Sharp-MAML: Sharpness-Aware Model-Agnostic Meta Learning

Momin Abbas [*1] **Quan Xiao** [*1] **Lisha Chen** [*1] **Pin-Yu Chen** [2] **Tianyi Chen** [1]

Abstract

Model-agnostic meta learning (MAML) is currently one of the dominating approaches for few-shot meta-learning. Albeit its effectiveness, the optimization of MAML can be challenging due to the innate bilevel problem structure. Specifically, the loss landscape of MAML is much more complex with possibly more saddle points and local minimizers than its empirical risk minimization counterpart. To address this challenge, we leverage the recently invented sharpness-aware minimization and develop a sharpness-aware MAML approach that we term Sharp-MAML. We empirically demonstrate that Sharp-MAML and its computation-efficient variant can outperform the plain-vanilla MAML baseline (e.g., $+3\%$ accuracy on Mini-Imagenet). We complement the empirical study with the convergence rate analysis and the generalization bound of Sharp-MAML. To the best of our knowledge, this is the first empirical and theoretical study on sharpness-aware minimization in the context of bilevel learning. The code is available at https://github.com/mominabbass/Sharp-MAML.

1. Introduction

Humans tend to easily learn new concepts using only a handful of samples. In contrast, modern deep neural networks require thousands of samples to train a model that generalizes well to unseen data (Krizhevsky et al., 2012). Meta learning is a remedy to such a problem whereby new concepts can be learned using a limited number of samples (Schmidhuber, 1987; Vilalta & Drissi, 2002). Meta learning offers fast adaptation to unseen tasks (Thrun & Pratt, 2012; Novak & Gowin, 1984) and has been widely studied to produce state of the art results in a variety of few-shot learning settings including language and vision tasks (Munkhdalai & Yu, 2017; Nichol & Schulman, 2018; Snell et al., 2017; Wang et al., 2016; Li & Malik, 2017; Vinyals et al., 2016; Andrychowicz et al., 2016; Brock et al., 2018; Zintgraf et al., 2019a; Wang et al., 2019; Achille et al., 2019; Li et al., 2018a; Hsu et al., 2018; Obamuyide & Vlachos, 2019). In particular, model-agnostic meta learning (MAML) is one of the most popular optimization-based meta learning frameworks for few-shot learning (Finn et al., 2017; Vuorio et al., 2019; Yin et al., 2020; Obamuyide & Vlachos, 2019). MAML aims to learn an initialization such that after applying only a few number of gradient descent updates on the initialization, the adapted task-specific model can achieve desired performance on the validation dataset. MAML has been successfully implemented in various data-limited applications including medical image analysis (Maicas et al., 2018), language modelling (Huang et al., 2018), and object detection (Wang et al., 2020).

Despite its recent success on some applications, MAML faces a variety of optimization challenges. For example, MAML incurs high computation cost due to second-order derivatives, requires searching for multiple hyperparameters, and is sensitive to neural network architectures (Antoniou et al., 2019). Even if various optimization techniques can potentially overcome these training challenges (e.g., making training error small), whether the meta-model learned with limited training samples can lead to small generalization error or testing error in unseen tasks with unseen data, is not guaranteed (Rothfuss et al., 2021).

These training and generalization challenges of MAML are partially due to the nested (e.g., *bilevel*) structure of the problem, where the upper-level optimization problem learns shared model initialization and the lower-level problem optimizes task-specific models (Finn et al., 2017; Rajeswaran et al., 2019a). This is in sharp contrast to the more widely known *single-level* learning framework - empirical risk minimization (ERM). As a result, the training and generalization challenges in ERM will not only remain in MAML but may be also exacerbated by the bilevel structure of MAML. For example, as we will show later, the nonconvex loss landscape of MAML contains possibly more saddle points and local minimizers than its ERM counterpart, many of which do not have good generalization performance. Recent works have proposed various useful techniques to improve the generalization performance (Grant et al., 2018a; Park & Oliva,

[*]Equal contribution [1]Rensselaer Polytechnic Institute, Troy, NY [2]IBM Thomas J. Watson Research Center, NY, USA. Correspondence to: Tianyi Chen <chentianyi19@gmail.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

2019; Antoniou et al., 2019; Kao et al., 2021), but none of them are from the perspective of the optimization landscape.

Given the nested nature of *bilevel* learning models such as MAML, this paper aims to answer the following question:

> How can we find nonconvex bilevel learning models such as MAML that generalize well?

In an attempt to provide a satisfactory answer to this question, we use MAML as a concrete case of bilevel learning and incorporate a recently proposed sharpness-aware minimization (SAM) algorithm (Foret et al., 2021) into the MAML baseline. Originally designed for *single-level* problems such as ERM, SAM improves the generalization ability of non-convex models by leveraging the connection between generalization and sharpness of the loss landscape (Foret et al., 2021). We demonstrate the power of integrating SAM into MAML by: i) empirically showing that it outperforms the popular MAML baseline; and, ii) theoretically showing it leads to the potentially improved generalization bound. To the best of our knowledge, this is the first study on sharpness-aware minimization in the context of bilevel optimization.

1.1. Our contributions

We summarize our contributions below.

(C1) Sharpness-aware optimization for MAML with improved empirical performance. We theoretically and empirically discover that the loss landscape of bilevel models such as MAML is more involved than its ERM counterpart with possibly more saddle points and local minimizers. To overcome this challenge, we develop a sharpness-aware MAML approach that we term Sharp-MAML and its computation-efficient variant. Intuitively, Sharp-MAML avoids the sharp local minima of MAML loss functions and achieves better generalization performance. We empirically demonstrate that Sharp-MAML can outperform the plain-vanilla MAML baseline.

(C2) Optimization analysis of Sharp-MAML including MAML as a special case. We establish the $\mathcal{O}(1/\sqrt{T})$ convergence rate of Sharp-MAML through the lens of recent bilevel optimization analysis (Chen et al., 2021a), where T is the number of iterations. This corresponds to $\mathcal{O}(\epsilon^{-2})$ sample complexity as a fixed number of samples are used per iteration. The convergence rate and sample complexity match those of training single-level ERM models, and improves the known $\mathcal{O}(\epsilon^{-3})$ sample complexity of MAML.

(C3) Generalization analysis of Sharp-MAML demonstrating its improved generalization performance. We quantify the generalization performance of models learned by Sharp-MAML through the lens of a recently developed probably approximately correct (PAC)-Bayes framework (Farid & Majumdar, 2021). The generalization bound justifies the desired empirical performance of models learned from Sharp-MAML, and provides some insights on why models learned through Sharp-MAML can have better generalization performance than that from MAML.

1.2. Technical challenges

Due to the bilevel structure of both SAM and MAML, formally quantifying the optimization and generalization performance of Sharp-MAML is highly nontrivial.

Specifically, the state-of-the-art convergence analysis of bilevel optimization (e.g., (Chen et al., 2021a)) only applies to the case where the upper- and lower-level are both minimization problems. Unfortunately, this prerequisite is not satisfied in Sharp-MAML. In addition, the existing analysis of MAML in (Fallah et al., 2020) requires the growing batch size and thus results in a suboptimal $\mathcal{O}(\epsilon^{-3})$ sample complexity. From the theoretical perspective, this work not only broadens the applicability of the recent analysis of bilevel optimization (Chen et al., 2021a) to tackle Sharp-MAML problems, but also tightens the analysis of the original MAML (Fallah et al., 2020). For the generalization analysis of Sharp-MAML, different from the classical PAC-Bayes analysis for single-level problems as in SAM (Foret et al., 2021), both the lower and upper level problems of MAML contribute to the generalization error. Going beyond the PAC-Bayes analysis in (Foret et al., 2021), we further discuss how the choice of the perturbation radius in SAM affects the bound, providing insights on why Sharp-MAML improves over MAML in terms of generalization ability.

1.3. Related work

We review related work from the following three aspects.

Loss landscape of non-convex optimization. The connection between the flatness of minima and the generalization performance of the minimizers has been studied both theoretically and empirically; see e.g., (Dziugaite & Roy, 2016; Dinh et al., 2017; Keskar et al., 2017; Neyshabur et al., 2019). In a recent study, (Jiang et al., 2019) has showed empirically that sharpness-based measure has the highest correlation with generalization. Furthermore, (Izmailov et al., 2018) has showed that averaging model weights during training yields flatter minima that can generalize better.

Sharpness-aware minimization. Motivated by the connection between sharpness of a minimum and generalization performance, (Foret et al., 2021) developed the SAM algorithm that encourages the learning algorithm to converge to a flat minimum, thereby improving its generalization performance. Recent follow-up works on SAM showed the efficacy of SAM in various settings. Notably, (Bahri

et al., 2021) used SAM to improve the generalization performance of language models like text-to-text Transformer (Raffel et al., 2020) and its multilingual counterpart (Xue et al., 2020). More importantly, they empirically showed that the gains achieved by SAM are even more when *the training data are limited*. Furthermore, (Chen et al., 2021b) showed that vision models such as transformers (Dosovitskiy et al., 2020) and MLP-mixers (Tolstikhin et al., 2021) suffer from sharp loss landscapes that can be better trained via SAM. They showed that the generalization performance of resultant models improves across various tasks including supervised, adversarial, contrastive, and transfer learning (e.g., 11.0% increase in top-1 accuracy). However, existing efforts have been focusing on improving generalization performance in *single-level* problems such as ERM (Bahri et al., 2021; Chen et al., 2021b). Different from these works based on single-level ERM, we study SAM in the context of MAML through the lens of bilevel optimization. Recent works aim to reduce the computation overhead of SAM. In (Du et al., 2021), two new variants of SAM have been proposed namely, *Stochastic Weight Perturbation* and *Sharpness-sensitive Data Selection*, both of which improve the efficiency of SAM without sacrificing generalization performance. While this work showed remarkable improvement on a standard ERM model, whether it can improve the computation overhead (without sacrificing generalization) of a MAML-model is unknown.

Model-agnostic meta learning. Since it was first developed in (Finn et al., 2017), MAML has been one of the most popular optimization-based meta learning tools for fast few-shot learning. Recent studies revealed that the choice of the lower-level optimizer affects the generalization performance of MAML (Grant et al., 2018a; Antoniou et al., 2019; Park & Oliva, 2019). (Antoniou et al., 2019) pointed out a variety of issues of training MAML, such as sensitivity to neural network architectures that leads to instability during training and high computational overhead at both training and inference times. They proposed multiple ways to improve the generalization error, and stabilize training MAML, calling the resulting framework MAML++. Many recent works focus on analyzing the generalization ability of MAML (Farid & Majumdar, 2021; Denevi et al., 2018; Rothfuss et al., 2021; Chen & Chen, 2022) and improving the generalization performance of MAML (Finn & Levine, 2018; Gonzalez & Miikkulainen, 2020; Park & Oliva, 2019). However, these works do not take into account the geometry of the loss landscape of MAML. In addition to generalization-ability, recent works (Wang et al., 2021; Goldblum et al., 2020; Xu et al., 2020) investigated MAML from another important perspective of *adversarial robustness* the capabilities of a model to defend against adversarial perturbed inputs (also known as adversarial attacks in some literature). However, we focus on improving the generalization performance of the models trained by MAML with theoretical guarantees.

2. Preliminaries and Motivations

In this section, we first review the basics of MAML and describe the optimization difficulty of learning MAML models, followed by introducing the SAM method.

2.1. Problem formulation of MAML

The goal of few-shot learning is to train a model that can quickly adapt to a new task using only a few datapoints (usually 1-5 samples per task). Consider M few-shot learning tasks $\{\mathcal{T}_m\}_{m=1}^M$ drawn from a distribution $p(\mathcal{T})$. Each task m has a fine-tuning training set $\mathcal{D}_m = \cup_{i=1}^n \{(x_i, y_i)\}$ and a separate validation set $\mathcal{D}'_m = \cup_{i=1}^n \{(x_i, y_i)\}$, where data are independently and identically distributed (i.i.d.) and drawn from the per-task data distribution $\mathcal{P}_m$. MAML seeks to learn a good initialization of the model parameter θ (called the meta-model) such that fine-tuning θ via a small number of gradient updates will lead to fast learning on a new task. Consider a per datum loss $l : \Theta \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$; define the generic *empirical loss* over a finite-sample dataset $\mathcal{D}$ as $\mathcal{L}(\theta; \mathcal{D}) = \frac{1}{n}\sum_{i=1}^n l(\theta, x_i, y_i)$ and the generic *population loss* over a data distribution $\mathcal{P}$ as $\mathcal{L}(\theta; \mathcal{P}) = \mathbb{E}_{(x,y)\sim\mathcal{P}}[l(\theta, x, y)]$. For a particular task m, they become $\mathcal{L}(\theta; \mathcal{D}_m)$ or $\mathcal{L}(\theta; \mathcal{D}'_m)$ and $\mathcal{L}(\theta; \mathcal{P}_m)$.

MAML can be formulated as a bilevel optimization problem, where the fine-tuning stage forms a task-specific lower-level problem while the meta-model θ optimization forms a shared upper-level problem. Namely, the optimization problem of MAML is (Rajeswaran et al., 2019a):

$$\min_{\theta} \ \frac{1}{M}\sum_{m=1}^{M} \mathcal{L}(\theta_m^*(\theta); \mathcal{D}'_m) \tag{1a}$$

$$\text{s.t.} \ \ \theta_m^*(\theta) = \arg\min_{\theta_m} \mathcal{L}(\theta_m; \mathcal{D}_m) + \frac{\|\theta_m - \theta\|^2}{2\beta_{\text{low}}}, \ \forall m \tag{1b}$$

where β_{low} denotes the lower-level step size.

The bilevel optimization problem in (1) is difficult to solve because each upper-level update (1a) requires calling lower optimization oracle multiple times (1b). There exist many MAML algorithms to solve (1) efficiently, such as Reptile (Nichol & Schulman, 2018) and first-order MAML (Finn et al., 2017) which is an approximation to MAML obtained by ignoring second-order derivatives. We instead use the one-step gradient update (Finn et al., 2017) to approximate the lower-level problem:

$$\min_{\theta} \ F(\theta) \triangleq (1a) \ \ \text{s.t.} \ \theta'_m(\theta) = \theta - \beta_{\text{low}} \nabla \mathcal{L}(\theta; \mathcal{D}_m). \tag{2}$$

Generalization performance. We are particularly interested in the generalization performance of a meta-model θ

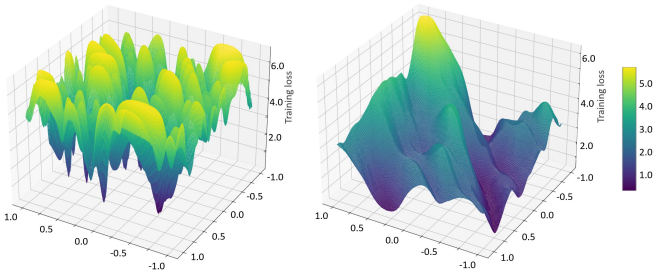

Figure 1. Loss landscapes of MAML and ERM for a single task on CIFAR-100 dataset. Details of the architecture are given in Section 5.1. We use the cross-entropy loss following the process in (Li et al., 2018b). **Left:** Loss landscape of a MAML model (5-way 1-shot). **Right:** Loss landscape of a standard ERM model.

obtained by solving the empirical MAML problem (2). The generalization performance of a meta-model θ is measured by the expected population loss

$$\mathcal{L}(\theta_m(\theta);\mathcal{P}) \triangleq \mathbb{E}_{\mathcal{T}_m}\mathbb{E}_{(x,y)\sim\mathcal{P}_m}[\mathcal{L}(\theta_m(\theta;\mathcal{D}_m);\mathcal{D}'_m)] \quad (3)$$

where the expectation is taken over the randomness of the sampled tasks as well as data in the training and validation datasets per sampled task. For notation simplicity, we define the marginal data distribution for variable (x, y) as

$$\mathcal{P}(x,y) \triangleq \mathbb{E}_{\mathcal{T}_m}[\mathcal{P}_m(x,y)] = \int P(x,y \mid \mathcal{T}_m)P(\mathcal{T}_m)d\mathcal{T}_m \quad (4)$$

and we use $\mathcal{P}$ to represent $\mathcal{P}(x, y)$ thereafter.

2.2. Local minimizer of ERM implies that of MAML

Nevertheless, even with the simple lower-level gradient descent step (2), training the upper-level meta-model θ still requires differentiating the lower update. In other words, the meta-model requires the second-order information (i.e., the Hessian) of the objective function with respect to θ, making the problem (1) more involved than an ERM formulation of the multi-task learning (called ERM thereafter), given by

$$\min_{\theta} \frac{1}{M}\sum_{m=1}^{M}\mathcal{L}(\theta;\mathcal{D}_m). \quad (5)$$

To understand the difficulty of the MAML objective in (2), we visualize its loss landscape of a particular task m and compare it with that of ERM for the same task m. Figure 1 shows the per-task loss landscapes of a meta-model θ in (1a) and a standard ERM model in (5) on CIFAR-100 dataset. We find that the loss landscape of a meta-model is indeed much more involved with more local minima, making the optimization problem difficult to solve. The following lemma also characterizes the complex landscape of meta-model on a particular task m and its proof is deferred in Appendix B.

Lemma 1 (Local minimizers of MAML)**.** *Consider the one-step gradient fine-tuning step (2). For any $m \in \mathcal{M}$, assume $\mathcal{L}(\theta;\mathcal{D}_m)$ has continuous third-order derivatives. Then for a particular task m, the following two statements hold*
a) the stationary points for $\mathcal{L}(\theta;\mathcal{D}_m)$ are also the stationary points for $\mathcal{L}(\theta'_m(\theta);\mathcal{D}_m)$; and,
b) the local minimizers for $\mathcal{L}(\theta;\mathcal{D}_m)$ are also the local minimizers for $\mathcal{L}(\theta'_m(\theta);\mathcal{D}_m)$.

Lemma 1 shows that for a given task m, the number of stationary points and local minimizers for ERM's loss $\mathcal{L}(\theta;\mathcal{D}_m)$ are fewer than that of MAML's loss $\mathcal{L}(\theta'_m(\theta);\mathcal{D}_m)$, which is aligned with the empirical observations in Figure 1. While some of the local minimizers in MAML's loss landscape are indeed effective few-shot learners, there are a number of sharp local minimizers in MAML that may have undesired generalization performance. It also suggests that the optimization of MAML can be more challenging than its ERM counterpart.

2.3. Sharpness aware minimization

SAM is a recently developed technique that leverages the geometry of the loss landscape to improve the generalization performance by simultaneously minimizing the loss value and the loss sharpness (Foret et al., 2021). Given the empirical loss $\mathcal{L}(\theta;\mathcal{D})$, the goal of training is to choose θ having low population loss $\mathcal{L}(\theta;\mathcal{P})$. SAM achieves this through the following optimization problem

$$\min_{\theta} \mathcal{L}^{\text{sam}}(\theta;\mathcal{D}) \text{ with } \mathcal{L}^{\text{sam}}(\theta;\mathcal{D}) \triangleq \max_{||\epsilon||_2\leq\alpha} \mathcal{L}(\theta+\epsilon;\mathcal{D}). \quad (6)$$

Given θ, the maximization in (6) seeks to find the weight perturbation ϵ in the Euclidean ball with radius α that maximizes the empirical loss. If we define the **sharpness** as

$$\max_{||\epsilon||_2\leq\alpha} [\mathcal{L}(\theta+\epsilon;\mathcal{D}) - \mathcal{L}(\theta;\mathcal{D})] \quad (7)$$

then (6) essentially minimizes the sum of the sharpness and the empirical loss $\mathcal{L}(\theta;\mathcal{D})$. While the maximization in (6) is generally costly, a closed-form approximate maximizer has been proposed in (Foret et al., 2021) by invoking the Taylor expansion of the empirical loss. In such case, SAM seeks to find a flat minimum by iteratively applying the following two-step procedure at each iteration t, that is

$$\epsilon(\theta^t) = \alpha\nabla\mathcal{L}(\theta^t;\mathcal{D})/||\nabla\mathcal{L}(\theta^t;\mathcal{D})||_2 \quad (8a)$$
$$\theta^{t+1} = \theta^t - \beta^t(\nabla\mathcal{L}(\theta^t+\epsilon(\theta^t);\mathcal{D})) \quad (8b)$$

where β^t is an appropriately scheduled learning rate. In (8b) and thereafter, the notation $\nabla\mathcal{L}(\theta+\epsilon_m(\theta))$ means $\nabla\mathcal{L}(\theta+\epsilon_m(\theta)) \triangleq \nabla_x\mathcal{L}(x)|_{x=\theta+\epsilon_m(\theta)}$. SAM works particularly well for complex and non-convex problems having a myriad of local minima, and where different minima yield models with different generalization abilities.

3. Sharp-MAML: Sharpness-aware MAML

As discussed in Section 2.1, MAML has a complex loss landscape with multiple local and global minima, that may yield similar values of empirical loss $\mathcal{L}(\theta;\mathcal{D})$ while having significantly different generalization performance. Therefore, we propose integrating SAM with MAML, which is a new bilevel optimization problem.

3.1. Problem formulation of Sharp-MAML

We propose a unified optimization framework for Sharpness-aware MAML that we term Sharp-MAML by using two hyperparameters $\alpha_{\text{up}} \geq 0$ and $\alpha_{\text{low}} \geq 0$, that is:

$$\textbf{(P)} \quad \min_{\theta} \max_{||\epsilon||_2 \leq \alpha_{\text{up}}} \sum_{m=1}^{M} \mathcal{L}(\theta_m^*(\theta+\epsilon);\mathcal{D}'_m) \quad \text{(upper)}$$
$$\text{s.t.} \quad \theta_m^*(\theta) = \arg\min_{\theta_m} \max_{||\epsilon_m||_2 \leq \alpha_{\text{low}}} \mathcal{L}(\theta_m+\epsilon_m;\mathcal{D}_m) + \frac{\|\theta_m-\theta\|^2}{2\beta_{\text{low}}}, \ m=1,...,M. \quad \text{(lower)}$$

Compared with the bilevel formulation of MAML in (1), the above Sharp-MAML formulation is a four-level problem. However, in our algorithm design, we will efficiently approximate the two maximizations in (**P**) so that the cost of Sharp-MAML is almost the same as that of MAML.

In what follows, we list three main technical questions that we aim to address.

Q1. The choice of α_{up}, α_{low} determines the specific scenario of integrating SAM with MAML. Applying SAM to both fine-tuning and meta-update stages would be computationally very expensive. Spurred by that, we ask: *Is it possible to achieve better generalization by incorporating SAM into only either upper- or lower-level problem?*

Q2. Both MAML in (1) and SAM in (6) are bilevel optimization problems requiring several lower optimization steps. Thus, we also study whether or not the computationally-efficient alternatives (e.g. ESAM (Du et al., 2021), ANIL (Raghu et al., 2020)) can promise good generalization.

Q3. The theoretical motivation for SAM has been illustrated in (Foret et al., 2021) by bounding generalization ability in terms of neighborhood-wise training loss. Spurred by that, we further ask: *Can we explain and theoretically justify why integrating SAM with MAML is effective in promoting generalization performance of MAML models?*

3.2. Algorithm development

Based on (**P**), we focus on three variants of Sharp-MAML that differ in their respective computational complexity:

(a) Sharp-MAML$_{\text{low}}$: SAM is applied to only the fine-tuning step, i.e., $\alpha_{\text{low}} > 0$ and $\alpha_{\text{up}} = 0$.
(b) Sharp-MAML$_{\text{up}}$: SAM is applied to only the meta-update step, i.e., $\alpha_{\text{low}} = 0$ and $\alpha_{\text{up}} > 0$.
(c) Sharp-MAML$_{\text{both}}$: SAM is applied to both fine-tuning and meta-update steps, i.e., $\alpha_{\text{up}}, \alpha_{\text{low}} > 0$.

Below we only introduce Sharp-MAML$_{\text{both}}$ in detail and leave the pseudo-code of the other two variants in Appendix A since the other two variants can be deduced from Sharp-MAML$_{\text{both}}$. For the sake of convenience, we define the biased mini-batch gradient descent (BGD) at point $\theta^t+\epsilon$ using gradient at $\theta^t+\epsilon+\epsilon_m$ as

$$\text{BGD}_m(\theta^t,\epsilon,\epsilon_m) \triangleq \theta^t+\epsilon-\beta_{\text{low}}\widetilde{\nabla}\mathcal{L}(\theta^t+\epsilon+\epsilon_m;\mathcal{D}_m) \tag{9}$$

where ϵ and ϵ_m are perturbation vectors that can be computed accordingly to different Sharp-MAML, and $\widetilde{\nabla}\mathcal{L}(\,\cdot\,;\mathcal{D}_m)$ is an unbiased estimator of $\nabla\mathcal{L}(\,\cdot\,;\mathcal{D}_m)$ which can be assessed by mini-batch evaluation. Moreover, letting $\tilde{\theta}_m(\theta^t)=\text{BGD}_m(\theta^t,\epsilon,\epsilon_m)$, we define

$$\nabla_{\theta_t}\mathcal{L}(\tilde{\theta}_m(\theta^t);\mathcal{D}'_m) \triangleq (I-\beta_{\text{low}}\nabla^2\mathcal{L}(\theta^t+\epsilon+\epsilon_m;\mathcal{D}_m))\nabla\mathcal{L}(\tilde{\theta}_m(\theta^t);\mathcal{D}'_m) \tag{10}$$

and $\nabla^2\mathcal{L}(\theta^t+\epsilon+\epsilon_m;\mathcal{D}_m)$ is the Hessian matrix of $\mathcal{L}(\,\cdot\,;\mathcal{D}_m)$ at $\theta^t+\epsilon+\epsilon_m$.

Sharp-MAML$_{\text{both}}$. For each task m, we compute its corresponding perturbation $\epsilon_m(\theta^t)$ as follows:

$$\epsilon_m(\theta^t)=\alpha_{\text{low}}\widetilde{\nabla}\mathcal{L}(\theta^t;\mathcal{D}_m)/||\widetilde{\nabla}\mathcal{L}(\theta^t;\mathcal{D}_m)||_2. \tag{11}$$

Thereafter, the fine-tuning step is carried out by performing gradient descent at θ^t using the gradient at the maximum point $\theta^t+\epsilon_m(\theta^t)$ using (9):

$$\tilde{\theta}_m^1(\theta^t)=\text{BGD}_m(\theta^t,0,\epsilon_m(\theta^t)). \tag{12}$$

After we obtain $\tilde{\theta}_m^1(\theta^t)$ for all tasks, we compute the mini-batch gradient estimator of the upper loss i.e., $\nabla h = \widetilde{\nabla}_{\theta^t}\sum_{m=1}^M\mathcal{L}(\tilde{\theta}_m^1(\theta^t);\mathcal{D}'_m)$ which is an unbiased estimator of the upper-level gradient $\nabla_{\theta^t}\sum_{m=1}^M\mathcal{L}(\tilde{\theta}_m^1(\theta^t);\mathcal{D}'_m)$, and use it to compute the meta perturbation $\epsilon(\theta^t)$ by:

$$\epsilon(\theta^t)=\alpha_{\text{up}}\nabla h/||\nabla h||_2. \tag{13}$$

Afterwards, we compute the new perturbed fine-tuned parameter, denoted by $\tilde{\theta}_m^2(\theta^t)$, using the gradient at the maximum point $\theta^t+\epsilon(\theta^t)+\epsilon_m(\theta^t)$ in (9), that is:

$$\tilde{\theta}_m^2(\theta^t)=\text{BGD}_m(\theta^t,\epsilon(\theta^t),\epsilon_m(\theta^t)). \tag{14}$$

Finally, for the meta-update stage, we evaluate the upper loss using the fine-tuned parameter $\tilde{\theta}_m^2(\theta^t)$ obtained from

Algorithm 1 Pseudo-code for **Sharp-MAML$_{\text{both}}$**; red lines need to be modified for **Sharp-MAML$_{\text{up}}$**; blue lines need to be modified for **Sharp-MAML$_{\text{low}}$**

Require: $p(\mathcal{T})$: distribution over tasks
Require: $\beta_{\text{low}}, \beta_{\text{up}}$: step sizes
Require: $\alpha_{\text{low}} > 0, \alpha_{\text{up}} > 0$: perturbation radii
for $t = 1, \cdots, T$ **do**
  Sample batch of tasks $\mathcal{T}_m \sim p(\mathcal{T})$
  for all $\mathcal{T}_m$ **do**
    Sample K examples from $\mathcal{D}_m$
    Evaluate $\widetilde{\nabla}\mathcal{L}(\theta^t; \mathcal{D}_m)$
    Compute perturbation $\epsilon_m(\theta^t)$ via (11)
    Compute fine-tuned parameter $\tilde{\theta}^1_m(\theta^t)$ via (12)
    Sample data from $\mathcal{D}'_m$ for meta-update
  end for
  Compute $\sum_{m=1}^M \widetilde{\nabla}\mathcal{L}(\tilde{\theta}^1_m(\theta^t); \mathcal{D}'_m)$
  Compute perturbation $\epsilon(\theta^t)$ via (13)
  Update θ via (15) using $\hat{\theta}^2_m(\theta^t)$ from (14)
end for

(14) and update the meta-parameter θ via:

$$\theta^{t+1} = \theta^t - \beta_{\text{up}} \sum_{m=1}^{M} \widetilde{\nabla}_{\theta^t} \mathcal{L}(\tilde{\theta}^2_m(\theta^t); \mathcal{D}'_m). \tag{15}$$

See the pseudocode of Sharp-MAML$_{\text{both}}$ in Algorithm 1. The algorithms for Sharp-MAML$_{\text{up}}$ and Sharp-MAML$_{\text{low}}$ can be deduced by setting $\epsilon_m(\theta_t) = 0$ and $\epsilon(\theta_t) = 0$ in Algorithm 1, respectively, which are formally stated in Algorithm 3 and Algorithm 2 in Appendix A.

4. Theoretical Analysis of Sharp-MAML

In this part, we rigorously analyze the performance of the proposed Sharp-MAML method in terms of the convergence rate and the generalization error.

4.1. Optimization analysis

To quantify the optimization performance of solving the one-step version of (1), we introduce the following assumptions.

Assumption 1 (Lipschitz continuity)**.** *Assume that $\mathcal{L}(\theta; \mathcal{D}'_m), \nabla\mathcal{L}(\theta; \mathcal{D}_m), \nabla\mathcal{L}(\theta; \mathcal{D}'_m), \nabla^2\mathcal{L}(\theta; \mathcal{D}_m), \forall m$ are Lipschitz continuous with constant $\ell_0, \ell_1, \ell_1, \ell_2$.*

Assumption 2 (Stochastic derivatives)**.** *Assume that $\widetilde{\nabla}\mathcal{L}(\theta; \mathcal{D}_m), \widetilde{\nabla}^2\mathcal{L}(\theta; \mathcal{D}_m), \widetilde{\nabla}\mathcal{L}(\theta; \mathcal{D}'_m)$ are unbiased estimator of $\nabla\mathcal{L}(\theta; \mathcal{D}_m), \nabla^2\mathcal{L}(\theta; \mathcal{D}_m), \nabla\mathcal{L}(\theta; \mathcal{D}'_m)$ respectively and their variances are bounded by σ^2.*

Assumptions 1–2 also appear similarly in the convergence analysis of meta learning and bilevel optimization (Finn et al., 2019; Rajeswaran et al., 2019a; Fallah et al., 2020; Chen et al., 2021a; Ji et al., 2022; Chen et al., 2022).

With the above assumptions, we introduce a novel biased MAML framework which includes MAML and sharp-MAML as special cases and get the following theorem. The proof is deferred in Appendix C.

Theorem 1. *Under Assumption 1–2, and choosing stepsizes and perturbation radii $\beta_{\text{low}}, \beta_{\text{up}}, \alpha_{\text{up}} = \mathcal{O}(\frac{1}{\sqrt{T}}), \alpha_{\text{low}} = \mathcal{O}(1)$, with some proper constants, we can get that the iterates $\{\theta^t\}$ generated by Sharp-MAML$_{\text{up}}$, Sharp-MAML$_{\text{low}}$ and Sharp-MAML$_{\text{both}}$ satisfy*

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left[\|\nabla F(\theta^t)\|^2\right] = \mathcal{O}\left(\frac{1}{\sqrt{T}}\right) \tag{16}$$

where $F(\theta)$ is the objective function of MAML in (2).

Theorem 1 implies that by choosing a proper perturbation threshold, all three versions of Sharp-MAML can still find ϵ stationary points for MAML objective (2) with $\mathcal{O}(\epsilon^{-2})$ iterations and $\mathcal{O}(\epsilon^{-2})$ samples, which matches or even improves the state-of-the-art sample complexity of MAML (Rajeswaran et al., 2019a; Fallah et al., 2020; Ji et al., 2022).

4.2. Generalization analysis

To analyze the generalization error of Sharp-MAML, we make similar assumptions to Theorem 2 in (Foret et al., 2021). Recall the *population loss* $\mathcal{L}(\theta; \mathcal{P}) = \mathbb{E}_{(x,y)\sim\mathcal{P}}[l(\theta, x, y)]$. Denote the stationary point obtained by Sharp-MAML$_{\text{up}}$ algorithm as $\hat{\theta}$. Note that the Sharp-MAML adopts gradient descent (GD) as the lower level algorithm, which is uniformly stable based on Definition 1.

Definition 1 ((Hardt et al., 2016))**.** *An algorithm A is γ-uniformly stable if for all data sets $S, S' \in Z^n$ such that S and S' differ in at most one example, we have*

$$\sup_S \left|\mathbb{E}_S\left[l\left(A(S); x, y\right) - l\left(A\left(S'\right); x, y\right)\right]\right| \leq \gamma \tag{17}$$

where $A(S)$ and $A(S')$ are the outputs of the algorithm A given datasets S and S'.

With the above definition of uniform stability, we are ready to establish the generalization performance. We defer the proof of Theorem 2 to Appendix D.

Theorem 2. *Assume loss function $\mathcal{L}(\cdot)$ is bounded: $0 \leq \mathcal{L}(\theta'_m; \mathcal{D}) \leq 1$, for θ'_m defined in (2), and any $\mathcal{D}$. Define $F(\theta; \mathcal{P}) = \mathbb{E}_{\mathcal{D}\sim\mathcal{P}}[F(\theta; \mathcal{D})]$. Assume $F(\hat{\theta}; \mathcal{P}) \leq \mathbb{E}_{\epsilon\sim\mathcal{N}(0,\alpha^2\mathbf{I})}\left[F(\hat{\theta}+\epsilon; \mathcal{P})\right]$ at the stationary point of the Sharp-MAML$_{\text{up}}$ denoted by $\hat{\theta}$. For parameter $\theta'_m(\hat{\theta}; \mathcal{D})$ learned with γ_{A} uniformly stable algorithm A from $\hat{\theta} \in \mathbb{R}^k$, with probability $1-\delta$ over the choice of the training set*

$\mathcal{D} \sim \mathcal{P}$, with $|\mathcal{D}| = nM$, it holds that

$$F(\hat{\theta};\mathcal{P}) \leq \max_{\|\epsilon\|_2 \leq \alpha} F(\hat{\theta}+\epsilon;\mathcal{D}) + \gamma_{\mathrm{A}} + \sqrt{\frac{k\ln\left(1+\frac{\|\hat{\theta}\|_2^2}{\alpha^2}\left(1+\sqrt{\frac{\ln(nM)}{k}}\right)^2\right)+2\ln\frac{1}{\delta}+5\ln(nM)}{4nM}}. \tag{18}$$

Improved upper bound of generalization error. Theorem 2 shows that the difference between the population loss and the empirical loss of Sharp-MAML$_{\mathrm{up}}$ is bounded by the stability of the lower-level update γ_A and another $\tilde{\mathcal{O}}(k/nM)$ term that vanishes as the number of meta-training data goes to infinity. The lower-level update GD has uniform stability of order $\mathcal{O}(1/n)$ (Hardt et al., 2016). Also, it is worth noting that the upper bound of the population loss on the right-hand side (RHS) of (18), is a function of α. And for any sufficiently small $\alpha_0 > 0$, we can find some $\alpha_1 > \alpha_0$, where this function takes smaller value than at α_0 (see the proof in Appendix D). This suggests that a choice of α arbitrarily close to zero, in which case Sharp-MAML reduces to the original MAML method, is not optimal in terms of the generalization error upper bound. Therefore, it shows Sharp-MAML has smaller generalization error upper bound than conventional MAML. The analysis can be extended to Sharp-MAML$_{\mathrm{both}}$ in a similar way.

5. Numerical Results

In this section, we demonstrate the effectiveness of Sharp-MAML by comparing it with several popular MAML baselines in terms of generalization and computation cost. We evaluate Sharp-MAML on 5-way 1-shot and 5-way 5-shot settings on the Mini-Imagenet dataset and present the results on Omniglot dataset in Appendix E.

5.1. Experiment setups

Our model follows the same architecture used by (Vinyals et al., 2016), comprising of 4 modules with a 3×3 convolutions with 64 filters followed by batch normalization (Ioffe & Szegedy, 2015), a ReLU non-linearity, and a 2×2 max-pooling. We follow the experimental protocol in (Finn et al., 2017). The models were trained using the SAM[1] algorithm with Adam as the base optimizer and learning rate $\alpha = 0.001$. Following (Ravi & Larochelle, 2017), 15 examples per class were used to evaluate the post-update meta-gradient. The values of α_{low}, α_{up} are taken from a set of $\{0.05, 0.005, 0.0005, 0.00005\}$ and each experiment is run on each value for three random seeds. We choose the inner gradient steps from a set of $\{3, 5, 7, 10\}$. The step size is chosen from a set of $\{0.1, 0.01, 0.001\}$. For Sharp-MAML$_{\mathrm{both}}$ we use the same value of α_{low}, α_{up} in each experiment. We report the best results in Tables 1 and 2.

[1]We used the open-source SAM PyTorch implementation available at https://github.com/davda54/sam

Table 1. Results on Mini-Imagenet (5-way 1-shot). Our reproduced result of MAML is close to that of the original*.

ALGORITHMS	ACCURACY
MATCHING NETS	43.56%
IMAML (RAJESWARAN ET AL., 2019B)	49.30 %
CAVIA (ZINTGRAF ET AL., 2019B)	47.24 %
REPTILE (NICHOL & SCHULMAN, 2018)	49.97 %
FOMAML (NICHOL & SCHULMAN, 2018)	48.07 %
LLAMA (GRANT ET AL., 2018B)	49.40 %
BMAML (YOON ET AL., 2018)	49.17 %
MAML (REPRODUCED)	47.13%
SHARP-MAML$_{\mathrm{low}}$	49.72%
SHARP-MAML$_{\mathrm{up}}$	49.56%
SHARP-MAML$_{\mathrm{both}}$	50.28%

* reproduced using the Torchmeta (Deleu et al., 2019) library

Table 2. Results on Mini-Imagenet (5-way 5-shot). Our reproduced result of MAML is close to that of the original*.

ALGORITHMS	ACCURACY
MATCHING NETS	55.31%
CAVIA (ZINTGRAF ET AL., 2019B)	59.05%
REPTILE (NICHOL & SCHULMAN, 2018)	65.99%
FOMAML (NICHOL & SCHULMAN, 2018)	63.15 %
BMAML (YOON ET AL., 2018)	64.23 %
MAML (REPRODUCED)	62.20%
SHARP-MAML$_{\mathrm{low}}$	63.18%
SHARP-MAML$_{\mathrm{up}}$	63.06%
SHARP-MAML$_{\mathrm{both}}$	65.04%

* reproduced using the Torchmeta (Deleu et al., 2019) library

One Sharp-MAML update executes two backpropagation operations (i.e., one to compute $\epsilon(\theta)$ and another to compute the final gradient). Therefore, for a fair comparison, we execute each MAML training run twice as many epochs as each Sharp-MAML training run and report the best score achieved by each MAML training run across either the standard epoch count or the doubled epoch count.

5.2. Sharp-MAML versus MAML baselines

Regarding baselines, we use the MAML (Finn et al., 2017), Matching Nets (Vinyals et al., 2016), CAVIA (Zintgraf et al., 2019b), Reptile (Nichol & Schulman, 2018), FOMAML (Nichol & Schulman, 2018), and BMAML (Yoon et al., 2018).

In Tables 1 and 2, we report the accuracy of three variants of Sharp-MAML and other baselines on the Mini-Imagenet dataset in the 5-way 1-shot and 5-way 5-shot settings respectively. We observe that Sharp-MAML$_{\mathrm{low}}$ outperforms MAML in all cases, exhibiting the advantage of our methods. The results on the Omniglot dataset are reported in Table 4 and Table 5 of Appendix. Our results verify the efficacy

Table 3. Results on Mini-Imagenet (5-way 1-shot).

ALGORITHMS	ACCURACY	TIME†
MAML (REPRODUCED)	47.13%	x1
SHARP-MAML$_{\text{low}}$	49.72%	x2.60
SHARP-MAML$_{\text{up}}$	49.56%	x3.60
SHARP-MAML$_{\text{both}}$	50.28%	x4.60
SHARP-MAML$_{\text{low}}$ -ANIL	49.19%	x1.40
ESHARP-MAML$_{\text{low}}$	48.90%	x2.20
ESHARP-MAML$_{\text{low}}$ -ANIL	49.03%	x1.20

† execution time is normalized to MAML training time

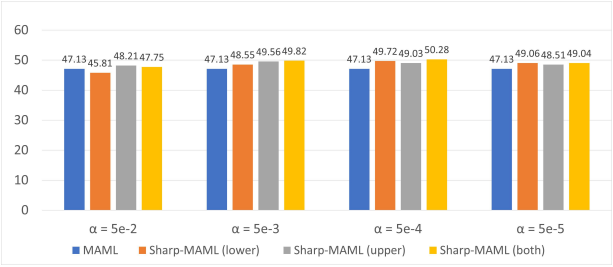

Figure 2. Performance under different values of α_{low}, α_{up} on Mini-Imagenet (5-way 1-shot). For Sharp-MAML$_{\text{both}}$, we used the same value of α_{low} and α_{up} (i.e., $\alpha_{\text{low}} = \alpha_{\text{up}}$).

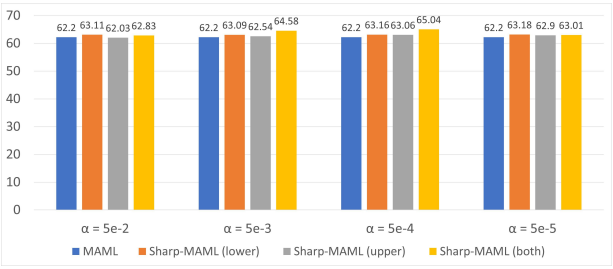

Figure 3. Performance under different values of α_{low}, α_{up} on Mini-Imagenet (5-way 5-shot). For Sharp-MAML$_{\text{both}}$, we used the same value of α_{low} and α_{up} (i.e., $\alpha_{\text{low}} = \alpha_{\text{up}}$)

of all the three variants of Sharp-MAML, suggesting that SAM indeed improves the generalization performance of bi-level models like MAML by seeking out flatter minima.

Since Sharp-MAML requires one more gradient computation per iteration than the original MAML, for a fair comparison, we report the execution times in Table 3. The results show that Sharp-MAML$_{\text{low}}$ requires the least amount of additional computation while still achieving significant performance gains. Sharp-MAML$_{\text{up}}$ and Sharp-MAML$_{\text{both}}$ also improves the performance significantly but both approaches have a higher computation than Sharp-MAML$_{\text{low}}$ since the computation of additional Hessians is needed for the meta-update gradient.

5.3. Ablation study and loss landscape visualization

We conduct an ablation study on the effect of perturbation radii α_{low} and α_{up} on the three Sharp-MAML variants. Figure 2 and Figure 3 summarize the results on the Mini-Imagenet dataset. We observe that all the three Sharp-MAML variants outperform the original MAML for almost all the values of α_{low}, α_{up} we used in our experiments. Therefore, integrating SAM at any/both stage(s) gives better performance than the original SAM for a wide range of values of the perturbation sizes, reducing the need to fine-tune these hyperparameters. This also suggests that SAM is effectively avoiding bad local minimum in MAML loss landscape (cf. Figure 1) for a wide range of perturbation radii. In Figure 4, we plot the loss landscapes of MAML and Sharp-MAML, and observe that Sharp-MAML indeed seeks out landscapes that are smoother as compared to the landscape of original MAML and hence, meets our theoretical characterization of improved generalization performance. Furthermore, the generalization error of Sharp-MAML$_{\text{both}}$ is found to be 34.58%/8.56% as compared to 37.46%/11.58% of MAML for the 5-way 1-shot/5-shot Mini-Imagenet, which explains the advantage of our approach.

5.4. Computationally-efficient version of Sharp-MAML

Next we investigate if the computational overhead of Sharp-MAML$_{\text{low}}$ can be further reduced by leveraging the computationally-efficient MAML – an almost-no-inner-loop (ANIL) method (Raghu et al., 2020) and the computationally-efficient SAM – ESAM (Du et al., 2021). Sharp-MAML-ANIL is the case when we use ANIL with Sharp-MAML$_{\text{low}}$; ESharp-MAML is the case when we use ESAM with Sharp-MAML$_{\text{low}}$; ESharp-MAML-ANIL is Sharp-MAML$_{\text{low}}$ with both ANIL and ESAM.

In ANIL, fine-tuning is only applied to the task-specific head with a frozen representation network from the meta-model. Motivated by (Raghu et al., 2020), we ask if incorporating Sharp-MAML$_{\text{low}}$ in ANIL can ameliorate the computational overhead while preserving the performance gains obtained using the model trained on Sharp-MAML$_{\text{low}}$. ANIL decomposes the meta-model θ into two parts: the representation encoding network denoted by θ_r and the classification head denoted by θ_c i.e., $\theta \triangleq [\theta_r, \theta_c]$. Different from (1b), ANIL then only fine-tunes θ_c over a specific task m, given by:

$$\theta'_m(\theta) = \arg \min_{\theta_m;\, \theta_{r,m}=\theta_r} \mathcal{L}(\theta_{c,m}, \theta_{r,m}; \mathcal{D}_m). \qquad (19)$$

In other words, the initialized representation θ_r, which comprises most of the network, is unchanged during fine-tuning.

ESAM leverages two training strategies, Stochastic Weight Perturbation (SWP) and Sharpness-Sensitive Data Selection (SDS). SWP saves computation by stochastically selecting set of weights in each iteration, and SDS judiciously selects a subset of data that is sensitive to sharpness. To be specific, SWP uses a gradient mask $\mathbf{v} = (v_1, ..., v_M)$ where $v_i \overset{\text{i.i.d.}}{\sim}$

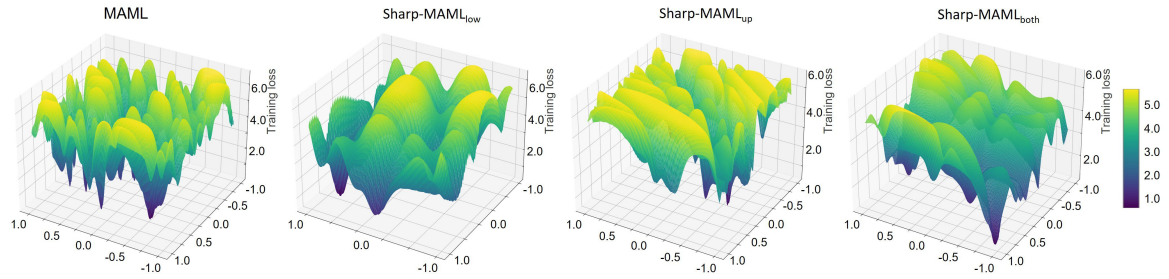

Figure 4. One-task cross entropy loss landscapes of MAML and different variants of Sharp-MAML trained on CIFAR-100 dataset (5-way 1-shot setting) using class one. The plots are generated following (Li et al., 2018b). Details of the architecture are given in Section 5.1.

Bern(ξ), $\forall i$. In SDS, instead of computing $\mathcal{L}_{\mathcal{N}}(\theta+\epsilon(\theta))$ over all samples, $\mathcal{N}$, a *subset* of samples, $\mathcal{N}^+$, is selected, whose loss values increase the most with $\epsilon(\theta)$; that is,

$$\mathcal{N}^+ \triangleq \{(x_i,y_i) : l(\theta+\epsilon;x_i,y_i) - l(\theta;x_i,y_i) > \tau\}$$
$$\mathcal{N}^- \triangleq \{(x_i,y_i) : l(\theta+\epsilon;x_i,y_i) - l(\theta;x_i,y_i) < \tau\}$$

where the threshold τ controls the size of $\mathcal{N}^+$. Furthermore, $\mu = |\mathcal{N}^+|/|\mathcal{N}|$ is ratio of number of selected samples with respect to the batch size and determines the exact value of τ. In practice, μ is selected to maximize efficiency while preserving generalization performance.

In Table 3, we report our results on three computationally efficient versions of Sharp-MAML. We find that Sharp-MAML-ANIL is comparable in performance to Sharp-MAML while requiring almost 86% less computation. ESharp-MAML also reduces the computation, but has slight performance loss. We suspect that this is due to the nested structure of the meta-learning problem that adversely affects the two training strategies used in ESAM. We further investigate the effect of both ANIL and ESAM on Sharp-MAML and observe significant reduction in computation (116% faster) with slight degradation in performance as compared to Sharp-MAML. When compared to MAML, ESharp-MAML-ANIL performs considerably better (+1.90$\%$ gain in accuracy) while requiring only 20% more computation.

6. Conclusions

In this paper, we study sharpness-aware minimization (SAM) in the context of model-agnostic meta-learning (MAML) through the lens of bilevel optimization. We name our new MAML method Sharp-MAML. Through a systematic empirical and theoretical study, we find that adding SAM into any/both fine-tuning or/and meta-update stages improves the generalization performance. We further find that incorporating SAM in the fine-tuning stage alone is the best trade-off between performance and computation. To further save computation overhead, we leverage the techniques such as efficient SAM and almost no inner loop to speed up Sharp-MAML, without sacrificing generalization.

Acknowledgements

The work was partially supported by NSF MoDL-SCALE Grant 2134168 and the Rensselaer-IBM AI Research Collaboration (http://airc.rpi.edu), part of the IBM AI Horizons Network (http://ibm.biz/AIHorizons).

References

Achille, A., Lam, M., Tewari, R., Ravichandran, A., Maji, S., Fowlkes, C. C., Soatto, S., and Perona, P. Task2vec: Task embedding for meta-learning. In *Proc. International Conference on Computer Vision*, pp. 6430–6439, Seoul, South Korea, 2019.

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., and d. Freitas, N. Learning to learn by gradient descent by gradient descent. In *Proc. Advances in Neural Information Processing Systems*, pp. 3981–3989, Barcelona, Spain, December 2016.

Antoniou, A., Edwards, H., and Storkey, A. How to train your maml. In *Proc. International Conference on Learning Representations*, New Orleans, LA, 2019.

Bahri, D., Mobahi, H., and Tay, Y. Sharpness-aware minimization improves language model generalization. *arXiv preprint:2110.08529*, 2021.

Brock, A., Lim, T., Ritchie, J. M., and Weston, N. SmaSH: One-shot model architecture search through hypernetworks. In *Proc. International Conference on Learning Representations*, Vancouver, Canada, April 2018.

Chen, L. and Chen, T. Is Bayesian model-agnostic meta learning better than model-agnostic meta learning, provably? In *Proc. International Conference on Artificial Intelligence and Statistics*, March 2022.

Chen, T., Sun, Y., and Yin, W. Closing the gap: Tighter analysis of alternating stochastic gradient methods for

bilevel problems. In *Proc. Advances in Neural Information Processing Systems*, volume 34, Virtual, 2021a.

Chen, T., Sun, Y., Xiao, Q., and Yin, W. A single-timescale method for stochastic bilevel optimization. In *Proc. International Conference on Artificial Intelligence and Statistics*, pp. 2466–2488, March 2022.

Chen, X., Hsieh, C., and Gong, B. When vision transformers outperform resnets without pretraining or strong data augmentations. *arXiv preprint:2106.0154*, 2021b.

Deleu, T., Würfl, T., Samiei, M., Cohen, J. P., and Bengio, Y. Torchmeta: A Meta-Learning library for PyTorch, 2019. URL https://arxiv.org/abs/1909.06576.

Denevi, G., Ciliberto, C., Stamos, D., and Pontil, M. Learning to learn around a common mean. In *Proc. Advances in Neural Information Processing Systems*, volume 31, Montreal, Canada, 2018.

Dinh, L., Pascanu, R., S.Bengio, and Bengio, Y. Sharp minima can generalize for deep nets. In *Proc. International Conference on Machine Learning*, pp. 1019–1028, Sydney, Australia, August 2017.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., S. Gelly, J. U., and Houlsby, N. An image is worth 16$\times$ 16 words: Transformers for image recognition at scale. In *Proc. International Conference on Learning Representations*, Virtual, April 2020.

Du, J., Yan, H., Feng, J., Zhou, J. T., Zhen, L., Goh, R. S. M., and Tan, V. Y. F. Efficient sharpness-aware minimization for improved training of neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Virtual, June 2021.

Dziugaite, G. and Roy, D. M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proc. Conference on Uncertainty in Artificial Intelligence*, Sydney, Australia, August 2016.

Fallah, A., Mokhtari, A., and Ozdaglar, A. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pp. 1082–1092, 2020.

Farid, A. and Majumdar, A. Generalization bounds for meta-learning via pac-bayes and uniform stability. *Advances in Neural Information Processing Systems*, 34, 2021.

Finn, C. and Levine, S. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. In *Proc. International Conference on Learning Representations*, Vancouver, Canada, 2018.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. International Conference on Machine Learning*, Sydney, Australia, 2017.

Finn, C., Rajeswaran, A., Kakade, S., and Levine, S. Online meta-learning. In *International Conference on Machine Learning*, pp. 1920–1930, 2019.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. In *Proc. International Conference on Learning Representations*, Virtual, April 2021.

Goldblum, M., Fowl, L., and Goldstein, T. Adversarially robust few-shot learning: A meta-learning approach. In *Proc. Advances in Neural Information Processing Systems*, Virtual, December 2020.

Gonzalez, S. and Miikkulainen, R. Improved training speed, accuracy, and data utilization through loss function optimization. In *Proc. IEEE Congress on Evolutionary Computation*, pp. 1–8, Glasgow, United Kingdom, 2020.

Grant, E., Finn, C., Levine, S., Darrell, T., and Griffiths, T. Recasting gradient-based meta-learning as hierarchical bayes. In *Proc. International Conference on Learning Representations*, April 2018a.

Grant, E., Finn, C., Levine, S., Darrell, T., and Griffiths, T. Recasting gradient-based meta-learning as hierarchical bayes. In *Proc. International Conference on Learning Representations*, Vancouver, Canada, 2018b.

Hardt, M., Recht, B., and Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pp. 1225–1234, 2016.

Hsu, K., Levine, S., and Fin, C. Unsupervised learning via meta-learning. In *Proc. International Conference on Learning Representations*, Vancouver, Canada, April 2018.

Huang, P., Wang, C., Singh, R., Yih, W., and He, X. Natural language to structured query generation via meta-learning. In *Proc. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 732–738, New Orleans, LA, 2018.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. International Conference on Machine Learning*, Lille, France, June 2015.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima

and better generalization. In *Proc. Conference on Uncertainty in Artificial Intelligence*, pp. 876–885, Monterey, CA, 2018.

Ji, K., Yang, J., and Liang, Y. Theoretical convergence of multi-step model-agnostic meta-learning. *Journal of Machine Learning Research*, 23(29):1–41, 2022.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. Fantastic generalization measures and where to find them. In *Proc. International Conference on Learning Representations*, New Orleans, LA, 2019.

Kao, C.-H., Chiu, W.-C., and Chen, P.-Y. Maml is a noisy contrastive learner. *arXiv preprint arXiv:2106.15367*, 2021.

Keskar, N., D.Mudigere, Noceda, J., Smelyanskiy, M., and Tang, P. On large-batch training for deep learning: Generalization gap and sharp minima. In *Proc. International Conference on Learning Representations*, Toulon, France, May 2017.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Proc. Advances in Neural Information Processing Systems*, pp. 1097–1105, Lake Tahoe, NV, 2012.

Langley, P. Crafting papers on machine learning. In *Proc. International Conference on Machine Learning*, pp. 1207–1216, Stanford, CA, 2000.

Laurent, B. and Massart, P. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302 – 1338, 2000.

Li, D., Yang, Y., Song, Y., and Hospedales, T. M. Learning to generalize: Meta-learning for domain generalization. In *Proc. AAAI Conference on Artificial Intelligence*, New Orleans, LA, February 2018a.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. Visualizing the loss landscape of neural nets. In *Proc. Advances in Neural Information Processing Systems*, Montreal, Canada, December 2018b.

Li, K. and Malik, J. Learning to optimize. In *Proc. International Conference on Learning Representations*, Toulon, France, May 2017.

Maicas, G., Bradley, A. P., Nascimento, J. C., Reid, I., and Carneiro, G. Training medical image analysis systems like radiologists. In *Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 546–554, Granada, Spain, 2018.

Munkhdalai, T. and Yu, H. Meta networks. In *Proc. International Conference on Machine Learning*, pp. 2554–2563, Sydney, Australia, August 2017.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. Exploring general-ization in deep learning. In *Proc. Advances in Neural Information Processing Systems*, pp. 5947–5956, Vancouver, Canada, 2019.

Nichol, A. and Schulman, J. Reptile: a scalable meta learning algorithm. *arXiv preprint arXiv:1803.02999*, 2018.

Novak, J. D. and Gowin, D. B. *Learning how to learn*. Cambridge University Press, 1984.

Obamuyide, A. and Vlachos, A. Model-agnostic meta-learning for relation classification with limited supervision. In *Proc. Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, July 2019.

Park, E. and Oliva, J. B. Meta-curvature. In *Proc. Advances in Neural Information Processing Systems*, Vancouver, Canada, December 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67, 2020.

Raghu, A., Raghu, M., and Bengio, S. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In *Proc. International Conference on Learning Representations*, Virtual, April 2020.

Rajeswaran, A., Finn, C., Kakade, S., and Levine, S. Meta-learning with implicit gradients. In *Proc. Advances in Neural Information Processing Systems*, pp. 113–124, Vancouver, Canada, December 2019a.

Rajeswaran, A., Finn, C., Kakade, S. M., and Levine, S. Meta-learning with implicit gradients. In *Proc. Advances in Neural Information Processing Systems*, pp. 113–124, Vancouver, Canada, 2019b.

Ravi, S. and Larochelle, H. Optimization as a model for few-shot learning. In *Proc. International Conference on Learning Representations*, Toulon, France, 2017.

Rothfuss, J., Fortuin, V., Josifoski, M., and Krause, A. PACOH: Bayes-optimal meta-learning with pac-guarantees. In *Proc. International Conference on Machine Learning*, pp. 9116–9126, virtual, 2021.

Schmidhuber, J. Evolutionary principles in self-referential learning. on learning now to learn: The meta-meta-meta...-hook. *http://www.idsia.ch/ juergen/diploma.html*, 1987.

Snell, J., Swersky, K., and Zemel, R. Prototypical networks for few-shot learning. In *Proc. Advances in Neural Information Processing Systems*, pp. 4077–4087, Long Beach, CA, December 2017.

Thrun, S. and Pratt, L. Learning to learn. *Springer Science & Business Media*, 2012.

Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Keysers, D., Uszkoreit, J., Lucic, M., and Dosovitskiy, A. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint:2105.01601*, 2021.

Vilalta, R. and Drissi, Y. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 2002.

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., and Wierstra, D. Matching networks for one shot learning. In *Proc. Advances in Neural information processing systems*, Barcelona, Spain, December 2016.

Vuorio, R., Sun, S., Hu, H., and Lim, J. J. Multimodal model-agnostic metalearning via task-aware modulation. In *Proc. Advances in Neural Information Processing Systems*, pp. 1–12, Vancouver, Canada, December 2019.

Wang, G., Luo, C., Sun, X., Xiong, Z., and Zeng, W. Tracking by instance detection: A meta-learning approach. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6288–6297, Virtual, 2020.

Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., Blundell, C., Kumaran, D., and Botvinick, M. Learning to reinforcement learn. *arXiv preprint:1611.05763*, 2016.

Wang, R., Xu, K., Liu, S., Chen, P., Weng, T., Gan, C., and Wang, M. On fast adversarial robustness adaptation in model-agnostic meta-learning. In *Proc. International Conference on Learning Representations*, Virtual, May 2021.

Wang, Y., Ramanan, D., and Hebert, M. Meta-learning to detect rare objects. In *Proc. International Conference on Computer Vision*, Seoul, Korea, 2019.

Xu, H., Li, Y., Liu, X., Liu, H., and Tang, J. Yet meta learning can adapt fast, it can also break easily. *arXiv preprint: 2009.01672*, 2020.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. mT5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint:2010.11934*, 2020.

Yin, M., Tucker, G., Zhou, M., Levine, S., and Finn, C. Meta-learning without memorization. In *Proc. International Conference on Learning Representations*, Virtual, April 2020.

Yoon, J., Kim, T., Dia, O., Kim, S., Bengio, Y., and Ahn, S. Bayesian model-agnostic meta-learning. In *Proc. Advances in Neural Information Processing Systems*, volume 31, Montreal, Canada, 2018.

Zintgraf, L., Shiarli, K., Kurin, V., Hofmann, K., and Whiteson, S. Fast context adaptation via meta-learning. In *Proc. International Conference on Machine Learning*, pp. 7693–7702, Long Beach, CA, June 2019a.

Zintgraf, L., Shiarli, K., Kurin, V., Hofmann, K., and Whiteson, S. Fast context adaptation via meta-learning. In *Proc. International Conference on Machine Learning*, pp. 7693–7702, Long Beach, CA, 2019b.

Supplementary Material for "Sharp-MAML: Sharpness-Aware Model-Agnostic Meta Learning"

A. Omitted Pseudo-code in The Main Manuscript

In this section, we present the omitted pseudo-code of MAML and two Sharp-MAML algorithms.

A.1. MAML algorithm

The pseudo-code of plain-vanilla MAML is summarized in Algorithm 2.

Algorithm 2 MAML for few-shot supervised learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: $\beta_{\text{low}}, \beta_{\text{up}}$: step sizes
for $t = 1, \cdots, T$ **do**
  Sample batch of tasks $\mathcal{T}_m \sim p(\mathcal{T})$
  for all $\mathcal{T}_m$ **do**
    Sample K examples from $\mathcal{D}_m = \{x_i, y_i\}$
    Evaluate $\widetilde{\nabla}\mathcal{L}(\theta^t; \mathcal{D}_m)$
    Compute fine-tuned parameter $\theta'_m(\theta^t)$ via (2)
    Sample datapoints from $\mathcal{D}'_m = \{x_i, y_i\}$ for meta-update
  end for
  Update the meta-model θ by $\theta^{t+1} = \theta^t - \beta_{\text{up}} \widetilde{\nabla}_{\theta^t} \sum_{m=1}^{M} \mathcal{L}(\theta'_m(\theta^t); \mathcal{D}'_m)$
end for

Algorithm 3 Sharp-MAML$_{\text{up}}$

Require: $p(\mathcal{T})$: distribution over tasks
Require: $\beta_{\text{low}}, \beta_{\text{up}}$: step sizes
Require: $\alpha_{\text{low}} > 0, \alpha_{\text{up}} > 0$: perturbation radii
for $t = 1, \cdots, T$ **do**
  Sample batch of tasks $\mathcal{T}_m \sim p(\mathcal{T})$
  for all $\mathcal{T}_m$ **do**
    Sample K examples from $\mathcal{D}_m$
    Evaluate $\widetilde{\nabla}\mathcal{L}(\theta^t; \mathcal{D}_m)$
    Compute fine-tuned parameter $\tilde{\theta}^1_m(\theta^t) = \theta^t - \beta_{\text{low}} \widetilde{\nabla}\mathcal{L}(\theta^t; \mathcal{D}_m)$
    Sample data from $\mathcal{D}'_m$ for meta-update
  end for
  Compute $\widetilde{\nabla}_{\theta^t} \sum_{m=1}^{M} \mathcal{L}(\tilde{\theta}^1_m(\theta^t); \mathcal{D}'_m)$
  Compute perturbation $\epsilon(\theta^t)$ via (20)
  Update θ^{t+1} via (21)
end for

A.2. Sharp-MAML$_{\text{up}}$ algorithm

In this case, $\epsilon_m(\theta^t) = 0, \forall m, t$, so we have that $\tilde{\theta}^1_m(\theta^t) = \theta^t - \beta_{\text{low}} \widetilde{\nabla}\mathcal{L}(\theta^t; \mathcal{D}_m)$ and $\tilde{\theta}^2_m(\theta^t) = \theta^t + \epsilon(\theta^t) - \beta_{\text{low}} \widetilde{\nabla}\mathcal{L}(\theta^t + \epsilon(\theta^t); \mathcal{D}_m)$. Defining $\nabla h = \widetilde{\nabla}_{\theta^t} \sum_{m=1}^{M} \mathcal{L}(\tilde{\theta}^1_m(\theta^t); \mathcal{D}'_m)$, the upper perturbation $\epsilon(\theta^t)$ can be computed by:

$$\epsilon(\theta^t) = \alpha_{\text{up}} \nabla h / ||\nabla h||_2. \tag{20}$$

Algorithm 4 Sharp-MAML$_{\text{low}}$

Require: $p(\mathcal{T})$: distribution over tasks
Require: $\beta_{\text{low}}, \beta_{\text{up}}$: step sizes
Require: $\alpha_{\text{low}} > 0, \alpha_{\text{up}} > 0$: perturbation radii
for $t = 1, \cdots, T$ **do**
  Sample batch of tasks $\mathcal{T}_m \sim p(\mathcal{T})$
  for all $\mathcal{T}_m$ **do**
    Sample K examples $\mathcal{D}_m$ from $\mathcal{T}_m$
    Evaluate $\widetilde{\nabla}\mathcal{L}(\theta^t; \mathcal{D}_m)$
    Compute perturbation $\epsilon_m(\theta^t)$ via (11)
    Compute fine-tuned parameter $\tilde{\theta}^1_m(\theta^t)$ via (22)
    Sample data $\mathcal{D}'_m$ for meta-update
  end for
  Update the meta-model θ^{t+1} via (23)
end for

Let $\epsilon^t = \epsilon(\theta^t)$, then the final meta update can be written as

$$\theta^{t+1} = \theta^t - \beta_{\text{up}} \widetilde{\nabla}_{\theta^t} \sum_{m=1}^{M} \mathcal{L}(\theta^t + \epsilon^t - \beta_{\text{low}} \widetilde{\nabla}\mathcal{L}(\theta^t + \epsilon^t; \mathcal{D}_m); \mathcal{D}'_m). \tag{21}$$

The pseudo-code is summarized in Algorithm 3.

A.3. Sharp-MAML$_{\text{low}}$ algorithm

In this case, $\epsilon(\theta^t) = 0, \forall t$, so we have that $\tilde{\theta}^1_m(\theta^t) = \tilde{\theta}^2_m(\theta^t) = \theta^t - \beta_{\text{low}} \widetilde{\nabla}\mathcal{L}(\theta^t + \epsilon_m(\theta^t); \mathcal{D}_m)$. Then the final meta update can be written as

$$\tilde{\theta}^1_m(\theta^t) = \theta^t - \beta_{\text{low}} \widetilde{\nabla}\mathcal{L}(\theta^t + \epsilon_m(\theta^t); \mathcal{D}_m) \tag{22}$$

$$\theta^{t+1} = \theta^t - \beta_{\text{up}} \widetilde{\nabla}_{\theta^t} \sum_{m=1}^{M} \mathcal{L}(\tilde{\theta}^1_m(\theta^t); \mathcal{D}'_m) \tag{23}$$

The pseudo-code is summarized in Algorithm 4.

B. Proof of Lemma 1

Proof. Since $\mathcal{L}(\theta; \mathcal{D}_m) \in \mathcal{C}^3$, the stationary point of $\mathcal{L}(\theta; \mathcal{D}_m)$ satisfies

$$\nabla\mathcal{L}(\theta; \mathcal{D}_m) = 0 \tag{24}$$

and the local minimizer of $\mathcal{L}(\theta; \mathcal{D}_m)$ satisfies

$$\nabla\mathcal{L}(\theta; \mathcal{D}_m) = 0 \ \text{ and } \ \nabla^2\mathcal{L}(\theta; \mathcal{D}_m) \succeq 0. \tag{25}$$

Next we compute the gradient of $\mathcal{L}(\theta'_m(\theta); \mathcal{D}_m)$ according to the chain rule, that is

$$\nabla\mathcal{L}(\theta'_m(\theta); \mathcal{D}_m) = (I - \beta_{\text{low}}\nabla^2\mathcal{L}(\theta; \mathcal{D}_m))\nabla\mathcal{L}(\theta - \beta_{\text{low}}\nabla\mathcal{L}(\theta; \mathcal{D}_m); \mathcal{D}_m) \tag{26}$$

and the Hessian of $\mathcal{L}(\theta'_m(\theta); \mathcal{D}_m)$, that is

$$\begin{aligned}
\nabla^2\mathcal{L}(\theta'_m(\theta); \mathcal{D}_m) &= \nabla\left((I - \beta_{\text{low}}\nabla^2\mathcal{L}(\theta; \mathcal{D}_m))\nabla\mathcal{L}(\theta - \beta_{\text{low}}\nabla\mathcal{L}(\theta; \mathcal{D}_m); \mathcal{D}_m)\right) \\
&= -\beta_{\text{low}}\nabla^3\mathcal{L}(\theta; \mathcal{D}_m)\nabla\mathcal{L}(\theta - \beta_{\text{low}}\nabla\mathcal{L}(\theta; \mathcal{D}_m); \mathcal{D}_m) \\
&\quad + (I - \beta_{\text{low}}\nabla^2\mathcal{L}(\theta; \mathcal{D}_m))^2\nabla^2\mathcal{L}(\theta - \beta_{\text{low}}\nabla\mathcal{L}(\theta; \mathcal{D}_m); \mathcal{D}_m).
\end{aligned} \tag{27}$$

Plugging (24) to (26), we get that $\nabla_\theta \mathcal{L}(\theta'_m(\theta); \mathcal{D}_m) = 0$ which implies θ is also a stationary point for $\mathcal{L}(\theta'_m(\theta); \mathcal{D}_m)$.

Moreover, plugging (25) to (27), we get that $\nabla_\theta \mathcal{L}(\theta'_m(\theta); \mathcal{D}_m) = 0$ and

$$\nabla^2_\theta \mathcal{L}(\theta'_m(\theta); \mathcal{D}_m) = (I - \beta_{\text{low}} \nabla^2 \mathcal{L}(\theta; \mathcal{D}_m))^2 \nabla^2 \mathcal{L}(\theta; \mathcal{D}_m) \succeq 0$$

which implies θ is also a local minimizer of $\mathcal{L}(\theta'_m(\theta); \mathcal{D}_m)$.

If θ is the stationary point for $\mathcal{L}(\theta; \mathcal{D}_m), \forall m \in \mathcal{M}$, we know that θ is also the stationary point for $\mathcal{L}(\theta'_m(\theta); \mathcal{D}_m), \forall m \in \mathcal{M}$. Thus, θ is also the stationary point for $\sum_{m=1}^M \mathcal{L}(\theta'_m(\theta); \mathcal{D}_m)$. Likewise, the statement is also true for local minimizer. □

C. Convergence Analysis

C.1. Convergence analysis of MAML (Finn et al., 2017)

We provide theoretical analysis for MAML (Finn et al., 2017). First, we state the exact form of MAML as follows.

$$\min_\theta \frac{1}{M} \sum_{m=1}^M \mathcal{L}(\theta'_m(\theta); \mathcal{D}'_m) \tag{28a}$$

$$\text{s.t. } \theta'_m(\theta) = \theta - \beta_{\text{low}} \nabla \mathcal{L}(\theta; \mathcal{D}_m), \ \forall m \in \mathcal{M}. \tag{28b}$$

The problem (28) can be reformulated as

$$\min_\theta \quad F(\theta) = \frac{1}{M} \sum_{m=1}^M \mathcal{L}(\theta'_m(\theta); \mathcal{D}'_m) \tag{29a}$$

$$\text{s.t. } \theta'_m(\theta) = \arg\min_{\theta_m} \left\{ \nabla \mathcal{L}(\theta; \mathcal{D}_m)^\top (\theta_m - \theta) + \frac{1}{2\beta_{\text{low}}} \|\theta_m - \theta\|^2 \right\}. \tag{29b}$$

Next, to show the connection between MAML formulation and ALSET (Chen et al., 2021a), we can concatenate θ_m as a new vector $\phi = [\theta_1^\top, \cdots, \theta_M^\top]^\top$ and define

$$F(\theta) = f(\phi'(\theta)), \quad f(\phi) = \frac{1}{M} \sum_{m=1}^M \mathcal{L}(\theta_m; \mathcal{D}'_m), \quad g(\theta, \phi) = \frac{1}{M} \sum_{m=1}^M \left\{ \nabla \mathcal{L}(\theta; \mathcal{D}_m)^\top (\theta_m - \theta) + \frac{\|\theta_m - \theta\|^2}{2\beta_{\text{low}}} \right\}$$

where $\phi'(\theta) = \arg\min_\phi g(\theta, \phi)$. Then the Jacobian and Hessian of f and g can be computed by

$$\nabla_\phi f(\phi) = \begin{pmatrix} \nabla \mathcal{L}(\theta_1; \mathcal{D}'_1) \\ \vdots \\ \nabla \mathcal{L}(\theta_M; \mathcal{D}'_M) \end{pmatrix}, \quad \nabla_{\phi\theta} g(\theta, \phi) = \begin{pmatrix} \nabla^2 \mathcal{L}(\theta_1; \mathcal{D}_1) - \beta_{\text{low}}^{-1} I \\ \vdots \\ \nabla^2 \mathcal{L}(\theta_M; \mathcal{D}_M) - \beta_{\text{low}}^{-1} I \end{pmatrix}, \quad \nabla_{\phi\phi} g(\theta, \phi) = \beta_{\text{low}}^{-1} I$$

where I denotes the identity matrix. According to the expression of $\nabla F(\theta)$ in ALSET (Chen et al., 2021a), we can verify that MAML's gradient has the following form

$$\begin{aligned} \nabla F(\theta) &= -\nabla_{\theta\phi} g(\theta, \phi) \nabla^{-1}_{\phi\phi} g(\theta, \phi) \nabla_\phi f(\phi) \\ &= \frac{1}{M} \sum_{m=1}^M (I - \beta_{\text{low}} \nabla^2 \mathcal{L}(\theta; \mathcal{D}_m)) \nabla \mathcal{L}(\theta_m; \mathcal{D}'_m). \end{aligned} \tag{30}$$

Moreover, since $g(\theta, \phi)$ is a quadratic function with respect to ϕ, the strongly convexity and Lipschitz continuity assumptions hold.[2] Assumptions about upper-level function also holds under Assumption 1.

Then, for notational simplicity, we consider the single-sample case with $K = 1$ and define three independent samples for stochastic gradient and Hessian computation as $\xi_m := (x, y) \sim \mathcal{D}_m, \psi_m := (x, y) \sim \mathcal{D}_m, \xi'_m := (x, y) \sim \mathcal{D}'_m$, so the

[2] $\nabla^2 g$ is Lipschitz continuous in Assumption 1 in (Chen et al., 2021a) can be reduced to $\nabla_{\phi\phi} g$ and $\nabla_{\phi\theta} g$ is Lipschitz continuous, which can be satisfied under Assumption 1.

corresponding K-batch gradient and Hessian estimators used in MAML algorithms can be written as

$$\nabla \mathcal{L}(\theta; \mathcal{D}_m, \xi_m) = \frac{1}{K} \sum_{\xi_m \sim \mathcal{D}_m} \nabla l(\theta, x, y),$$
$$\nabla^2 \mathcal{L}(\theta; \mathcal{D}_m, \psi_m) = \frac{1}{K} \sum_{\psi_m \sim \mathcal{D}_m} \nabla^2 l(\theta, x, y),$$
$$\nabla \mathcal{L}(\theta; \mathcal{D}'_m, \xi'_m) = \frac{1}{K} \sum_{\xi'_m \sim \mathcal{D}'_m} \nabla l(\theta, x, y).$$

Based on these notations, we can write the stochastic update of MAML algorithm (Finn et al., 2017) as

$$\theta^{t+1} = \theta^t - \beta_{\text{up}} \widetilde{\nabla}_{\theta^t} \mathcal{L}(\theta^t; \mathcal{D}_m) = \theta^t - \frac{\beta_{\text{up}}}{M} \sum_{m=1}^{M} (I - \beta_{\text{low}} \nabla^2 \mathcal{L}(\theta^t; \mathcal{D}_m, \psi_m)) \nabla \mathcal{L}(\theta_m^{t+1}; \mathcal{D}'_m, \xi'_m)$$
$$\theta_m^{t+1} = \theta^t - \beta_{\text{low}} \nabla \mathcal{L}(\theta^t; \mathcal{D}_m, \xi_m).$$

Then MAML algorithm can be seen as a special case of ALSET, so we have the following Lemma.

Lemma 2. *Under Assumption 1–2, and choosing stepsizes $\beta_{\text{low}}, \beta_{\text{up}} = \mathcal{O}(\frac{1}{\sqrt{T}})$ with some proper constants, we can get that the iterates $\{\theta^t\}$ generated by MAML (Finn et al., 2017) satisfy*

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \left[\|\nabla F(\theta^t)\|^2 \right] = \mathcal{O}\left(\frac{1}{\sqrt{T}} \right).$$

C.2. Convergence analysis of a generic biased MAML

Since Sharp-MAML can be treated as biased update version of MAML (Finn et al., 2017), we first analyze the general biased MAML algorithm. Suppose that biased MAML update with

$$\theta^{t+1} = \theta^t - \frac{\beta_{\text{up}}}{M} \sum_{m=1}^{M} (I - \beta_{\text{low}} \hat{\nabla}^2 \mathcal{L}(\theta^t; \mathcal{D}_m, \psi_m)) \nabla \mathcal{L}(\hat{\theta}_m^{t+1}; \mathcal{D}'_m, \xi'_m)$$
$$\hat{\theta}_m^{t+1} = \theta^t - \beta_{\text{low}} \hat{\nabla} \mathcal{L}(\theta^t; \mathcal{D}_m, \xi_m)$$

where $\hat{\nabla}^2 \mathcal{L}(\theta^t; \mathcal{D}_m, \psi_m), \hat{\nabla} \mathcal{L}(\theta^t; \mathcal{D}_m, \xi_m)$ are biased estimator of $\nabla^2 \mathcal{L}(\theta^t; \mathcal{D}_m), \nabla \mathcal{L}(\theta^t; \mathcal{D}_m)$, respectively. With the notation that

$$\hat{\nabla} \mathcal{L}(\theta; \mathcal{D}_m) \triangleq \mathbb{E}_{\xi_m} \left[\hat{\nabla} \mathcal{L}(\theta; \mathcal{D}_m, \xi_m) \right], \hat{\nabla}^2 \mathcal{L}(\theta; \mathcal{D}_m) \triangleq \mathbb{E}_{\psi_m} \left[\hat{\nabla}^2 \mathcal{L}(\theta; \mathcal{D}_m, \psi_m) \right],$$

we make the following assumptions.

Assumption 3 (Stochastic derivatives)**.** *Assume that $\hat{\nabla} \mathcal{L}(\theta; \mathcal{D}_m, \xi_m), \hat{\nabla}^2 \mathcal{L}(\theta; \mathcal{D}_m, \psi_m)$ are unbiased estimator of $\hat{\nabla} \mathcal{L}(\theta; \mathcal{D}_m), \hat{\nabla}^2 \mathcal{L}(\theta; \mathcal{D}_m)$ respectively and their variances are bounded by σ_b^2.*

Assumption 4. *Assume that $\|\hat{\nabla}^2 \mathcal{L}(\theta^t; \mathcal{D}_m) - \nabla^2 \mathcal{L}(\theta^t; \mathcal{D}_m)\| \leq \gamma_h$, $\|\hat{\nabla} \mathcal{L}(\theta^t; \mathcal{D}_m) - \nabla \mathcal{L}(\theta^t; \mathcal{D}_m)\| \leq \gamma_g, \forall m \in \mathcal{M}$.*

Throughout the proof, we use

$$\mathcal{F}_t = \sigma \left\{ \hat{\theta}_1^0, \cdots, \hat{\theta}_M^0, \theta^0, \ldots, \theta^t, \hat{\theta}_1^{t+1}, \ldots, \hat{\theta}_M^{t+1} \right\}, \quad \mathcal{F}'_t = \sigma \left\{ \hat{\theta}_1^0, \cdots, \hat{\theta}_M^0, \theta^0, \ldots, \theta^t \right\}$$

where $\sigma\{\cdot\}$ denotes the σ-algebra generated by random variables. We also denote

$$h^t \triangleq \frac{1}{M} \sum_{m=1}^{M} (I - \beta_{\text{low}} \hat{\nabla}^2 \mathcal{L}(\theta^t; \mathcal{D}_m, \psi_m)) \nabla \mathcal{L}(\hat{\theta}_m^{t+1}; \mathcal{D}'_m, \xi'_m)$$
$$\bar{h}^t \triangleq \mathbb{E} \left[\frac{1}{M} \sum_{m=1}^{M} (I - \beta_{\text{low}} \hat{\nabla}^2 \mathcal{L}(\theta^t; \mathcal{D}_m, \psi_m)) \nabla \mathcal{L}(\hat{\theta}_m^{t+1}; \mathcal{D}'_m, \xi'_m) \Big| \mathcal{F}_t \right]$$
$$= \frac{1}{M} \sum_{m=1}^{M} (I - \beta_{\text{low}} \hat{\nabla}^2 \mathcal{L}(\theta^t; \mathcal{D}_m)) \nabla \mathcal{L}(\hat{\theta}_m^{t+1}; \mathcal{D}'_m).$$

Lemma 3. *Under Assumption 1–4, we have that*

$$\mathbb{E}\left[\|\nabla F(\theta^t)-\bar{h}^t\|^2\right] \le 4\ell_1^2\beta_{\rm low}^2(\gamma_g^2+\sigma_b^2)+4\beta_{\rm low}^2\left(\ell_0^2\gamma_h^2+4(\gamma_h^2+\ell_1^2)\ell_1^2\beta_{\rm low}^2(\gamma_g^2+\sigma_b^2)\right). \tag{31}$$

Proof. Since $\mathbb{E}\left[\hat{\theta}_m^{t+1}|\mathcal{F}_t'\right]=\theta^t-\beta_{\rm low}\hat{\nabla}\mathcal{L}(\theta^t;\mathcal{D}_m)$, then from Assumption 4, we have $\left\|\theta_m'(\theta^t)-\mathbb{E}\left[\hat{\theta}_m^{t+1}|\mathcal{F}_t'\right]\right\|\le\beta_{\rm low}\gamma_g$. Taking expectation with respect to $\mathcal{F}_t'$, we get

$$\mathbb{E}\left[\|\theta_m'(\theta^t)-\hat{\theta}_m^{t+1}\|^2\right]\le 2\mathbb{E}\left[\|\theta_m'(\theta^t)-\mathbb{E}\left[\hat{\theta}_m^{t+1}|\mathcal{F}_t'\right]\|^2\right]+2\mathbb{E}\left[\|\mathbb{E}\left[\hat{\theta}_m^{t+1}|\mathcal{F}_t'\right]-\hat{\theta}_m^{t+1}\|^2\right]\le 2\beta_{\rm low}^2\gamma_g^2+2\beta_{\rm low}^2\sigma_b^2.$$

Then using Lipschitz continuity of $\nabla\mathcal{L}(\theta;\mathcal{D}_m')$, we obtain

$$\mathbb{E}\left\|\nabla\mathcal{L}(\theta_m'(\theta^t);\mathcal{D}_m')-\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}_m')\right\|^2\le 2\ell_1^2\beta_{\rm low}^2(\gamma_g^2+\sigma_b^2). \tag{32}$$

On the other hand, by observing that

$$\left\|\hat{\nabla}^2\mathcal{L}(\theta^t;\mathcal{D}_m)\right\|^2\le 2\left\|\nabla^2\mathcal{L}(\theta^t;\mathcal{D}_m)\right\|^2+2\left\|\nabla^2\mathcal{L}(\theta^t;\mathcal{D}_m)-\hat{\nabla}^2\mathcal{L}(\theta^t;\mathcal{D}_m)\right\|^2\le 2(\gamma_h^2+\ell_1^2),$$

we get

$$\begin{aligned}
&\mathbb{E}\left\|\nabla^2\mathcal{L}(\theta^t;\mathcal{D}_m)\nabla\mathcal{L}(\theta_m'(\theta^t);\mathcal{D}_m')-\hat{\nabla}^2\mathcal{L}(\theta^t;\mathcal{D}_m)\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}_m')\right\|^2\\
&\le 2\mathbb{E}\left\|\nabla^2\mathcal{L}(\theta^t;\mathcal{D}_m)-\hat{\nabla}^2\mathcal{L}(\theta^t;\mathcal{D}_m)\right\|^2\left\|\nabla\mathcal{L}(\theta_m'(\theta^t);\mathcal{D}_m')\right\|^2+2\mathbb{E}\left\|\hat{\nabla}^2\mathcal{L}(\theta^t;\mathcal{D}_m)\right\|^2\left\|\nabla\mathcal{L}(\theta_m'(\theta^t);\mathcal{D}_m')-\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}_m')\right\|^2\\
&\le 2\ell_0^2\gamma_h^2+8(\gamma_h^2+\ell_1^2)\ell_1^2\beta_{\rm low}^2(\gamma_g^2+\sigma_b^2).
\end{aligned} \tag{33}$$

Thus, using (32) and (33), we get

$$\begin{aligned}
&\mathbb{E}\left[\|\nabla F(\theta^t)-\bar{h}^t\|^2\right]\\
&=\mathbb{E}\left\|\frac{1}{M}\sum_{m=1}^M\left[(I-\beta_{\rm low}\nabla^2\mathcal{L}(\theta^t;\mathcal{D}_m))\nabla\mathcal{L}(\theta_m'(\theta^t);\mathcal{D}_m')-(I-\beta_{\rm low}\hat{\nabla}^2\mathcal{L}(\theta^t;\mathcal{D}_m))\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}_m')\right]\right\|^2\\
&\le\frac{2}{M}\sum_{m=1}^M\mathbb{E}\left\|\nabla\mathcal{L}(\theta_m'(\theta^t);\mathcal{D}_m')-\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}_m')\right\|^2\\
&\quad+\frac{2\beta_{\rm low}^2}{M}\sum_{m=1}^M\mathbb{E}\left\|\nabla^2\mathcal{L}(\theta^t;\mathcal{D}_m)\nabla\mathcal{L}(\theta_m'(\theta^t);\mathcal{D}_m')-\hat{\nabla}^2\mathcal{L}(\theta^t;\mathcal{D}_m)\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}_m')\right\|^2\\
&\le 4\ell_1^2\beta_{\rm low}^2(\gamma_g^2+\sigma_b^2)+4\beta_{\rm low}^2\left(\ell_0^2\gamma_h^2+4(\gamma_h^2+\ell_1^2)\ell_1^2\beta_{\rm low}^2(\gamma_g^2+\sigma_b^2)\right)
\end{aligned}$$

from which the proof is complete. □

Lemma 4. *Under Assumption 1–4, and choosing stepsizes $\beta_{\rm low},\beta_{\rm up}=\mathcal{O}(\frac{1}{\sqrt{T}})$, and $\gamma_g,\gamma_h=\mathcal{O}(1)$ with some proper constants, we can get that the iterates $\{\theta^t\}$ generated by biased MAML satisfy*

$$\frac{1}{T}\sum_{t=1}^T\mathbb{E}\left[\|\nabla F(\theta^t)\|^2\right]=\mathcal{O}\left(\frac{1}{\sqrt{T}}\right).$$

Proof. First we bound the variance of stochastic biased meta gradient estimator h^t as

$$\begin{aligned}
\mathbb{E}\left[\|h^t-\bar{h}^t\|^2\Big|\mathcal{F}_t\right]\le\frac{2}{M}\sum_{m=1}^M\mathbb{E}\Big[&\|(I-\beta_{\rm low}\hat{\nabla}^2\mathcal{L}(\theta^t;\mathcal{D}_m,\psi_m))\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}_m',\xi_m')\\
&-(I-\beta_{\rm low}\hat{\nabla}^2\mathcal{L}(\theta^t;\mathcal{D}_m))\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}_m')\|^2\Big|\mathcal{F}_t\Big]
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{4}{M}\sum_{m=1}^{M}\mathbb{E}\left[\|\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}'_m,\xi'_m)-\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}'_m)\|^2\Big|\mathcal{F}_t\right]\\
&\quad+\frac{4\beta_{\text{low}}^2}{M}\sum_{m=1}^{M}\mathbb{E}\left[\|\hat{\nabla}^2\mathcal{L}(\theta^t;\mathcal{D}_m,\psi_m)\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}'_m,\xi'_m)-\hat{\nabla}^2\mathcal{L}(\theta^t;\mathcal{D}_m)\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}'_m)\|^2\Big|\mathcal{F}_t\right]\\
&\leq 4\ell_1^2\sigma^2+\frac{8\beta_{\text{low}}^2}{M}\sum_{m=1}^{M}\mathbb{E}\left[\|\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}'_m,\xi'_m)\|^2\|\hat{\nabla}^2\mathcal{L}(\theta^t;\mathcal{D}_m,\psi_m)-\hat{\nabla}^2\mathcal{L}(\theta^t;\mathcal{D}_m)\|^2\Big|\mathcal{F}_t\right]\\
&\quad+\frac{8\beta_{\text{low}}^2}{M}\sum_{m=1}^{M}\mathbb{E}\left[\|\hat{\nabla}^2\mathcal{L}(\theta^t;\mathcal{D}_m)\|^2\|\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}'_m,\xi'_m)-\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}'_m)\|^2\Big|\mathcal{F}_t\right]\\
&\leq 4\ell_1^2\sigma^2+8\beta_{\text{low}}^2\sigma_b^2\mathbb{E}\left[\|\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}'_m,\xi'_m)\|^2\Big|\mathcal{F}_t\right]+16(\gamma_h^2+\ell_1^2)\beta_{\text{low}}^2\sigma^2\\
&\leq 4\ell_1^2\sigma^2+16(\gamma_h^2+\ell_1^2)\beta_{\text{low}}^2\sigma^2+16\beta_{\text{low}}^2\sigma_b^2(\sigma^2+\ell_0^2)\triangleq\tilde{\sigma}^2 \qquad (34)
\end{aligned}
$$

where (34) comes from

$$
\mathbb{E}\left[\|\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}'_m,\xi'_m)\|^2\Big|\mathcal{F}_t\right]\leq 2\|\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}'_m)\|^2+2\mathbb{E}\left[\|\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}'_m,\xi'_m)-\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}'_m)\|^2\Big|\mathcal{F}_t\right]\leq 2(\ell_0^2+\sigma^2).
$$

Then according to Lemma 4 in (Chen et al., 2021a), F is smooth with constant $L_F=\mathcal{O}(1)$. Using the smoothness property with Lemma 2 in (Chen et al., 2021a), it follows that

$$
\begin{aligned}
\mathbb{E}\left[F(\theta^{t+1})|\mathcal{F}_t\right]&\leq F(\theta^t)+\mathbb{E}\left[\left\langle\nabla F(\theta^t),\theta^{t+1}-\theta^t\right\rangle|\mathcal{F}_t\right]+\frac{L_F}{2}\mathbb{E}\left[\left\|\theta^{t+1}-\theta^t\right\|^2|\mathcal{F}_t\right]\\
&\leq F(\theta^t)-\beta_{\text{up}}\left\langle\nabla F(\theta^t),\bar{h}^t\right\rangle+\frac{L_F\beta_{\text{up}}^2}{2}\mathbb{E}\left[\|h^t\|^2|\mathcal{F}_t\right]\\
&\overset{(a)}{\leq} F(\theta^t)-\beta_{\text{up}}\left\langle\nabla F(\theta^t),\bar{h}^t\right\rangle+\frac{L_F\beta_{\text{up}}^2}{2}\left\|\bar{h}^t\right\|^2+\frac{L_F\beta_{\text{up}}^2\tilde{\sigma}^2}{2}\\
&\overset{(b)}{=} F(\theta^t)-\frac{\beta_{\text{up}}}{2}\left\|\nabla F(\theta^t)\right\|^2-\left(\frac{\beta_{\text{up}}}{2}-\frac{L_F\beta_{\text{up}}^2}{2}\right)\left\|\bar{h}^t\right\|^2+\frac{\beta_{\text{up}}}{2}\left\|\nabla F(\theta^t)-\bar{h}^t\right\|^2+\frac{L_F\beta_{\text{up}}^2\tilde{\sigma}^2}{2}
\end{aligned}
$$

where (a) uses $\mathbb{E}\left[\|A\|^2|B\right]=(\mathbb{E}\left[A|B\right])^2+\mathbb{E}\left[\|A-\mathbb{E}\left[A|B\right]\|^2|B\right]$ and (34), and (b) uses $2a^\top b=\|a\|^2+\|b\|^2-\|a-b\|^2$. Then using the result in Lemma 3 and choosing $\beta_{\text{up}}\leq\frac{1}{L_F}$, telescoping and rearranging it, we obtain that

$$
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla F(\theta^t)\right\|^2\right]\leq\frac{2F(\theta^1)}{\beta_{\text{up}}T}+L_F\beta_{\text{up}}\tilde{\sigma}^2+4\beta_{\text{low}}^2\left(\ell_1^2(\gamma_g^2+\sigma_b^2)+\ell_0^2\gamma_h^2+4(\gamma_h^2+\ell_1^2)\ell_1^2\beta_{\text{low}}^2(\gamma_g^2+\sigma_b^2)\right). \qquad (35)
$$

Choosing $\beta_{\text{up}},\beta_{\text{low}}=\mathcal{O}(\frac{1}{\sqrt{T}})$ and $\gamma_g,\gamma_h=\mathcal{O}(1)$, we can get the $\mathcal{O}(\frac{1}{\sqrt{T}})$ convergence results of biased MAML. □

C.3. Convergence analysis of Sharp-MAML

Thanks to the discussion in Section C.2, the convergence of Sharp-MAML is straightforward. Here we only prove for Sharp-MAML$_{\text{both}}$ since same results for the other two variants can be derived by setting $\alpha_{\text{up}}=0$ or $\alpha_{\text{low}}=0$ accordingly.

Lemma 5. *Under Assumption 1–2, and choosing stepsizes $\beta_{\text{low}},\beta_{\text{up}}=\mathcal{O}(\frac{1}{\sqrt{T}})$ and perturbation radii $\alpha_{\text{up}}=\mathcal{O}(\frac{1}{\sqrt{T}})$, $\alpha_{\text{low}}=\mathcal{O}(1)$, with some proper constants, we can get that the iterates $\{\theta^t\}$ generated by Sharp-MAML$_{\text{both}}$ satisfy*

$$
\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla F(\theta^t)\|^2\right]=\mathcal{O}\left(\frac{1}{\sqrt{T}}\right).
$$

Proof. Recalling the update of Sharp-MAML$_{\text{both}}$, we rewrite it as follows.

$$
\theta^{t+1}=\theta^t-\frac{\beta_{\text{up}}}{M}\sum_{m=1}^{M}(I-\beta_{\text{low}}\nabla^2\mathcal{L}(\theta^t+\epsilon(\theta^t)+\epsilon_m(\theta^t);\mathcal{D}_m,\psi_m))\nabla\mathcal{L}(\hat{\theta}_m^{t+1};\mathcal{D}'_m,\xi'_m);
$$

$$\begin{aligned}\hat{\theta}_m^{t+1} &= \theta^t + \epsilon(\theta^t) - \beta_{\text{low}} \nabla \mathcal{L}(\theta^t + \epsilon(\theta^t) + \epsilon_m(\theta^t); \mathcal{D}_m, \xi_m) \\ &= \theta^t - \beta_{\text{low}} \left(\nabla \mathcal{L}(\theta^t + \epsilon(\theta^t) + \epsilon_m(\theta^t); \mathcal{D}_m, \xi_m) - \frac{\epsilon(\theta^t)}{\beta_{\text{low}}} \right).\end{aligned}$$

Since $\nabla \mathcal{L}(\theta; \mathcal{D}_m), \nabla^2 \mathcal{L}(\theta; \mathcal{D}_m)$ are Lipschitz continuous with ℓ_1, ℓ_2 according to Assumption 1, then we have that

$$\begin{aligned}&\|\nabla \mathcal{L}(\theta^t + \epsilon(\theta^t) + \epsilon_m(\theta^t); \mathcal{D}_m) - \frac{\epsilon(\theta^t)}{\beta_{\text{low}}} - \nabla \mathcal{L}(\theta^t; \mathcal{D}_m)\| \leq \ell_1(\alpha_{\text{up}} + \alpha_{\text{low}}) + \frac{\alpha_{\text{up}}}{\beta_{\text{low}}} \\ &\|\nabla^2 \mathcal{L}(\theta^t + \epsilon(\theta^t) + \epsilon_m(\theta^t); \mathcal{D}_m) - \nabla^2 \mathcal{L}(\theta^t; \mathcal{D}_m)\| \leq \ell_2(\alpha_{\text{up}} + \alpha_{\text{low}}),\end{aligned}$$

which satisfies the condition in Lemma 4 if $\alpha_{\text{up}} = \mathcal{O}(\frac{1}{\sqrt{T}}), \alpha_{\text{low}} = \mathcal{O}(1)$. Thus, we arrive at the conclusion. □

D. Generalization Analysis

We build on the recently developed PAC-Bayes bound for meta learning (Farid & Majumdar, 2021), as restated below.

Lemma 6. *Assume the loss function $\mathcal{L}(\cdot)$ is bounded: $0 \leq \mathcal{L}(h, \mathcal{D}) \leq 1$ for any h in the hypothesis space, and any $\mathcal{D}$ in the sample space. For hypotheses $h_{A(\theta,\mathcal{D})}$ learned with γ_{A} uniformly stable algorithm A, data-independent prior $P_{\theta,0}$ over initializations θ, and $\delta \in (0,1)$, with probability at least $1-\delta$ over a sampling of the meta-training dataset $\mathcal{D} \sim \mathcal{P}$, and $|\mathcal{D}| = n$, which include meta-training data from all tasks, the following holds simultaneously for all distributions P_θ over θ:*

$$\mathop{\mathbb{E}}_{\mathcal{D}\sim\mathcal{P}} \mathop{\mathbb{E}}_{\theta\sim P_\theta} \mathcal{L}(h_{A(\theta,\mathcal{D})}, \mathcal{D}) \leq \frac{1}{n} \sum_{i=1}^{n} \mathop{\mathbb{E}}_{\theta\sim P_\theta} \mathcal{L}(h_{A(\theta,\mathcal{D})}, (x_i, y_i)) + \sqrt{\frac{D_{\text{KL}}(P_\theta \| P_{\theta,0}) + \ln \frac{2\sqrt{n}}{\delta}}{2n}} + \gamma_{\text{A}}.$$

We can obtain a generalization bound below.

Theorem 3. *Assume loss function $\mathcal{L}(\cdot)$ is bounded: $0 \leq \mathcal{L}(\theta_m; \mathcal{D}) \leq 1$ for any θ_m, and any $\mathcal{D}$, and $\mathcal{L}(\theta_m(\hat{\theta}); \mathcal{P}) \leq \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\alpha^2\mathbf{I})}[\mathcal{L}(\theta_m(\hat{\theta}+\epsilon); \mathcal{P})]$ at the stationary point of the Sharp-MAML$_{\text{up}}$ denoted by $\hat{\theta}$. For parameter $\theta_m(\hat{\theta}; \mathcal{D})$ learned with γ_{A} uniformly stable algorithm A from $\hat{\theta} \in \mathbb{R}^k$, with probability $1-\delta$ over the choice of the training set $\mathcal{D} \sim \mathcal{P}$, with $|\mathcal{D}| = n$, it holds that*

$$\mathcal{L}(\theta_m(\hat{\theta}); \mathcal{P}) \leq \max_{\|\epsilon\|_2 \leq \alpha} \mathcal{L}(\theta_m(\hat{\theta}+\epsilon); \mathcal{D}) + \gamma_{\text{A}} + \sqrt{\frac{k \ln\left(1 + \frac{\|\hat{\theta}\|_2^2}{\alpha^2}\left(1 + \sqrt{\frac{\ln n}{k}}\right)^2\right) + 2\ln\frac{1}{\delta} + 5\ln n + \mathcal{O}(1)}{4n}}.$$

Proof. Since Lemma 6 holds for any prior $P_{\theta,0}$ and posterior P_θ, let $P_{\theta,0} = P = \mathcal{N}(\mathbf{0}, \sigma_P^2 \mathbf{I})$, $P_\theta = Q = \mathcal{N}(\theta, \alpha^2 \mathbf{I})$, then

$$\begin{aligned}D_{KL}(Q\|P) &= \frac{1}{2}\left\{ \text{tr}\left(\boldsymbol{\Sigma}_P^{-1}\boldsymbol{\Sigma}_Q\right) + \left(\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q\right)^{\text{T}} \boldsymbol{\Sigma}_P^{-1}(\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q) - k + \ln\frac{|\boldsymbol{\Sigma}_P|}{|\boldsymbol{\Sigma}_Q|} \right\} \\ &= \frac{1}{2}\left[\frac{k\alpha^2 + \|\theta\|_2^2}{\sigma_P^2} - k + k\ln\left(\frac{\sigma_P^2}{\alpha^2}\right) \right].\end{aligned}$$

Let $T = \{c\exp((1-j)/k) \mid j \in \mathbb{N}\}$ be the set of values for σ_P^2. If for any $j \in \mathbb{N}$, the PAC-Bayesian bound in Lemma 6 holds for $\sigma_P^2 = c\exp((1-j)/k)$ with probability $1-\delta_j$ with $\delta_j = \frac{6\delta}{\pi^2 j^2}$, then by the union bound, all bounds w.r.t. all $\sigma_P^2 \in T$ hold simultaneously with probability at least $1 - \sum_{j=1}^{\infty} \frac{6\delta}{\pi^2 j^2} = 1 - \delta$.

First consider $\|\theta\|^2 \leq \alpha^2(\exp(4n/k) - 1)$, then $k\alpha^2 + \|\theta\|_2^2 \leq k\alpha^2(\exp(4n/k) + 1)$. Now set $j = \left\lfloor 1 - k\ln\left(\left(\alpha^2 + \|\theta\|_2^2/k\right)/c\right)\right\rfloor$. By setting $c = \alpha^2(1 + \exp(4n/k))$, then $\ln\left(\left(\alpha^2 + \|\theta\|_2^2/k\right)/c\right) < 0$, thus we can ensure that $j \in \mathbb{N}$. Furthermore, for $\sigma_P^2 = c\exp((1-j)/k)$, we have:

$$\alpha^2 + \|\theta\|_2^2/k \leq \sigma_P^2 \leq \exp(1/k)(\alpha^2 + \|\theta\|_2^2/k)$$

where the first inequality is derived from $1 - j = \lceil k\ln((\alpha^2 + \|\theta\|_2^2/k)/c)\rceil \geq k\ln((\alpha^2 + \|\theta\|_2^2/k)/c)$, the second inequality is derived from $1 - j = \lceil k\ln((\alpha^2 + \|\theta\|_2^2/k)/c)\rceil \leq k\ln((\alpha^2 + \|\theta\|_2^2/k)/c) + 1$.

The KL-divergence term can be further bounded as

$$\begin{aligned} D_{KL}(Q\|P) &= \frac{1}{2}\left[\frac{k\alpha^2+\|\theta\|_2^2}{\sigma_P^2} - k + k\ln\left(\frac{\sigma_P^2}{\alpha^2}\right)\right] \\ &\leq \frac{1}{2}\left[\frac{k\alpha^2+\|\theta\|_2^2}{\alpha^2+\|\theta\|_2^2/k} - k + k\ln\left(\frac{\exp(1/k)\left(\alpha^2+\|\theta\|_2^2/k\right)}{\alpha^2}\right)\right] \\ &= \frac{1}{2}\left[k\ln\left(\frac{\exp(1/k)\left(\alpha^2+\|\theta\|_2^2/k\right)}{\alpha^2}\right)\right] \\ &= \frac{1}{2}\left[1 + k\ln\left(1+\frac{\|\theta\|_2^2}{k\alpha^2}\right)\right]. \end{aligned}$$

Given the bound that corresponds to j holds with probability $1-\delta_j$ for $\delta_j = \frac{6\delta}{\pi^2 j^2}$, the $\ln$ term above can be written as:

$$\begin{aligned} \ln\frac{1}{\delta_j} &= \ln\frac{1}{\delta} + \ln\frac{\pi^2 j^2}{6} \\ &\leq \ln\frac{1}{\delta} + \ln\frac{\pi^2 k^2 \ln^2\left(c/\left(\alpha^2+\|\theta\|_2^2/k\right)\right)}{6} \\ &\leq \ln\frac{1}{\delta} + \ln\frac{\pi^2 k^2 \ln^2\left(c/\alpha^2\right)}{6} \\ &= \ln\frac{1}{\delta} + \ln\frac{\pi^2 k^2 \ln^2(1+\exp(4n/k))}{6} \\ &\leq \ln\frac{1}{\delta} + \ln\frac{\pi^2 k^2 (4n/k)^2}{6} \leq \ln\frac{1}{\delta} + 2\ln(6n). \end{aligned}$$

Therefore for $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, the generalization bound is

$$\begin{aligned} \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})}\left[\mathcal{L}(\theta_m(\theta+\epsilon);\mathcal{P})\right] \leq& \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})}\left[\mathcal{L}(\theta_m(\theta+\epsilon);\mathcal{D})\right] + \sqrt{\frac{\frac{1}{2}k\ln\left(1+\frac{\|\theta\|_2^2}{k\sigma^2}\right)+\frac{1}{2}+\ln\frac{2\sqrt{n}}{\delta}+2\ln(6n)}{2n}} + \gamma_{\mathrm{A}} \\ =& \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})}\left[\mathcal{L}(\theta_m(\theta+\epsilon);\mathcal{D})\right] + \sqrt{\frac{\frac{1}{2}k\ln\left(1+\frac{\|\theta\|_2^2}{k\sigma^2}\right)+\frac{1}{2}+\ln 72+\ln\frac{1}{\delta}+\frac{5}{2}\ln n}{2n}} + \gamma_{\mathrm{A}}. \end{aligned}$$

By Lemma 1 in (Laurent & Massart, 2000), we have that for $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ and any positive t :

$$P\left(\|\epsilon\|_2^2 - k\sigma^2 \geq 2\sigma^2\sqrt{kt} + 2t\sigma^2\right) \leq \exp(-t).$$

Therefore, with probability $1-1/\sqrt{n}$ we have that:

$$\|\epsilon\|_2^2 \leq \sigma^2(2\ln(\sqrt{n}) + k + 2\sqrt{k\ln(\sqrt{n})}) \leq \sigma^2 k\left(1+\sqrt{\frac{\ln n}{k}}\right)^2 = \alpha^2.$$

At the stationary point $\hat{\theta}$ obtained by Sharp-MAML, we have

$$\begin{aligned}\mathcal{L}(\theta_m(\hat{\theta});\mathcal{P}) &\leq \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\alpha^2\mathbf{I})}\Big[\mathcal{L}(\theta_m(\hat{\theta}+\epsilon);\mathcal{P})\Big] \leq (1-1/\sqrt{n})\max_{\|\epsilon\|_2\leq\alpha}\mathcal{L}(\theta_m(\hat{\theta}+\epsilon);\mathcal{D}) + 1/\sqrt{n} \\ &+\sqrt{\frac{\frac{1}{2}k\ln\left(1+\frac{\|\hat{\theta}\|_2^2}{k\sigma^2}\right)+\frac{1}{2}+\ln 72+\ln\frac{1}{\delta}+\frac{5}{2}\ln n}{2n}}+\gamma_{\mathrm{A}} \\ &\leq \max_{\|\epsilon\|_2\leq\alpha}\mathcal{L}(\theta_m(\hat{\theta}+\epsilon);\mathcal{D}) + \sqrt{\frac{k\ln\left(1+\frac{\|\hat{\theta}\|_2^2}{\alpha^2}\left(1+\sqrt{\frac{\ln n}{k}}\right)^2\right)+14+2\ln\frac{1}{\delta}+5\ln n}{4n}}+\gamma_{\mathrm{A}}\end{aligned}$$

where the last inequality holds due to $1-1/\sqrt{n}\leq 1$ and Jensen's inequality.

And then consider $\|\hat{\theta}\|^2 > \alpha^2(\exp(4n/k)-1)$, apparently in (18), the RHS

$$\begin{aligned}&\max_{\|\epsilon\|_2\leq\alpha}\mathcal{L}(\theta_m(\hat{\theta}+\epsilon);\mathcal{D}) + \sqrt{\frac{k\ln\left(1+\frac{\|\hat{\theta}\|_2^2}{\alpha^2}\left(1+\sqrt{\frac{\ln n}{k}}\right)^2\right)+14+2\ln\frac{1}{\delta}+5\ln n}{4n}}+\gamma_{\mathrm{A}} \\ >&\max_{\|\epsilon\|_2\leq\alpha}\mathcal{L}(\theta_m(\hat{\theta}+\epsilon);\mathcal{D}) + \sqrt{\frac{4n+14+2\ln\frac{1}{\delta}+5\ln n}{4n}}+\gamma_{\mathrm{A}} \\ >&\max_{\|\epsilon\|_2\leq\alpha}\mathcal{L}(\theta_m(\hat{\theta}+\epsilon);\mathcal{D}) + 1+\gamma_{\mathrm{A}} \\ \geq&\mathcal{L}(\theta_m(\hat{\theta});\mathcal{P})\end{aligned}$$

which completes the proof. □

D.1. Discussion: choice of the perturbation radius α

The upper bound of the population loss on the RHS of (18), is a function of α. A choice of $\alpha>0$ close to zero, approximates the original MAML method without SAM. We explain why SAM improves the generalization ability of MAML by showing that for any sufficiently small $\alpha_0>0$, we can find $\alpha_1>\alpha_0$ where the upper bound of the population loss takes smaller value than at α_0.

Proof. Let $c=\|\theta\|_2^2\big(1+\sqrt{\frac{\ln n}{k}}\big)^2$. Denote

$$g(\alpha)=\max_{\|\epsilon\|_2\leq\alpha}\mathcal{L}(\theta+\epsilon;\mathcal{D})+\sqrt{\frac{k\ln(1+\frac{c}{\alpha^2})+2\ln\frac{1}{\delta}+5\ln n+O(1)}{4n}}+\gamma_{\mathrm{A}}.$$

Since $0\leq\mathcal{L}(\cdot)\leq 1$, it follows that for any $0<\alpha_0<(\frac{c}{\exp(4n/k)-1})^{1/2}$,

$$g(\alpha_0)\geq\sqrt{\frac{k\ln\left(1+\frac{c}{\alpha_0^2}\right)+2\ln\frac{1}{\delta}+5\ln n+O(1)}{4n}}+\gamma_{\mathrm{A}}.$$

Choose

$$\alpha_1>\Big(\frac{c}{\big(1+\frac{c}{\alpha_0^2}\big)\exp(-4n/k)-1}\Big)^{1/2}>\Big(\frac{c}{\big(1+\frac{c}{\alpha_0^2}\big)-1}\Big)^{1/2}=\alpha_0$$

then it follows that

$$\begin{aligned} g(\alpha_1) &\leq 1+\sqrt{\frac{k\ln\left(1+\frac{c}{\alpha_1^2}\right)+2\ln\frac{1}{\delta}+5\ln n+O(1)}{4n}}+\gamma_{\mathrm{A}} \\ &< 1+\sqrt{\frac{k\ln\left(\left(1+\frac{c}{\alpha_0^2}\right)\exp(-4n/k)\right)+2\ln\frac{1}{\delta}+5\ln n+O(1)}{4n}}+\gamma_{\mathrm{A}} \\ &= 1+\sqrt{\frac{-4n+k\ln\left(1+\frac{c}{\alpha_0^2}\right)+2\ln\frac{1}{\delta}+5\ln n+O(1)}{4n}}+\gamma_{\mathrm{A}} \\ &\leq \sqrt{\frac{k\ln\left(1+\frac{c}{\alpha_0^2}\right)+2\ln\frac{1}{\delta}+5\ln n+O(1)}{4n}}+\gamma_{\mathrm{A}} \\ &\leq g(\alpha_0) \end{aligned}$$

which completes the proof. □

D.2. Discussion: justification on the assumption

To obtain the generalization bound, we assume the population loss

$$\mathcal{L}(\theta_m(\hat{\theta});\mathcal{P}) \leq \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\alpha^2\mathbf{I})}[\mathcal{L}(\theta_m(\hat{\theta}+\epsilon);\mathcal{P})] \tag{36}$$

at the stationary point of the Sharp-MAML$_{\mathrm{up}}$ denoted by $\hat{\theta}$. We give some discussion next to justify this assumption.

If $\hat{\theta}$ is the local minimizer of $\mathcal{L}(\theta_m(\hat{\theta});\mathcal{D})$, then with high probability, $\|\epsilon\|^2 \leq \alpha^2$, $\mathcal{L}(\theta_m(\hat{\theta});\mathcal{D}) \leq \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\alpha^2\mathbf{I})}[\mathcal{L}(\theta_m(\hat{\theta});\mathcal{D})]$. Assume the empirically observed $\mathcal{D}$ is representative of $\mathcal{P}$ and preserves the local property of the loss landscape $\mathcal{L}(\theta_m(\hat{\theta});\mathcal{P})$ around $\hat{\theta}$, i.e. for $\mathcal{D}\sim\mathcal{P}$, $|\mathcal{D}|\to\infty$, $\mathcal{L}(\theta_m(\hat{\theta});\mathcal{D}) \leq \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\alpha^2\mathbf{I})}[\mathcal{L}(\theta_m(\hat{\theta});\mathcal{D})]$ with high probability, then we have $\mathcal{L}(\theta_m(\hat{\theta});\mathcal{P}) \leq \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\alpha^2\mathbf{I})}[\mathcal{L}(\theta_m(\hat{\theta});\mathcal{P})]$.

Table 4. Results on Omniglot (20-way 1-shot).

ALGORITHMS	ACCURACY
MATCHING NETS	93.8%
REPTILE (NICHOL & SCHULMAN, 2018)	89.43%
FOMAML (NICHOL & SCHULMAN, 2018)	89.40%
MAML (REPRODUCED)	91.77 %
SHARP-MAML$_{\mathrm{low}}$	92.89 %
SHARP-MAML$_{\mathrm{up}}$	92.96 %
SHARP-MAML$_{\mathrm{both}}$	93.47 %

Table 5. Results on Omniglot (20-way 5-shot).

ALGORITHMS	ACCURACY
MATCHING NETS	98.50%
FOMAML (NICHOL & SCHULMAN, 2018)	97.12%
REPTILE (NICHOL & SCHULMAN, 2018)	97.90%
MAML (REPRODUCED)	96.16%
SHARP-MAML$_{\mathrm{low}}$	96.59%
SHARP-MAML$_{\mathrm{up}}$	96.62%
SHARP-MAML$_{\mathrm{both}}$	96.64 %

E. Additional Experiments

In this section, we provide additional details of the experimental set-up and present our results on the Omniglot dataset.

Few-shot classification on Omniglot dataset. We used the same experimental setups in (Finn et al., 2017). We use only one inner gradient step with 0.1 learning rate for all our experiments for training and testing. The batch size was set to 16 for the 20-way learning setting. Following (Ravi & Larochelle, 2017), 15 examples per class were used to evaluate the post-update meta-gradient. The values of $\alpha_{\rm low}$ and $\alpha_{\rm up}$ are chosen from the grid search on the set $\{0.05, 0.005, 0.0005, 0.00005\}$ and each experiment is run on each value for three random seeds. We choose the inner gradient steps from a set of $\{3, 5, 7, 10\}$. The step size is chosen via the grid search from a set of $\{0.1, 0.01, 0.001\}$. For Sharp-MAML$_{\rm both}$ we use the same value of $\alpha_{\rm low}$ and $\alpha_{\rm up}$ in each experiment. The reproduced result of MAML for the 20-way 1-shot setting is close to that of MAML++ (Antoniou et al., 2019). For the 20-way 1-shot setting, we observe a similar trend where Sharp-MAML$_{\rm both}$ achieves the best accuracy of 93.47% as compared to 91.77% of MAML. The performance gain of Sharp-MAML on the Omniglot dataset is not as significant as the Mini-Imagenet dataset because the former task is much simpler.