FedMM: A Communication Efficient Solver for Federated Adversarial Domain Adaptation

Yan Shen*
University at Buffalo
Buffalo, New York, USA
yshen22@buffalo.edu

Jian Du*
TikTok Inc.
San Jose, California, USA
dujianeee@gmail.com

Han Zhao
University of Illinois Urbana
Champaign
Urbana, Illinois, USA
hanshao@illinois.edu

Zhanghexuan Ji
University at Buffalo
Buffalo, New York, USA
zhanghex@buffalo.edu

Chunwei Ma
University at Buffalo
Buffalo, New York, USA
chunweim@buffalo.edu

Mingchen Gao
University at Buffalo
Buffalo, New York, USA
mgao8@buffalo.edu

ABSTRACT
Federated adversary domain adaptation is a unique distributed minimax training task due to the heterogeneous data among different local clients, where each client only sees a subset of the data that merely belongs to either the source or target domain. Despite the extensive research in distributed minimax optimization, existing communication efficient solvers that exploit multiple steps of the local update are still not able to generate satisfactory solutions for federated adversarial domain adaptation because of the gradient divergence issue among clients. To tackle this problem, we propose a distributed minimax optimizer, referred to as FedMM, by introducing dual variables to bridge the gradient gap among clients. This algorithm is effective even in the extreme case where each client has different label classes and some clients only have unlabeled data. We prove that FedMM admits benign convergence to a stationary point under domain-shifted unlabeled data. On a variety of benchmark datasets, extensive experiments show that FedMM consistently achieves both better communication savings and significant accuracy improvements over existing federated optimizers based on the stochastic gradient descent ascent (SGDA) algorithm. When training from scratch, for example, it outperforms other SGDA based federated average methods by around 20% in accuracy over the same communication rounds; and it consistently outperforms when training from pre-trained models.

KEYWORDS
Federated Learning; Domain Adaptation; Adversarial Learning

1 INTRODUCTION
Federated Learning (FL) is gaining popularity because it enables multiple clients to train machine learning models without directly sharing the potentially sensitive data with other clients [11, 13].

A typical FL training pipeline involves exchanging local model parameters with a centralized server to update the global model parameter, and its communication overhead has been, in many cases, identified as the bottleneck [3, 20] of the training pipeline. Moreover, domain shift often exists between clients’ data [23], which is another inherent characteristic of FL training, where the data are sampled from different parts of the sample space on different clients. Because of the aforementioned unique features, FL training needs efficient optimizers that converge over heterogeneous data among clients with fewer communication rounds.

For data with distributional shifts, one of the most challenging settings is that each local client only has access to a subset of the label classes in order to train the global/common model. In this situation, the global model’s accuracy suffers considerably as a result of the gradient/model drift [20]. In the literature of domain adaptation, this problem is also known as label shift [29, 38]. Under the setting of FL, however, label shift is ubiquitous due to the imbalance between clients’ label distributions, with the extreme case being that individual clients have disjoint labels or only unlabeled data. While domain adversarial training is a classic technique in centralized settings [7, 31, 43], the distributed nature of FL makes the direct application of this line of approaches particularly challenging.

One method is to use the stochastic gradient descent ascent (SGDA) method [15] directly as if the data are homogeneous and centralized, where data are aggregated together to find the saddle point solutions [10, 16] of a minimax problem. However, because of the potential domain shifts among clients in FL settings, a single client cannot access an unbiased sample of the gradient from the global objective function. A natural solution would be to average each client’s gradients, which exactly corresponds to the FedSGDA approach in [22]. Its training efficiency, on the other hand, is low due to the requirement of large communication rounds between the server and clients. Without considering the issue of domain shift, there are several works on communication-efficient FL algorithms. A large spectrum of these algorithms are variations of the classic FedAvg [20]. Following this pipeline, a natural extension of
FedMM. In light of the above challenges, to optimize a federated minimax objective, we formulate this distributed saddle point optimization as a Federated MiniMax (FedMM) optimization on a sum of non-identical distributions. In particular, we use an augmented Lagrange function to enforce the global model consensus constraints. In each client’s local optimization oracle, FedMM deconstructs the global sum by solving the augmented Lagrange of each function individually in a finite number of steps. The collection of Lagrange multipliers is updated to enforce the global model consensus constraints.

Contributions:
(i) We present, FedMM, a stochastic federated optimizer tailored for federated minimax optimizations with non-separable minimization and maximization variables, as well as clients with imbalanced label class distributions. FedMM is effective in the extreme case where each client has disjoint classes of labels or unlabeled data.
(ii) Under the generic federated saddle point optimization problem with a nonconvex-concave global objective function assumption, we prove that FedMM converges asymptotically to a stationary point for the nonconvex-strongly-concave setting under local update residual errors and distribution shifts.
(iii) Empirically, we show that FedMM consistently achieves either significant communication savings or accuracy improvements over the federated SGDA method on a variety of benchmark datasets with varying adversarial domain adaptation networks. For example, when training from scratch, it outperforms other SGDA based federated average methods by around 20% in accuracy over the same communication rounds.

2 PRELIMINARIES
2.1 Adversarial Domain Adaptation
Domain adaptation refers to the process of transferring knowledge from a labeled source domain to an unlabeled target domain [2, 42]. Let $\mathcal{P}$ and $\mathcal{Q}$ be the source and target distributions, respectively. In a general formulation, the upper bound of the target prediction error is given by Ben-David et al. [2]

$$
\text{err}_{\mathcal{Q}}(\xi) \leq \text{err}_{\mathcal{P}}(\xi) + d_{\mathcal{H}}(\mathcal{P}, \mathcal{Q}) + \min_{\xi} \{\text{err}_{\mathcal{P}}(\xi) + \text{err}_{\mathcal{Q}}(\xi)\},
$$

where $\text{err}_{\mathcal{Q}}(\xi)$ denotes the population loss of $\xi$ under the target distribution $\mathcal{Q}$, i.e., $\text{err}_{\mathcal{Q}}(\xi) = \mathbb{E}_{(x,y) \sim \mathcal{Q}}[\ell(\xi(x), y)]$, and we use the parallel notation $\text{err}_{\mathcal{P}}(\xi)$ for the source domain error. Besides, $d_{\mathcal{H}}(\mathcal{P}, \mathcal{Q})$ is a discrepancy-based distance, known as the $\mathcal{H}$-divergence, and $\min_{\xi} \{\text{err}_{\mathcal{P}}(\xi) + \text{err}_{\mathcal{Q}}(\xi)\}$ is the optimal joint error, i.e., the sum of source and target domain’s population loss of $\xi$ in a hypothesis class $\mathcal{F}$. For the unsupervised domain adaptation problem, it has been proven that minimizing the upper bound, which is the l.h.s in (1), leads to an architecture consisting of a feature extractor parameterized by $\omega$, i.e., $\xi_{\omega}$, a label predictor, parameterized also by $\omega$, i.e., $\hat{\psi}_{\omega}$, and a domain classifier parameterized by $\psi$, i.e., $h_\psi$, as shown in Fig 2 [6, 43]. The feature extractor generates the domain-independent feature representations, which are then fed into the domain classifier and label predictor. The domain classifier then tries to determine whether the extracted features belong to the source or target domain. Meanwhile, the label predictor predicts instance labels based on the extracted features of the labeled source-domain instances.

Minimizing the upper bound in (1) encourages the extracted features to be both discriminative and invariant to changes between the source and target domains. The upper bound minimization corresponding to a saddle point over the parameter space of $\omega$ and $\psi$ has been demonstrated using $\hat{\omega} = \arg\min_{\omega} L_1(\omega) - vL_2(\omega, \hat{\psi})$ and $\hat{\psi} = \arg\min_{\psi} L_2(\hat{\omega}, \psi)$ with an equivalent minimax form as

$$
\min_{\omega} \max_{\psi} F(\omega, \psi) = \min_{\omega} \max_{\psi} L_1(\omega) - vL_2(\omega, \psi).
$$

In the majority of adversarial domain adaptation problems, $L_1(\omega) = \mathbb{E}_{(x,y) \sim \mathcal{Q}}[\ell(\xi_\omega(x), y)]$ is the supervised learning loss on $\xi_\omega$.

\[1\] The parameters of $\xi^1$ and $\zeta^1$ are not the same. In this case, we abuse the notation to simplify the expression.
where \( h_y \) is the feature and \( h_y \cdot \) : \( \mathbb{R}^D \rightarrow \{0, 1\} \) is the probabilistic prediction of the domain label. In general, \( \zeta''_i \) and \( h_y \cdot \), which include, but is not limited to, the following cases: (i) Domain-Adversarial Neural Networks (DANN) [6]: In DANN, the input of \( h_y \cdot \) is designed simply to be the domain invariant feature \( \zeta''_i \), i.e., \( h_y \cdot (\zeta''_i(x_i)) \). (ii) Margin Disparity Discrepancy (MDD) [41]: In the MDD, the input of \( h_y \cdot \) is the concatenation of \( \zeta''_i \) and \( \arg\max_c \zeta''_i(x_i; c) \) with \( c \) the class type i.e., \( h_y \cdot \{\zeta''_i(x_i), \arg\max_c \zeta''_i(x_i; c)\} \). (iii) Conditional Domain Adaptation Network (CDAN) [17]: In CDAN, the input of \( h_y \cdot \) is from the cross-product space of \( \zeta''_i(x_i) \) and \( \zeta''_i(x_i) \), i.e., \( h_y \cdot (\zeta''_i(x_i) \otimes \zeta''_i(x_i)) \).

Our FedMM is a generic federated adversarial domain adaptation framework in which each client is equipped with \( h_y \cdot \) and \( \zeta''_i \) depending on the availability of source data, target data, or both.

The objective function in an adversarial domain adaptation problem is determined by whether the data is from the source domain or the target domain, i.e.,

\[
F_i(\omega, \psi; \xi^{(1)}_j) = \begin{cases} 
\ell(\zeta''_i(x_i), y_i) + \nu \log(1 - h_y \cdot (\zeta''_i(x_i))), & \text{if } \xi_j \in P, \\
\nu \log(h_y \cdot (\zeta''_i(x_i))), & \text{if } \xi_j \in Q.
\end{cases}
\]  

(3)

2.2 Federated Learning under Domain Shifts

We focus on the cross-silo FL adversarial domain adaptation problem, in which the training dataset is distributed across silos in a multi-organizational context, such as in healthcare, banking, finance and so on, where institutions hold users’ data but cannot share it directly with other institutions for collaborative learning.

As a result, federated adversarial domain adaptation addresses the problem by training a model among clients from a labeled source domain to an unlabeled target domain. A centralized server coordinates between the clients to solve the learning task. To express the federated adversarial domain adaptation objective, we convert the joint learning objective of (2) into the form of a centralized average of all the clients’ objective functions, as given by

\[
\min_{\omega} \max_\psi f(\omega, \psi) \triangleq \min_{\omega} \max_\psi \frac{1}{N} \sum_{i=1}^{N} f_i(\omega, \psi)
\]

(4)

where \( N \) is the number of clients, \( f_i(\omega, \psi) \) is the loss function at the \( i \)-th client, \( a_i \) is the weight coefficient, and \( F_i(\omega, \psi; \xi_j) \) is the loss function w.r.t. the data point \( \xi_j^{(i)} \triangleq (x_j, y_j) \) with specific form determined by whether the data is from the source domain or the target domain.

This novel problem structure introduces several unique challenges in federated adversarial domain adaptation that do not exist in existing adversarial domain adaptation problems or the FL literature: (i) Clients are restricted to compute the minimax optimization in a distributed manner rather than the centralized minimax optimization. (ii) To train a centralized model, both the set of feature extractor variables \( \omega \) and domain classifier variables \( \psi \) are non-separable cross-clients.

(iii) The marginal label distributions are class-imbalanced across clients due to the imbalanced distribution of source domain data and target domain data. In extreme cases, each client may only have access to data from the target domain or the source domain, but not both.

To address the above unique challenges in federated domain adaptation, we propose FedMM, a general algorithm that works for minimax optimization under FL. FedMM is designed for imbalanced label classes among clients in federated minimax training, a unique problem in domain adaptation.

3 FEDMM ALGORITHM

In this section, we look at the federated minimax problem by reformulating the centralized problem in (4) into the federated saddle-point optimization problem with consensus constraints given by

\[
\min_{\omega_i, \psi_i} \max_{\psi} f(\omega, \psi) = \frac{1}{N} \sum_{i=1}^{N} f_i(\omega_i, \psi_i)
\]

s.t. \( \omega_i = \omega_0 \), \( \psi_i = \psi_0 \), \( \forall i \in \{1,\ldots,N\} \).

(5)

The corresponding augmented Lagrangian form for each client is defined as

\[
L_i(\omega_i, \psi_i, \lambda_i, \psi_0, \psi_i, \beta_i) \triangleq f_i(\omega_i, \psi_i) + \langle \lambda_i, \omega_i - \omega_0 \rangle + \frac{\mu_1}{2} \|\omega_i - \omega_0\|^2 + \frac{\mu_2}{2} \|\psi_i - \psi_0\|^2.
\]

(6)

The centralized optimization problem in (4) is then transformed into a saddle-point minimax optimization of augmented Lagrangian functions over all primal-dual pairs, i.e., \( \{\omega_0, \omega_i, \lambda_i, \psi_i, \psi_0, \beta_i\} \) for all clients \( i \in \{1,\ldots,N\} \):

\[
\min_{\omega_0, \omega_i} \max_{\psi_i} \frac{1}{N} \sum_{i=1}^{N} L_i(\omega_0, \omega_i, \lambda_i, \psi_i, \psi_i, \beta_i) \triangleq \min_{\omega_0} \max_{\psi_0, \psi_i} \frac{1}{N} \sum_{i=1}^{N} L_i(\omega_0, \omega_i, \psi_i, \lambda_i, \psi_i, \beta_i).
\]

(7)

By fixing the global consensus variables \( \{\omega_0, \psi_0\} \), the above problem is separable w.r.t local pairs \( \{\omega_i, \lambda_i, \beta_i\} \) for all \( i \in \{1,\ldots,N\} \). And the decomposed task could be independently updated on local clients periodically without global communication. The only problem left is to align the update of global consensus \( \omega_0, \psi_0 \) and local updates \( \omega_i, \psi_i \) for all \( i \in \{1,\ldots,N\} \). Next, we demonstrate how to achieve distributed local updates and align local updates with global consensus. By substituting (6) into (7), we obtain the augmented Lagrangian functions over all primal-dual parameters:

\[
\min_{\omega_0, \psi_0} \max_{\psi_i} \frac{1}{N} \sum_{i=1}^{N} f_i(\omega_i, \psi_i) + \langle \lambda_i, \omega_i - \omega_0 \rangle + \frac{\mu_1}{2} \|\omega_i - \omega_0\|^2 + \frac{\mu_2}{2} \|\psi_i - \psi_0\|^2.
\]

(8)

where \( \mu_1 \) and \( \mu_2 \) are the penalty parameters. The minimax optimization w.r.t. the global consensus variable \( \omega_0 \) and \( \psi_0 \) is given...
by:

\[
\tilde{\omega}_0 = \arg\min_{\omega_0} \frac{1}{N} \sum_{i=1}^{N} \left( f_i(\omega_i, \psi_i) + \langle \lambda_i, \omega_i - \omega_0 \rangle + \frac{\mu_1}{2} \|\omega_i - \omega_0\|_2^2 - \langle \beta_i, \psi_i - \psi_0 \rangle - \frac{\mu_2}{2} \|\psi_i - \psi_0\|_2^2 \right)
\]

where the closed-form solution is due to the quadratic optimization. Similarly, we obtain

\[
\tilde{\psi}_0 = \frac{1}{N} \sum_{i=1}^{N} \left( \psi_i + \frac{1}{\mu_2} \beta_i \right).
\]

Eqn. (9) and (10) provide guidance for local update alignment with global consensus. More specifically, in each round, we optimize each client’s individual \(\omega_i\) and \(\psi_i\), by fixing the local consensus constraints \((\omega_0, \psi_0)\) and dual parameters \((\lambda_i, \beta_i)\). Taking the \((t+1)\)-th round update for example. Client \(i\) receives the global parameters \((\omega_0, \psi_0)\) from the server and sets local parameters \(\omega_i^m = \tilde{\omega}_0, \psi_i^m = \tilde{\psi}_0\). Then, the local saddle-point optimization of (8) w.r.t. \((\omega_i, \psi_i)\) is updated by multiple-step SGDA to reduce the communication rounds between a client and the server:

\[
\begin{align*}
\omega_i^{m+1} &= \omega_i^m - \eta_i \nabla_{\omega_i} f_i(\omega_i^m, \psi_i^m) + \lambda_i (\omega_i^m - \omega_0^0) + \lambda_i^1 \\
\psi_i^{m+1} &= \psi_i^m + \eta_i \nabla_{\psi_i} f_i(\omega_i^m, \psi_i^m) - \beta_i (\psi_i^m - \psi_0^0) - \beta_i^1
\end{align*}
\]

where \(m \in \{M_t\}\). We denote \(\omega_i^{t+1} = \tilde{\omega}_i^m\) and \(\psi_i^{t+1} = \tilde{\psi}_i^m\) for the results of \(M_t\)-th step local update. The dual parameters are then updated using SGDA by

\[
\begin{align*}
\lambda_i^{t+1} &= \lambda_i^t + \mu_1 (\omega_i^{t+1} - \omega_0^0), \\
\beta_i^{t+1} &= \beta_i^t + \mu_2 (\psi_i^{t+1} - \psi_0^0).
\end{align*}
\]

To align with the global consensus constraint obtained in (9) and (10), we set

\[
\begin{align*}
\omega_i^{t+1} &= \omega_i^{t+1} + \frac{\eta_i}{\mu_1} \lambda_i^{t+1}, \\
\psi_i^{t+1} &= \psi_i^{t+1} + \frac{\eta_i}{\mu_2} \beta_i^{t+1}.
\end{align*}
\]

Note that different from vanilla augmented Lagrangian, we introduce the decay factor \(\eta_i < 1\), which helps FedMM converge with smaller local steps. Therefore, the global consensus constraint is satisfied by the global update at the server with

\[
\begin{align*}
\omega_0^{t+1} &= \frac{1}{N} \sum_{i=1}^{N} \omega_i^{t+1}, \quad \text{and} \quad \psi_0^{t+1} = \frac{1}{N} \sum_{i=1}^{N} \psi_i^{t+1}.
\end{align*}
\]

We can now summarize one round of the FedMM algorithm, which consists of three major steps: (i) Parallel saddle-point optimization on all local augmented Lagrangian function \(\mathcal{L}_i\)’s. One optimization oracle example is based on SGDA, as shown in (11) and (12). (ii) Local gradient descent and ascent updates on dual variable \((\{\beta_i, \lambda_i\})\) as shown in (14). (iii) Aggregation to update global consensus variables.

Algorithm 1 FedMM Algorithm

\[
\begin{align*}
\text{Require:} & \quad \text{Initialize } \omega_0^0, \psi_0^0, \mu_1, \mu_2, \eta_1, \eta_2, \{M_t\} \in \mathbb{N}. \\
& \text{for } t = 0, \ldots, T - 1 \text{ do} \\
& \text{for each client } i \in [N] \text{ in parallel do} \\
& \quad \omega_i^{t+1} = \omega_i^t; \quad \psi_i^{t+1} = \psi_i^t \\
& \quad \# \text{Local Update:} \\
& \text{for } m = 0, \ldots, M_t - 1 \text{ do} \\
& \quad \# \text{Stochastic Gradient Descent:} \\
& \quad \omega_i^{m+1} = \omega_i^m - \eta_i \left[ \nabla_{\omega_i} f_i(\omega_i^m, \psi_i^m) + \mu_1 (\omega_i^m - \omega_0^0) + \lambda_i^1 \right] \\
& \quad \# \text{Stochastic Gradient Ascent} \\
& \quad \omega_i^{m+1} = \omega_i^m + \eta_i \left[ \nabla_{\psi_i} f_i(\omega_i^m, \psi_i^m) - \mu_2 (\psi_i^m - \psi_0^0) - \beta_i^1 \right] \\
& \text{end for} \\
& \quad \omega_i^{t+1} = \omega_i^t; \quad \psi_i^{t+1} = \psi_i^t \\
& \quad \# \text{Dual Descent:} \\
& \quad \lambda_i^{t+1} = \lambda_i^t + \mu_1 (\omega_i^{t+1} - \omega_0^0) \\
& \quad \# \text{Dual Ascent:} \\
& \quad \beta_i^{t+1} = \beta_i^t + \mu_2 (\psi_i^{t+1} - \psi_0^0) \\
& \text{end for} \\
& \quad \# \text{Global Update:} \\
& \quad \omega_0^{t+1} = \frac{1}{N} \sum_{i=1}^{N} \omega_i^{t+1}; \quad \psi_0^{t+1} = \frac{1}{N} \sum_{i=1}^{N} \psi_i^{t+1} \\
& \text{end for}
\end{align*}
\]

Figure 3: The FedMM algorithm addresses the federated adversarial domain adaptation as shown in the flowchart. Each source and target client has unique local minimax objectives due to domain distribution differences. Clients conduct local optimization, upload parameters to the server, and receive averaged parameter updates in parallel, completing one-round updates.
We first investigate how a change in label imbalance affects model worst-case scenario, in which the source domain data and target strength of our proposed FedMM algorithm on the Federated do-parameter settings are provided in Appendix C and our code is are trained from random initialization in all settings. The hyper-distribution, as shown in Fig. 4, and performs well even in the 𝑝peration on server. On Office-31, we use a three-layer convolutional network as the invariant feature extractor, and the network models are trained from random initialization on server. On Office-31, we use the pre-trained MobileNetV2 [28] on ImageNet [27] as the feature extractor. For a fair comparison, the pre-trained MobileNetV2 is downloaded from the pre-trained one by [4]. Our experiments are primarily concerned with the training communication overhead and test accuracy on the label-free target data set. Appendix B contains detailed descriptions of the dataset used in this experiment.

**Experiment Setup.** On MNISTM, we use a three-layer convolutional network as the invariant feature extractor, and the network models are trained from random initialization on server. On Office-31, we use the pre-trained MobileNetV2 [28] on ImageNet [27] as the feature extractor. For a fair comparison, the pre-trained MobileNetV2 is downloaded from the pre-trained one by [28]. Both the task classifier and the domain classifier are two-layer fully-connected neural networks. The domain classifier’s parameter are trained from random initialization in all settings. The hyper-parameter settings are provided in Appendix C and our code is available at github.com/fedmm/FedMM.

**4.1 Ablation Experiment on FedMM**

We first investigate how a change in label imbalance affects model training performance. Consider the case of two clients and set the ratios for source and target data assigned to client 1 as 𝑝 and 1 − 𝑝, respectively. The commonly used adversarial domain adaptation models including DANN [6], CDAN [41], and MDD [17] are tested separately as the local model. The label imbalance degree varies from 0.5 to 1. FedMM is robust to variations in label distribution, as shown in Fig. 4, and performs well even in the worst-case scenario, in which the source domain data and target domain data are allocated to different clients separately, i.e., 𝑝 = 1. In practice, 𝑝 = 1 occurs frequently because different silos contain data from distinct domains. We will focus on the 𝑝 = 1 case for the remainder of the experiment to test the effectiveness of FedMM.

Unlike the traditional augmented Lagrangian method, FedMM introduces 𝜖3 < 1 for the FL setting to reduce the need for large local update steps 𝑀𝑖 for convergence. When 𝜖3 = 1, as shown in Fig. 6, large local steps with 𝑀𝑖 > 50 are required. With appropriate 𝜖3, one can reduce 𝑀𝑖 from 50 to 25 with negligible performance loss for all three adversarial domain adaptation models. Note that if 𝜖3 is less than the feasible range, the outcome will be suboptimal.

**4.2 FedMM is the State-of-the-Art of Federated Domain Adaptation**

With extensive experiments, we show that FedMM achieves SOTA performance in terms of accuracy and communication overhead. We include the following two kinds of baselines:

(i) Recent work on distributed minimax optimization including [4, 26, 30, 35] with extensive studies for different adversarial domain adaptation tasks shown in Table 1. As MDD is the SOTA network of centralized domain adaptation in current stage, we only compare these works on federated domain adaptation with MDD loss.

(ii) Peng et al. [22] proposed FedSGDA to extend FedSGD to SGDA for federated domain adaptation, where the single step per communication round in SGDA resulted in massive communication overhead as observed in experiments. Though, to the best of our knowledge, there is no communication-efficient federated domain adaptation algorithm. To reduce communication overhead, we inspire from the existing multi-step federated minimization optimizers like FedAvg [20], and FedProx [14], which leads to FedAvgSGDA and FedProxSGDA in Algorithm 3 as described in Appendix A.

**Performance comparison.** FedMM has far fewer global communication rounds than FedSGDA, as illustrated in Fig. 6. FedMM saves more than 90% of the rounds required to achieve the same level of test accuracy on the target domain as FedSGDA in all three categories of representative domain adaptation networks due to the deliberately designed FedMM’s local multi-step minimax optimization at each client.

Moreover, FedMM not only reduces communication overhead but also ensures accuracy. Gradient drift causes severe performance degradation in multi-step local SGD on existing baselines. In Fig. 7, we compare the convergence property of our proposed FedMM to FedAvgSGDA and FedProxSGDA with 𝑀𝑖 = 20 for different source/target client settings. While both the FedAvgSGDA and FedProxSGDA algorithms converge in a communication-saving manner at the expense of severe performance decay, FedMM consistently outperforms them by more than 20% in terms of test accuracy for training from scratch with all the three domain adaptation networks. The results clearly show how that FedMM addresses the problem of gradient drifts in multi-steps local minimax, which have not been observed in any previous FL problems other than federated adversarial domain adaptation.

We compare FedMM to all of the above baselines on commonly used domain adaptation tasks on Office-31, including A → W, etc. six tasks with pretrained model, i.e., MobileNetV2 [28] on ImageNet [27]. When compared to the training from scratch case in Fig. 7, it is expected that FedMM’s performance improvement will be reduced.
The comparison is based on different adversarial domain networks, i.e., DANN, MDD, and CDAN. Models are trained from scratch on the global objective function $f$ concave case. Asymptotically to the stationary point for the nonconvex strongly-concave case. Our main convergence results and show that FedMM converges dual method, on the other hand, is more challenging. We establish invariance bounding client’s drift from global parameter via primal-dual method. Convergence analysis for a federated optimizer, such as FedMM that involves bounding client’s drift from global parameter via primal-dual method. They do not have an evident performance gain over FedAvgSGDA. Because the feature extractor parameters in these pre-trained models have approached optimal values. FedMM achieves SOTA with significant accuracy improvement even though other multi-step local update distributed minimax optimization methods including [4, 26, 30] save the communication overhead, they do not have a evident performance gain over FedAvgSGDA.

5 CONVERGENCE ANALYSIS

For a theoretical analysis, finding a global saddle point, i.e., min$_{\omega}$ max$_{\psi} F(x, y)$, in general is intractable [16]. One approach is to equivalently reformulate the problem by min$_{\omega}$ $\Phi(x) := \max_{y \in Y} F(x, y)$, and define an optimality notion for the local surrogate of global optimum of $\Phi$. A series of theoretical analyses on the stationary point convergence condition of $\Phi$ with first-order algorithm were carried out to extend the convex-concave assumption to assumptions of nonconvex-strongly-concave [18, 24, 34], nonconvex-concave [16, 21], and nonconvex-nonconcave [10]. Convergence analysis for a federated optimizer, such as FedMM that involves bounding client’s drift from global parameter via primal-dual method, on the other hand, is more challenging. We establish our main convergence results and show that FedMM converges asymptotically to the stationary point for the nonconvex-strongly-concave case.

Let $\psi^*(\omega) := \arg \max_{\psi} F(\omega, \psi)$ be the optimal value of $\psi$ for the global objective function $f$ w.r.t $\omega$. Then (4) can be reformulated as

$$\min_{\omega} f(\omega, \psi) = \min_{\omega} \frac{1}{M} \sum_i \Phi_i(\omega) \text{ with }$$

$$\Phi_i(\omega) \equiv f_i(\omega, \psi^*(\omega)), \quad \Phi(\omega) \equiv \frac{1}{N} \sum_{i=1}^N \Phi_i(\omega). \quad \text{(17)}$$

In this way, we equivalently reformulate the problem as

$$\min_{\omega} \{ \Phi(\omega) = \max_{\psi} f(\omega, \psi) \}.$$ To ease the presentation, we further define the augmented Lagrange function of $\Phi_i$ by

$$L^\delta_i(\omega^i, o^i, \psi^i) = \Phi_i(\omega^i) + \langle \lambda^i, o^i - o^i_0 \rangle + \frac{\mu_i}{2} \| o^i - o^i_0 \|^2. \quad \text{(18)}$$

5.1 Assumptions

Note that we concentrate on the convergence analysis of the federated nonconvex-strongly-concave case, which is difficult even in a centralized setting and has recently received increased attention in the literature [16, 19]. Thus, we set the standard assumptions by following the minimax optimization literature [10, 15, 19] to impose customary conditions on the gradients of local functions.

Assumption 1. (Lipschitz continuous gradients) For all $i \in [N]$, there exists positive constants $L_{11}, L_{12}, L_{21}$, and $L_{22}$ such that for any $\omega, \omega' \in \mathbb{R}^{d_1}$ and $\psi, \psi' \in \mathbb{R}^{d_2}$, we have

$$\| \nabla_{\omega} f_i(\omega, \psi) - \nabla_{\omega} f_i(\omega', \psi) \| \leq L_{11} \| \omega - \omega' \|,$nabla_{\omega} f_i(\omega, \psi) - \nabla_{\omega} f_i(\omega', \psi') \| \leq L_{12} \| \psi - \psi' \|,$nabla_{\psi} f_i(\omega, \psi) - \nabla_{\psi} f_i(\omega', \psi') \| \leq L_{21} \| \omega - \omega' \|,$nabla_{\psi} f_i(\omega, \psi) - \nabla_{\psi} f_i(\omega', \psi') \| \leq L_{22} \| \psi - \psi' \|.$
Assumption 2. (Strongly concave $f_i(\omega, \psi)$) For all $i \in [N]$, $f_i(\omega, \psi)$ are strongly concave on $\psi$ with constant $B > 0$ such that for any $\omega \in \mathbb{R}^{d_1}$ and $\psi, \psi' \in \mathbb{R}^{d_2}$, we have
\[
\left\langle \nabla f_i(\omega, \psi) - \nabla f_i(\omega, \psi'), \psi - \psi' \right\rangle \leq -B \| \psi - \psi' \|^2.
\] (19)

Assumption 3. The $\kappa$-Lipschitz continuity of $\psi^\ast(\omega)$, i.e.,
\[
\| \psi^\ast(\omega^1) - \psi^\ast(\omega^t) \| \leq \kappa \| \omega^1 - \omega^t \|, \quad \forall t \in [T].
\] (20)

Next, we make the following assumptions that $M_i$ in FedMM is chosen that local objective are sufficiently trained that $\omega_i^1, \psi_i^1$ is $\epsilon$ stationary on $L_i$.

Assumption 4. (Sufficient local training) For all $i \in [N]$, after $M_i$-step update, the gradients w.r.t. $\omega_i$ and $\psi_i$ are finite and denoted by
\[
\| \nabla L_i(\omega_i^1, \psi_i^1) \| \leq \epsilon \quad \forall t \in [T].
\] (21)

Theorem 1. (Convergence on $\Phi(\omega)$) With Assumption 1, 2, 20 and 4 holds. Then there exist positive constants $E_1$, $E_2$, and $E_3$, which are independent of $T$, such that after $T$ rounds of global updates, the upper bound for the accumulate descent of $\Phi(\omega^0)$ is given by
\[
\Phi(\omega^0) - \Phi(\omega^T) \leq -E_1 \sum_{t=1}^{T} \| \nabla \Phi(\omega^t) \|^2 + E_2 T \epsilon
\]
\[
+ E_2 T \epsilon + E_3 \sum_{i=1}^{N} \| \psi_i^0 - \psi^\ast(\omega_i^0) \|^2.
\] (22)

In particular, this implies $\limsup_{T \to \infty} \| \nabla \Phi(\omega^T) \| = O(\epsilon)$ with a local residue gradient error bound, i.e., $\| \nabla L_i(\omega_i^T, \psi_i^T) \|^2 \leq \epsilon$.

Remark Because the l.h.s. of (22) admits a lower bound, so is the r.h.s. As a result, $\limsup_{T \to \infty} \sum_{i=1}^{T} \| \nabla \Phi(\omega_i^T) \|$ must converge.
which implies that $\Phi(\omega_0)$ converges to a $\epsilon$-stationary point. More specifically, dividing both sides of (22) by $T$ and taking lim sup$_{T \to \infty}$, we obtain lim sup$_{T \to \infty} \frac{\sum_{t=1}^{T} \|\nabla \Phi(\omega_0)\|}{T} \leq \frac{E_{\epsilon,\epsilon}^T}{T}$, which implies that $\sum_{t=1}^{T} \frac{\|\nabla \Phi(\omega_0)\|}{T} = O(T\epsilon)$ for sufficiently large $T$. $\nabla \Phi(\omega_0) = O(\epsilon)$. In the special case of $\epsilon = 0$, i.e., strict optimality is obtained at each local client, this result shows that the limiting point is a stationary point.

### 6 RELATED WORK

**Distributed minimization**: Since the invention of FedSGD and its communication efficient version FedAvg [29], several work has been developed to address the suboptimality of FedAvg over non-i.i.d data, including FedProx [14], FedPD [40], SCAFFOLD [12], FedNova [32], dynamic gradient aggregation [5], and FedDyn [1]. These works aim to minimize a sum of non-identical functions, where each function can only be accessed locally. Auto-FedAvg [33] adjusted weights at the aggregation during training. These results cannot be directly applied to federated saddle point optimization problems, such as the federated adversarial domain adaptation, which seeks a federated minimax optimization. Similarly, off-the-shelf distributed augmented Lagrangian minimization methods and convergence analyses in Jakovetić et al. [8, 9], Yue et al. [37] fails to address the unique challenges in FL minimax optimization including the domain adaptation problem.

**Distributed minimax**: Several works [4, 25, 26] have improved communication efficiency in FedAvg-based minimax optimization, including the federated GAN. However, FedAvgSGDA is sensitive to data imbalance in our federated domain adaptation problem, unlike in the federated GAN where the binary classification function works well since there is no label-imbalanced problem. Note that the FLRA algorithm in Reisizadeh et al. [26] also corresponds to FedAvgSGDA, and its convergence analysis cannot be directly adapted to our setting because each local client in our study is optimized on the augmented Lagrangian function with the specific assumption that a centralized server. This research was conducted during his time at Ant Group.

### 7 CONCLUSIONS

We have proposed FedMM for federated adversarial domain adaptation. FedMM is designed specifically for federated minimax optimizations with non-separable minimization and maximization variables, as well as clients with uneven label class distributions. We have symptomatically performed a theoretical analysis on the convergence property of our proposed FedMM. Experiments show that FedMM outperforms state-of-the-art algorithms in terms of communication rounds and test accuracy on various benchmark datasets.

### 8 ACKNOWLEDGEMENT

This research was supported in part by NSF through grant IIS-1910492. Han Zhao would like to thank the support from a Facebook research award and Amazon AWS Cloud Credits. Jian Du’s work was conducted during his time at Ant Group.