infinite-time case guarantees a stationary point. This indicates that a stationary point achieved by limited training data in the adaptation step is not always the best choice, because the limited training data might easily overfit the model, thus achieving worse test results.

5 CONCLUSION

We develop meta-learning in RKHS, and propose two practical algorithms allowing efficient adaptation in the function space by avoiding the explicit adaptation as in traditional methods. We show connections between our proposed methods and existing ones. Extensive experiments suggest that our proposed methods are more effective, achieve better generalization and are more robust against adversarial attacks and out-of-distribution generalization.

REFERENCES

Marcin Andrychowicz, Misha Denil, Sergio Gómez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, pp. 3981–3989. 2016.

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, 2019.

J. C. Butcher and F. H. Chipman. Generalized padé approximations to the exponential function. *BIT Numerical Mathematics*, 32:118–130, 1992.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems 32*, pp. 2937–2947. 2019.

Giulia Denevi, Carlo Ciliberto, Riccardo Grazzi, and Massimiliano Pontil. Learning-to-learn stochastic gradient descent with biased regularization. In *https://arxiv.org/abs/1903.10399*, 2019.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, 2020.

Chelsea Finn and Sergey Levine. Meta-learning: from few-shot learning to rapid reinforcement learning. In *ICML 2019 Meta-Learning Tutorial*, 2019.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017.

Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*. 2018.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. In *https://arxiv.org/abs/1410.5401*, 2014.

Chuan Guo, Jacob R. Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Q. Weinberger. Simple black-box adversarial attacks. In *International Conference on Machine Learning*, 2019.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8571–8580, 2018.

Mikhail Khodak, Maria-Florina Balcan, and Ameet Talwalkar. Provable guarantees for gradient-based meta-learning. In *International Conference on Machine Learning*, 2019a.

Mikhail Khodak, Maria-Florina Balcan, and Ameet Talwalkar. Adaptive gradient-based meta-learning methods. In *Advances in Neural Information Processing Systems*, 2019b.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Hae Beom Lee, Hayeon Lee, Donghyun Na, Saehoon Kim, Minseop Park, Eunho Yang, and Sung Ju Hwang. Learning to balance: Bayesian meta-learning for imbalanced and out-of-distribution tasks. In *International Conference on Learning Representations*, 2020.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems 32*, pp. 8572–8583. Curran Associates, Inc., 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ArXiv*, abs/1706.06083, 2017.

M.Arioli, B.Codenotti, and C.Fassino. The $Padé$ method for computing the matrix exponential. *Linear Algebra and its Applications*, June 1996.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *International Conference on Learning Representations*, 2018.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. In *https://arxiv.org/abs/1803.02999*, 2018.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Sixth Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

Roman Novak, Lechao Xiao, Yasaman Bahri, Jaehoon Lee, Greg Yang, Jiri Hron, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-dickstein. Bayesian deep convolutional networks with many channels are gaussian processes. In *International Conference on Learning Representations*, 2019.

Boris N. Oreshkin, Pau Rodriguez, and Alexandre Lacoste. Tadam: Task dependent adaptive metric for improved few-shot learning. In *Advances in Neural Information Processing Systems*, 2018.

Aravind Rajeswaran, Chelsea Finn, Sham Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems*. 2019.

Sachin Ravi and Alex Beatson. Amortized bayesian meta-learning. In *International Conference on Learning Representations*, 2019.

Filippo Santambrogio. Euclidean, Metric, and Wasserstein gradient flows: an overview, 2016.

Jurgen Schmidhuber. Evolutionary principles in self-referential learning. Diploma thesis, Technische Universitat Munchen, Germany, 14 May 1987.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 2017.

Giacomo Spigler. Meta-learnt priors slow down catastrophic forgetting in neural networks. *arXiv e-prints*, art. arXiv:1909.04170, Sep 2019.

Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Utku Evci, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. In *International Conference on Learning Representations*, 2020.

Nilesh Tripuraneni, Chi Jin, and Michael I. Jordan. Provable meta-learning of linear representations. In *https://arxiv.org/abs/2002.11684*, 2020.

C Villani. *Optimal transport – Old and new*, volume 338, pp. xxii+973. 01 2008. doi: 10.1007/978-3-540-71050-9.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *https://arxiv.org/pdf/1606.04080.pdf*, 2016.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. In *Technical Report CNS-TR-2011-001, California Institute of Technology*, 2011.

Huaxiu Yao, Ying Wei, Junzhou Huang, and Zhenhui Li. Hierarchically structured meta-learning. In *International Conference on Machine Learning*, 2019.

Jaesik Yoon, Taesup Kim, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*. 2018.