ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10448144

META-LEARNING WITH VERSATILE LOSS GEOMETRIES FOR FAST ADAPTATION USING MIRROR DESCENT

Yilang Zhang, Bingcong Li, Georgios B. Giannakis

Dept. of ECE, University of Minnesota, Minneapolis, MN 55455, USA

ABSTRACT

Utilizing task-invariant prior knowledge extracted from related tasks, meta-learning is a principled framework that empowers learning a new task especially when data records are limited. A fundamental challenge in meta-learning is how to quickly "adapt" the extracted prior in order to train a task-specific model within *a few* optimization steps. Existing approaches deal with this challenge using a preconditioner that enhances convergence of the per-task training process. Though effective in representing locally a quadratic training loss, these simple linear preconditioners can hardly capture complex loss geometries. The present contribution addresses this limitation by learning a nonlinear mirror map, which induces a versatile distance metric to enable capturing and optimizing a wide range of loss geometries, hence facilitating the per-task training. Numerical tests on few-shot learning datasets demonstrate the superior expressiveness and convergence of the advocated approach.

***Index Terms*—** Meta-learning, bilevel optimization, mirror descent, loss geometries

1. INTRODUCTION

The success of deep learning relies heavily on large-scale and high-dimensional models, which require extensive training using a large number of data. However, this "data-driven learning" approach is not feasible in applications where data are scarce due to costly data collection and labelling process. Examples of such applications include drug discovery [1], machine translation [2], and robot manipulation [3].

In contrast, *meta-learning* offers a powerful approach for learning a task in data-limited setups. Specifically, meta-learning extracts *task-invariant prior* information from a collection of given tasks, that can subsequently aid learning of a new, albeit related task. Although this new task may have limited training data, the prior serves as a strong inductive bias that effectively transfers knowledge to aid its learning. In image classification for instance, a feature extractor learned from a collection of given tasks can act as a common prior, and thus benefit a variety of other image classification tasks.

This work was supported by NSF grants 2126052, 2128593, 2212318, 2220292, and 2312547; Emails: {zhan7453,lixx5599,georgios}@umn.edu

Depending on how this "data-limited learning" is performed, meta-learning algorithms can be categorized into neural network (NN)- and optimization-based ones. In NN-based ones, the per-task learning is viewed as an NN mapping from its training data to task-specific model parameters [4, 5]. The prior information is encoded in the NN weights, which are shared and optimized across tasks. With the universality of NNs in approximating complex mappings granted, their black-box structure challenges their reliability and interpretability. On the other hand, optimization-based meta-learning alternatives interpret "data-limited learning" as a cascade of a few optimization iterations (a.k.a. adaptation) over the model parameters. The prior here is captured by the shared hyperparameters of the iterative optimizer. A representative of these alternatives is the model-agnostic meta-learning (MAML) [6], which views the prior as a learnable task-invariant initialization of the optimizer. By starting from an informative initial point, the model parameters can rapidly converge to local minima within a few gradient descent (GD) steps. Building upon MAML, a series of variants have been proposed to learn different priors [7, 8, 9].

While optimization-based meta-learning has been proven effective numerically, recent studies suggest that its generalization and stability heavily rely on convergence of per-task optimization [7, 9]. This motivates one to grow the number of descent iterations. However, this can be infeasible as the overall complexity of meta-learning scales linearly with the number of GD steps [7]. Besides, using accelerated first-order optimizers, such as Adam [10], introduces extra backpropagation complexity when optimizing the prior. To improve the per-task convergence without markedly adding to the complexity, another line of research focuses on second-order optimization using a learnable precondition matrix having simple form [11, 12, 13, 14, 15, 16]. In fact, the precondition matrix captures the local quadratic curvature of the training loss, and linearly transforms the gradient based on this curvature. To acquire more expressive preconditioners, recent advances suggest replacing the linear matrix multiplication with a nonlinear NN transformation [17]. However, convergence of this NN-manipulated GD is an uncharted territory.

The present work advocates learning a generic distance metric induced by a strictly increasing nonlinear mirror map, which enables efficient optimization over generic loss geome-

tries. All in all, our contribution is three-fold.

i) Broadening linear preconditioners with guaranteed per-task convergence.
ii) Blockwise inverse autoregressive flow (blockIAF) ensuring monotonicity and scalability of the mirror map.
iii) Numerical tests showing superior performance and improved convergence compared to linear preconditioners.

2. PROBLEM SETUP

To enable "data-limited learning" of a new task, meta-learning forms task-invariant priors using a collection of given tasks indexed by $t = 1, \ldots, T$. Each task comprises a dataset $\mathcal{D}_t := \{(\mathbf{x}_t^n, y_t^n)\}_{n=1}^{N_t}$ consisting of N_t data-label pairs, which are split into a training subset $\mathcal{D}_t^{\text{trn}}$, and a disjoint validation subset $\mathcal{D}_t^{\text{val}}$. The new task, indexed by $\star$, contains a training subset $\mathcal{D}_\star^{\text{trn}}$, and a set of test data $\{\mathbf{x}_\star^n\}_{n=1}^{N_\star^{\text{tst}}}$ for which the corresponding labels $\{y_\star^n\}_{n=1}^{N_\star^{\text{tst}}}$ are to be predicted. The key premise of meta-learning is that all the aforementioned tasks share related model structures or data distributions. Thus, one can postulate a large model shared across all tasks, along with distinct model parameters $\boldsymbol{\phi}_t \in \mathbb{R}^d$ per individual task. But since the cardinality $N_t^{\text{trn}} := |\mathcal{D}_t^{\text{trn}}|$ can be much smaller than d, learning a task by directly optimizing $\boldsymbol{\phi}_t$ over $\mathcal{D}_t^{\text{trn}}$ is impractical. Fortunately, since T is considerably large, a task-invariant prior can be learned using $\{\mathcal{D}_t^{\text{val}}\}_{t=1}^T$ to render per-task learning well posed.

Letting $\boldsymbol{\theta} \in \mathbb{R}^{d'}$ denote the vector parameter of the prior, the meta-learning objective can be formulated as a bilevel optimization problem. The lower-level trains each task-specific model by optimizing $\boldsymbol{\phi}_t$ using $\mathcal{D}_t^{\text{trn}}$ and $\boldsymbol{\theta}$ from the upper-level. The upper-level adjusts $\boldsymbol{\theta}$ by evaluating the optimized $\boldsymbol{\phi}_t$ on the validation sets $\{\mathcal{D}_t^{\text{val}}\}_{t=1}^T$. The two levels depend on each other and yield the following nested objective

$$\min_{\boldsymbol{\theta}} \sum_{t=1}^{T} \mathcal{L}(\boldsymbol{\phi}_t^*(\boldsymbol{\theta}); \mathcal{D}_t^{\text{val}}) \tag{1a}$$

$$\text{s.t. } \boldsymbol{\phi}_t^*(\boldsymbol{\theta}) = \operatorname*{argmin}_{\boldsymbol{\phi}_t} \mathcal{L}(\boldsymbol{\phi}_t; \mathcal{D}_t^{\text{trn}}) + \mathcal{R}(\boldsymbol{\phi}_t; \boldsymbol{\theta}),\ \forall t \tag{1b}$$

where $\mathcal{L}$ is the loss function capturing each task-specific model fit, and $\mathcal{R}$ is the regularizer accounting for the task-invariant prior. From the Bayesian viewpoint, $\mathcal{L}$ and $\mathcal{R}$ represent the negative log-likelihood (nll), $-\log p(\mathbf{y}_t^{\text{trn}}|\boldsymbol{\phi}_t; \mathbf{X}_t^{\text{trn}})$, and the negative log-prior (nlp) $-\log p(\boldsymbol{\phi}_t; \boldsymbol{\theta})$, where $\mathbf{X}_t^{\text{trn}} := [\mathbf{x}_t^1, \ldots, \mathbf{x}_t^{N_t^{\text{trn}}}]$ and $\mathbf{y}_t^{\text{trn}} := [y_t^1, \ldots, y_t^{N_t^{\text{trn}}}]^\top$ ($^\top$ denotes transpose). Bayes' rule then implies $\boldsymbol{\phi}_t^* = \operatorname{argmin} -\log p(\boldsymbol{\phi}_t | \mathbf{y}_t^{\text{trn}}; \mathbf{X}_t^{\text{trn}}, \boldsymbol{\theta})$ is the maximum a posteriori (MAP) estimator.

Reaching the global optimum $\boldsymbol{\phi}_t^*$ is generally infeasible because the task-specific model is nonlinear. Hence, a prudent remedy is to rely on an approximate solver $\hat{\boldsymbol{\phi}}_t \approx \boldsymbol{\phi}_t^*$ obtained by a tractable optimizer. For instance, MAML replaces (1b) with a K-step GD minimizing the nll:

$$\boldsymbol{\phi}_t^{(k)}(\boldsymbol{\theta}) = \boldsymbol{\phi}_t^{(k-1)}(\boldsymbol{\theta}) - \alpha \nabla \mathcal{L}(\boldsymbol{\phi}_t^{(k-1)}(\boldsymbol{\theta}); \mathcal{D}_t^{\text{trn}}),\ \forall t \tag{2}$$

where $k = 1, \ldots, K$ indexes iterations; initialization $\boldsymbol{\phi}_t^{(0)} = \boldsymbol{\phi}^{(0)} = \boldsymbol{\theta}$; approximate solver $\hat{\boldsymbol{\phi}}_t(\boldsymbol{\theta}) = \boldsymbol{\phi}_t^{(K)}(\boldsymbol{\theta})$; and α denotes the step size. Although $\mathcal{R}(\boldsymbol{\phi}_t; \boldsymbol{\theta}) = 0$ in MAML, it has been shown that the GD solver satisfies [18]

$$\hat{\boldsymbol{\phi}}_t(\boldsymbol{\theta}) \approx \boldsymbol{\phi}_t^*(\boldsymbol{\theta}) = \operatorname*{argmin}_{\boldsymbol{\phi}_t} \mathcal{L}(\boldsymbol{\phi}_t; \mathcal{D}_t^{\text{trn}}) + \frac{1}{2}\|\boldsymbol{\phi}_t - \boldsymbol{\theta}\|_{\boldsymbol{\Lambda}_t}^2,\ \forall t$$

where the precision matrix $\boldsymbol{\Lambda}_t$ is determined by $\nabla^2 \mathcal{L}(\boldsymbol{\theta}; \mathcal{D}_t^{\text{trn}})$, α, and K. This indicates that MAML's optimization strategy (2) is approximately tantamount to an implicit Gaussian prior probability density function (pdf) $p(\boldsymbol{\phi}_t; \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}, \boldsymbol{\Lambda}_t^{-1})$, with the task-invariant initialization serving as the mean vector. Alongside implicit priors, their explicit counterparts have also been investigated with various prior pdfs [7, 9].

For both implicit and explicit priors, numerical studies [11, 13] and theoretical analyses [7, 9] demonstrate that the gradient error for optimizing $\boldsymbol{\theta}$ in (1a) relies on the convergence accuracy of $\hat{\boldsymbol{\phi}}_t$ relative to a stationary point. In addition, employing a large K or complicated optimizers could prohibitively escalate the overall complexity for solving (1). As a consequence, attention has been directed towards preconditioned GD (PGD) solvers, as in the update

$$\boldsymbol{\phi}_t^{(k)}(\boldsymbol{\theta}) = \boldsymbol{\phi}_t^{(k-1)}(\boldsymbol{\theta}) - \alpha \mathbf{P}(\boldsymbol{\theta}_P) \nabla \mathcal{L}(\boldsymbol{\phi}_t^{(k-1)}(\boldsymbol{\theta}); \mathcal{D}_t^{\text{trn}}) \tag{3}$$

where $\boldsymbol{\theta}_P$ parametrizes $\mathbf{P} \in \mathbb{R}^{d \times d}$, and the prior parameter is augmented as $\boldsymbol{\theta} := [\boldsymbol{\phi}^{(0)\top}, \boldsymbol{\theta}_P^\top]^\top$. To ensure (3) incurs affordable complexity after preconditioning, $\mathbf{P}$ must have a simple enough structure so that $\mathbf{P}(\boldsymbol{\theta}_P)\nabla \mathcal{L}(\boldsymbol{\phi}_t^{(k-1)}; \mathcal{D}_t^{\text{trn}})$ incurs computational complexity $\mathcal{O}(d)$. Examples of such structures include diagonal [11, 12], block-diagonal [13, 14], and NN-based [15] matrices. A more generic preconditioner can be formed by replacing the linear transformation $\mathbf{P}(\boldsymbol{\theta}_P)\nabla \mathcal{L}(\boldsymbol{\phi}_t^{(k-1)}; \mathcal{D}_t^{\text{trn}})$ with a nonlinear NN $f(\nabla \mathcal{L}(\boldsymbol{\phi}_t^{(k-1)}; \mathcal{D}_t^{\text{trn}}); \boldsymbol{\theta}_P)$ [17], but unfortunately convergence of this alternative iterate may not be guaranteed.

Essentially, GD conducts a pre-step greedy search with a quadratic loss approximation. To see this, let $\text{lin}(\mathcal{L}(\boldsymbol{\phi}_t), \bar{\boldsymbol{\phi}}_t) := \mathcal{L}(\bar{\boldsymbol{\phi}}_t; \mathcal{D}_t^{\text{trn}}) + (\boldsymbol{\phi}_t - \bar{\boldsymbol{\phi}}_t)^\top \nabla \mathcal{L}(\bar{\boldsymbol{\phi}}_t; \mathcal{D}_t^{\text{trn}})$. Using this linearization of $\mathcal{L}$ at $\bar{\boldsymbol{\phi}}_t \in \mathbb{R}^d$, the GD update reduces to (cf. (2))

$$\boldsymbol{\phi}_t^{(k)} = \operatorname*{argmin}_{\boldsymbol{\phi}_t} \text{lin}(\mathcal{L}(\boldsymbol{\phi}_t), \boldsymbol{\phi}_t^{(k-1)}) + \frac{1}{2\alpha}\|\boldsymbol{\phi}_t - \boldsymbol{\phi}_t^{(k-1)}\|_2^2 \tag{4}$$

where dependencies on $\boldsymbol{\theta}$ are dropped hereafter for notational brevity. The term $\frac{1}{2\alpha}\|\boldsymbol{\phi}_t - \boldsymbol{\phi}_t^{(k-1)}\|_2^2$ implies the isotropic approximation $\nabla^2 \mathcal{L}(\boldsymbol{\phi}_t^{(k-1)}; \mathcal{D}_t^{\text{trn}}) \approx \frac{1}{\alpha}\mathbf{I}_d$, while (3) refines the approximation as a more informative matrix $\frac{1}{\alpha}\mathbf{P}^{-1}$ (if invertible). This quadratic local approximation is particularly effective when K is large and α is small, which gradually ameliorates $\boldsymbol{\phi}_t^{(k)}$ to a stationary point. In meta-learning however, the standard setup relies on a small K (e.g., 1 or 5) and a sufficiently large α, so that the model can quickly adapt to the task with low complexity. This tradeoff highlights the need for learning more expressive loss geometries.

3. LOSS GEOMETRIES USING MIRROR DESCENT

Instead of quadratic approximations of the local loss induced by certain norms (e.g., $\|\cdot\|_2$ and $\|\cdot\|_{\mathbf{P}^{-1}}$), our fresh idea is a data-driven distance metric that captures a broader spectrum of loss geometries. This is accomplished by learning the so-termed "mirror map," which will be introduced first. All the proofs are delegated to the Appendix.

3.1. Modeling the loss geometry using the mirror map

To generalize the (P)GD, we will replace the ℓ_2-norm in (4) with a generic metric D_h to arrive at

$$\boldsymbol{\phi}_t^{(k)} = \operatorname*{argmin}_{\boldsymbol{\phi}_t} \operatorname{lin}(\mathcal{L}(\boldsymbol{\phi}_t), \boldsymbol{\phi}_t^{(k-1)}) + \frac{1}{\alpha} D_h(\boldsymbol{\phi}_t, \boldsymbol{\phi}_t^{(k-1)}) \quad (5)$$

where $D_h(\boldsymbol{\phi}_t, \boldsymbol{\phi}_t^{(k-1)}) := h(\boldsymbol{\phi}_t) - \operatorname{lin}(h(\boldsymbol{\phi}_t), \boldsymbol{\phi}_t^{(k-1)})$ is the Bregman divergence, and the associated distance-generating function $h : \mathbb{R}^d \mapsto \mathbb{R}$ is strongly convex to ensure the existence and uniqueness of the minimizer. As a result, ∇h is strictly increasing, and thus invertible[1]. Then, applying the optimality condition leads to the mirror descent (MD) update

$$\boldsymbol{\phi}_t^{(k)} = (\nabla h)^{-1}\big(\nabla h(\boldsymbol{\phi}_t^{(k-1)}) - \alpha \nabla \mathcal{L}(\boldsymbol{\phi}_t^{(k-1)}; \mathcal{D}_t^{\mathrm{trn}})\big). \quad (6)$$

The invertible ∇h, dubbed mirror map, connects $\boldsymbol{\phi}_t$ in the primal space to $\nabla\mathcal{L}$ in the dual space under the endowed metric D_h. As a special case, when choosing $h(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, it is easy to verify that (6) boils down to (2) due to the self-duality of the ℓ_2-norm. Likewise, (3) can be obtained with $h(\cdot) = \frac{1}{2}\|\cdot\|_{\mathbf{P}^{-1}}^2$, where ∇h reduces to a linear mapping. Function h reflects our prior knowledge about the geometry of $\mathcal{L}$. In particular, letting $h(\cdot) = \mathcal{L}(\cdot; \mathcal{D}_t^{\mathrm{trn}})$ (even when $\mathcal{L}$ is not strong convex) in (5) gives $\boldsymbol{\phi}_t^{(k)} = \operatorname{argmin}_{\boldsymbol{\phi}_t} \mathcal{L}(\boldsymbol{\phi}_t; \mathcal{D}_t^{\mathrm{trn}})$, which is precisely the original nll minimization solved in (2) and (3). Thus, an ideal choice of h would yield $h \approx \mathcal{L}$ (up to a constant) within a sufficiently large region around $\boldsymbol{\phi}_t^{(k-1)}$.

Different from past works that rely on a simple preselected h to model loss geometries, we here acquire a data-driven h by learning a strictly increasing ∇h that best fits the given tasks. Interestingly, (6) can be reformulated to yield an update of the dual vector $\mathbf{z}_t := \nabla h(\boldsymbol{\phi}_t)$ as

$$\mathbf{z}_t^{(k)} = \mathbf{z}_t^{(k-1)} - \alpha \nabla \mathcal{L}\big((\nabla h)^{-1}(\mathbf{z}_t^{(k-1)}); \mathcal{D}_t^{\mathrm{trn}}\big) \quad (7)$$

with $\mathbf{z}_t^{(0)} = \nabla h(\boldsymbol{\phi}^{(0)})$ and $\hat{\boldsymbol{\phi}}_t = (\nabla h)^{-1}(\mathbf{z}_t^{(K)})$. Hence, it suffices to learn a strictly increasing $(\nabla h)^{-1}$ and a task-invariant dual initialization $\mathbf{z}^{(0)} := \nabla h(\boldsymbol{\phi}^{(0)})$, thus removing the need for directly calculating ∇h.

3.2. Learning the inverse mirror map via blockIAF

Inspired by this observation, a prudent option is to model $(\nabla h)^{-1}$ as an inverse autoregressive flow (IAF) [19]. The notable benefit of IAF lies in its efficient parallelization of forward computation, that makes it considerably faster than computing its inverse. However, directly applying the dimension-wise IAF to the high-dimensional $\mathbf{z}_t \in \mathbb{R}^d$ will incur prohibitively high complexity of $\Omega(d^2)$. For this reason, we introduce a novel blockIAF model that effectively reduces complexity by performing block-wise (nonlinear) autoregression on a low-dimensional space encoding $\mathbf{z}_t$. To this end, let $\{\mathcal{B}_i\}_{i=1}^B$ be a partition of the index set $\{1, \ldots, d\}$, and $[\mathbf{z}_t]_{\mathcal{B}_i}$ denote the subvector of $\mathbf{z}_t$ restricted to the block $\mathcal{B}_i$. The blockIAF model transforms $\mathbf{z}_t$ to $\boldsymbol{\phi}_t$ through

$$[\boldsymbol{\phi}_t]_{\mathcal{B}_i} = [\mathbf{z}_t]_{\mathcal{B}_i} \odot \sigma(\boldsymbol{\alpha}_i) + \boldsymbol{\mu}_i \quad (8a)$$

$$[\boldsymbol{\alpha}_i^\top, \boldsymbol{\mu}_i^\top]^\top = d_i\big(\{e_j([\mathbf{z}_t]_{\mathcal{B}_j})\}_{j=1}^{i-1}\big),\ i = 1, \ldots, B \quad (8b)$$

where nonlinearity σ is positive and upper bounded (e.g., logistic function), $\sigma(\boldsymbol{\alpha}_i), \boldsymbol{\mu}_i \in \mathbb{R}^{|\mathcal{B}_i|}$ are the scale and shift of $[\mathbf{z}_i]_{\mathcal{B}_i}$, e_i and d_i denote learnable encoder and decoder for the i-th block, and $\odot$ is the Hadamard (element-wise) product. In our implementation, $\{e_i\}_{i=1}^{B-1}$ and $\{d_i\}_{i=1}^B$ are multilayer perceptrons (MLPs) with ReLU activations. To further reduce complexity, all linear layers in MLPs are implemented by tensor mode product [13]. This technique is equivalent to a low-rank Kronecker approximation to MLPs' weight matrices. This lowers the per-step MD complexity to $\mathcal{O}(d)$.

The following theorem characterizes two important properties of the proposed blockIAF model.

Theorem 1. *Let $g : \mathbb{R}^d \mapsto \mathbb{R}^d$ be the blockIAF model* (8). *For any partition $\{\mathcal{B}_i\}_{i=1}^B$, g is strictly increasing, that is*

$$(\mathbf{z}_t - \mathbf{z}_t')^\top (g(\mathbf{z}_t) - g(\mathbf{z}_t')) > 0, \ \ \forall \mathbf{z}_t \neq \mathbf{z}_t'. \quad (9)$$

Moreover, there exists a constant $C > 0$ such that

$$\nabla(g^{-1})(\boldsymbol{\phi}_t) \succeq C. \quad (10)$$

Theorem 1 asserts that with $(\nabla h)^{-1} = g$, one ensures the desired strict monotonicity, and strong convexity of the induced h (by noting that $\nabla^2 h = \nabla(g^{-1})$). As a result, the per-task optimization (7) enjoys the standard convergence guarantee of MD. Although the convergence rate of MD is in the same order as GD, it outperforms GD markedly in the constant factor when d is large [20], and relies on more relaxed assumptions [21].

The meta-learning objective (1) is solved using alternating optimization. With $\boldsymbol{\theta}_g$ denoting the blockIAF parameters, let $\boldsymbol{\theta} := [\mathbf{z}^{(0)\top}, \boldsymbol{\theta}_g^\top]^\top$ be the prior parameter vector. In the (r)-th iteration of (1a), the optimizer has access to $\boldsymbol{\theta}^{(r-1)}$ provided by its last iteration, and a batch of randomly sampled tasks $\mathcal{T}^{(r)} \subset \{1, \ldots, T\}$. The optimizer first solves $\hat{\boldsymbol{\phi}}_t(\boldsymbol{\theta}^{(r-1)})$ for each $t \in \mathcal{T}^{(r)}$ leveraging the K-step MD (7). Then, $\boldsymbol{\theta}^{(r-1)}$ is updated using mini-batch stochastic GD with step size β:

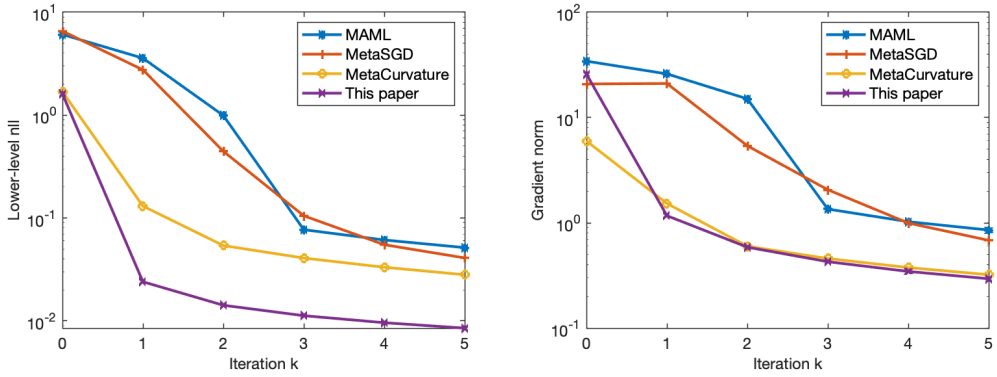
$$\boldsymbol{\theta}^{(r)} = \boldsymbol{\theta}^{(r-1)} - \beta \frac{T}{|\mathcal{T}^{(r)}|} \sum_{t \in \mathcal{T}^{(r)}} \nabla_{\boldsymbol{\theta}^{(r-1)}} \mathcal{L}(\hat{\boldsymbol{\phi}}_t(\boldsymbol{\theta}^{(r-1)}); \mathcal{D}_t^{\mathrm{val}}).$$

A summary of the algorithm can be found in the Appendix.

[1] When ∇h is discontinuous but h is proper, the inverse $(\nabla h)^{-1}$ is defined as $\nabla h^*(\mathbf{z}) := \operatorname{argmax}_{\boldsymbol{\phi}} \boldsymbol{\phi}^\top \mathbf{z} - h(\boldsymbol{\phi})$, where $h^*(\mathbf{z}) := \sup_{\boldsymbol{\phi}} \boldsymbol{\phi}^\top \mathbf{z} - h(\boldsymbol{\phi})$ is the Fenchel conjugate of h.

Table 1: Comparison of meta-learning algorithms with different loss geometry models on the 5-class miniImageNet dataset. Maximum and mean accuracies within its 95% confidence interval are in bold. (No model ensembling for a fair comparison.)

Method	Lower-level optimizer	Loss geometry model	5-class accuracies	
			1-shot	5-shot
MAML [6]	GD	identity matrix	$48.70 \pm 1.84\%$	$63.11 \pm 0.92\%$
MetaSGD [11]	PGD	diag. matrix	$50.47 \pm 1.87\%$	$64.03 \pm 0.94\%$
MT-net [14]	PGD	block diag. matrix	$51.70 \pm 1.84\%$	–
WarpGrad [15]	PGD	NN-based low-rank matrix	$52.3 \pm 0.8\%$	$68.4 \pm 0.6\%$
MetaCurvature [13]	PGD	block diag. & Kron. (low-rank) matrix	$54.23 \pm 0.88\%$	$67.99 \pm 0.73\%$
MetaKFO [17]	NN-transformed GD	NN-based gradient transformation	–	64.9%
ECML [16]	PGD	Gauss-Newton approximation	$48.94 \pm 0.80\%$	$65.26 \pm 0.67\%$
This paper's method	MD	blockIAF-based mirror map	$\mathbf{56.10 \pm 1.43\%}$	$\mathbf{69.59 \pm 0.71\%}$



(a) $\mathcal{L}(\boldsymbol{\phi}_\star^{(k)}; \mathcal{D}_\star^{\mathrm{trn}})$ versus k (b) $\|\nabla \mathcal{L}(\boldsymbol{\phi}_\star^{(k)}; \mathcal{D}_\star^{\mathrm{trn}})\|_2$ versus k

Fig. 1: Convergence comparison on randomly sampled new tasks.

4. NUMERICAL TESTS

Here we compare the empirical performance of optimization-based meta-learning using different lower-level optimizers, on the standard few-shot classification dataset miniImageNet [22], where "shots" signify the per-class training data for each t. The task-specific model is a standard 4-layer convolutional NN (CNN) [22, 6]. Each layer comprises a 3×3 convolution of 64 channels, batch normalization, ReLU activation, and 2×2 max pooling module. After the convolutional layers, a linear regressor with softmax activation is appended to perform classification. Subset $\mathcal{B}_i$ is formed by the weight indices of the i-th CNN layer. The autoregression in (8b) implies that "how to optimize weights of the i-th layer" depends on "how weights of previous layers have been optimized." This choice enables blockIAF to model the optimization dependency of high-level features (e.g., textures and patterns) on low-level ones (e.g., colors and edges). Test setups and hyperparameters can be found in the Appendix.

Table 1 lists various loss geometry models, where classification accuracy on new tasks is the figure of merit. For fairness, MAML is the backbone of all methods. By utilizing a more versatile loss geometry model, our approach outperforms the state-of-the-art ones by a large margin.

To further gauge the performance gain achieved by our novel approach, Fig. 1 visualizes the convergence of $\mathcal{L}(\boldsymbol{\phi}_\star^{(k)}; \mathcal{D}_\star^{\mathrm{trn}})$ averaged on $1,000$ random new tasks. The proposed method results in faster convergence to a lower and more stable nll compared with all three competitors. Moreover, Fig. 1a reveals that both the proposed method and MetaCurvature improve the initialization compared to MAML and MetaSGD. This confirms that convergence and generalization of (1a) relies on the convergence accuracy of $\hat{\boldsymbol{\phi}}_t$ [7, 9]. Fig. 1b further illustrates that although the initial gradients of different methods have comparable norms $\|\nabla \mathcal{L}(\boldsymbol{\phi}_\star^{(0)}; \mathcal{D}_\star^{\mathrm{trn}})\|_2$, our method can make better use of the gradient, leading to a rapid reduction of the nll as well as its gradient norm at $k = 1$. This improved gradient utilization highlights our method's superior modeling of loss geometries.

5. CONCLUSIONS AND OUTLOOK

Versatile loss geometry models can accelerate the lower-level convergence in meta-learning. A novel BlockIAF model is introduced to learn the inverse mirror map $(\nabla h)^{-1}$ induced by a strongly convex h. The resultant algorithm generalizes preconditioning-based meta-learning, captures versatile loss geometries, and improves lower-level convergence. Effectiveness of the novel approach was validated on a standard few-shot dataset. Future research includes bi-level convergence guarantees for the proposed method, and development of more expressive yet scalable inverse mirror maps.

6. REFERENCES

[1] Han Altae-Tran, Bharath Ramsundar, Aneesh S. Pappu, and Vijay Pande, "Low data drug discovery with one-shot learning," *ACS Central Science*, vol. 3, no. 4, pp. 283–293, 2017.

[2] Jiatao Gu, Yong Wang, Yun Chen, Kyunghyun Cho, and Victor OK Li, "Meta-learning for low-resource neural machine translation," *arXiv preprint arXiv:1808.08437*, 2018.

[3] Ignasi Clavera, Anusha Nagabandi, Simin Liu, Ronald S. Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn, "Learning to adapt in dynamic, real-world environments through meta-reinforcement learning," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[4] Sachin Ravi and Hugo Larochelle, "Optimization as a model for few-shot learning," in *Proc. Int. Conf. Learn. Represent.*, 2017.

[5] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel, "A simple neural attentive meta-learner," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, vol. 70, pp. 1126–1135.

[7] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine, "Meta-learning with implicit gradients," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32.

[8] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto, "Meta-learning with differentiable convex optimization," in *Proc. IEEE/CVF Conf. on Comp. Vis. and Pat. Recog.*, 2019.

[9] Yilang Zhang, Bingcong Li, Shijian Gao, and Georgios B. Giannakis, "Scalable bayesian meta-learning through generalized implicit gradients," in *Proc. AAAI Conf. Artif. Intel.*, 2023, vol. 37(9), pp. 11298–11306.

[10] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015.

[11] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li, "Meta-sgd: Learning to learn quickly for few-shot learning," *arXiv preprint arXiv:1707.09835*, 2017.

[12] Boyan Gao, Henry Gouk, Hae Beom Lee, and Timothy M Hospedales, "Meta mirror descent: Optimiser learning for fast convergence," *arXiv preprint arXiv:2203.02711*, 2022.

[13] Eunbyung Park and Junier B Oliva, "Meta-curvature," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, vol. 32.

[14] Yoonho Lee and Seungjin Choi, "Gradient-based meta-learning with learned layerwise metric and subspace," in *Proc. Int. Conf. Mach. Learn.*, 2018, vol. 80, pp. 2927–2936.

[15] Sebastian Flennerhag, Andrei A. Rusu, Razvan Pascanu, Francesco Visin, Hujun Yin, and Raia Hadsell, "Meta-learning with warped gradient descent," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[16] Markus Hiller, Mehrtash Harandi, and Tom Drummond, "On enforcing better conditioned meta-learning for rapid few-shot adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, vol. 35, pp. 4059–4071.

[17] Sébastien M. R. Arnold, Shariq Iqbal, and Fei Sha, "When maml can adapt fast and how to assist when it cannot," in *Proc. Int. Conf. Artif. Intel. and Stats.*, 2021, vol. 130, pp. 244–252.

[18] Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths, "Recasting gradient-based meta-learning as hierarchical Bayes," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[19] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling, "Improved variational inference with inverse autoregressive flow," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, vol. 29.

[20] Aharon Ben-Tal, Tamar Margalit, and Arkadi Nemirovski, "The ordered subsets mirror descent optimization method with applications to tomography," *SIAM Journal on Optimization*, vol. 12, no. 1, pp. 79–108, 2001.

[21] Zhengyuan Zhou, Panayotis Mertikopoulos, Nicholas Bambos, Stephen Boyd, and Peter W Glynn, "Stochastic mirror descent in variationally coherent optimization problems," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30.

[22] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, vol. 29.