

Inexact-ADMM Based Federated Meta-Learning for Fast and Continual Edge Learning

Sheng Yue
Central South University
Arizona State University
sheng.yue@csu.edu.cn

Ju Ren*
Central South University
renju@csu.edu.cn

Jiang Xin
Central South University
xinjiang@csu.edu.cn

Sen Lin
Arizona State University
sen.lin.1@asu.edu

Junshan Zhang
Arizona State University
junshan.zhang@asu.edu

ABSTRACT

In order to meet the requirements for performance, safety, and latency in many IoT applications, intelligent decisions must be made right here right now at the network edge. However, the constrained resources and limited local data amount pose significant challenges to the development of edge AI. To overcome these challenges, we explore continual edge learning capable of leveraging the knowledge transfer from previous tasks. Aiming to achieve fast and continual edge learning, we propose a platform-aided federated meta-learning architecture where edge nodes collaboratively learn a meta-model, aided by the knowledge transfer from prior tasks. The edge learning problem is cast as a regularized optimization problem, where the valuable knowledge learned from previous tasks is extracted as regularization. Then, we devise an ADMM based federated meta-learning algorithm, namely ADMM-FedMeta, where ADMM offers a natural mechanism to decompose the original problem into many subproblems which can be solved in parallel across edge nodes and the platform. Further, a variant of inexact-ADMM method is employed where the subproblems are ‘solved’ via linear approximation as well as Hessian estimation to reduce the computational cost per round to $O(n)$. We provide a comprehensive analysis of ADMM-FedMeta, in terms of the convergence properties, the rapid adaptation performance, and the forgetting effect of prior knowledge transfer, for the general non-convex case. Extensive experimental studies demonstrate the effectiveness and efficiency of ADMM-FedMeta, and showcase that it substantially outperforms the existing baselines.

CCS CONCEPTS

• **Theory of computation** → **Multi-agent learning**; • **Networks** → **Network services**.

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MobiHoc '21, July 26–29, 2021, Shanghai, China

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8558-9/21/07...\$15.00

<https://doi.org/10.1145/3466772.3467038>

KEYWORDS

edge intelligence, continual learning, federated meta-learning, regularization, ADMM

ACM Reference Format:

Sheng Yue, Ju Ren, Jiang Xin, Sen Lin, and Junshan Zhang. 2021. Inexact-ADMM Based Federated Meta-Learning for Fast and Continual Edge Learning. In *The Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '21)*, July 26–29, 2021, Shanghai, China. ACM, New York, NY, USA, 21 pages. <https://doi.org/10.1145/3466772.3467038>

1 INTRODUCTION

The past few years have witnessed an explosive growth of Internet of Things (IoT) devices. In many of these IoT applications, decisions must be made in real time to meet the requirements for safety, accuracy, and performance [46]. A general consensus is that the conventional cloud-based approach would not work well in these applications, calling for edge intelligence or edge AI [28, 48]. Built on a synergy of edge computing and AI, edge intelligence is expected to push the frontier of model training and inference processes to the network edge in the physical proximity of IoT devices and data sources. Nevertheless, it is highly nontrivial for a single edge node to achieve real-time edge intelligence, since AI model training usually requires extensive computing resources and a large number of data samples. To tackle these challenges, we resort to continual learning capable of leveraging the knowledge transfer from previous tasks in the cloud or by other edge nodes. Simply put, continual learning (CL) is a machine learning paradigm that is designed to sequentially learn from data samples corresponding to different tasks [27]. Rather than learning the new model from scratch, CL aims to design algorithms leveraging knowledge transfer from pre-trained models to the new learning task, assuming that the training data of previous tasks are unavailable for the newly coming task (this is the case for edge learning).

To facilitate edge learning, collaborative learning has recently been proposed to leverage the model knowledge distillation, including cloud-edge collaboration and edge-edge collaboration. More specifically, a distributionally robust optimization based edge learning framework has been introduced to build a cloud-edge synergy between the pre-trained model in the cloud and the local data samples at the edge [46]. Along a different avenue, building on the recent advances in meta-learning [9, 26] and federated learning

[24], a significant body of work has been devoted to federated meta-learning [5, 15, 21, 47] and personalized federated learning [8, 40], under a common theme of fostering edge-edge collaboration. In particular, federated meta-learning aims to learn a good model initialization (meta-model) across source edge nodes, such that the model of the new task can be learned via fast adaptation from the meta-initialization with only a few data samples at the target edge node.

Most of the existing works on federated meta-learning focus on the learning from a given set of tasks, each with its training data, but have not addressed the well-known *catastrophic forgetting* issue in continual learning [11] [27]. Further, the performance of the fast adaptation depends on the similarity among tasks [8, 21]. As a result, the meta-model obtained via federated meta-learning may not work well when the tasks on the target node are dissimilar to that at the source nodes. It is also worth noting that most of the existing federated meta-learning algorithms are gradient-based, which may suffer from some limitations such as vanishing gradients and sensitivity to poor conditioning [37]. It has been observed in practice that these gradient based algorithms often exhibit slow convergence in training the meta-model, especially on complex tasks [5, 8, 15], resulting in low communication efficiency and high computational cost.

To tackle the issues noted above, in this paper we study continual edge learning via federated meta-learning with regularization. Notably, regularization-based methods have been widely used in continual learning [17, 31, 44] and transfer learning [32, 34]. Inspired by theoretical neuroscience models via imposing constraints on the update of the neural weights [2], regularization approaches can help to alleviate catastrophic forgetting [27]. Meanwhile, valuable knowledge learned from previous tasks can be extracted as regularization to improve the training speed and the performance of the new task (so called “positive forward transfer” [22]). Accordingly, we cast the federated meta-learning problem as a regularized stochastic optimization problem, using Bregman divergence [4] to define the regularization. Further, to reduce the computational cost and to facilitate collaborative learning, we employ the alternative direction method of multipliers (ADMM) technique to decompose the problem into a set of subproblems which can be solved in parallel across edge nodes and the platform. In particular, by “decoupling the regularizer” from the computation at local edge nodes, it suffices to run the regularization only in the platform for global aggregation. Observe that the conventional ADMM technique requires the exact solutions to a set of (possibly non-convex) subproblems during each iteration, incurring possibly high computational cost. To overcome this challenge, we develop a variant of inexact-ADMM algorithm for the regularized federated meta-learning problem, namely ADMM-FedMeta, where we use linear approximation in each subproblem¹, as well as Hessian estimation, and then transform it into a quadratic form that can be solved with a closed-form solution, thus achieving computational complexity of $O(n)$ per round, with n being the model dimension.

We note that the error induced by linear approximation and Hessian estimation, complicates the proof of the convergence of

¹As shown in the proof of the convergence of ADMM-FedMeta, it is unnecessary to obtain the exact solutions in each iteration, and this is the underlying rationale of the inexact-ADMM.

the proposed algorithm, and the existing results [1, 12, 39] cannot be applied directly, simply because the *sufficient descent* condition of the Lagrangian function is violated. In this paper, we develop a new technical path to resolve this issue and establish the convergence guarantee for the general non-convex case. Further, we rigorously show that our method can mitigate the catastrophic forgetting and alleviate the performance degradation due to the dissimilarity between the source nodes and the target node. Besides, different from the previous approaches [8, 21], our algorithm can converge under mild conditions, i.e., without regular *similarity* assumptions on the training nodes. Therefore, it can be applied to unbalanced and heterogeneous local datasets, unleashing the potential in dealing with the inherent challenges in federated learning.

The main contributions of this work are summarized as follows:

- Aiming to facilitate fast and continual edge learning, we propose a platform-aided federated-meta learning architecture where edge nodes join force to learn a meta-model with the knowledge transfer from previous tasks. We cast the edge learning problem as a regularized optimization problem, in which the transferred knowledge is in the form of a regularization using Bregman divergence. We devise an inexact-ADMM based algorithm, called ADMM-FedMeta, where the ADMM technique is employed to decompose the problem into a set of subproblems which can be solved in parallel across edge nodes and the platform, and also it suffices to run the regularization only in the platform for global aggregation. Further, a variant of inexact-ADMM method is devised where the subproblems are ‘solved’ via linear approximation as well as Hessian estimation to reduce the computational cost of per round to $O(n)$, achieving lower computational complexity compared to most of the existing methods.
- We carry out a comprehensive analysis of the proposed algorithm for the general non-convex case, where we establish the convergence and characterize the performance of fast adaptation using local samples at the target node. We also quantify the forgetting effect of model knowledge transferred from previous tasks for a special case. Besides, we show that ADMM-FedMeta can mitigate performance degradation incurred by the dissimilarity between the source nodes and the target node.
- We evaluate the performance of the proposed algorithm on different models and benchmark datasets. Our extensive experimental results showcase that ADMM-FedMeta outperforms existing state-of-the-art approaches, in terms of convergence speed, adaptation performance, and the capability of learning without forgetting, especially with small sample sizes.

2 RELATED WORK

In this section, we briefly review the related work in the following three areas.

Meta-Learning. Meta-learning has emerged as a promising solution for few-shot learning. Ravi *et al.* [30] propose an LSTM-based meta-learning model to learn an optimization algorithm for training

neural networks. Different from [30], a gradient-based Model Agnostic Meta-Learning (MAML) algorithm is proposed in [9], which aims at learning a model initialization, based on which using a few gradient descent updates can achieve satisfactory performance on a new task. To reduce the computational complexity, Nichol *et al.* [26] introduce a first-order meta-learning algorithm called Reptile, which does not require the computation of the second-order derivatives. Multiple follow-up works extend MAML from different perspectives, e.g., [6, 10, 29, 33]. Along a different line, Fallah *et al.* [7] establish the convergence of one-step MAML for non-convex loss functions, and then proposes a Hessian-free MAML to reduce the computational cost with theoretical guarantee. The convergence for multi-step MAML is studied in [13]. Wang *et al.* [38] further characterize the gap between the stationary point and the global optimum of MAML in a general non-convex setting.

Federated Meta-Learning. Very recently, the integration of federated learning and MAML has garnered much attention. Chen *et al.* [5] propose a federated meta-learning framework called FedMeta based on FedAvg [24] and MAML-type algorithms, which improves the performance and convergence speed of FedAvg. Jiang *et al.* [15] analyze the connections between FedAvg and MAML, and proposes a federated meta-learning algorithm called personalized FedAvg. Lin *et al.* [21] analyze the convergence properties and computational complexity of federated meta-learning for a strongly convex setting. Another recent work [8] proposes a federated meta-learning algorithm called Per-FedAvg and provides the convergence guarantee for general non-convex setting. However, these studies focus on the collaborative learning on a given set of tasks without exploring the valuable knowledge transfer from the previous tasks [27].

ADMM. A number of existing works [12, 23, 36, 39] analyze the convergence of ADMM for the case where the solution to each subproblem is computed exactly. Wang *et al.* [35] extend the ADMM method from two-block to multi-block form. Besides, there also a few of works [1, 14, 20, 25] studying the performance of ADMM in an inexact and non-convex setting, by linearizing the subproblems that are difficult to solve exactly. It is worth noting that linear approximation is insufficient for the meta-learning problem which generally requires higher order information.

3 CONTINUAL EDGE LEARNING VIA FEDERATED META-LEARNING WITH REGULARIZATION

We consider a platform-aided federated meta-learning architecture for edge learning (as illustrated in Figure 1), where a set \mathcal{I} of source edge nodes joint force to learn a meta-model, aided by the valuable knowledge learned from previous tasks in the cloud. Specifically, the knowledge transfer is in the form of regularization using Bregman divergence on the prior model.

3.1 Problem Formulation

For ease of exposition, we consider a general supervised learning setting where each edge node $i \in \mathcal{I} \cup \{m\}$ has a labeled dataset $\mathcal{D}_i = \{(\mathbf{x}_i^j, \mathbf{y}_i^j)\}_{j=1}^{D_i}$ with total D_i samples. Here $(\mathbf{x}_i^j, \mathbf{y}_i^j) \in \mathcal{X}_i \times \mathcal{Y}_i$ is a sample point with input \mathbf{x}_i^j and true label \mathbf{y}_i^j , and follows

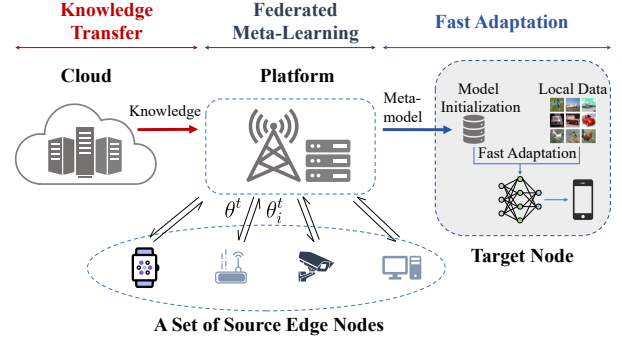


Figure 1: Illustration of the platform-aided federated meta-learning architecture with knowledge transfer.

an unknown underlying distribution P_i . For a model parameter $\phi_i \in \mathbb{R}^n$, the empirical loss function for a dataset \mathcal{D}_i is defined as $L_i(\phi_i, \mathcal{D}_i) \triangleq (1/D_i) \sum_{j=1}^{D_i} l_i(\phi_i, (\mathbf{x}_i^j, \mathbf{y}_i^j))$, where $l_i(\cdot, \cdot)$ is a general differentiable non-convex loss function. Note

Motivated by the recent success of regularization approaches in transfer learning and continual learning [27], we use regularization for extracting the valuable knowledge from the prior model to facilitate fast edge training and to alleviate catastrophic forgetting. More specially, for a model parameter $\theta \in \mathbb{R}^n$, we denote $\theta_p \in \mathbb{R}^n$ as the prior model parameter, and use the Bregman divergence $D_h(\theta, \theta_p)$ [4] as the regularization, given by:

$$D_h(\theta, \theta_p) \triangleq h(\theta) - h(\theta_p) - \langle \nabla h(\theta_p), \theta - \theta_p \rangle, \quad (1)$$

for some continuously-differentiable strictly convex function $h(\cdot)$. It is worth noting that Bregman divergence is a dissimilarity measure between two objects (e.g., vectors, matrices, distributions, etc.). It encompasses a rich class of divergence metrics, including squared Euclidean distance, squared Mahalanobis distance, Kullback-Leibler (KL) divergence, and Itakura-Saito (IS) distance, which are widely used in machine learning to encode the dissimilarity from different perspectives, and is particularly useful for the regularization approaches [16, 32, 42]. Note that while for ease of exposition, in this paper we consider the regularizer on the model parameters using Bregman divergence, the same methodology can be applied to generalize the regularization to be in terms of Bregman divergence between two functions of θ and θ_p , respectively (see Assumption 2).

Following the same line as in MAML [9], we divide the dataset \mathcal{D}_i for the source edge node $i \in \mathcal{I}$ into two disjoint sets, i.e., the support set \mathcal{D}_i^s and the query set \mathcal{D}_i^q . Based on the prior model, we can formulate the federated meta-learning with knowledge transfer among the source edge nodes as the following regularized optimization problem:

$$\begin{aligned} \min_{\theta} \quad & \sum_{i \in \mathcal{I}} w_i L_i(\phi_i(\theta), \mathcal{D}_i^q) + \lambda D_h(\theta, \theta_p) \\ \text{s.t.} \quad & \phi_i(\theta) = \theta - \alpha \nabla L_i(\theta, \mathcal{D}_i^s), \quad i \in \mathcal{I} \end{aligned} \quad (2)$$

where $w_i \triangleq D_i / \sum_{i \in \mathcal{I}} D_i$, α is the learning rate, and λ is a penalty parameter which can be used to balance the trade-off between the loss and the regularization. In this formulation, we aim to find a good meta-model such that slight updating, i.e., one-step gradient descent, results in substantial performance improvement for

any task across the edge nodes. That is, the source edge nodes collaboratively learn how to learn fast with a few data samples. Further, by penalizing changes in the model via regularization, the learned model from (2)-(3) is confined to stay ‘close’ to the prior model for enabling collaborative edge learning without forgetting prior knowledge, thus the learned meta-model can widely adapt to different types of tasks.

In the fast adaptation stage, the platform transfers the learned meta-model θ to the target node (denoted by m) after solving the regularized federated meta-learning problem (2)-(3). Based on θ , the target node m can use its local data set \mathcal{D}_m^s to quickly compute a new model ϕ_m by performing one-step stochastic gradient descent, i.e.,

$$\phi_m = \theta - \alpha \nabla L_m(\theta, \mathcal{D}_m^s). \quad (4)$$

Note that the target node also can execute a few steps of stochastic gradient descent updates for better performance when needed.

3.2 An Inexact-ADMM Based Algorithm for Regularized Federated Meta-Learning

As alluded to earlier, general gradient-based federated meta-learning approaches cannot handle the regularized optimization problem (2)-(3) well. To address this problem, we propose an inexact-ADMM based federated meta-learning algorithm (ADMM-FedMeta) to solve (2)-(3).

Observe that the federated meta-learning problem (2)-(3) is equivalent to the following constrained optimization problem:

$$\begin{aligned} \min_{\{\theta_i\}, \theta} \quad & \sum_{i \in \mathcal{I}} w_i L_i(\phi_i(\theta_i), \mathcal{D}_i^q) + \lambda D_h(\theta, \theta_p) \\ \text{s.t.} \quad & \theta_i - \theta = 0, \quad i \in \mathcal{I} \end{aligned} \quad (5)$$

where $\phi_i(\theta_i)$ is given by (3). To solve (5), we form the augmented Lagrangian function as follows:

$$\begin{aligned} \mathcal{L}(\{\theta_i, y_i\}, \theta) \triangleq \quad & \sum_{i \in \mathcal{I}} \left(w_i L_i(\phi_i(\theta_i), \mathcal{D}_i^q) + \langle y_i, \theta_i - \theta \rangle \right. \\ & \left. + \frac{\rho_i}{2} \|\theta_i - \theta\|^2 \right) + \lambda D_h(\theta, \theta_p), \end{aligned} \quad (6)$$

where $y_i \in \mathbb{R}^n$ is a dual variable and $\rho_i > 0$ is a penalty parameter for each $i \in \mathcal{I}$.

When the classical ADMM method is applied [3], the variables θ_i , θ and y_i are updated alternatively in solving (5) as follows:

$$\begin{cases} \theta^{t+1} = \arg \min_{\theta} \mathcal{L}(\{\theta_i^t, y_i^t\}, \theta), \\ \theta_i^{t+1} = \arg \min_{\theta_i} \mathcal{L}_i(\theta_i, y_i^t, \theta^{t+1}), \\ y_i^{t+1} = y_i^t + \rho_i(\theta_i^{t+1} - \theta^{t+1}), \end{cases} \quad (7)$$

where $\mathcal{L}_i(\theta_i, y_i, \theta) \triangleq w_i L_i(\phi_i(\theta_i), \mathcal{D}_i^q) + \langle y_i, \theta_i - \theta \rangle + \frac{\rho_i}{2} \|\theta_i - \theta\|^2$. The conventional ADMM decomposes the problem (5) into a set of subproblems that can be solved in parallel, while computing $D_h(\theta, \theta_p)$ and $L_i(\phi_i(\theta_i), \mathcal{D}_i^q)$ separately. Based on that, to fully take advantage of the combined computation power of the local edge nodes and the platform, we provide the following alternating updating strategy: 1) updating θ at the platform and 2) updating $\{\theta_i, y_i\}$ at the source edge nodes in a distributed manner. Particularly, in this way, the computation corresponding to the regularizer can be decoupled from the edge nodes to the platform. However,

attaining the exact solution to each subproblem is computationally costly, especially with a complex DNN model. *To tackle this challenge, we devise the inexact-ADMM based federated meta-learning (ADMM-FedMeta) below.*

Specifically, in communication round $t = 0$, the platform initializes θ^0 and sends it to edge nodes. Each node $i \in \mathcal{I}$ initializes y_i^{-1} locally.

- **Local update of $\{\theta_i, y_i\}$.** After receiving θ^t from the platform at communication round $t \in \mathbb{N}$, each edge node $i \in \mathcal{I}$ would do the following updates:

(1) *Update node-specific model ϕ_i .* Based on the dataset \mathcal{D}_i^s , ϕ_i^t is updated as:

$$\phi_i^t = \theta^t - \alpha \nabla L_i(\theta^t, \mathcal{D}_i^s). \quad (8)$$

(2) *Update local parameter θ_i .* Based on (7), given the meta-model θ^t and local dual variable y_i^{t-1} from last communication round, the local parameter θ_i should be updated as:

$$\begin{aligned} \theta_i^t = \arg \min_{\theta_i} \quad & \left\{ w_i L_i(\phi_i(\theta_i), \mathcal{D}_i^q) + \langle y_i^{t-1}, \theta_i - \theta^t \rangle \right. \\ & \left. + \frac{\rho_i}{2} \|\theta_i - \theta^t\|^2 \right\}. \end{aligned} \quad (9)$$

To simplify the computation, we use linear approximation (i.e., first-order Taylor expansion) around θ^t to relax this subproblem, i.e.,

$$\begin{aligned} \theta_i^t = \arg \min_{\theta_i} \quad & \left\{ w_i L_i(\phi_i^t, \mathcal{D}_i^q) \right. \\ & + \langle w_i (I - \alpha \nabla^2 L_i(\theta^t, \mathcal{D}_i^s)) \nabla L_i(\phi_i^t, \mathcal{D}_i^q) \\ & \left. + y_i^{t-1}, \theta_i - \theta^t \rangle + \frac{\rho_i}{2} \|\theta_i - \theta^t\|^2 \right\}, \end{aligned} \quad (10)$$

where ϕ_i^t is from (8). Nevertheless, (10) is still insufficient since the computational complexity of the Hessian-gradient product $\nabla^2 L_i(\theta^t, \mathcal{D}_i^s) \nabla L_i(\phi_i^t, \mathcal{D}_i^q)$ is $\mathcal{O}(n^2)$. To further reduce the computational cost, as in [7, 13], we replace the Hessian-gradient product by a first-order estimator, i.e.,

$$g_i^t \triangleq \frac{\nabla L_i(\theta^t + \delta_{i,t} r_i^t, \mathcal{D}_i^s) - \nabla L_i(\theta^t - \delta_{i,t} r_i^t, \mathcal{D}_i^s)}{2\delta_{i,t}}, \quad (11)$$

where $r_i^t \triangleq \nabla L_i(\phi_i^t, \mathcal{D}_i^q)$ and $\delta_{i,t} > 0$ is the degree of freedom capturing the estimation accuracy. In a nutshell, the local parameter θ_i^t is updated as follows:

$$\theta_i^t = \theta^t - \frac{y_i^{t-1} + w_i (\nabla L_i(\phi_i^t, \mathcal{D}_i^q) - \alpha g_i^t)}{\rho_i}, \quad (12)$$

where (12) is derived by the optimality of (10) after replacing $\nabla^2 L_i(\theta^t, \mathcal{D}_i^s) \nabla L_i(\phi_i^t, \mathcal{D}_i^q)$ with g_i^t .

(3) *Update local dual variable y_i .* Based on θ^t and the updated local parameter θ_i^t , the auxiliary dual variable y_i^t is next updated according to:

$$y_i^t = y_i^{t-1} + \rho_i(\theta_i^t - \theta^t). \quad (13)$$

- **Global Aggregation towards Meta-Model θ .** Each edge node $i \in \mathcal{I}$ sends the updated local parameters θ_i^t and y_i^t to the platform. With the prior model θ_p transferred from the

Algorithm 1: Inexact-ADMM Based Meta-Learning Algorithm (ADMM-FedMeta)

Input: $\theta_p, \alpha, \lambda, \rho_i, \mathcal{D}_i^s, \mathcal{D}_i^q$ for $i \in \mathcal{I}$
Output: Final meta-model θ

- 1 Each edge node $i \in \mathcal{I}$ initializes y_i^{-1} ;
- 2 Platform initializes θ^0 and sends it to all edge nodes;
- 3 **for** $t = 0$ **to** T **do**
- 4 **for** $i = 1$ **to** I **do**
- 5 Compute $\phi_i^t \leftarrow \theta^t - \alpha \nabla L_i(\theta^t, \mathcal{D}_i^s)$;
- 6 Compute θ_i^t by (12);
- 7 Compute $y_i^t \leftarrow y_i^{t-1} + \rho_i(\theta_i^t - \theta^t)$;
- 8 Send θ_i^t and y_i^t back to the platform;
- 9 **end**
- 10 Platform updates θ^{t+1} by (14) and sends it to all edge nodes $i \in \mathcal{I}$;
- 11 **end**
- 12 Platform transfers θ^T to target node for fast adaptation;

cloud, the platform performs a global update of the model initialization θ based on:

$$\theta^{t+1} = \frac{\sum_{i \in \mathcal{I}} (y_i^t + \rho_i \theta_i^t) - \lambda \nabla_{\theta^t} D_h(\theta^t, \theta_p)}{\sum_{i \in \mathcal{I}} \rho_i}, \quad (14)$$

where (14) is derived from the optimality of the linearized $\mathcal{L}(\{\theta_i^t, y_i^t\}, \theta)$ around θ^t similar to (12). Then, the platform sends θ^{t+1} back to all edge nodes for the next communication round.

- **Fast Adaptation.** After the training phase, the platform transfers the learned meta-model θ^T to the target node m . Based on θ^T , the target node performs one or a few steps of stochastic gradient descent on its own dataset to obtain a new model.

The details of ADMM-FedMeta are summarized in Algorithm 1. Note that due to linearizing all decomposed subproblems and estimating Hessian by its first-order estimation, we enable the computation complexity of ADMM-FedMeta to be $O(n)$ per round, which maintains the lowest among all existing federated meta-learning approaches.

4 PERFORMANCE ANALYSIS

In this section, we analyze the performance of ADMM-FedMeta. First, we study the convergence properties and characterize the communication complexity for ADMM-FedMeta. Then, we quantify the forgetting effect to previous tasks of the meta-model, and analyze the fast adaptation performance at the target edge node.

4.1 Convergence Analysis

For convenience, we denote the objective function of (2) as $F(\theta)$:

$$F(\theta) \triangleq \sum_{i \in \mathcal{I}} w_i L_i(\phi_i(\theta), \mathcal{D}_i^q) + \lambda D_h(\theta, \theta_p), \quad (15)$$

where $\phi_i(\theta) = \theta - \alpha \nabla L_i(\theta, \mathcal{D}_i^s)$. Next, we characterize the convergence and communication complexity of the proposed algorithm for finding a first-order stationary point of function $F(\theta)$. Formally,

the definition of an ϵ -approximate first-order stationary point is given as follows.

Definition 1 (ϵ -FOSP). A solution $\theta \in \mathbb{R}^n$ is called an ϵ -approximate first-order stationary point (ϵ -FOSP) of (2), if $\|\nabla F(\theta)\| \leq \epsilon$, for $\epsilon > 0$.

The above definition implies that if a solution θ by any algorithm is a ϵ -FOSP, then the gradient norm of the objective function is bounded above by ϵ .

Note that the first-order estimator of Hessian introduced in the subproblem (12) inevitably complicates the convergence analysis of ADMM-FedMeta, making the existing analysis methods of ADMM [1] not suitable here. To establish the convergence of ADMM-FedMeta, we impose the following standard assumptions in the literature.

Assumption 1. $F(\theta)$ is lower-bounded, i.e., $F(\theta) > -\infty$, for all $\theta \in \mathbb{R}^n$.

Assumption 2 (Smoothness and Bounded Gradient). For each $i \in \mathcal{I} \cup \{m\}$, any \mathcal{D}_i^s , and $\theta_p \in \mathbb{R}^n$, both $L_i(\cdot, \mathcal{D}_i^s)$ and $D_h(\cdot, \theta_p)$ are twice continuously differentiable and smooth, i.e., for any $x, y \in \mathbb{R}^n$, there exist constants $\mu_i > 0$ and $\mu_r > 0$ such that:

$$\|\nabla L_i(x, \mathcal{D}_i^s) - \nabla L_i(y, \mathcal{D}_i^s)\| \leq \mu_i \|x - y\|, \quad (16)$$

$$\|\nabla_x D_h(x, \theta_p) - \nabla_y D_h(y, \theta_p)\| \leq \mu_r \|x - y\|. \quad (17)$$

Besides, the gradient norms of $L_i(\cdot, \mathcal{D}_i^s)$ is bounded by a positive constant $\beta_i > 0$, i.e., for any $x \in \mathbb{R}^n$, the following holds:

$$\|\nabla L_i(x, \mathcal{D}_i^s)\| \leq \beta_i. \quad (18)$$

Assumption 3 (Lipschitz Continuous Hessian). For any $i \in \mathcal{I}$ and any \mathcal{D}_i^s , the Hessian of $L_i(\cdot, \mathcal{D}_i^s)$ is ζ_i -Lipschitz continuous, i.e., for any $x, y \in \mathbb{R}^n$, we have:

$$\|\nabla^2 L_i(x, \mathcal{D}_i^s) - \nabla^2 L_i(y, \mathcal{D}_i^s)\| \leq \zeta_i \|x - y\|. \quad (19)$$

Assumption 4 (Bounded Variance). For any $i \in \mathcal{I} \cup \{m\}$ and $\theta \in \mathbb{R}^n$, the stochastic gradient $\nabla l_i(\theta, (\mathbf{x}, \mathbf{y}))$ and Hessian $\nabla^2 l_i(\theta, (\mathbf{x}, \mathbf{y}))$ with respect to data point $(\mathbf{x}, \mathbf{y}) \in \mathcal{X}_i \times \mathcal{Y}_i$ have bounded variances, i.e.,

$$\mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim P_i} \{\|\nabla l_i(\theta, (\mathbf{x}, \mathbf{y})) - \nabla L_i(\theta)\|^2\} \leq (\sigma_i^g)^2, \quad (20)$$

$$\mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim P_i} \{\|\nabla^2 l_i(\theta, (\mathbf{x}, \mathbf{y})) - \nabla^2 L_i(\theta)\|^2\} \leq (\sigma_i^h)^2, \quad (21)$$

for some positive constants $\sigma_i^g > 0$ and $\sigma_i^h > 0$.

Assumptions 1-4 are standard in the state-of-the-art studies on the analysis of federated learning algorithms [8, 21, 45]. In particular, (18) is critical for analyzing the convergence as it enables characterizing the estimation error of the Hessian. Assumption 3 implies the high-order smoothness of $L_i(\cdot, \mathcal{D}_i^s)$ for dealing with the second-order information in the update steps of Algorithm 1. Furthermore, Assumption 4 provides the upper bounds of the variances of the gradient and Hessian estimations.

To quantify the convergence behavior of ADMM-FedMeta, we first study the properties of the objective function $F(\theta)$. Denote $F_i(\theta) \triangleq L_i(\phi_i(\theta), \mathcal{D}_i^q)$. Based on Assumptions 2 and 3, we have the following result about the smoothness of F_i as in the standard analysis of federated meta-learning approaches.

Lemma 1. Given Assumptions 2 and 3, for each $i \in \mathcal{I} \cup \{m\}$, F_i is proper and v_i -smooth, i.e.,

$$\|\nabla F_i(x) - \nabla F_i(y)\| \leq v_i \|x - y\|, \quad \forall x, y \in \mathbb{R}^n, \quad (22)$$

where v_i is defined as follows:

$$v_i \triangleq (1 + \alpha\mu_i)(1 + \mu_i)\mu_i + \alpha\beta_i\zeta_i. \quad (23)$$

PROOF. The proof is standard. The detailed proof is provided in Appendix A of the technical report [43]. \square

Next, we impose the assumptions on the hyper-parameters.

Assumption 5. For all $i \in \mathcal{I}$, ρ_i is large enough such that:

$$\frac{\rho_i}{2} - 4w_i v_i > 0, \quad (24)$$

$$\frac{\rho_i}{2} - 2w_i^2 v_i^2 \left(\frac{4w_i v_i}{\rho_i^2} + \frac{1}{\rho_i} \right) - \frac{\lambda\mu_r}{2I} > 0, \quad (25)$$

$$\rho_i - 3v_i > 0, \quad (26)$$

where v_i is a smooth scalar defined in (23). Besides, for all $i \in \mathcal{I}$, the additional degree of freedom parameter $\{\delta_{i,t}\}$ for the approximation of Hessian-gradient products is chosen to be a monotonically non-increasing positive sequence and satisfies $\sum_{t=1}^{\infty} \delta_{i,t} < \infty$.

We impose Assumption 5 on the penalty parameter ρ_i and the degree of freedom parameter $\delta_{i,t}$. Intuitively, (24)-(26) imply that a large ρ_i is required to balance the error caused by the linear approximation and Hessian estimation in (12).

Based on Lemma 1, we are ready to establish the convergence and characterize the communication complexity for Algorithm 1.

Theorem 1 (Convergence and Communication Complexity). Under Assumptions 1-5, we have the following results based on Algorithm 1:

- (i) $\{\theta^t\}$ has at least one limit point and each limit point θ^* is a stationary solution of (2), i.e., $\|\nabla F(\theta^*)\| = 0$.
- (ii) Algorithm 1 finds an ϵ -FOSP of Problem (2)-(3) after at most $O(1/\epsilon^2)$ communication rounds.

PROOF. For part (i), we first characterize the successive difference of the augmented Lagrangian function. Based on that, we next show $\lim_{t \rightarrow \infty} \|\theta_i^{t+1} - \theta_i^t\| = 0$ and $\lim_{t \rightarrow \infty} \|\theta^{t+1} - \theta^t\| = 0$. Lastly, we bound $\|\nabla F(\theta^t)\|$ via $\|\theta_i^{t+1} - \theta_i^t\|$ and $\|\theta^{t+1} - \theta^t\|$. We prove (ii) via dividing the sum of the Lagrangian successive difference into two finite parts. The detailed proof is presented in Appendix F of the technical report [43]. \square

Theorem 1 indicates that Algorithm 1 always converges to a stationary point of (2). Besides, to find an ϵ -FOSP of Problem (2)-(3), Algorithm 1 requires $O(1/\epsilon^2)$ communication rounds between edge nodes and the platform. It is worth to note that in contrast to the previous methods [7, 8, 21], ADMM-FedMeta can converge under mild conditions, i.e., not depending on the similarity assumptions (i.e., Assumption 6) across different edge nodes. This implies that Algorithm 1 can be applied to unbalanced and heterogeneous local datasets, revealing the potential in dealing with the inherent challenges in federated learning.

To characterize the impact of local data samples on the expected performance on the source nodes, we provide the following corollary.

Corollary 1. Given Assumptions 1-5, the ϵ -FOSP solution θ_ϵ found by Algorithm 1 satisfies that:

$$\begin{aligned} & \mathbb{E} \left\{ \left\| \sum_{i \in \mathcal{I}} w_i L_i(\theta_\epsilon - \alpha \nabla L_i(\theta_\epsilon)) + \lambda D_h(\theta_\epsilon, \theta_p) \right\| \right\} \\ & \leq \epsilon + \sum_{i \in \mathcal{I}} w_i \sigma_i^g \left(\frac{\alpha \mu_i}{\sqrt{D_i^s}} + \frac{1}{\sqrt{D_i^g}} \right), \end{aligned} \quad (27)$$

where $w_i = 1/I$ and $L_i(\cdot)$ is the expected loss denoted by:

$$L_i(\theta) \triangleq \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim P_i} \{l_i(\theta, (\mathbf{x}, \mathbf{y}))\}. \quad (28)$$

PROOF. The detailed proof is presented in Appendix G of the technical report [43]. \square

Corollary 1 implies that despite ϵ -FOSP of the deterministic loss (2) can be obtained within $O(1/\epsilon^2)$ rounds, a performance degradation may happen due to a small sample size.

4.2 Performance of Rapid Adaptation at Target Node

While the task similarity assumption is not required to guarantee the convergence of Algorithm 1, we impose such an assumption to study the fast adaptation performance at the target node m .

Assumption 6 (Task Similarity). There exist positive constants $\psi_i^g > 0$ and $\psi_i^h > 0$ such that for any $i \in \mathcal{I}$ and $\theta \in \mathbb{R}^n$, the following holds:

$$\|\nabla L_m(\theta) - \nabla L_i(\theta)\| \leq \psi_i^g, \quad (29)$$

$$\|\nabla^2 L_m(\theta) - \nabla^2 L_i(\theta)\| \leq \psi_i^h, \quad (30)$$

where $L_i(\cdot)$ is defined in (28), and the same applies to $L_m(\cdot)$.

Assumption 6 indicates that the variations of the gradients between the loss of source edge nodes and the target edge node are bounded by some constants, which capture the similarity of the tasks corresponding to non-IID data and holds for many practical loss functions [45], such as logistic regression and hyperbolic tangent functions. In particular, ψ_i^g and ψ_i^h can be roughly seen as a distance between data distributions P_m and P_i [7].

Next, we present the following result about the performance of rapid adaptation.

Theorem 2 (Fast Adaptation Performance). Suppose that Assumptions 1-6 hold. For any $\epsilon > 0$, the ϵ -FOSP solution θ_ϵ obtained by Algorithm 1 satisfies that:

$$\begin{aligned} \mathbb{E} \{ \|\nabla F_m(\theta_\epsilon)\| \} & \leq \epsilon + \alpha\beta_m \sum_{i \in \mathcal{I}} w_i \psi_i^h + (\alpha\mu + 1)^2 \sum_{i \in \mathcal{I}} w_i \psi_i^g \\ & \quad + \alpha\mu(\alpha\mu + 1) \sum_{i \in \mathcal{I}} w_i \sigma_i^g \left(\frac{1}{\sqrt{D_m^g}} + \frac{1}{\sqrt{D_i^g}} \right) \\ & \quad + (\alpha\mu + 1) \sum_{i \in \mathcal{I}} w_i \sigma_i^g \left(\frac{1}{\sqrt{D_m^s}} + \frac{1}{\sqrt{D_i^s}} \right) \\ & \quad + \alpha\beta_m \sum_{i \in \mathcal{I}} w_i \sigma_i^h \left(\frac{1}{\sqrt{D_m^s}} + \frac{1}{\sqrt{D_i^s}} \right), \end{aligned} \quad (31)$$

where $F_m(\theta) \triangleq L_m(\theta - \alpha \nabla L_m(\theta, \mathcal{D}_m^s), \mathcal{D}_m^g) + \lambda D_h(\theta, \theta_p)$ for any \mathcal{D}_m^s and \mathcal{D}_m^g with respect to distribution P_m , and $\mu = \max_{i \in \mathcal{I}} \{\mu_i\}$.

PROOF. The detailed proof is presented in Appendix H of the technical report [43]. \square

Theorem 2 sheds light on the performance of fast adaptation with the previous knowledge, which depends on the size of datasets, the variance of stochastic gradient and Hessian, and the similarity between target node and source nodes. In particular, if $D_i^s = \mathcal{O}(\epsilon^{-2})$ and $D_i^g = \mathcal{O}(\epsilon^{-2})$ for $i \in \mathcal{I} \cup \{m\}$, then an $\mathcal{O}(\epsilon + \sum_{i \in \mathcal{I}} w_i(\psi_i^h + \psi_i^g))$ -FOSP can be obtained at the target node. However, it is clear that the larger the dataset of source nodes dissimilar to the target node is, the worse the rapid adaptation performs. In the next subsection, we will show this issues can be alleviated via regularization with a good prior model.

4.3 Forgetting effect to Prior Knowledge

In this section, we quantify the forgetting effect of the previous task of Algorithm 1 in a special case, where the regularizer is squared Euclidean distance, i.e., $D_h(\theta, \theta_p) = \|\theta - \theta_p\|^2$. To do so, we first derive an upper bound of $D_h(\theta, \theta_p)$ via the following lemma.

Lemma 2. *Given Assumptions 1-5, for any $\epsilon > 0$, the ϵ -FOSP solution θ_ϵ obtained by Algorithm 1 satisfies that:*

$$\mathbb{E}\{D_h(\theta_\epsilon, \theta_p)\} \leq \frac{1}{\lambda} \left(\epsilon + \sum_{i \in \mathcal{I}} w_i \left(\beta_i + \frac{\alpha \mu_i \sigma_i^g}{\sqrt{D_i^s}} + \frac{\sigma_i^g}{\sqrt{D_i^g}} \right) \right) \|\theta_\epsilon - \theta_p\|. \quad (32)$$

Particularly, suppose $D_h(\theta, \theta_p)$ is strongly convex with respect to θ , i.e., there exists $M > 0$ such that for $x, y \in \mathbb{R}^n$, the following holds:

$$\langle \nabla D_h(x, \theta_p) - \nabla D_h(y, \theta_p), x - y \rangle \geq M \|x - y\|^2. \quad (33)$$

Then (32) can be written as:

$$\mathbb{E}\{D_h(\theta_\epsilon, \theta_p)\} \leq \frac{1}{M\lambda^2} \left(\epsilon + \sum_{i \in \mathcal{I}} w_i \left(\beta_i + \frac{\alpha \mu_i \sigma_i^g}{\sqrt{D_i^s}} + \frac{\sigma_i^g}{\sqrt{D_i^g}} \right) \right)^2. \quad (34)$$

PROOF. The detailed proof is presented in Appendix I of the technical report [43]. \square

For a current model parameter θ , we define the *forgetting cost* of θ on the previous task p as $L_p(\theta)$ [18], where $L_p(\cdot)$ is the expected loss over the data distribution of task p (defined by (28)). Based on Lemma 2, we next characterize the forgetting cost of the ϵ -FOSP solution.

Theorem 3. *Suppose that $L_p(\cdot)$ is μ_p -smooth, $\|\nabla L_p(\theta_p)\| \leq \epsilon_p$ for some $\epsilon_p > 0$, and $D_h(\theta, \theta_p) = \|\theta - \theta_p\|^2$. Under Assumptions 1-5, for the ϵ -FOSP solution θ_ϵ , we have the following result:*

$$\|\nabla L_p(\theta_\epsilon)\| \leq \epsilon_p + \frac{v_p}{2\lambda} \left(\epsilon + \sum_{i \in \mathcal{I}} w_i \left(\beta_i + \frac{\alpha \mu_i \sigma_i^g}{\sqrt{D_i^s}} + \frac{\sigma_i^g}{\sqrt{D_i^g}} \right) \right). \quad (35)$$

PROOF. The result can be directly obtained by Lemma 2. \square

Based on Theorem 3 and corollary 1, it is clear that by selecting a suitable λ , the regularizer enables the meta-model to learn on current task while maintaining a good performance on the previous task. On the other hand, combined with Theorem 2, Theorem 3 also implies that due to the independence of (35) and similarity conditions, a good prior reference model (e.g., with a relatively small $\|\nabla L_m(\theta_p)\|$) can effectively alleviate the significant performance degradation caused by the dissimilarity between the source nodes and the target nodes.

5 EXPERIMENTAL RESULTS

In this section, we evaluate the experimental performance of ADMM-FedMeta on different datasets and models. In particular, our experimental studies are designed to evaluate the performance of the proposed ADMM-FedMeta algorithm in challenging edge learning settings where edge nodes have limited data samples. Specifically, we assume that each source node has only tens of data samples during the training stage, and that in the testing phase, each target node has only 10-20 data samples. Clearly, edge learning in these settings is highly nontrivial, particularly for sophisticated datasets (e.g., CIFAR-100).

Datasets and models. We evaluate the performance of ADMM-FedMeta on three widely-used benchmarks, including Fashion-MNIST [41], CIFAR-10 [19], and CIFAR-100 [19]. Specifically, the data is distributed among I edge nodes as follows: (1) Each node has samples from only two random classes [21]; (2) The number of samples per node follows a discrete uniform distribution, i.e., $D_i \sim U(a, b)$ for $i \in \mathcal{I}$. Here we set $a = 20$, $b = 40$, $I = 50$ for Fashion-MNIST and CIFAR-10, and $I = 100$ for CIFAR-100. We randomly select 80% and 20% nodes as the source nodes and the target nodes respectively. For each node, we divide the local dataset into a support set and a query set (i.e., \mathcal{D}_i^s and \mathcal{D}_i^g), each with 50% of the local data. We set the meta-step stepsize as $\alpha = 0.01$, the penalty parameters $\rho = 0.3$ for Fashion-MNIST, and $\rho = 0.7$ for CIFAR-10 and CIFAR-100, where $\rho_i = \rho$ for all $i \in \mathcal{I}$. We set the regularizer as squared ℓ_2 -norm, and the degree of freedom parameter $\delta_{i,t} = 1/(10t + 100)$ with $t = 1, 2, \dots, 100$ for $i \in \mathcal{I}$. For Fashion-MNIST, we use a convolutional neural network (CNN) with max-pooling operation and Exponential Linear Unit (ELU) activation function, which contains two convolutional layers with size 32 and 64 followed by a fully connected layer and softmax. The strides are set as 1 for convolution operation and 2 for pooling operation. For CIFAR-10 and CIFAR-100, we use a CNN containing three convolutional layers with size 32, 64, and 128, and a CNN containing four convolutional layers with size 32, 64, 128, and 256, respectively, while keeping all the other setup the same as that in Fashion-MNIST.

Implementation. We implement the code in TensorFlow Version 1.14 on a server with two Intel® Xeon® Golden 5120 CPUs and one Nvidia® Tesla-V100 32G GPU. Please refer to <https://github.com/XinJiang1994/HFmaml> for full details.

Baselines. We consider two existing baseline algorithms, i.e., FedAvg [24] and Per-FedAvg [8] with one or multiple local update steps. For the sake of fair comparison, we test different hyperparameters of Per-FedAvg from $\{0.001, 0.005, 0.01, 0.05, 0.1\}$ (i.e., β

Dataset	# local updates	FedAvg	Per-FedAvg	ADMM-FedMeta
Fashion-MNIST	1	83.99% \pm 3.90%	87.55% \pm 2.42%	95.69% \pm 0.37%
	5	88.86% \pm 1.57%	89.65% \pm 3.26%	N/A
	10	85.29% \pm 1.93%	90.95% \pm 2.71%	N/A
CIFAR-10	1	41.97% \pm 1.33%	60.53% \pm 1.12%	74.61% \pm 2.19%
	5	56.58% \pm 2.27%	65.93% \pm 9.97%	N/A
	10	56.58% \pm 1.15%	67.43% \pm 0.99%	N/A
CIFAR-100	1	42.35% \pm 1.55%	48.19% \pm 2.18%	63.56% \pm 0.87%
	5	50.00% \pm 1.09%	49.56% \pm 1.09%	N/A
	10	49.97% \pm 1.04%	48.73% \pm 1.23%	N/A

Table 1: Comparison of the accuracy on target nodes of different algorithms.

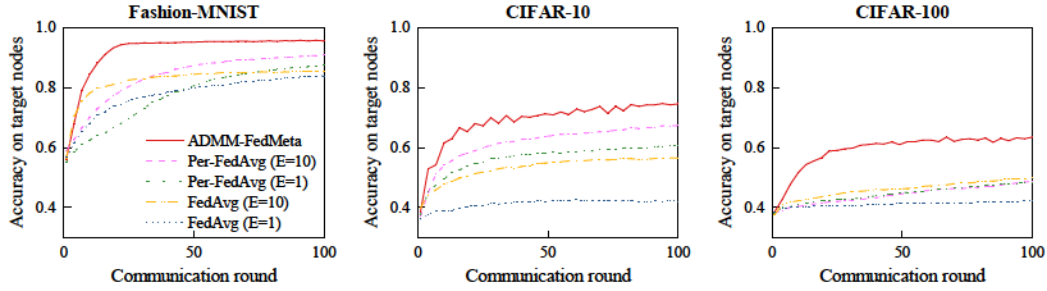


Figure 2: Comparison of the convergence speed of different algorithms.

in [8, Algorithm 1]), and select the best for the experiments, i.e., 0.005 for Fashion-MNIST and CIFAR-10, and 0.001 for CIFAR-100.

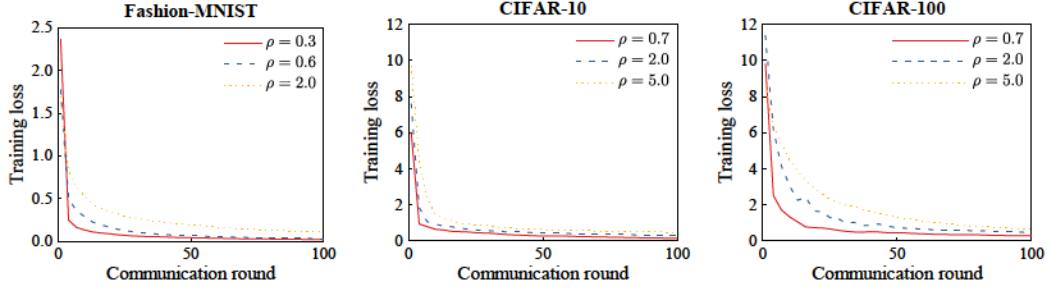
To demonstrate the impact of the knowledge transfer and the inexact-ADMM based methods respectively, we first remove the regularization term (i.e., letting $\lambda = 0$) and compare the convergence speed and adaptation performance between ADMM-FedMeta and the baselines. Then, we conduct the experiment under using a prior model for regularization, and show the performance improvement in terms of convergence, adaptation, and forgetting effect.

Performance and computational efficiency. To be fair, we set $\lambda = 0$ to remove the benefit of using the regularization in ADMM-FedMeta. We repeat the experiments for 10 times, then show the comparison of the accuracy along with 95% confidence intervals in Table 1. We have the following observations. (1) ADMM-FedMeta substantially outperforms Per-FedAvg and FedAvg, especially on sophisticated datasets. Specifically, ADMM-FedMeta achieves 7.7% over FedAvg and 5.2% over Per-FedAvg on Fashion-MNIST, 31.87% over FedAvg and 10.65% over Per-FedAvg on CIFAR-10, and 27.12% over FedAvg and 28.25% over Per-FedAvg on CIFAR-100. Note that the computation costs of each local update are $O(n)$, $O(n^2)$, and $O(n)$ for FedAvg, Per-FedAvg, and ADMM-FedMeta, respectively. This performance improvement clearly indicates that ADMM-FedMeta is more computationally efficient on non-convex loss and heterogeneous datasets with a small number of data samples. (2) It should be noted that the gaps between ADMM-FedMeta and the two baselines on CIFAR-10 are smaller when the number of local updates increases. The underlying rationale is that with more local update steps, the number of the overall iterations in the two baseline algorithms actually increases, thus resulting in a better model. However,

a large number of local update steps would lead to high computational cost and may cause failure to convergence [8, Theorem 4.5].

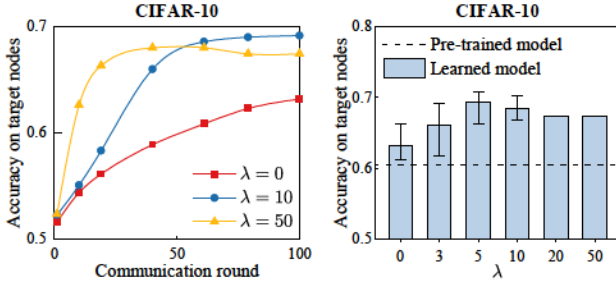
Convergence speed and hyper-parameter. As shown in Figure 2, ADMM-FedMeta converges significantly faster than the existing methods, often requiring tens of rounds to obtain a high-quality meta-model, which indicates ADMM-FedMeta can achieve a great communication efficiency. Besides, Figure 2 also suggest that despite the sample size is small, edge nodes can obtain a satisfactory model via federated meta-learning with only one-step stochastic gradient descent. Further, we investigate the impact of the hyper-parameter ρ on the convergence of ADMM-FedMeta (we let ρ_i be the same across different nodes). It can be seen from Figure 3 that ADMM-FedMeta has a relative faster convergence speed with a smaller ρ in terms of the training loss. In particular, a small change of ρ does not greatly affect the convergence properties of the algorithm, which implies that ADMM-FedMeta is robust to the hyper-parameters.

Forgetting effect to the previous task. To demonstrate the forgetting effect to the prior task on different algorithms, we pre-train a model θ_p with satisfactory performance on the data of the first five classes as a prior task, and then use θ_p as the initialization to train the meta-models by different algorithms on the data of the last five classes as a new task. Similarly, over CIFAR-100, we use the first fifty classes and the last fifty classes as the prior and new tasks respectively. After that, We test the adaptation performance of the meta-models on the prior and new tasks to show the forgetting effect. We set $\lambda = 0.5$ for Fashion-MNIST, $\lambda = 1$ for CIFAR-10 and CIFAR-100, and use squared Euclidean distance as the regularizer.

Figure 3: Impact of penalty parameter ρ .

Dataset	Task	Prior Model	FedAvg	Per-FedAvg	ADMM-FedMeta
Fashion-MNIST	Prior	95.63%	41.27%	49.60%	92.86%
	New	49.21%	94.05%	94.84%	94.04%
CIFAR-10	Prior	75.74%	41.08%	42.08%	62.38%
	New	17.33%	55.45%	47.03%	71.29%
CIFAR-100	Prior	66.27%	35.32%	37.62%	59.52%
	New	45.63%	40.48%	57.92%	63.10%

Table 2: Comparison of the accuracy on prior and current tasks.

Figure 4: Impact of θ_p and λ .

As illustrated in Table 2, the existing methods suffer from the catastrophic forgetting on the previous task due to lack of mechanisms to extract the knowledge from the prior model. Clearly, ADMM-FedMeta can effectively mitigate this issue via a regularization with prior model, while also maintaining a satisfactory performance on the new task.

Impact of prior knowledge. To quantify the impact of the knowledge transfer on the convergence of model training and the adaptation performance of target nodes, we pre-train a prior model as θ_p using images of 3-10 classes on CIFAR-10. Then, we train the meta-model θ on source nodes with images of 1-8 classes and test the accuracy on the target nodes with all 1-10 classes. In this way, the pre-trained model can be considered as containing valuable knowledge of the current task. As shown in Figure 4, with the useful knowledge transferred from the previous task to the edge nodes, faster convergence and higher adaptation performance are achieved by appropriately selecting the trade-off parameter λ . In other words, the regularization with useful knowledge transfer can help get a high-quality meta-model and achieve faster edge learning.

6 CONCLUSION

In this paper, we presented an inexact-ADMM based federated meta-learning approach for fast and continual edge learning. More specifically, we first proposed a platform-aided federated meta-learning architecture enabling edge nodes to collaboratively learn a meta-model with the knowledge transfer of previous tasks. We cast the federated meta-learning problem as a regularized optimization problem, where the previous knowledge is extracted as regularization. Then, we devised an ADMM based algorithm, namely ADMM-FedMeta, in which the original problem is decomposed into many subproblems which can be solved in parallel across edge nodes and the platform. Further, we developed a variant of inexact-ADMM method to reduce the computational cost per round to $O(n)$ via employing linear approximation as well as Hessian estimation. We provided a comprehensive analysis and empirical results to demonstrate the effectiveness and efficiency of ADMM-FedMeta. The advantages of the proposed algorithm are summarized as follows: First, it can decouple the regularizer from edge nodes to the platform, which helps to alleviate the local computational cost while exploiting the resources between local devices and the server effectively. Besides, by inexact-ADMM technique, we further reduce the computational complexity during local update and global aggregation. We show that ADMM-FedMeta can converge under mild conditions, particularly, with weak *task similarity* assumptions. Lastly, empirical results show that ADMM-FedMeta converges faster than existing benchmark algorithms.

There are a number of interesting questions and directions for future work. First, it is of interest to incorporate the experience replay method and parameter isolation approaches into the ADMM-FedMeta to further mitigate the catastrophic forgetting. Secondly, despite ADMM-FedMeta can be directly applied to reinforcement learning with policy gradient, it may lead to poor sample efficiency. It remains largely open to develop efficient collaborative reinforcement learning for edge learning. Moreover, our experimental results indicate that even without the regularization term, in practice ADMM-FedMeta can still converge faster than the existing gradient-based methods, especially on small sample sizes. It is intriguing to get a more deep understanding of this phenomenon.

7 ACKNOWLEDGMENTS

This research was supported in part by NSF under Grants CNS-2003081 and CPS-1739344, National Key R&D Program of China

under Grant No. 2019YFA0706403, National Natural Science Foundation of China under Grants No. 62072472, 61702562 and U19A2067, Natural Science Foundation of Hunan Province, China under Grant No. 2020JJ2050, 111 Project under Grant No. B18059, the Young Elite Scientists Sponsorship Program by CAST under Grant No. 2018QNRC001, the Young Talents Plan of Hunan Province of China under Grant No. 2019RS2001, and also financially supported by China Scholarship Council (CSC).

REFERENCES

- [1] Rina Foygel Barber and Emil Y Sidky. 2020. Convergence for nonconvex ADMM, with applications to CT imaging. *arXiv preprint arXiv:2006.07278* (2020).
- [2] Pablo Barros, German I Parisi, Di Fu, Xun Liu, and Stefan Wermter. 2018. Expectation learning for adaptive crossmodal stimuli association. *arXiv preprint arXiv:1801.07654* (2018).
- [3] Stephen Boyd, Neal Parikh, and Eric Chu. 2011. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc.
- [4] Lev M Bregman. 1967. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics* 7, 3 (1967), 200–217.
- [5] Fei Chen, Mi Luo, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. 2018. Federated meta-learning with fast convergence and efficient communication. *arXiv preprint arXiv:1802.07876* (2018).
- [6] Liam Collins, Aryan Mokhtari, and Sanjay Shakkottai. 2020. Distribution-Agnostic Model-Agnostic Meta-Learning. *arXiv preprint arXiv:2002.04766* (2020).
- [7] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *International Conference on Artificial Intelligence and Statistics*. 1082–1092.
- [8] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948* (2020).
- [9] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400* (2017).
- [10] Chelsea Finn, Kelvin Xu, and Sergey Levine. 2018. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*. 9516–9527.
- [11] Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences* 3, 4 (1999), 128–135.
- [12] Mingyi Hong, Zhi-Quan Luo, and Meisam Razaviyayn. 2016. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization* 26, 1 (2016), 337–364.
- [13] Kaiyi Ji, Junjie Yang, and Yingbin Liang. 2020. Multi-Step Model-Agnostic Meta-Learning: Convergence and Improved Algorithms. *arXiv preprint arXiv:2002.07836* (2020).
- [14] Bo Jiang, Tianyi Lin, Shiqian Ma, and Shuzhong Zhang. 2019. Structured nonconvex and nonsmooth optimization: algorithms and iteration complexity analysis. *Computational Optimization and Applications* 72, 1 (2019), 115–157.
- [15] Yihan Jiang, Jakub Konečný, Keith Rush, and Sreeram Kannan. 2019. Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488* (2019).
- [16] Heechul Jung, Jeongwoo Ju, Minju Jung, and Junmo Kim. 2016. Less-forgetting learning in deep neural networks. *arXiv preprint arXiv:1607.00122* (2016).
- [17] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114, 13 (2017), 3521–3526.
- [18] R Krishnan and Prasanna Balaprakash. 2020. Meta Continual Learning via Dynamic Programming. *arXiv preprint arXiv:2008.02219* (2020).
- [19] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).
- [20] Alessandro Lanza, Serena Morigi, Ivan Selesnick, and Fiorella Sgallari. 2017. Nonconvex nonsmooth optimization via convex-nonconvex majorization-minimization. *Numer. Math.* 136, 2 (2017), 343–381.
- [21] Sen Lin, Guang Yang, and Junshan Zhang. 2020. A Collaborative Learning Framework via Federated Meta-Learning. *arXiv preprint arXiv:2001.03229* (2020).
- [22] David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. In *Advances in neural information processing systems*. 6467–6476.
- [23] Sindre Magnússon, Pradeep Chaturanga Weeraddana, Michael G Rabbat, and Carlo Fischione. 2015. On the convergence of alternating direction lagrangian methods for nonconvex structured optimization problems. *IEEE Transactions on Control of Network Systems* 3, 3 (2015), 296–309.
- [24] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*. 1273–1282.
- [25] Mahesh Chandra Mukkamala, Peter Ochs, Thomas Pock, and Shoham Sabach. 2020. Convex-Concave Backtracking for Inertial Bregman Proximal Gradient Algorithms in Nonconvex Optimization. *SIAM Journal on Mathematics of Data Science* 2, 3 (2020), 658–682.
- [26] Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999* (2018).
- [27] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks* 113 (2019), 54–71.
- [28] Jihong Park, Sumudu Samarakoon, Mehdi Bennis, and Mérouane Debbah. 2019. Wireless network intelligence at the edge. *Proc. IEEE* 107, 11 (2019), 2204–2239.
- [29] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. 2019. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157* (2019).
- [30] Sachin Ravi and Hugo Larochelle. 2016. Optimization as a model for few-shot learning. (2016).
- [31] Jonathan Schwarz, Jelena Luketina, Wojciech M Czarnecki, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. 2018. Progress & compress: A scalable framework for continual learning. *arXiv preprint arXiv:1805.06370* (2018).
- [32] Si Si, Dacheng Tao, and Bo Geng. 2009. Bregman divergence-based regularization for transfer subspace learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 7 (2009), 929–942.
- [33] Xingyou Song, Wenbo Gao, Yuxiang Yang, Krzysztof Choromanski, Aldo Pacchiano, and Yunhao Tang. 2019. Es-maml: Simple hessian-free meta learning. *arXiv preprint arXiv:1910.01215* (2019).
- [34] Masaaki Takada and Hironori Fujisawa. 2020. Transfer Learning via ℓ_1 Regularization. *Advances in Neural Information Processing Systems* 33 (2020).
- [35] Fenghui Wang, Wenfei Cao, and Zongben Xu. 2018. Convergence of multi-block Bregman ADMM for nonconvex composite problems. *Science China Information Sciences* 61, 12 (2018), 122101.
- [36] Fenghui Wang, Zongben Xu, and Hong-Kun Xu. 2014. Convergence of Bregman alternating direction method with multipliers for nonconvex composite problems. *arXiv preprint arXiv:1410.8625* (2014).
- [37] Junxiang Wang, Fuxun Yu, Xiang Chen, and Liang Zhao. 2019. Admm for efficient deep learning with global convergence. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 111–119.
- [38] Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. 2020. On the Global Optimality of Model-Agnostic Meta-Learning. *arXiv preprint arXiv:2006.13182* (2020).
- [39] Yu Wang, Wotao Yin, and Jinshan Zeng. 2019. Global convergence of ADMM in nonconvex nonsmooth optimization. *Journal of Scientific Computing* 78, 1 (2019), 29–63.
- [40] Qiong Wu, Kaiwen He, and Xu Chen. 2020. Personalized federated learning for intelligent iot applications: A cloud-edge based framework. *IEEE Computer Graphics and Applications* (2020).
- [41] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017).
- [42] Dong Yu, Kaisheng Yao, Hang Su, Gang Li, and Frank Seide. 2013. KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 7893–7897.
- [43] Sheng Yue, Ju Ren, Jiang Xin, Sen Lin, and Junshan Zhang. 2020. Inexact-ADMM based Federated Meta-Learning for Fast and Continual Edge Learning. *arXiv:2012.08677* [cs.LG]
- [44] Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. *Proceedings of machine learning research* 70 (2017), 3987.
- [45] Xinwei Zhang, Mingyi Hong, Sairaj Dhople, Wotao Yin, and Yang Liu. 2020. FedPD: A Federated Learning Framework with Optimal Rates and Adaptivity to Non-IID Data. *arXiv preprint arXiv:2005.11418* (2020).
- [46] Zhaofeng Zhang, Sen Lin, Mehmet Dedeoglu, Kemi Ding, and Junshan Zhang. 2020. Data-driven Distributionally Robust Optimization for Edge Intelligence. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2619–2628.
- [47] Wenbo Zheng, Lan Yan, Chao Gou, and Fei-Yue Wang. 2020. Federated Meta-Learning for Fraudulent Credit Card Detection. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)* (2020).
- [48] Zhi Zhou, Xu Chen, En Li, Liekang Zeng, Ke Luo, and Junshan Zhang. 2019. Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proc. IEEE* 107, 8 (2019), 1738–1762.