7th ICML Workshop on Automated Machine Learning (2020)

# Weighted Meta-Learning*

**Diana Cai**
DCAI@CS.PRINCETON.EDU
*Princeton University*
*Princeton, NJ 08544*

**Rishit Sheth**
RISHET@MICROSOFT.COM
*Microsoft Research New England*
*Cambridge, MA 02142*

**Lester Mackey**
LMACKEY@MICROSOFT.COM
*Microsoft Research New England*
*Cambridge, MA 02142*

**Nicolo Fusi**
FUSI@MICROSOFT.COM
*Microsoft Research New England*
*Cambridge, MA 02142*

## 1. Introduction

The applicability of machine learning techniques to real-world problems is often limited by the quantity of labeled data available. This is particularly detrimental when high-accuracy, high-capacity models are needed for a given application, since their requirements on the amount of data are particularly onerous. As a result, examples of this issue are wide-ranging and can be identified in vision (Koch, 2015), language modeling (Vinyals et al., 2016), content recommendation (Vartak et al., 2017), character generation (Lake et al., 2015), and health care (Zhang et al., 2019; Altae-Tran et al., 2017). One crucial observation to overcome this challenge is that while data on the *target* task may be limited, other *source* tasks can be used to help with learning. In meta-learning, multiple source tasks are used to provide a good "initialization" to learn on a target task. In gradient-based meta-learning (Ravi and Larochelle, 2016; Finn et al., 2017; Nichol et al., 2018), the goal is to learn an initialization from a set of source tasks that can be quickly adapted to a new target task with a small number of gradient steps, and model-agnostic meta-learning (MAML) (Finn et al., 2017) is a popular approach that leverages data from a collection of source tasks to learn an initial model that can be quickly adapted to some target data task.

An important assumption in many meta-learning methods is that the source and target tasks are drawn from the same task distribution. Since the true task distribution is usually unknown, implicit in this assumption is that future target tasks will be uniformly similar to the source tasks. In practice, this assumption is encoded in the algorithm as uniformly sampling from the source tasks during meta-training (Finn et al., 2017; Nichol et al., 2018). However, a target task may be similar to only a few of the source tasks, or even just one, and applying equal weighting to all sources during meta-learning can be detrimental. Indeed, recent research in extending the MAML framework by modeling hierarchical task

*Longer version available at `https://arxiv.org/abs/2003.09465`

distributions (Yao et al., 2019), task non-stationarity (Nagabandi et al., 2018), and multi-modality (Vuorio et al., 2018) attempts to address this shortcoming with more complex meta-learners, and other meta-learning methods have noted the importance task similarity (Achille et al., 2019; Jomaa et al., 2019). In many practical applications, the target task is available during training, and we focus minimizing the loss of the specific target task during the *entire* training procedure, rather than just the adaptation step, using the labeled target task samples during meta-training to learn a better initialization for the target task.

We study a class of meta-learning methods that can be described by a task-weighted meta-objective, which includes a variety of gradient-based meta-learning objectives, such as joint training and MAML (and first-order variants), as well as weighted variants of joint training and MAML. Without assumptions on the distribution of the source and target tasks, we provide data-dependent error bounds on the expected target risk in terms of an empirical integral probability metric (IPM) and Rademacher complexity. The resulting generalization bound leads naturally to a learning algorithm incorporating weight optimization. We show that the IPM calculation can be bounded by selecting a kernel that generates a reproducing kernel Hilbert space (RKHS) ball containing the class of functions described by composing the model class with the loss function. We provide examples on how to construct such an RKHS ball for squared loss (regression) and hinge loss (binary classification) with linear basis function models, which apply to weighted MAML and weighted ERM.

## 2. Weighted meta-learning and generalization bounds

Let $\mathcal{X}$ and $\mathcal{Y}$ represent input and output spaces respectively, and define $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$. Suppose we have $J$ independently drawn *source* tasks $\{Z^{(j)}\}_{j=1}^{J}$, where the $j$-th task $Z^{(j)} := \{z_i^{(j)}\}_{i=1}^{N^{(j)}}$ is defined by a set of data points $z_i^{(j)} \in \mathcal{Z}$. Let $\{z_i^{(j)}\}$ be instasnces of a source $j$ drawn i.i.d. from some unknown distribution $\mathbb{S}^{(j)}$. The objective is to use the source tasks to learn an initial model, $f : \mathcal{X} \to \mathcal{Y}$, that generalizes well with respect to a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ and an unknown *target* distribution $\mathbb{T}$ over $\mathcal{X} \times \mathcal{Y}$. That is, the expected target risk $\mathbb{E}_{\mathbb{T}}\ell(y, f(x))$ is small.

Importantly, we assume that a small i.i.d. sample from the target distribution, $Z^T = \{z_i^T\}_{i=1}^{N^{(T)}}$, is available and can be utilized during training, where $N^{(T)} \ll N^{(j)}$, for all $1 \leq j \leq J$. Rather than using only the source tasks during meta-training, we additionally use this labeled target sample $Z^T$ to learn the initial model.

In the following, let $\delta_z$ denote the Dirac measure at $z \in \mathcal{Z}$. Denote the $j$-th empirical source distribution and the empirical target distribution by $\hat{\mathbb{S}}^{(j)} := \frac{1}{N^{(j)}} \sum_{i=1}^{N^{(j)}} \delta_{z_i^{(j)}}$ and $\hat{\mathbb{T}} := \frac{1}{N^{(T)}} \sum_{i=1}^{N^{(T)}} \delta_{z_i^T}$ respectively. Given weights $\alpha \in \Delta^{J-1} := \{\alpha \in [0,1]^J : \sum_{j=1}^{J} \alpha_j = 1\}$, we define the empirical $\alpha$-mixture distribution among the $J$ source samples as $\hat{\mathbb{S}}_\alpha := \sum_{j=1}^{J} \alpha_j \hat{\mathbb{S}}^{(j)}$. The empirical risk of a model on a source task is given by $\mathbb{E}_{\hat{\mathbb{S}}^{(j)}}\ell(y, f(x))$, the empirical risk on the target task by $\mathbb{E}_{\hat{\mathbb{T}}}\ell(y, f(x))$, and the empirical risk on an $\alpha$-mixture of source samples by $\mathbb{E}_{\hat{\mathbb{S}}_\alpha}\ell(y, f(x))$.

Let $\mathcal{G}$ be a function class with members mapping from $\mathcal{Z}$ to $\mathbb{R}$. We consider a class of meta-learning algorithms that learns the initial model by minimizing the task-weighted meta-objective $\sum_{j=1}^{J} \alpha_j \mathbb{E}_{\hat{\mathbb{S}}^{(j)}} g(z)$, where $\alpha \in \Delta^{J-1}$. Let $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ denote a loss function and $\mathcal{F} = \{f(x;\theta) : \theta \in \Theta\}$ denote a parameterized predictor or model class

with $f(\cdot;\theta) : \mathcal{X} \to \mathcal{Y}$. Joint training (i.e., standard ERM with uniform weights on the tasks) is instantiated in this framework with uniform weights $\alpha_j = 1/J$ and the function class $\mathcal{G} = \left\{g(x,y) = \ell(y, f(x;\theta)) : \theta \in \Theta\right\}$. MAML is instantiated with $\alpha_j = 1/J$ and $\mathcal{G} = \left\{g(x,y) = \ell(y, f(x;U(\theta))) : \theta \in \Theta\right\}$, where $U$ is an adaptation function defined by $U(\theta) := \theta - \eta\nabla_\theta \mathbb{E}_{\hat{\mathbb{S}}_\alpha} \ell(y, f(x;\theta))$ and $\eta$ is a global step-size parameter.

### 2.1 Data-dependent bound for weighted meta-learning

We now provide a data-dependent upper bound on the distance between the empirical risk of an $\alpha$-mixture of source tasks and the expected risk of the target task, which directly yields (i) a generalization bound for target risk in terms of an empirical IPM between weighted source samples and the target sample and (ii) a computable algorithm for finding the weights $\alpha$ that minimize the bound. A key component in our bound is the integral probability metric (IPM) (Müller, 1997), which measures the distance between the distributions of the weighted sources and the target.

**Definition 2.1.** *The integral probability metric (IPM) between two probability distributions $\mathbb{P}$ and $\mathbb{Q}$ on $\mathcal{Z}$ with respect to the class of real-valued functions $\mathcal{G}$ is defined as*

$$\gamma_{\mathcal{G}}(\mathbb{P}, \mathbb{Q}) := \sup_{g \in \mathcal{G}} \left| \mathbb{E}_{\mathbb{P}} g(z) - \mathbb{E}_{\mathbb{Q}} g(z) \right|. \tag{1}$$

Many popular metrics between probability distributions can be cast in terms of an IPM with respect to a specific class of functions $\mathcal{G}$, such as the total variation distance and the Wasserstein distance, and the kernel distance (Sriperumbudur et al. (2012, Table 1)).

**Definition 2.2.** *The empirical Rademacher complexity of a function class $\mathcal{G}$ with respect to a sample $\{z_i\}_{i=1}^N$ drawn i.i.d. from a distribution $\mathbb{P}$ is defined as*

$$\mathcal{R}(\mathcal{G}|z_1, \ldots, z_N) := \mathbb{E} \sup_{g \in \mathcal{G}} \frac{1}{N} \left| \sum_{i=1}^N \sigma_i g(z_i) \right|,$$

*where the expectation is taken w.r.t. the i.i.d. Rademacher random variables $\{\sigma_i\}$.*

We now present the following data-dependent upper bound on $\gamma_{\mathcal{G}}(\hat{\mathbb{S}}_\alpha, \mathbb{T})$, which decomposes into a sum of the IPM between the empirical distribution of the $\alpha$-mixture of sources $\hat{\mathbb{S}}_\alpha$ and empirical target distribution $\hat{\mathbb{T}}$ and the empirical Rademacher complexity with respect to the target distribution.

**Theorem 2.3.** *Let $\mathcal{G}$ denote a class of functions whose members map from $\mathcal{Z}$ to $[a, b]$, and suppose that the source tasks $\{Z^{(j)}\}_{j=1}^J$ and target task $Z^T$ are independent, and that the data instances of each are i.i.d. within a sample. Let $\epsilon > 0$. Then with probability at least $1 - \epsilon$ over the draws of the source and target samples,*

$$\gamma_{\mathcal{G}}(\hat{\mathbb{S}}_\alpha, \mathbb{T}) \leq \gamma_{\mathcal{G}}(\hat{\mathbb{S}}_\alpha, \hat{\mathbb{T}}) + 2\mathcal{R}(\mathcal{G}|z_1, \ldots, z_{N^{(T)}}) + 3\sqrt{\frac{(b-a)^2 \log(2/\epsilon)}{2N^{(T)}}}, \tag{2}$$

*where $\mathcal{R}(\mathcal{G}|z_1, \ldots, z_{N^{(T)}})$ denotes the empirical Rademacher complexity of the function class $\mathcal{G}$ w.r.t. the target sample.*

The bound in Theorem 2.3 involves purely empirical quantities, i.e., the empirical IPM and empirical Rademacher complexity. Note that only the empirical IPM in the first term involves the $\alpha$-weights.

### 2.2 Weight selection using empirical kernel distances

While the upper bound given in Theorem 2.3 leads naturally to an algorithm for computing the weights by minimizing the bound, the IPM is in general not computable for arbitary function classes $\mathcal{G}$. Instead, we optimize a surrogate distance bound using the kernel distance, which is an IPM defined with respect to the class of functions given by the unit ball of a reproducing kernel Hilbert space (RKHS). That is, $\mathcal{G}_{\text{RKHS}} := \{g : \|g\|_{\mathcal{K}_k} \leq 1\}$, where $\mathcal{K}_k$ is a Hilbert space associated with a reproducing kernel $k : \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ and $\|\cdot\|_{\mathcal{K}_k}$ is the norm induced by the inner product on $\mathcal{K}_k$.

Let $\gamma_{\mathcal{G}_{\text{RKHS}}}(\mathbb{P}, \mathbb{Q})$ denote the kernel distance with respect to the probability distributions $\mathbb{P}$ and $\mathbb{Q}$. In order to upper bound the IPMs defined with respect to $\mathcal{G}$, we need to find an RKHS ball $\mathcal{G}_{\text{RKHS}}$ associated with a kernel $k$ such that the function class is contained in the RKHS ball, i.e., $\mathcal{G} \subseteq \mathcal{G}_{\text{RKHS}}$. Then respective IPMs can then be bounded as

$$\gamma_{\mathcal{G}}(\hat{\mathbb{S}}_\alpha, \hat{\mathbb{T}}) \leq \gamma_{\mathcal{G}_{\text{RKHS}}}(\hat{\mathbb{S}}_\alpha, \hat{\mathbb{T}}) := \gamma_k(\hat{\mathbb{S}}_\alpha, \hat{\mathbb{T}}), \tag{3}$$

where $\gamma_k(\cdot, \cdot)$ is the empirical kernel distance, or maximum mean discrepancy, The empirical kernel distance between the $\alpha$-weighted source distribution and the target distribution $\gamma_k(\hat{\mathbb{S}}_\alpha, \hat{\mathbb{T}})$ can be easily computed (Sriperumbudur et al. (2012, Theorem 2.4)) as

$$\gamma_k(\hat{\mathbb{S}}_\alpha, \hat{\mathbb{T}}) = \sqrt{v_\alpha^\top K_J \, v_\alpha}, \qquad v_\alpha := \left[\frac{\alpha_1}{N^{(1)}}, \ldots, \frac{\alpha_J}{N^{(J)}}, \frac{-1}{N^{(T)}}\right]^\top \in \mathbb{R}^{J+1}, \tag{4}$$

where $K_J \in \mathbb{R}^{(J+1)\times(J+1)}$ is a kernel gram matrix between tasks, with $[K_J]_{j,j'} = \sum_{i,i'} k(z_i^{(j)}, z_{i'}^{(j')})$.

Minimizing a bound based on the kernel distance in Equation (4) involves solving a quadratic program with simplex constraints, which has time complexity $O(J^3)$ when the gram matrix $K_J$ is positive definite.

**Upper bounds on the IPM for linear basis models.** Now that we have a surrogate bound that is computable, we now show how to construct a class of functions $\mathcal{G}_{\text{RKHS}}$ such that $\mathcal{G} \subseteq \mathcal{G}_{\text{RKHS}}$ for regression and binary classification settings. Let $\psi : \mathcal{X} \to \mathbb{R}^d$ denote a basis function, and consider the class of linear basis function models composed with a loss $\ell$,

$$\mathcal{G}^\ell := \{g((x, y)) = \ell(y, w^\top \psi(x; \theta)) : w \in \mathcal{W}, \theta \in \Theta\},$$

where $\mathcal{W} \subset \mathbb{R}^d$ denotes a constraint set and $\Theta$ denotes the parameter space for the basis function $\psi$. We consider selecting kernels for the class of functions $\mathcal{G}^\ell$ such that $\mathcal{G}^\ell \subseteq \mathcal{G}_{\text{RKHS}}$, where $\mathcal{G}_{\text{RKHS}}$ is a RKHS ball associated with the kernel. To do so, we define a feature map $\phi$ mapping $\mathbb{R}^{d+1}$ to a Euclidean feature space, and define $\mathcal{G}_{\text{RKHS}}$ to be a ball of an RKHS $\mathcal{K}_k$ constructed from the kernel $k(z, z') = \langle \phi(\psi(x), y), \phi(\psi(x'), y') \rangle$.

In the following, let $z$ denote a point $(\psi(x), y)$, and let $\text{vec}(\cdot)$ denote the vectorization operator. First we consider the class of functions $\mathcal{G}^\ell$ when $\ell$ is a square loss function.

**Lemma 2.4** (Square loss). *Let $\mathcal{W} = \{w \in \mathbb{R}^d : \|w\|_2 \leq 1\}$. For $\ell(y, y') = \frac{1}{2}(y - y')^2$, construct an RKHS from the feature map $\phi : \mathbb{R}^{d+1} \to \mathbb{R}^{d^2+d+1}$,*

$$\phi((\psi(x), y)) = (vec(\psi(x)\psi(x)^\top), \sqrt{2}y\psi(x), y^2)^\top.$$

*Then, $\mathcal{G}^\ell \subseteq \mathcal{G}_{RKHS}$ for the kernel $k$ associated with the feature map $\phi$.*

**Algorithm 1** Meta-training procedure for $\alpha$-meta-learning

1: **Input:** kernel $k$, source tasks $\{Z_j\}_{j=1}^J$, target task $Z^T$
2: Compute empirical kernel distance $\gamma_k(\hat{\mathbb{S}}_\alpha, \hat{\mathbb{T}}) = \sqrt{v_\alpha^\top K_J v_\alpha}$
3: Compute $\hat{\alpha} := \arg\min_{\alpha \in \Delta^{J-1}} \gamma_k(\hat{\mathbb{S}}_\alpha, \hat{\mathbb{T}})$
4: Learn initial model by minimizing $\sum_{j=1}^J \hat{\alpha}_j \mathbb{E}_{\hat{\mathbb{S}}^{(j)}} g(z)$
5: **Output:** weights $\hat{\alpha}$ and initial model $\hat{g}$

Now we consider $\mathcal{G}^\ell$ where $\ell$ is a hinge loss function, with constraints on the domain and parameter spaces. This allows us to, e.g., utilize a penalized SVM with sufficiently small penalty $C$ on the solution norm, i.e., $\|w\|^2 \leq C$.

**Lemma 2.5** (Hinge loss)**.** *Let $\mathcal{W} = \{w \in \mathbb{R}^d : \|w\|_2 \leq 1\}$, $\Theta = \{\theta : \|\psi(x;\theta)\|_2 \leq 1\}$, $\mathcal{Y} = [-1, 1]$. Under the constraints on the input and output spaces, $\ell(y, y') = max(1 - y\psi(x)^\top w, 0) = 1 - y\psi(x)^\top w$. Construct an RKHS from the feature map $\phi : \mathbb{R}^{d+1} \to \mathbb{R}^{d+1}$,*

$$\phi((\psi(x), y)) = (y\psi(x), 1)^\top.$$

*Then, $\mathcal{G}^\ell \subseteq \mathcal{G}_{RKHS}$ for the kernel $k$ associated with the feature map $\phi$.*

This construction encodes a natural notion for task similarity: when the kernel distance between two tasks is relatively small, this implies that the model class cannot distinguish between these tasks with respect to the associated loss function.

The examples above examine classes of linear basis functions composed with a loss $\mathcal{G}^\ell$ without explicitly considering an adaptation function $U$. Finn et al. (2019) summarizes sufficient conditions of under which the projection in $U$ is equivalent to a contraction, ensuring that model updates during training remain within $\mathcal{G}^\ell$. More generally, a projection step back into $\mathcal{G}^\ell$ can be utilized during optimization. Algorithm 1 summarizes the meta-learning procedure used learn the $\alpha$ weight values and an initial model. Note that in an adaptive basis setup, steps 2–4 are iterated, since selecing $g$ changes the basis function $\psi$.

## 3. Experiments

We present several regression examples on synthetic and real data tasks in Appendix C. In the following section, we examine a synthetic sine wave target task with random sine wave sources, and compare the performance of MAML with uniform weights with the $\alpha$-weighted MAML algorithm. Additional details on the setup and model used are in Appendix C.1.

Figure 1 shows one target task where 10 training samples (denoted by black points) are drawn from the target (denoted by the solid black curve). The red and blue curves denote the resulting predictions from the learned intializations of uniformly-weighted MAML and $\alpha$-weighted MAML. In the bottom plot, the intializations are adapted to the 10 target samples. In this plot, we observe that only after a larger number of gradient steps ($\sim$100) is the uniform weighting able to achieve a comparable mean squared error on the held-out target samples as the $\alpha$-weighted initialization.

In Table 1, we report the average RMSE for each method before and after fast adaptation for MAML and ERM with 1) uniform weights, 2) $\alpha$-weights (Algorithm 1), and 3) threshold weights (i.e., closest source selection). In the table, we see that the predictions from the

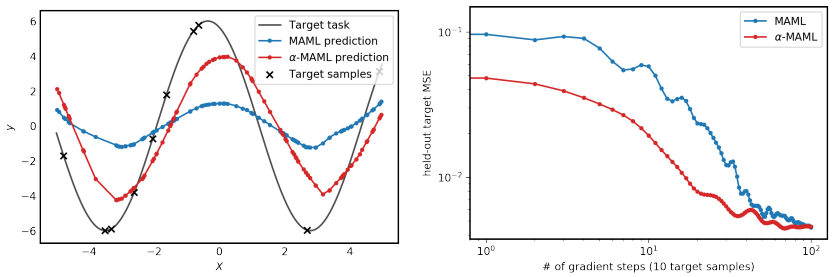


Figure 1: Sine wave regression with 10 labeled target examples. **Left:** Predictions after training MAML and weighted MAML for 10,000 meta-iterations. **Right:** Held-out target mean squared error after $L$ gradient steps of fast adaptation.

Table 1: RMSE of sine wave predictions using (1) the initial meta-model and (2) after 10 gradient steps (denoted by †) for 5-shot, 10-shot, and 20-shot target training scenarios, averaged over 4 random trials.

| | 5-shot | 10-shot | 20-shot |
|---|---|---|---|
| MAML | $3.90 \pm 0.85$ | $3.57 \pm 0.66$ | $4.11 \pm 0.94$ |
| $\alpha$-MAML | $3.21 \pm 1.12$ | $2.93 \pm 0.75$ | $3.05 \pm 1.09$ |
| Threshold | $2.83 \pm 1.04$ | $3.17 \pm 1.05$ | $3.26 \pm 1.08$ |
| MAML† | $4.24 \pm 1.00$ | $1.90 \pm 0.26$ | $2.06 \pm 0.39$ |
| $\alpha$-MAML† | $2.65 \pm 1.34$ | $1.68 \pm 0.77$ | $1.67 \pm 0.75$ |
| Threshold† | $2.35 \pm 1.75$ | $2.01 \pm 0.83$ | $2.01 \pm 0.88$ |

initializations are fairly close for all methods, with the threshold method and $\alpha$-MAML achieving lower RMSE on the predicted values than uniformly-weighted MAML on average. On average, the $\alpha$-MAML is able to adapt better in 10 gradient steps than the uniformly and single-source threshold MAML intializations, and the threshold method still is competitive for fast adaptation, especially relative to uniformly weighted MAML.

## 4. Discussion and future work

We presented a class of weighted meta-learning methods, where the weights are selected by minimizing a data-dependent bound involving an empirical IPM between the weighted sources and target risks. Using this bound, we developed a computable algorithm based on minimizing an empirical kernel distance, providing examples for basis regression models with square loss and hinge loss. A number of promising future directions remain. One direction is to generalize our approach to arbitrary loss functions, beyond the square and hinge loss, and to extend the method to multi-class classification problems; here it would be necessary to develop additional computational improvements. Additionally, one could consider only use the labeled target task examples during training, but also unlabeled target information to help quickly adapt the tasks. Finally, exploring the use of this method in other applications, such as a continual learning paradigm, remains a fruitful direction.

Acknowledgments

This work was partially completed while Diana Cai was at Microsoft Research New England. Diana Cai is supported in part by a Google Ph.D. Fellowship in Machine Learning.

## Appendix A. Related work

A number of recent works have established guarantees for gradient-based meta-learning algorithms (Finn et al., 2019; Khodak et al., 2019a,b) developed from the perspective of online convex optimization. Further work has also established guarantees for non-convex loss functions (Fallah et al., 2019). In these frameworks, task similarity is either not considered, or is fundamentally defined as distance between model parameters in some metric space (Khodak et al., 2019a,b). Li et al. (2017); Xu et al. (2019) incorporate the use of task-weighted loss functions within MAML meta-training but provide no guarantees.

A separate line of work in domain adaptation studies the problem of combining multiple source tasks with target task data. Early bounds for classification were established by Ben-David et al. (2010) in terms of an $\mathcal{H}$-divergence. Zhang et al. (2013, 2012) extend these results by considering general loss functions and deriving bounds in terms of a population IPM (and subsequently study convergence in this setting). Separately, Mansour et al. (2009b) considered the mixture adaptation problem of combining the predictions of given source models and showed that a distribution-weighted combining rule will achieve performance close to the lowest performing source model assuming the target is a mixture of sources. The ensemble generative adversarial network of Adlam et al. (2019) utilizes a discrepancy distance (Mansour et al., 2009a; Cortes and Mohri, 2014) to compute task weights, but utilizes fixed models in the ensemble to generate data for a target task, whereas we learn task weights to optimize a model for a target task directly. Similar to our setting, Pentina et al. (2019) also develop a data-dependent bound for meta-learning with weighted tasks; their bound, however, contains interaction terms between task weights and unobservable quantities (the minimum possible combined source/target error of a single hypothesis) which, unlike this work, precludes optimization with respect to task weights.

In a similar spirit to our work, Shui et al. (2019) consider the $\mathcal{H}$-divergence and Wasserstein distance as task similarity measures to develop generalization bounds in the setting of multi-task learning with finite VC- and pseudo-dimension model classes. In our construction, we embed the model class composed with loss function within a RKHS, allowing the task similarity measure to be efficiently computed by kernel distance. Finally, there are other lines of work that capture notions of task similarity for meta-learning through proxy measures such as distance between embedded tasks (Achille et al., 2019; Jomaa et al., 2019).

## Appendix B. Proofs

### B.1 Proof of Theorem 2.3

The following is a standard uniform deviation bound based on Rademacher complexity (Bartlett and Mendelson (2002)):

**Lemma B.1** (Uniform deviation with empirical Rademacher complexity)**.** *Let the sample* $\{z_1, \ldots, z_N\}$ *be drawn i.i.d. from a distribution* $\mathbb{P}$ *over* $\mathcal{Z}$ *and let* $\mathcal{G}$ *denote a class of functions*

*on $\mathcal{Z}$ with members mapping from $\mathcal{Z}$ to $[a, b]$. Then for $\epsilon > 0$, we have that with probability at least $1 - \epsilon$ over the draw of the sample,*

$$\sup_{g \in \mathcal{G}} \left| \mathbb{E}_{\hat{\mathbb{P}}} g(z) - \mathbb{E}_{\mathbb{P}} g(z) \right| \leq 2\,\mathcal{R}(\mathcal{G}|z_1, \ldots, z_N) + 3\sqrt{\frac{(b-a)^2 \log(2/\epsilon)}{2N}}, \tag{5}$$

*where $\hat{\mathbb{P}}$ represents the empirical distribution of the sample, and $\mathcal{R}(\mathcal{G}|z_1, \ldots, z_N)$ denotes the empirical Rademacher complexity of the function class $\mathcal{G}$ w.r.t. the sample.*

*Proof of Theorem 2.3.* With probability 1 over the draw of target sample, we have

$$\begin{aligned}
\gamma_{\mathcal{G}}(\hat{\mathbb{S}}_\alpha, \mathbb{T}) &= \sup_{g \in \mathcal{G}} \left| \mathbb{E}_{\hat{\mathbb{S}}_\alpha} g(z) + \mathbb{E}_{\hat{\mathbb{T}}} g(z) - \mathbb{E}_{\hat{\mathbb{T}}} g(z) - \mathbb{E}_{\mathbb{T}} g(z) \right| \leq \sup_{g \in \mathcal{G}} \left[ \left| \mathbb{E}_{\hat{\mathbb{S}}_\alpha} g(z) - \mathbb{E}_{\hat{\mathbb{T}}} g(z) \right| + \left| \mathbb{E}_{\hat{\mathbb{T}}} g(z) - \mathbb{E}_{\mathbb{T}} g(z) \right| \right] \\
&\leq \sup_{g \in \mathcal{G}} \left| \mathbb{E}_{\hat{\mathbb{S}}_\alpha} g(z) - \mathbb{E}_{\hat{\mathbb{T}}} g(z) \right| + \sup_{g \in \mathcal{G}} \left| \mathbb{E}_{\hat{\mathbb{T}}} g(z) - \mathbb{E}_{\mathbb{T}} g(z) \right| = \gamma_{\mathcal{G}}(\hat{\mathbb{S}}_\alpha, \hat{\mathbb{T}}) + \sup_{g \in \mathcal{G}} \left| \mathbb{E}_{\hat{\mathbb{T}}} g(z) - \mathbb{E}_{\mathbb{T}} g(z) \right|,
\end{aligned}$$

where in the first inequality, we applied the triangle inequality, the second inquality, we split the supremum terms, and and in the last line, we applied Definition 2.1. The term $\sup_{g \in \mathcal{G}} \left| \mathbb{E}_{\hat{\mathbb{T}}} g(z) - \mathbb{E}_{\mathbb{T}} g(z) \right|$ can be bounded in a variety of ways. Here, we use a standard bound via the empirical Rademacher complexity (Lemma B.1) to yield the result. □

The bound in Theorem 2.3 involves purely empirical quantities, i.e., the empirical IPM and empirical Rademacher complexity. Note that only the empirical IPM in the first term involves the $\alpha$-weights.

**Corollary B.2.** *Assume the conditions of Theorem 2.3 hold. Then with probability at least $1 - \epsilon$,*

$$\gamma_{\mathcal{G}}(\hat{\mathbb{S}}_\alpha, \mathbb{T}) \leq \sum_{j=1}^{J} \alpha_j \gamma_{\mathcal{G}}(\hat{\mathbb{S}}^{(j)}, \hat{\mathbb{T}}) + 2\mathcal{R}(\mathcal{G}|z_1, \ldots, z_{N^{(T)}}) + 3\sqrt{\frac{(b-a)^2 \log(2/\epsilon)}{2N^{(T)}}}. \tag{6}$$

*Proof.* Since $\alpha \in \Delta^{J-1}$, it follows that

$$\begin{aligned}
\gamma_{\mathcal{G}}(\hat{\mathbb{S}}_\alpha, \hat{\mathbb{T}}) &= \sup_{g \in \mathcal{G}} \left| \sum_{j=1}^{J} \alpha_j \mathbb{E}_{\hat{\mathbb{S}}^{(j)}} g(z) - \mathbb{E}_{\hat{\mathbb{T}}} g(z) \right| \leq \sup_{g \in \mathcal{G}} \sum_{j=1}^{J} \alpha_j \left| \mathbb{E}_{\hat{\mathbb{S}}^{(j)}} g(z) - \mathbb{E}_{\hat{\mathbb{T}}} g(z) \right| \\
&\leq \sum_{j=1}^{J} \alpha_j \sup_{g \in \mathcal{G}} \left| \mathbb{E}_{\hat{\mathbb{S}}^{(j)}} g(z) - \mathbb{E}_{\hat{\mathbb{T}}} g(z) \right| = \sum_{j=1}^{J} \alpha_j \gamma_{\mathcal{G}}(\hat{\mathbb{S}}^{(j)}, \hat{\mathbb{T}}).
\end{aligned}$$

□

While the weighted empirical IPM in Corollary B.2 results in a looser bound, it leads to an even simpler and computationally cheaper weight selection rule, which may be sufficient for some problems; we discuss this further in Section 2.2. Corollary B.2 can be interpreted as an empirical version of the bound in Zhang et al. (2013, Theorem 5.2) for the function class $\mathcal{G}$ defined in the joint training objective.

### B.2 Proof of Lemma 2.4

Recall that the Hilbert space $\mathcal{K}_k$ associated with a reproducing kernel $k$ has the properties that (1) for all $z \in \mathcal{Z}$, $k(\cdot, z) \in \mathcal{K}_k$ and (2) for all $z \in \mathcal{Z}$ and for all functions $g \in \mathcal{K}_k$, $g(z) = \langle g, k(\cdot, z)\rangle_{\mathcal{K}_k}$.

*Proof.* Let $g \in \mathcal{G}^\ell$. Fix $w \in \mathcal{W}$ and let $a_1 = \frac{1}{2}, z_1 = (-w, 1)$. Then

$$g(z) = \ell(y, w^\top \psi(x)) = \frac{1}{2}(w^\top \psi(x) - y)^2 = \frac{1}{2}(\mathrm{vec}(\psi(x)\psi(x)^\top)^\top \mathrm{vec}(ww^\top) - 2y\psi(x)^\top w + y^2)$$
$$= a_1 \phi(z)^\top \phi(z_1) = a_1 k(z, z_1) \in \mathcal{K}_k. \tag{7}$$

Applying Equation (7) and Property (2) of the RKHS, $g$ has bounded norm:

$$\|g\|^2_{\mathcal{K}_k} = \langle g, g\rangle_{\mathcal{K}_k} = a_1 \langle g, k(\cdot, z_1)\rangle_{\mathcal{K}_k} = a_1^2 k(z_1, z_1) = a_1^2(\|w\|_2^2 + 2\|w\|_2 + 1) \le 1,$$

where the inequality follows from the assumption that $\|w\|_2 \le 1$. Thus, $\mathcal{G}^\ell \subseteq \mathcal{G}_{\mathrm{RKHS}}$. □

### B.3 Proof of Lemma 2.5

*Proof.* Fix $w \in \mathcal{W}$ and let $z_1 = (-w, 1)$. Let $g \in \mathcal{G}^\ell$. Then, $g$ is an element of the RKHS $\mathcal{K}_k$:

$$g(z) = \ell(y, w^\top \psi(x)) = 1 - y\psi(x)^\top w = k(z, z_1) \in \mathcal{K}_k,$$

which, along with Property (2) of the RKHS, implies that $g$ has bounded norm:

$$\|g\|^2_{\mathcal{K}_k} = \langle g, g\rangle_{\mathcal{K}_k} = \langle g, k(\cdot, z_1)\rangle_{\mathcal{K}_k} = k(z_1, z_1) = \|w\|_2^2 + 1 \le 2,$$

where we applied the assumption that $\|w\|_2 \le 1$. Thus, $\mathcal{G}^\ell \subseteq \mathcal{G}_{\mathrm{RKHS}}$. □

Note that in Lemma 2.5, $\mathcal{G}_{\mathrm{RKHS}}$ is a $\sqrt{2}$-RKHS ball; the extra constant factor only scales the kernel distance computation and does not affect the computation of the weights.

## Appendix C. Additional experiments and details

### C.1 Sine regression with $\alpha$-MAML

The target task was assigned a fixed amplitude of 6 with a small number of samples (5, 10, 20) for training and fast adaptation, and 100 data points were randomly sampled from the target task for evaluation. For the source tasks, the amplitudes were drawn according to a gamma$(1, 2)$ distribution. For both target and source tasks, the phase parameter was drawn uniformly from $(0, \pi)$, as in the setup in Finn et al. (2017). From each source task, 40 samples were drawn, where the $x$ values were sampled uniformly from $(-5, 5)$.

Following Finn et al. (2017), we used a fully-connected neural network with 2 hidden layers of size 40 with ReLU non-linearities. For all experiments, Adam was used as the meta-optimizer with an inner-loop learning rate of 0.01 and an outer-loop learning rate of 0.001. In each iteration of weighted MAML, we sample a mini-batch of $T$ tasks, compute embeddings for $T$ tasks and the target task, compute weights by optimizing the bound (and computing kernel distances using the computed embeddings of the sources and target),

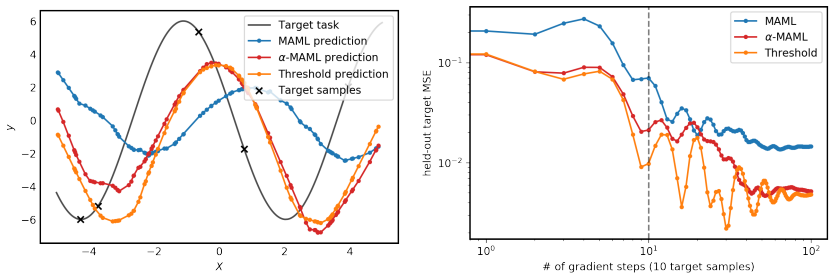


Figure 2: Results of sine wave regression for a 5-shot target task. **Left:** Learned meta-initializations for uniform, $\alpha$, and single-source weights after 10,000 meta-iterations. **Right:** Target task MSE on 100 held-out target samples after fast adaptation using 5 samples from the target.

compute weighted loss using the optimal weights, and lastly, update the model parameters. We used the same as above for MAML but with uniformly weighted source tasks. For both MAML and weighted MAML, the mini-batch size was set to $T = 100$ tasks.

Here we present additional results for 5-shot target training sizes using the adaptive version of Algorithm 1 (based on optimizing an upper bound on Theorem 2.3. We also evaluate a variant that we refer to as the "threshold" meta-learning method, that is based on optimizing an upper bound on Corollary B.2, which corresponds to weighting only the closest source task.

In each random trial, a random sine task was drawn according to the task distribution described in the main paper, and random samples from the target were also drawn (with fixed amplitude and random phase). Each method was trained for 10,000 meta-iterations. The intializations are evaluated on 1000 held-out samples from each sine task.

In Figure 2, we present a single trial from a 5-shot target task with the learned initializations of each method and the held-out MSE after fast adaptation. In this example, for all methods, fast adaptation helps, implying that the learned meta-initialization is useful for learning this task. However, even after a large number of target tasks, the uniformly-weighted MAML initialization is unable to achieve the same MSE as the non-uniformly weighted initializations (i.e., $\alpha$- and threshold-MAML).

### C.2 The analytical $\alpha$-weighted meta-learning solution

The solution to the weighted MAML (and weighted ERM) meta-objective is available in closed form, as we show in this section. We assume a linear model and squared loss for every task. We follow Finn et al. (2019, Appendix A), who provide a derivation of the analytical solution for the uniformly-weighted case of ERM (i.e., joint training) and MAML for linear regression with squared loss.

Denote the MAML adaptation function of the predictors $w \in \mathbb{R}^d$ as $U_j(w) := w - \eta(A_j w - b_j)$, where $A_j := X_j^\top X_j$, $b_j := X_j^\top w$, $X_j \in \mathbb{R}^{N^{(j)} \times d}$ is the covariate matrix of the $j$-th source task, and $\eta > 0$ is the step size.

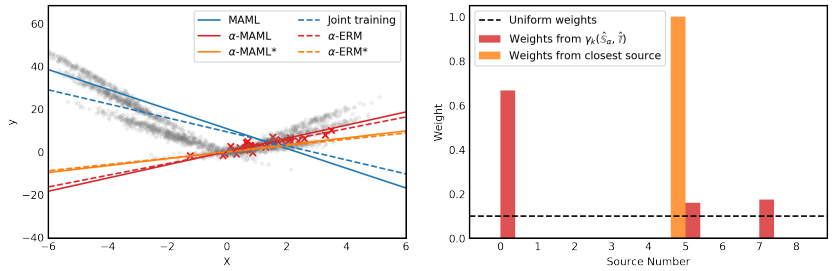


Figure 3: **Left:** Linear regression with MAML, joint training, $\alpha$-MAML, and $\alpha$-ERM solutions. Red x's denote the target task, and gray points denote the 9 source tasks. **Right:** The various weightings obtained from uniform weighting, $\alpha$-weighting according to the kernel IPM between the $\alpha$-mixture of sources and target, or weighting only the closest source.

The weighted MAML objective can be written as a function of the predictors $w$ as follows

$$F(w) = \frac{1}{2} w^\top \left( \sum_{j=1}^{J} \alpha_j (I - \eta A_j)^\top A_j (I - \eta A_j) \right) w + w^\top \left( \sum_{j=1}^{J} \alpha_j (I - \eta A_j)^\top b_j \right).$$

Defining $\tilde{A}_j := (I - \eta A_j)$, $\tilde{A} := \sum_{j=1}^{J} \alpha_j \tilde{A}_j^\top A_j \tilde{A}_j$, and $\tilde{b} := \sum_{j=1}^{J} \alpha_j \tilde{A}_j^\top b_j$, we have that the gradient of the meta-objective is $\nabla F(w) = \tilde{A} w - \tilde{b}$, and so the solution is $w_{\alpha\text{-MAML}} = \tilde{A}^{-1} \tilde{b}$. When the MAML learning rate $\eta = 0$, we recover the solution for the $\alpha$-weighted ERM, where $U(w) = w$.

### C.3 Synthetic linear regression

First we examine a 1-dimensional linear regression setting. We generated 9 source tasks and 1 target task as follows. The task sizes were generated according to a multinomial distribution with a uniform prior on the multinomial parameter. For each source, the covariates were generated from a gaussian with mean $\mu_j$ and variance 1, where $\mu_j \sim \text{uniform}(-5, 5)$. The slope of the $j$-th task was set to $2\mu_j$, and the response of the $j$-th source was then drawn according to $y_j \sim 2\mu_j + \epsilon$, where $\epsilon \sim \mathrm{N}(0, 1)$. We note that for weighted MAML and weighted ERM, an analytical solution to the meta-objective can be computed, see Appendix C.2 for a derivation. Thus, the analytical solution is used to compute an initialization, and we compare the resulting initializations from $\alpha$-weighted meta-learning and uniform weighting. All MAML solutions were computed with $\eta = 0.0001$ step size.

In Figure 3, we plot the initializations obtained from each method. Here $\alpha$-MAML and $\alpha$-ERM denote the initializations from minimizing the bound with the kernel distance, whereas $\alpha$-MAML$^*$ and $\alpha$-ERM$^*$ denote the initializations obtained from placing all weight on the closest source. The kernel distance was computed using 20 target training examples (denoted by red points). Lines denote inferred hypotheses using uniform, $\alpha$, and closest source weightings. We observe that the uniformly-weighted intializations (blue) correspond to an average model learned from all the sources, whereas unequally weighted sources (red, orange) are able to use the task similarity to better represent the target task. The weights

Table 2: RMSE of initializations for linear regression on sources and target using 20 labeled target training examples to compute the weights (before fast adaptation).

| | Diabetes | Boston |
|---|---|---|
| MAML | 50.44 | 3.64 |
| $\alpha$-MAML | 49.36 | 3.59 |
| Joint training | 50.31 | 3.58 |
| $\alpha$-ERM | 49.24 | 3.32 |
| target | 92.31 | 15.47 |

are shown in the bottom plot in Figure 3, where the weights obtained from minimizing the kernel distance $\gamma_k(\hat{\mathbb{S}}_\alpha, \hat{\mathbb{T}})$ place weight on 3 sources.

### C.4 Weighted meta-learning for real data tasks

**Multi-dimensional linear regression.** We examined two multi-dimensional regression data sets. The first uses the diabetes data set studied by Efron et al. (2004), which contains 10 covariates. We split the diabetes data set into multiple sources by grouping on the following age groups: $[19, 29)$, $[29, 39)$, $[39, 49)$, $[53, 59)$, $[59, 64)$, $[64, 79)$. The target age group included data from the age group $[49, 52]$. The goal is to predict a real-valued response that measures disease progression one year after baseline. We split the data set into separate source tasks by grouping on age, leading to a total of 6 source tasks, using the remaining covariates in each source. We picked a separate age group for the target task, using 20 target samples for computing the kernel distance and the remaining target samples were used for testing.

The second data set is the Boston house prices data of Harrison Jr and Rubinfeld (1978), which includes 13 covariates, and the response variable is the median value of owner-occupied homes. To form sources, we grouped on the attribute age, and separated the full data into 6 source tasks, where each source contained a group of 50 ages, and the remaining 12 covariates were used in each source. The target task contained 30 target samples for training, and the remaining samples were used for testing. The Boston housing prices data set was split into the sources by grouping the sources tasks on the following age groups: $[2.9, 29.1)$, $[29.1, 42.3)$, $[42.3, 58.1)$, $[72.5, 84.4)$, $[84.4, 92.4)$, and $[92.4, 100.0)$. The target age group included data from the age group $[58.1, 72.5)$.

The root mean squared error (RMSE) of each of the initializations obtained are presented in Table 2, where all MAML-related computations used $\eta = 0.0001$ for the step size. This is a setting where predicting on only the target training samples performs quite poorly and using the source data sets improves performance for this particular target task. Furthermore, weighting sources by kernel distance seems to also improve prediction error. We found that typically, similar age ranges were upweighted more than further away age groups.

**Basis linear regression.** Next we examined the sales data set studied by Tan and San Lau (2014). The data set consists of a collection of products with sales information over 52 weeks. We included the 300 products as source tasks, and used a single product as the target task.

Table 3: RMSE of sales data for 5-shot and 10-shot target training sample sizes, where the remaining data was used for evaluation. Mean and standard deviation computed over 20 target tasks.

| | 5-shot | 10-shot |
|---|---|---|
| MAML | $12.86 \pm 3.79$ | $12.69 \pm 3.69$ |
| $\alpha$-MAML | $2.43 \pm 2.09$ | $2.41 \pm 1.92$ |
| thresh-MAML | $2.53 \pm 1.94$ | $2.50 \pm 2.03$ |
| ERM | $12.09 \pm 3.53$ | $11.92 \pm 3.43$ |
| $\alpha$-ERM | $2.52 \pm 2.21$ | $2.50 \pm 2.04$ |
| thresh-ERM | $2.45 \pm 1.81$ | $2.45 \pm 1.97$ |
| target | $83.63 \pm 104.07$ | $209.06 \pm 378.88$ |

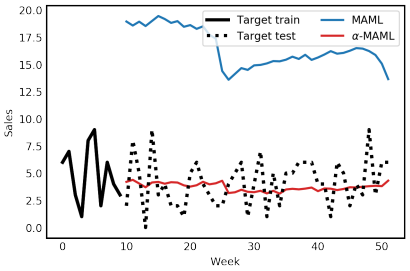


Figure 4: Product sales over 52 weeks. Example target task from the sales data set; the solid black line denotes the data examples used for training, and the dashed black line denotes the data used for testing. The blue and red lines denote the learned intializations (i.e., before fast adaptation).

For the target task, we used the first 10 weeks as labeled target training data, and the last 42 weeks as test data for evaluation. We computed random Fourier features (Rahimi and Recht, 2008) for the weeks and used these features when computing the $\alpha$-weights in the kernel distance.

In Table 3, we report the RMSE on held-out target data for target tasks with 5 and 10 weeks of data, i.e., 5- and 10-shot target tasks, averaged over 20 different product target tasks. In this example, the table shows that predicting on the target data alone performs very poorly but that meta-learning helps improve performance.

Here $\alpha$-MAML and $\alpha$-ERM are the methods used in Algorithm 1 for MAML and ERM, respectively, whereas thresh-MAML and thresh-ERM correspond to the threshold method that weights the closest source only. In this setting, both weighted methods, i.e., $\alpha$-based and thresh-based meta-learning, outperform the uniformly-weighted methods.

In Figure 4 we show the learned initializations from $\alpha$-MAML vs uniformly-weighted MAML, where the learning rate parameter was set as $\eta = 0.0001$. Here the MAML initialization learns a task that is an average of many of the tasks; in constrast, the $\alpha$-MAML initialization upweights products with more similar sources and patterns as the target. As a result, the $\alpha$-MAML initization is able to better predict future data coming from that task.

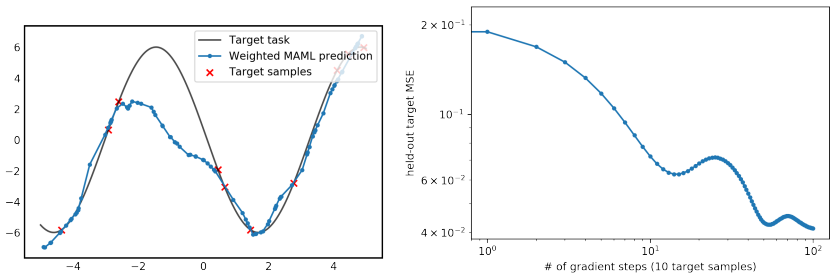


Figure 5: Results of sine wave regression obtained from directly optimizing the generalization bound in Equation (8) during meta-training. **Left:** Learned weighted MAML initialization after 20,000 meta-iterations. **Right:** Held-out MSE ater fast adaptation using 10 samples of the target.

### C.5 Generalization bound optimization

In Section 2.2 of the main paper, we describe a high-level algorithm for optimizing for the $\alpha$ weight values, given by $\hat{\alpha} := \arg\min_{\alpha \in \Delta^{J-1}} \gamma_k(\hat{\mathbb{S}}_\alpha, \hat{\mathbb{T}})$, i.e., minimizing a kernel distance between the empirical distributions of the $\alpha$-mixture of sources and the target. An alterative algorithm could also be derived from directly optimizing a generalization bound. Indeed, Theorem 2.3 implies that for all $g \in \mathcal{G}$, $\mathbb{E}_{\mathbb{T}}(g) \leq \mathbb{E}_{\hat{\mathbb{S}}_\alpha}(g) + \gamma_{\mathcal{G}}(\hat{\mathbb{S}}_\alpha, \hat{\mathbb{T}}) + 2\mathcal{R}(\mathcal{G}|z_1, \ldots, z_{N^{(T)}}) + 3\sqrt{\frac{(b-a)^2 \log(2/\epsilon)}{2N^{(T)}}}$, and similarly, Corollary B.2 implies that for all $g \in \mathcal{G}$, $\mathbb{E}_{\mathbb{T}}(g) \leq \mathbb{E}_{\hat{\mathbb{S}}_\alpha}(g) + \sum_{j=1}^{J} \alpha_j \gamma_{\mathcal{G}}(\hat{\mathbb{S}}^{(j)}, \hat{\mathbb{T}}) + 2\mathcal{R}(\mathcal{G}|z_1, \ldots, z_{N^{(T)}}) + 3\sqrt{\frac{(b-a)^2 \log(2/\epsilon)}{2N^{(T)}}}$.

Thus, an alternative algorithm to the one proposed in Algorithm 1 would involve directly optimizing the generalization bound above:

$$\hat{\alpha}, \hat{g} := \underset{\alpha \in \Delta^{J-1}, g \in \mathcal{G}}{\arg\min} \; \mathbb{E}_{\hat{\mathbb{S}}_\alpha}(g) + \gamma_k(\hat{\mathbb{S}}_\alpha, \hat{\mathbb{T}}). \tag{8}$$

We can also propose a variant of the looser bound given by optimizing

$$\hat{\alpha}, \hat{g} := \underset{\alpha \in \Delta^{J-1}, g \in \mathcal{G}}{\arg\min} \; \mathbb{E}_{\hat{\mathbb{S}}_\alpha}(g) + \sum_{j=1}^{J} \alpha_j \gamma_k(\hat{\mathbb{S}}^{(j)}, \hat{\mathbb{T}}). \tag{9}$$

An advantage of optimizing the generalization bound directly rather than the two-step procedure in Algorithm 1 is that we can jointly optimize for the $\alpha$ weight values and model parameters via gradient descent.

We examined the performance of the direct bound optimization above, i.e., optimizing Equation (8) in the sine wave regression setting. We sampled 200,000 source tasks in advance, according to the same task distribution described in Section 3, and used the same 2-layer neural network model as before. Adam was used for the meta-optimizer, with the same learning rates of the main document. In order to speed up the computation, we used mini-batches of size 150. Meta-training was performed for 20,000 meta-iterations. In Figure 5, we show the results of the direct bound optimization for the sine wave regression example. The top plot shows that the initialization learned is close to the target task, though it does not

seem to be able to capture areas where there are no samples as well. By contrast, while the predictions obtained according to Algorithm 1 (see main document, Figure 1) are able to better capture the overall shape of the sine wave task in only 10,000 meta-iterations. The bottom plot shows that the initialization is able to benefit from fast adaptation, as it can be adapted to the target with a small number of gradient steps; however, the initialization obtained from Algorithm 1 is able to adapt more quickly for this task.

## References

A. Achille, M. Lam, R. Tewari, A. Ravichandran, S. Maji, C. C. Fowlkes, S. Soatto, and P. Perona. Task2vec: Task embedding for meta-learning. In *ICCV*, pages 6430–6439, 2019.

B. Adlam, C. Cortes, M. Mohri, and N. Zhang. Learning GANs and ensembles using discrepancy. In *NeurIPS*, pages 5788–5799, 2019.

H. Altae-Tran, B. Ramsundar, A. S. Pappu, and V. Pande. Low data drug discovery with one-shot learning. *ACS central science*, 3(4):283–293, 2017.

P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *JMLR*, pages 463–482, 2002.

S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010.

C. Cortes and M. Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, 2014.

B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004.

A. Fallah, A. Mokhtari, and A. Ozdaglar. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. *arXiv e-print 1908.10400*, 2019.

C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, pages 1126–1135, 2017.

C. Finn, A. Rajeswaran, S. Kakade, and S. Levine. Online meta-learning. In *ICML*, pages 1920–1930, 2019.

D. Harrison Jr and D. L. Rubinfeld. Hedonic housing prices and the demand for clean air. 1978.

H. S. Jomaa, J. Grabocka, and L. Schmidt-Thieme. Dataset2vec: Learning dataset meta-features. *arXiv e-print 1905.11063*, 2019.

M. Khodak, M.-F. Balcan, and A. Talwalkar. Provable guarantees for gradient-based meta-learning. In *ICML*, 2019a.

M. Khodak, M.-F. F. Balcan, and A. S. Talwalkar. Adaptive gradient-based meta-learning methods. In *NeurIPS*, pages 5915–5926, 2019b.

G. Koch. Siamese neural networks for one-shot image recognition. 2015.

B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Z. Li, F. Zhou, F. Chen, and H. Li. Meta-SGD: Learning to learn quickly for few-shot learning. *arXiv e-print 1707.09835*, 2017.

Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv e-print 0902.3430*, 2009a.

Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation with multiple sources. In *NeurIPS*, pages 1041–1048, 2009b.

A. Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997.

A. Nagabandi, C. Finn, and S. Levine. Deep online learning via meta-learning: Continual adaptation for model-based rl. *arXiv e-print 1812.07671*, 2018.

A. Nichol, J. Achiam, and J. Schulman. On first-order meta-learning algorithms. *arXiv e-print 1803.02999*, 2018.

A. Pentina, E. SDSC, and C. H. Lampert. Multi-source domain adaptation with guarantees. In *NeurIPS 2019 Workshop on Machine Learning with Guarantees*, 2019.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *NeurIPS*, pages 1177–1184, 2008.

S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2016.

C. Shui, M. Abbasi, L.-É. Robitaille, B. Wang, and C. Gagné. A principled approach for learning task similarity in multitask learning. *arXiv e-print 1903.09109*, 2019.

B. K. Sriperumbudur, K. Fukumizu, A. Gretton, B. Schölkopf, G. R. Lanckriet, et al. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6:1550–1599, 2012.

S. C. Tan and J. P. San Lau. Time series clustering: A superior alternative for market basket analysis. In *DaEng*, pages 241–248, 2014.

M. Vartak, A. Thiagarajan, C. Miranda, J. Bratman, and H. Larochelle. A meta-learning perspective on cold-start recommendations for items. In *NeurIPS*, pages 6904–6914, 2017.

O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, pages 3630–3638, 2016.

R. Vuorio, S.-H. Sun, H. Hu, and J. J. Lim. Toward multimodal model-agnostic meta-learning. *arXiv e-print 1812.07172*, 2018.

Z. Xu, L. Cao, and X. Chen. Meta-learning via weighted gradient update. *IEEE Access*, 7:110846–110855, 2019.

H. Yao, Y. Wei, J. Huang, and Z. Li. Hierarchically structured meta-learning. *arXiv e-print 1905.05301*, 2019.

C. Zhang, L. Zhang, and J. Ye. Generalization bounds for domain adaptation. In *NeurIPS*, pages 3320–3328, 2012.

C. Zhang, L. Zhang, and J. Ye. Generalization bounds for domain adaptation. *arXiv e-print 1304.1574*, 2013.

X. S. Zhang, F. Tang, H. H. Dodge, J. Zhou, and F. Wang. Metapred: Meta-learning for clinical risk prediction with limited patient electronic health records. In *KDD*, pages 2487–2495, 2019.