FDR-SVM: A Federated Distributionally Robust Support Vector Machine via a Mixture of Wasserstein Balls Ambiguity Set

Michael Ibrahim[1] **Heraldo Rozas**[2] **Nagi Gebraeel**[1] **Weijun Xie**[1]

[1] H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA
[2] Department of Electrical Engineering, University of Chile, Santiago, Chile

Abstract

We study a federated classification problem over a network of multiple clients and a central server, in which each client's local data remains private and is subject to uncertainty in both the features and labels. To address these uncertainties, we develop a novel Federated Distributionally Robust Support Vector Machine (FDR-SVM), robustifying the classification boundary against perturbations in local data distributions. Specifically, the data at each client is governed by a unique true distribution that is unknown. To handle this heterogeneity, we develop a novel Mixture of Wasserstein Balls (MoWB) ambiguity set, naturally extending the classical Wasserstein ball to the federated setting. We then establish theoretical guarantees for our proposed MoWB, deriving an out-of-sample performance bound and showing that its design preserves the separability of the FDR-SVM optimization problem. Next, we rigorously derive two algorithms that solve the FDR-SVM problem and analyze their convergence behavior as well as their worst-case time complexity. We evaluate our algorithms on industrial data and various UCI datasets, whereby we demonstrate that they frequently outperform existing state-of-the-art approaches.

1 INTRODUCTION

Original equipment manufacturers (OEMs) of industrial equipment (used in manufacturing, energy, and healthcare) sell their products to multiple customers under lucrative service contracts that demand stringent reliability standards—an especially critical requirement for systems such as aircraft engines or turbines in power plants. In order to meet these standards, OEMs must provide accurate diagnostics of impending faults, including their types and severities [Dutta et al., 2023, Yang et al., 2025, Lei et al., 2020]. Most customers are unwilling to share their operational data due to confidentiality and security concerns, particularly in industries critical to national security (e.g., nuclear power plants). Consequently, OEMs face the challenge of leveraging dispersed data sources to develop analytic models capable of improving fault detection and classification.

Distributed fault diagnosis [Du et al., 2024] via federated learning (FL) [McMahan et al., 2017] offers a promising solution to this problem. FL enables the training of a global model on data that remains at each client, thereby preserving privacy while allowing OEMs to draw on broader information sources for more robust fault diagnosis. Despite these advantages, industrial data can be extremely noisy—due to harsh operating conditions and sensor limitations—and often suffers from labeling errors caused by variations in operator expertise. As a result, industrial datasets tend to be among the *most uncertain and poorly labeled*.

Uncertainty in a binary classification problem can be modeled by considering the features $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^P$ and labels $y \in \{-1, +1\}$ as random variables governed by an underlying distribution $\mathbb{P}$ [Shafieezadeh Abadeh et al., 2015]. An ideal classifier with parameters $\boldsymbol{w} \in \mathbb{R}^P$ and a loss function $\ell(\boldsymbol{w}; (\boldsymbol{x}, y))$ minimizes the expected risk $\mathbb{E}^{\mathbb{P}}[\ell(\boldsymbol{w}; (\boldsymbol{x}, y))]$. Since $\mathbb{P}$ is typically unknown, it is common practice to rely on an empirical distribution $\widehat{\mathbb{P}}_N$ derived from N IID samples and then minimize the empirical risk $\mathbb{E}^{\widehat{\mathbb{P}}_N}[\ell(\boldsymbol{w}; (\boldsymbol{x}, y))]$. However, in cases where the training data is noisy or limited, the resulting model can be highly suboptimal, leading to poor out-of-sample performance [Kuhn et al., 2019, Shafieezadeh Abadeh et al., 2015].

Distributionally robust optimization (DRO) [Scarf et al., 1957, Delage and Ye, 2010, Bayraksan and Love, 2015, Shapiro, 2017, Mohajerin Esfahani and Kuhn, 2018, Shafieezadeh-Abadeh et al., 2019, Kuhn et al., 2019] addresses these challenges by specifying an ambiguity set $\mathcal{A}$ of plausible data distributions. The model is trained by minimizing the worst-case risk $\sup_{\mathbb{Q} \in \mathcal{A}} \mathbb{E}^{\mathbb{Q}}[\ell(\boldsymbol{w}; (\boldsymbol{x}, y))]$ at-

tained by any distribution $\mathbb{Q} \in \mathcal{A}$. A particularly popular approach is to define $\mathcal{A}$ as a Wasserstein ball around $\widehat{\mathbb{P}}_N$ [Kuhn et al., 2019]. Wasserstein-based DRO (WDRO) has attracted growing attention in machine learning [Shafieezadeh-Abadeh et al., 2019, Nietert et al., 2023, Gao and Kleywegt, 2023]. However, most WDRO research remains limited to centralized settings, and extending it to federated environments introduces significant challenges and computational complexities [Cherukuri and Cortés, 2020].

Contributions. We develop a *federated distributionally robust support vector machine* (FDR-SVM) that can be trained to global optimality on data distributed across G clients. Using DRO allows our model to be robust to uncertainties in both features and labels. More importantly, our model does not rely on restrictive assumptions, such as Lipschitz smoothness or strong convexity, which are often imposed by existing FL approaches. Although differential privacy is vital in FL, our work focuses on robustness to distributional uncertainties. To the best of our knowledge, this is the first effort utilizing WDRO to robustify a FL model under such general conditions. The main contributions of this paper are:

1. We propose a **M**ixture **of** **W**asserstein **B**alls (MoWB) ambiguity set that generalizes the Wasserstein ball to the distributed setting. This lays the foundation for robustifying a variety of FL models under the DRO paradigm. We then prove that the true data distribution belongs to MoWB with a certain confidence level under a mild assumption, and we use it to derive our separable FDR-SVM formulation.

2. We propose a subgradient method-based (SM) algorithm for training our FDR-SVM, where we rigorously derive the subgradient of the *infinite-dimensional* worst-case risk problem at each client assuming the compactness of the feature support set. We then prove the convergence of this algorithm to global optimality, and derive its worst-case time complexity.

3. We also propose an alternating direction method of multipliers-based (ADMM) algorithm for training our FDR-SVM, where we derive a convex, tractable optimization problem and a closed-form for local and global model updates, respectively. We show that this algorithm is only guaranteed convergence under the addition of a strongly convex term to each client's objective. While this may affect final model performance, convergence is achieved in fewer rounds than SM and without the need for feature support assumptions.

4. We evaluate our proposed methods on an industrial dataset and various popular UCI repository datasets, where we study their hyperparameter sensitivity and demonstrate that the FDR-SVM typically outperforms state-of-the-art (SOTA) baselines.

2 BACKGROUND AND PRIOR WORK

Distributionally Robust Optimization. DRO has gained popularity recently due to its applications in various areas of optimization and ML [Kuhn et al., 2019]. The general 1-WDRO problem is mathematically formulated as

$$\inf_{\boldsymbol{w}\in\mathcal{W}} \sup_{\mathbb{Q}\in\mathcal{A}_{\varepsilon,1,d}(\Xi)} \mathbb{E}^{\mathbb{Q}}[\ell], \tag{1}$$

where ℓ is the loss function parameterized by $\boldsymbol{w} \in \mathcal{W}$, and $\mathbb{Q}$ is any distribution within ambiguity set $\mathcal{A}_{\varepsilon,1,d}(\Xi)$, which is defined as

$$\mathcal{A}_{\varepsilon,1,d}(\Xi) \coloneqq \left\{\mathbb{Q} \in \mathcal{P}(\Xi) : W_{d,1}\left(\mathbb{Q}, \widehat{\mathbb{P}}_N\right) \leq \varepsilon\right\}, \tag{2}$$

where $\mathcal{P}(\Xi)$ is the set of all distributions supported on Ξ, and $W_{d,1}(\cdot,\cdot)$ is the type-1 Wasserstein distance equipped with transportation cost function $d(\boldsymbol{\xi}, \boldsymbol{\xi}')$. It has been shown in various works [Mohajerin Esfahani and Kuhn, 2018, Kuhn et al., 2019, Shapiro, 2017, Gao and Kleywegt, 2023] that the DRO problem (1) admits tractable, convex reformulations in many cases of practical interest. Moreover, it was also demonstrated by Kuhn et al. [2019] that the Wasserstein ambiguity set enjoys various attractive properties, such as its ability to assign point mass anywhere in the support set, and its interpretation as a confidence interval for $\mathbb{P}$. We provide further background on Wasserstein DRO in Appendix A.

Many efforts have successfully utilized DRO to robustify various classifiers. For example, Shafieezadeh-Abadeh et al. [2019] develop Wasserstein DR logistic regression (LR) and SVM. Further, Selvi et al. [2022] extend the DR LR model to data with mixed features. Faccini et al. [2022] utilize a moment-based ambiguity set to derive a DR version of the SVM. Finally, Sagawa et al. [2020] utilize group DRO to mitigate the tendency of classification deep neural networks (DNNs) to learn spurious correlations, relying on manual training data grouping. This is advanced by Wu et al. [2023], who utilize a DNN to perform the data grouping. All these efforts implicitly assume the availability of the training data at a central location, making them difficult to extend to FL settings [Cherukuri and Cortés, 2020].

Federated Learning. Since its introduction by Konečný et al. [2016], McMahan et al. [2017], FL has garnered much attention due to its practical utility. The `FedAvg` algorithm introduced by McMahan et al. [2017] relies on local stochastic gradient descent (SGD) updates by clients, and subsequent aggregation and rebroadcasting of model by the server. The work also introduces `FedSGD`, where each client only performs one local update step. `FedProx` [Li et al., 2020] adds a proximal term to the objective function to mitigate client heterogeneity issues. Wang et al. [2020] develop `FedNova`, where client updates are normalized to address data heterogeneity without impacting convergence. Alternatively, Karimireddy et al. [2020] propose `SCAFFOLD`, where client drift is addressed with the introduction of control variables. Personalized FL algorithms

include FedPer [Arivazhagan et al., 2019] which introduces a local personalization layer at each client, FedEM [Marfoq et al., 2021] which models local data distributions as a mixture of unknown distributions, FedPer++ [Xu et al., 2022] which utilizes regularization to prevent local overfitting, and FedL2P [Lee et al., 2023] which uses meta-learning to learn a personalization strategy for each client.

Distributionally Robust Federated Learning. Recently, many efforts have combined ideas from DRO and FL. For example, Deng et al. [2020] develop a DR version of FedAvg, hedging against uncertainty in client weights. Wu et al. [2022] propose mixup techniques in the local training stages, addressing noisy and heterogeneous client data. Further, Zecchin et al. [2023] develop an efficient algorithm for a DR FedAvg algorithm with no central server. Alternatively, Huang et al. [2021] combine FL with stochastic compositional optimization (CO), transforming the DR FedAvg algorithm into a CO problem. A FedDRO algorithm is proposed by Khanduri et al. [2023] as an extension of FedAvg for CO problems. Lau and Liu [2022] construct a Wasserstein ambiguity set from distributed data using barycenters, which may not exist and can be difficult to compute in a distributed fashion if they do. Moreover, Cherukuri and Cortés [2020] and Le et al. [2024] propose distributed WDRO formulations. However, the earlier relies on peer-to-peer communication, while the latter assumes the Lipschitz smoothness and strong convexity of the loss function.

3 PROBLEM SETTING

We consider the problem of classifying data of the form $\boldsymbol{\xi} = (\boldsymbol{x}, y)$ distributed over G clients, where $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^P$ is the feature vector and $y \in \{-1, +1\}$ is the label. With such data, a commonly-used transportation cost function is

$$d(\boldsymbol{\xi}, \boldsymbol{\xi}') := ||\boldsymbol{x} - \boldsymbol{x}'|| + \kappa \mathbb{1}_{\{y \neq y'\}}, \tag{3}$$

where $||\cdot||$ is any common norm on $\mathbb{R}^P$, and κ is a hyperparameter corresponding to label flipping cost.

We consider classification via the binary SVM characterized by the hinge loss function $\ell_H(\boldsymbol{w}; \boldsymbol{\xi})$, which is parameterized by $\boldsymbol{w} \in \mathbb{R}^P$ and defined as $\ell_H(\boldsymbol{w}; \boldsymbol{\xi}) = \max\{0, 1 - y \cdot \boldsymbol{w}^\mathsf{T} \boldsymbol{x}\}$. We choose the SVM classifier as it is a well-established model that is commonly used in fault classification settings [Dutta et al., 2023, Mathew et al., 2018]. Moreover, its simple formulation allows for the rigorous derivation of a DR version.

We study the FL setting where clients can only communicate with the central server but not with each other. Clients do not share their data with the central server, but they can transmit insights from locally trained models, such as local (sub)gradients or model parameters. In this context, we assume the existence of a local training set $\mathcal{S}_g = \{\widehat{\boldsymbol{\xi}}_{n_g}\}_{n_g=1}^{N_g} = \{(\widehat{\boldsymbol{x}}_{n_g}, \widehat{y}_{n_g})\}_{n_g=1}^{N_g}$ at each client g. We denote the empirical distribution of the N_g IID local training samples and their true distribution as $\widehat{\mathbb{P}}_{N_g}$ and $\mathbb{P}_g$, respectively. Finally, we denote the total number of training samples available at all clients as $N = \sum_g^G N_g$.

4 MOWB AMBIGUITY SET

4.1 PROBLEM SEPARABILITY

In this section, we extend the classical Wasserstein amiguity set to the distributed setting via the novel MoWB ambiguity set $\mathcal{A}_G$ defined next.

Definition 1 (MoWB ambiguity set)**.** The Mixture of Wasserstein Balls (MoWB) ambiguity set contains mixture distributions whose constituents are distributions from local Wasserstein balls defined at each client, and is expressed as

$$\mathcal{A}_G := \Big\{ \mathbb{Q} \colon \mathbb{Q} = \sum_{g=1}^G \alpha_g \mathbb{Q}_g,\ \alpha_g \geq 0,\ \sum_{g=1}^G \alpha_g = 1, \\ \mathbb{Q}_g \in \mathcal{A}^{(g)}_{\varepsilon_g, 1, d}(\Xi) \Big\}, \tag{4}$$

where α_g is client g's weight, and $\mathcal{A}^{(g)}_{\varepsilon_g,1,d}(\Xi)$ is the type-1 Wasserstein ball of radius ε_g supported on Ξ, centered at $\widehat{\mathbb{P}}_{N_g}$, and defined via cost function $d(\boldsymbol{\xi}, \boldsymbol{\xi}')$ shown in (3).

Remark 1. Observe that when $G = N$, our ambiguity set models worst-case perturbations in individual training samples in a fashion similar to robust optimization (RO). Alternatively, when $G = 1$, our ambiguity set reduces to the classical Wasserstein ball $\mathcal{A}_{\varepsilon,1,d}(\Xi)$ defined in (2). This suggests that our proposed ambiguity set offers more flexibility in modeling the uncertainty than the classical Wasserstein ambiguity set, which can allow it to achieve improved performance in some settings. This also suggests that our proposed ambiguity set naturally extends the classical Wasserstein ball to the FL setting. Indeed, we show in Proposition 1 that when equipped with the MoWB ambiguity set, the DRO problem enjoys a naturally distributed formulation.

Proposition 1 (Problem Separability)**.** *The original DRO problem in* (1) *equipped with the MoWB ambiguity set defined in* (4) *admits the following reformulation:*

$$\inf_{\boldsymbol{w}} \sup_{\mathbb{Q} \in \mathcal{A}_G} \mathbb{E}^{\mathbb{Q}}[\ell_H(\boldsymbol{w}; \boldsymbol{\xi})] \\ = \inf_{\boldsymbol{w}} \sum_{g=1}^G \alpha_g \sup_{\mathbb{Q}_g \in \mathcal{A}^{(g)}_{\varepsilon_g,1,d}(\Xi)} \mathbb{E}^{\mathbb{Q}_g}[\ell_H(\boldsymbol{w}; \boldsymbol{\xi})]. \tag{5}$$

Proof. Proof is provided in Appendix B.2.1. □

4.2 OUT-OF-SAMPLE PERFORMANCE GUARANTEES

Since the MoWB ambiguity set $\mathcal{A}_G$ relies on local Wasserstein balls $\mathcal{A}^{(g)}_{\varepsilon_g,1,d}(\Xi)$, it inherits desirable out-of-sample performance guarantees shown by Kuhn et al. [2019]. Indeed, we show in Proposition 2 that the true distribution $\mathbb{P} = \sum_{g=1}^{G} \alpha_g \mathbb{P}_g$ is contained within the MoWB ambiguity set with a certain confidence level, thereby allowing for the reduction of the *true risk* without knowing $\mathbb{P}$. This relies on Assumption 1, which allows for tighter concentration inequalities for $\mathbb{P}_g$, ensuring that they can indeed be modeled as a perturbation of the empirical distributions $\widehat{\mathbb{P}}_{N_g}$.

Assumption 1 (Light-tailed Distribution)**.** The true distribution $\mathbb{P}_g$ of the data at client g is light-tailed. That is, there exists $a > 1$ with $A_g := \mathbb{E}^{\mathbb{P}_g}[\exp(||2\boldsymbol{x}||^{a_g})] < +\infty$.

Proposition 2 (Out-of-Sample Performance)**.** *Suppose Assumption 1 holds and the local Wasserstein ball radius ε_g at client g is set as [Kuhn et al., 2019]*

$$\varepsilon_{N_g}(\eta_g) = \left(\frac{\log(c_{1_g}\eta_g^{-1})}{c_{2_g}N_g}\right)^{\frac{1}{a_g}} \mathbb{1}_{\left\{N_g < \frac{\log(c_{1_g}\eta_g^{-1})}{c_{2_g}c_{3_g}}\right\}} + \left(\frac{\log(c_{1_g}\eta_g^{-1})}{c_{2_g}N_g}\right)^{\frac{1}{P}} \mathbb{1}_{\left\{N_g \geq \frac{\log(c_{1_g}\eta_g^{-1})}{c_{2_g}c_{3_g}}\right\}},$$

where $c_{1_g}, c_{2_g}, c_{3_g} \in \mathbb{R}_+$ are constants that depend on a_g, A_g, P (dimension of the feature space), and the transportation cost given by (3)*. Then the MoWB ambiguity set $\mathcal{A}_G$ defined in* (4) *enjoys the following property*

$$\mathbb{P}^N\{\mathbb{P} \in \mathcal{A}_G\} \geq \prod_{g=1}^{G}(1-\eta_g),$$

where η_g is such that $\mathbb{P}^{N_g}\{\mathbb{P}_g \in \mathcal{A}^{(g)}_{\varepsilon_g,1,d}(\Xi)\} \geq (1-\eta_g)$.

Proof. Proof is provided in Appendix B.2.2. □

5 SOLUTION ALGORITHMS

We introduce two algorithms to solve problem (5). Given that our motivating scenario involves manufacturing plants (clients) with ample local compute resources and reliable communication with a central server, we adopt Assumption 2 to guarantee convergence of our algorithms.

Assumption 2 (Synchronous Training)**.** The distributed optimization problem in (5) is solved synchronously. That is, the central server only performs an update step once all the clients have completed solving their local problems and communicated their insights to the central server.

5.1 SUBGRADIENT-BASED ALGORITHM (SM)

The subgradient-based (**SM**) algorithm begins by initializing the global model parameters $\boldsymbol{w}$. Next, each client g seeks to obtain a subgradient for their inner maximization problem from (5), to be sent to the server for aggregation and model update. This requires each client g to obtain a worst-case distribution $\mathbb{Q}_g^*$ from its local ambiguity set $\mathcal{A}^{(g)}_{\varepsilon_g,1,d}(\Xi)$, allowing the worst-case risk to be directly expressed as an expectation with respect to $\mathbb{Q}_g^*$. However, Kuhn et al. [2019] show that the worst-case distribution cannot be obtained from a type-1 Wasserstein ambiguity set centered around an empirical distribution $\widehat{\mathbb{P}}_{N_g}$ if $\mathcal{X}$ is not compact. Therefore, we make Assumption 3 *only* for the SM algorithm, ensuring the compactness of $\mathcal{X}$.

Assumption 3 (Support of Feature Vector)**.** The feature vector $\boldsymbol{x}$ is such that: $0 \leq e_p^\mathsf{T}\boldsymbol{x} \leq 1\ \forall p \in [P]$, where e_p are the standard unit vectors.

Note that Assumption 3 is not very restrictive in practice. Indeed, real-world data is often bounded by sensor ranges, and can therefore be easily normalized. Given Assumption 3 holds and the global model parameters $\boldsymbol{w}$ are fixed, then by Shafieezadeh-Abadeh et al. [2019] it can be shown that the worst-case distribution $\mathbb{Q}_g^*$ for client g is

$$\mathbb{Q}_g^* = \frac{1}{N_g}\sum_{n_g=1}^{N_g}\Big(\beta_{n_g}^{+\,*}\delta_{(\widehat{\boldsymbol{x}}_{n_g}-\boldsymbol{q}_{n_g}^{+\,*}/\beta_{n_g}^{+\,*},\widehat{y}_{n_g})} + \beta_{n_g}^{-\,*}\delta_{(\widehat{\boldsymbol{x}}_{n_g}-\boldsymbol{q}_{n_g}^{-\,*}/\beta_{n_g}^{-\,*},-\widehat{y}_{n_g})}\Big), \quad (6)$$

where $\delta_{(\boldsymbol{x},y)}$ is the Dirac density function that assigns probability mass 1 at sample $\boldsymbol{\xi} = (\boldsymbol{x}, y)$, and $\beta_{n_g}^+$, $\beta_{n_g}^-$, $\boldsymbol{q}_{n_g}^+$, and $\boldsymbol{q}_{n_g}^-$ are maximizers of the following optimization problem:

$$\max_{\substack{\beta_{n_g}^+,\beta_{n_g}^- \\ \boldsymbol{q}_{n_g}^+,\boldsymbol{q}_{n_g}^-}} H_g(\boldsymbol{w}) := \begin{cases} \max\limits_{\substack{\beta_{n_g}^+,\beta_{n_g}^- \\ \boldsymbol{q}_{n_g}^+,\boldsymbol{q}_{n_g}^-}} & \frac{1}{N_g}\sum\limits_{n_g=1}^{N_g}\big(-(\beta_{n_g}^+ - \beta_{n_g}^-)\widehat{y}_{n_g}\boldsymbol{w}^\mathsf{T}\widehat{\boldsymbol{x}}_{n_g} - \widehat{y}_{n_g}\boldsymbol{w}^\mathsf{T}(\boldsymbol{q}_{n_g}^+ - \boldsymbol{q}_{n_g}^-)\big) \\ \text{s. t.} & \sum\limits_{n_g=1}^{N_g}\big(||\boldsymbol{q}_{n_g}^+|| + ||\boldsymbol{q}_{n_g}^-|| + \kappa_g\beta_{n_g}^-\big) \leq N_g\varepsilon_g \\ & \beta_{n_g}^+ + \beta_{n_g}^- = 1 \qquad \forall n_g \in [N_g] \\ & 0 \leq \beta_{n_g}^+\widehat{\boldsymbol{x}}_{n_g} - \boldsymbol{q}_{n_g}^+ \leq \beta_{n_g}^+ \qquad \forall n_g \in [N_g] \\ & 0 \leq \beta_{n_g}^-\widehat{\boldsymbol{x}}_{n_g} - \boldsymbol{q}_{n_g}^- \leq \beta_{n_g}^- \qquad \forall n_g \in [N_g] \\ & \beta_{n_g}^+, \beta_{n_g}^- \geq 0 \qquad \forall n_g \in [N_g] \end{cases}, \quad (7)$$

where $||\cdot||$ is the norm used in the definition of the transportation cost function (3). Armed with the discrete worst-case

distribution $\mathbb{Q}_g^*$, each client g can compute a subgradient, $\boldsymbol{v}_g$, of their local maximization problem. Proposition 3 presents a closed-form for obtaining $\boldsymbol{v}_g$.

Proposition 3 (Local Subgradient Computation)**.** *Suppose the worst-case distribution $\mathbb{Q}_g^*$ is known to client g. Then, they can compute a subgradient $\boldsymbol{v}_g$ for their respective maximization problem from* (5) *as any vector that obeys*

$$\boldsymbol{v}_g \in \frac{1}{N_g} \sum_{n_g=1}^{N_g} \left(\mathcal{B}^+ + \mathcal{B}^-\right),$$

where $+$ is the Minkowski sum and $\mathcal{B}^+$, $\mathcal{B}^-$ are defined as

$$\mathcal{B}^\pm := \begin{cases} \mathbf{0} & \text{if } \widehat{r}_{n_g}^\pm < 0 \\ \mp \beta_{n_g}^{\pm\,*} \widehat{y}_{n_g} \widehat{\boldsymbol{z}}_{n_g}^\pm & \text{if } \widehat{r}_{n_g}^\pm > 0 \\ \text{conv}\left(\{\mathbf{0}, \mp \beta_{n_g}^{\pm\,*} \widehat{y}_{n_g} \widehat{\boldsymbol{z}}_{n_g}^\pm\}\right) & \text{if } \widehat{r}_{n_g}^\pm = 0 \end{cases},$$

where $\widehat{\boldsymbol{z}}_{n_g}^\pm = \widehat{\boldsymbol{x}}_{n_g} - \boldsymbol{q}_{n_g}^{\pm\,*}/\beta_{n_g}^{\pm\,*}$, $\widehat{r}_{n_g}^\pm = 1 \mp \widehat{y}_{n_g} \cdot \boldsymbol{w}^\mathsf{T} \widehat{\boldsymbol{z}}_{n_g}^\pm$, and conv$(\Theta)$ is the convex hull of set Θ.

Proof. Proof is provided in Appendix B.2.3. □

The subgradients $\boldsymbol{v}_g$ from the clients are then aggregated by the server, and used to update the global model $\boldsymbol{w}$ and broadcast it back to the clients. This process repeats for T rounds. The pseudocode of the SM algorithm is given in 1.

Algorithm 1 SM Algorithm

Input: $\boldsymbol{w}^{(0)}$
Parameters: Number of rounds T, step-size $\gamma(t)$ at round t
Output: $\boldsymbol{w}^*$

for $t = 1, \ldots, T$ **do**
  Client Update:
  for clients $g = 1, \ldots, G$ **do**
    Solve for $[\beta_{n_g}^{+\,*}, \beta_{n_g}^{-\,*}, \boldsymbol{q}_{n_g}^{+\,*}, \boldsymbol{q}_{n_g}^{-\,*}] \leftarrow \arg\max H_g(\boldsymbol{w}^{(t)})$
    Compute $\mathbb{Q}_g^* \leftarrow \frac{1}{N_g} \sum_{n_g=1}^{N_g} \beta_{n_g}^{+\,*} \delta_{(\widehat{\boldsymbol{x}}_{n_g} - \boldsymbol{q}_{n_g}^{+\,*}/\beta_{n_g}^{+\,*}, \widehat{y}_{n_g})} + \beta_{n_g}^{-\,*} \delta_{(\widehat{\boldsymbol{x}}_{n_g} - \boldsymbol{q}_{n_g}^{-\,*}/\beta_{n_g}^{-\,*}, -\widehat{y}_{n_g})}$
    Compute any local subgradient $\boldsymbol{v}_g$ via Proposition 3
    Send $\boldsymbol{v}_g$ to central server.
  end for
  Server Update:
  $\boldsymbol{w}^{(t+1)} \longleftarrow \boldsymbol{w}^{(t)} - \gamma(t) \sum_{g=1}^{G} \alpha_g \boldsymbol{v}_g$
  Broadcast $\boldsymbol{w}^{(t+1)}$ to all clients
end for

Convergence. It is known that the subgradient method converges to an optimal objective value under certain conditions [Boyd et al., 2003]. We present Lemmas 2, 3, 4 in Appendix B.1, proving the convexity, Lipschitz continuity, and coercivity of problem (5)'s objective in $\boldsymbol{w}$. Theorem 1 then asserts that these properties satisfy the convergence criteria of the subgradient method given that the step-size diminishes appropriately, proving the convergence of the SM algorithm. We also derive the SM algorithm's worst-case time complexity in Theorem 2, showing that it converges in polynomial time with a sublinear number of communication rounds.

Theorem 1 (SM Convergence)**.** *The SM Algorithm 1 converges to an optimal solution $\boldsymbol{w}^*$ of problem* (5) *within an arbitrary tolerance $\epsilon_1 > 0$, provided the step-size $\gamma(t) \to 0$ as $t \to \infty$ and $\sum_{t=1}^{\infty} \gamma(t) = \infty$.*

Proof. Proof is provided in Appendix B.2.4. □

Theorem 2 (SM Time Complexity)**.** *Suppose the ℓ_∞-norm is used in problem* (7)*, and that it is solved via the barrier method equipped with the log barrier function and Newton updates. Then, the SM algorithm 1 with the diminishing step-size in Theorem 1 has an overall worst-case time complexity of $\mathcal{O}\left(\epsilon_1^{-2}\left[N_{g^*}^{3.5} P^{3.5} \log(\epsilon_2^{-1}) + GP\right]\right)$ (with $\mathcal{O}(\epsilon_1^{-2})$ communication rounds), where ϵ_1, $\epsilon_2 > 0$ are tolerances on the solutions of the subgradient method and problem* (7)*, respectively, and N_{g^*} is the greatest number of samples at any client.*

Proof. Proof is provided in Appendix B.2.5. □

5.2 ADMM-BASED ALGORITHM (ADMM)

The ADMM-based (**ADMM**) algorithm requires each client g to solve their local problem and send their optimal local model $\boldsymbol{w}_g^*$ to the server. There, the local models are aggregated to obtain optimal global model $\boldsymbol{w}^*$, which is broadcast to the clients. This repeats for T rounds. To guarantee theoretical convergence, we also create a modified version of this algorithm with strongly convex client objectives, denoted as **ADMM-SC**. Further detail on this is given in the convergence discussion. Deriving this algorithm begins by introducing a decision variable $\boldsymbol{w}_g$ for each client g, and rewriting problem (5) to enforce client concensus as follows.

$$\begin{aligned} \inf_{\boldsymbol{w_g}, \boldsymbol{w}} \quad & \sum_{g=1}^{G} \alpha_g \sup_{\mathbb{Q}_g \in \mathcal{A}_{\varepsilon_g,1,d}^{(g)}(\Xi)} \mathbb{E}^{\mathbb{Q}_g}[\ell_H(\boldsymbol{w}_g; \boldsymbol{\xi})] \\ \text{s.t.} \quad & \boldsymbol{w}_g - \boldsymbol{w} = 0 \qquad \forall g \in [G]. \end{aligned} \tag{8}$$

Next, we express the Augmented Lagrangian parameterized by scale parameter ρ for the problem in (8) as follows:

$$\mathcal{L}_\rho(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_G, \boldsymbol{w}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_G) = \sum_{g=1}^{G} \alpha_g \mathcal{L}_{\rho_g}(\boldsymbol{w}_g, \boldsymbol{w}, \boldsymbol{\mu}_g),$$

where $\boldsymbol{\mu_g}$ are client g's scaled Lagrange multipliers, and

$$\mathcal{L}_{\rho_g}(\boldsymbol{w}_g, \boldsymbol{w}, \boldsymbol{\mu}_g) = \sup_{\mathbb{Q}_g \in \mathcal{A}_{\varepsilon_g,1,d}^{(g)}(\Xi)} \mathbb{E}^{\mathbb{Q}_g}[\ell_H(\boldsymbol{w}_g; \boldsymbol{\xi})] + \frac{\rho}{2} ||\boldsymbol{w}_g - \boldsymbol{w} + \boldsymbol{\mu}_g||_2^2.$$

Given the Augmented Lagrangian, client g and the server can obtain their model updates by minimizing it with respect

to $\boldsymbol{w}_g$ and $\boldsymbol{w}$, respectively. Proposition 4 presents a *tractable, convex* problem that is solved by each client for local model updates. Proposition 5 presents a closed-form expression for the server's update of the global model.

Proposition 4 (ADMM Client Update)**.** *Provided with the updated global model $\boldsymbol{w}^*$, client g can obtain their updated local model $\boldsymbol{w}_g^*$ as the minimizer to the following problem*

$$J_g(\boldsymbol{w}, \boldsymbol{\mu}_g) = \begin{cases} \min\limits_{\boldsymbol{w}_g, \lambda_g, s_{n_g}} & \lambda_g \varepsilon_g + \dfrac{1}{N_g} \displaystyle\sum_{n_g=1}^{N_g} s_{n_g} \\ & \qquad + \dfrac{\rho}{2} ||\boldsymbol{w}_g - \boldsymbol{w}^* + \boldsymbol{\mu}_g||_2^2 \\ \text{s. t.} & \ell_H(\boldsymbol{w}_g; (\widehat{\boldsymbol{x}}_{n_g}, \widehat{y}_{n_g})) \leq s_{n_g} \; \forall n_g \in [N_g] \\ & \ell_H(\boldsymbol{w}_g; (\widehat{\boldsymbol{x}}_{n_g}, -\widehat{y}_{n_g})) - \kappa \lambda_g \leq s_{n_g} \\ & \qquad \forall n_g \in [N_g] \\ & \lambda \geq ||\boldsymbol{w}_g||_* \end{cases}, \tag{9}$$

where $||\cdot||_*$ is the dual to the norm used in the transportation cost function (3)*.*

Proof. Proof is provided in Appendix B.2.6. □

Proposition 5 (ADMM Server Update)**.** *Provided with the updated local models $\boldsymbol{w}_g^*$ and scaled Lagrange multipliers $\boldsymbol{\mu}_g^*$, the central server can obtain the updated global model $\boldsymbol{w}^*$ via the following*

$$\boldsymbol{w}^* = \sum_{g=1}^{G} \alpha_g (\boldsymbol{w}_g^* + \boldsymbol{\mu}_g^*).$$

Proof. Proof is provided in Appendix B.2.7. □

The server then broadcasts $\boldsymbol{w}^*$ to the clients, where they update their Lagrange multipliers and the process repeats. We provide the pseudocode for the ADMM algorithm in 2.

Convergence. Even if the objective function is closed and proper convex as we show in Lemma 5 in Appendix B.1, it has been established in the literature that the global convergence of the multi-block (i.e. $G \geq 3$) ADMM algorithm is generally not guaranteed. Indeed, a counterexample is presented by Chen et al. [2016] demonstrating that the multi-block ADMM with a separable convex objective function can fail to converge. However, the convergence of multi-block ADMM in practical cases such as [Tao and Yuan, 2011] has motivated works to investigate conditions under which it is guaranteed convergence [Deng et al., 2017, Lin et al., 2015]. In Theorem 3, we introduce an additional strongly convex term to be added to each client's objective, and we denote the ADMM algorithm with strongly convex client objectives as ADMM-SC. We then show that

Algorithm 2 ADMM/ADMM-SC Algorithm

Input: $\boldsymbol{w}^{(0)}, \boldsymbol{w}_g^{(0)}, \boldsymbol{\mu}_g^{(0)}$
Parameters: Number of rounds T, scale parameter ρ
Output: $\boldsymbol{w}^*$

for $t = 1, \ldots, T$ **do**
  Client Update:
  for clients $g = 1, \ldots, G$ **do**
    Solve for $\boldsymbol{w}_g^{(t+1)} \leftarrow \boldsymbol{w}_g^*$ minimizer of $J_g(\boldsymbol{w}^{(t)}, \boldsymbol{\mu}_g^{(t)})$ (9) (or $J_g^{\text{SC}}(\boldsymbol{w}^{(t)}, \boldsymbol{\mu}_g^{(t)})$ in Appendix B.3.1 for ADMM-SC)
    Send $\boldsymbol{w}_g^{(t+1)}$ to central server
  end for
  Server Update:
  Update $\boldsymbol{w}^{(t+1)} \leftarrow \sum_{g=1}^{G} \alpha_g (\boldsymbol{w}_g^{(t+1)} + \boldsymbol{\mu}_g^{(t)})$
  Broadcast $\boldsymbol{w}^{(t+1)}$ to all clients
  Client Update:
  for clients $g = 1, \ldots, G$ **do**
    $\boldsymbol{\mu}_g^{(t+1)} \leftarrow \boldsymbol{\mu}_g^{(t)} + \boldsymbol{w}_g^{(t+1)} - \boldsymbol{w}^{(t+1)}$
  end for
end for

ADMM-SC indeed converges as it obeys the criteria given by Lin et al. [2015]. Subsequently, we present worst-case time complexity of ADMM-SC in Theorem 4, showing that it too converges in polynomial time, but requires fewer communication rounds than the SM algorithm.

Theorem 3 (ADMM-SC Convergence)**.** *Suppose the local client problem in* (9) *is modified by adding a $\tau_g ||\boldsymbol{w}_g||_2^2$ term to the objective function, where τ_g is a user-defined hyper-parameter, resulting in the modified client problem with a strongly convex objective $J_g^{SC}(\boldsymbol{w}^{(t)}, \boldsymbol{\mu}_g^{(t)})$ in Appendix B.3.1. Suppose further that the ADMM-SC algorithm in 2 is used to train the FDR-SVM with the modified objective. Then, ADMM-SC converges to an optimal solution $\boldsymbol{w}^*$ of the modified problem with arbitrary tolerance $\epsilon_1 > 0$ if $\rho \leq \min_{g=1,\ldots,G-1} \left\{ \frac{4\alpha_g \tau_g}{g(2G+1-g)}, \frac{4\alpha_G \tau_G}{(G-1)(G+2)} \right\}$.*

Proof. Proof is provided in Appendix B.2.8. □

Remark 2. The additional $\tau_g ||\boldsymbol{w}_g||_2^2$ terms are redundant regularization terms, potentially impacting the performance of the final model as shown empirically in Section 6.

Theorem 4 (ADMM-SC Time Complexity)**.** *Suppose that the ℓ_1-norm is used in the strongly convex variant of the local model problem* (9)*, and that it is solved via the barrier method with the log barrier function and Newton updates. Then, the ADMM-SC algorithm 2 equipped with ρ chosen according to Theorem 3 has an overall worst-case time complexity of $\mathcal{O}\left(\epsilon_1^{-1}\left[(N_{g^*} + P)^{3.5} \log(\epsilon_2^{-1}) + GP\right]\right)$ (with $\mathcal{O}(\epsilon_1^{-1})$ communication rounds), where ϵ_1, $\epsilon_2 > 0$ are tolerances on the solutions of ADMM and the strongly convex variant of the local problem* (9)*, respectively, and N_{g^*} is the greatest number of samples at any client.*

Proof. Proof is provided in Appendix B.2.9. □

Table 1: F-1 Score Attained by Classification Models on 7 UCI Datasets.

Model	Banknote	BCW	CB	MM	Parkinson's	Rice	UKM
Central (DR-SVM)	$\mathbf{.950} \pm .011$	$.964 \pm .013$	$.773 \pm .052$	$.792 \pm .017$	$.904 \pm .025$	$\mathbf{.938} \pm .005$	$.845 \pm .027$
FedSGD (ℓ_2-SVM)	$\mathbf{.950} \pm .011$	$.914 \pm .019$	$.765 \pm .045$	$.624 \pm .161$	$.752 \pm .204$	$.856 \pm .013$	$.808 \pm .019$
FedAvg (ℓ_2-SVM)	$\mathbf{.950} \pm .011$	$.929 \pm .016$	$.788 \pm .051$	$.787 \pm .024$	$.816 \pm .132$	$.936 \pm .006$	$.847 \pm .027$
FedProx (ℓ_2-SVM)	$\mathbf{.950} \pm .011$	$.929 \pm .019$	$.780 \pm .075$	$.782 \pm .052$	$.735 \pm .210$	$.931 \pm .006$	$.847 \pm .028$
FedDRO (KL)	$.945 \pm .011$	$.925 \pm .019$	$.738 \pm .036$	$.783 \pm .019$	$.864 \pm .025$	$.859 \pm .009$	$.718 \pm .001$
SM (FDR-SVM)	$.855 \pm .017$	$.957 \pm .015$	$.769 \pm .054$	$.797 \pm .023$	$\mathbf{.920} \pm .023$	$.936 \pm .006$	$.840 \pm .031$
ADMM (FDR-SVM)	$\mathbf{.950} \pm .011$	$\mathbf{.967} \pm .014$	$\mathbf{.792} \pm .047$	$\mathbf{.798} \pm .017$	$.911 \pm .021$	$\mathbf{.938} \pm .005$	$\mathbf{.848} \pm .027$
ADMM-SC (FDR-SVM)	$\mathbf{.950} \pm .011$	$.966 \pm .014$	$.765 \pm .048$	$.797 \pm .019$	$.902 \pm .026$	$\mathbf{.938} \pm .005$	$.846 \pm .027$

6 NUMERICAL EXPERIMENTS

We discuss numerical experiments that examine the performance of our algorithms and compare them to SOTA baselines. Unless otherwise stated, all numbers reported are averages over 50 repetitions with randomized data shuffling, and all error bars or confidence intervals represent one standard deviation. We use equivalent client weights α_g and local hyperparameters for all clients. Additional experimental details and a **scalability experiment** are provided in Appendix C. All code is available at this link.

Our Methods. Our algorithms include: i) *SM*: the FDR-SVM model trained via the SM algorithm in 1 with a diminishing step-size, ii) *ADMM*: the FDR-SVM model trained via the ADMM algorithm in 2, and iii) *ADMM-SC*: the FDR-SVM model with modified client objectives according to Theorem 3, trained via the ADMM-SC algorithm in 2.

6.1 UCI DATA EXPERIMENT

This experiment compares the performance of our methods to various SOTA benchmarks. We use $G = 4$ clients for all datasets. Performance is measured in terms of F-1 score.

Datasets. We utilize 7 popular dataset from the UCI repository. For all datasets 70% of the samples are used for training and the remainder is used for testing.

Baselines. We use the DR SVM model by Shafieezadeh-Abadeh et al. [2019] as a centralized benchmark model. For federated baselines, we compare to the popular `FedSGD`, `FedAvg` [McMahan et al., 2017], and `FedProx` [Li et al., 2020] used to train an ℓ_2-squared regularized SVM. We also compare to `FedDRP` [Khanduri et al., 2023] used to train a DR-SVM with a KL divergence ambiguity set.

Hyperparameters. We tune the Wasserstein radius ε and label-flipping cost κ for the centralized baseline, and the initial learning rate $\gamma(0)$ and number of rounds T for all federated baselines. We also tune the number of rounds T and the hyperparameters ρ and γ for our methods. We utilize 5-fold cross-validation for hyperparameter tuning.

Results. Table 1 presents the performance achieved by each model. Our proposed models consistently outperform the federated benchmark models on most datasets, often by a substantial margin. This underscores the value of DR in modeling uncertainty, and the benefits of using algorithms specifically designed for the FDR-SVM problem. We note that one or more of our FDR-SVM algorithms attains the highest F-1 score for all datasets. Additionally, the ADMM algorithm generally outperforms SM algorithm on most datasets, except for Parkinson's, which suggests that ADMM often converges in practice, in many settings, even if theoretical convergence is not guaranteed.

We also observe that in some cases, ADMM-SC performs much worse than ADMM (e.g., on BCW and UKM) but can also closely match its performance (e.g., on Banknote, MM, and Rice). This suggests that pursuing guaranteed theoretical convergence comes at the cost of stronger regularization, and thus, potentially weaker performance. One notable observation is that the ADMM or SM algorithms can sometimes outperform the centralized model. This suggests that our proposed MoWB ambiguity set can **outperform** the classical Wasserstein ball in modeling uncertainty in some settings as hypothesized in Remark 1. Finally, we note that `FedAvg` and `FedProx` failed to consistently converge for the Rice dataset despite extensive hyperparameter tuning and a diminishing learning rate. This suggests a lack of stability potentially due to the non-smoothness of the hinge loss, which further highlights the benefits of our algorithms. We highlight through a one-sided Wilcoxon singed-rank test in Appendix C.4 that performance improvements offered by our model are statistically significant.

6.2 INDUSTRIAL DATA EXPERIMENT

We utilize industrial data from degrading pumps to examine the performance of our models. We explore 5 settings: i) nominal: training data is distributed evenly across clients and classes, ii) client imbalance: training data distribution across clients is $[70\%, 15\%, 10\%, 5\%]$, iii) class imbalance: training data distribution across classes is $[90\%, 10\%]$, iv) client+class imbalance: a combination of the previous two settings, and v) noisy labels: 15% of the training labels are flipped. This experiment contains two distinct components:

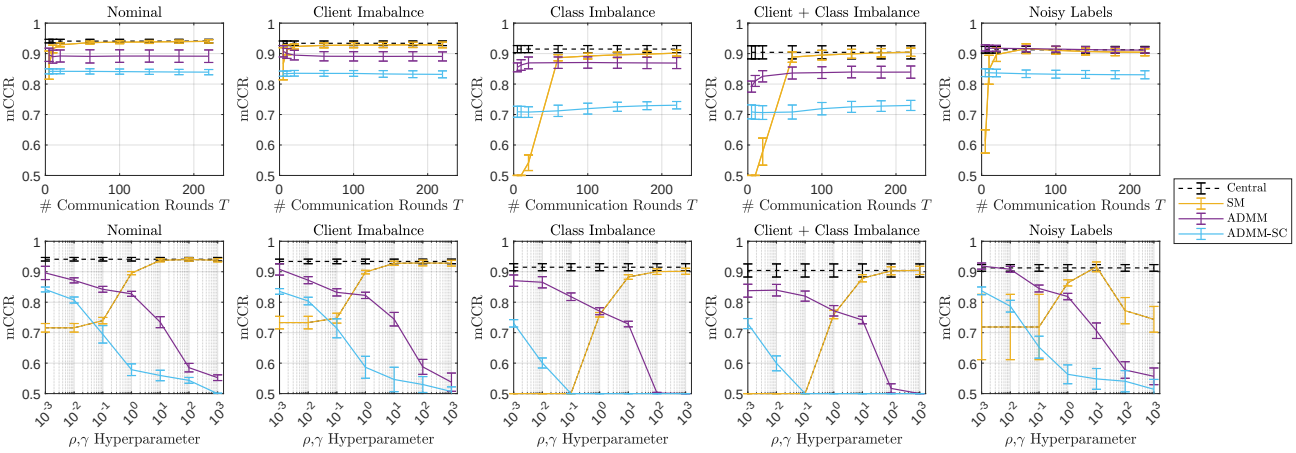

Figure 1: mCCR vs. the Global Hyperparameters, Comparing Our Proposed Methods to the Best-Performing Central Model.

1) a sensitivity analysis, and 2) a benchmarking study. Performance is evaluated in terms of mean correct classification rate (mCCR) and F-1 score in the sensitivity analysis and benchmarking study, respectively.

Dataset. We utilize industrial data generated via a physics-driven pump model [Mathworks, n.d.]. The data contains healthy and leak fault classes. Client heterogeneity is simulated by generating different fault severities per client. We use $G = 4$, $P = 14$, $N = 400$, and $N_{Test} = 1000$ test samples. The test set contains 500 healthy samples, and 125 samples from each of the 4 fault severities.

6.2.1 Sensitivity Analysis

Baseline. We compare our models to the the central DR-SVM benchmark by Shafieezadeh-Abadeh et al. [2019].

Hyperparameters. In this part of the experiment, we plot each algorithm's performance as a function of its hyperparameters. We examine global and local hyperparameters, and vary each of them separately. The global ones are initial step-size γ for the SM algorithm, scale parameter ρ for the ADMM algorithms, and total number of rounds T for all algorithms. The local hyperparameters are the label flipping cost κ_g, and the local Wasserstein ball radius factor β_g, where the radius is $\varepsilon_g = \frac{1}{\beta_g} N_g$. This is used as a simplifying heuristic to relate the radius to the number of training samples. We also vary the central's ε and κ, showing only the best performance as a benchmark line on the plots.

Results. The *global hyperparameters* effects are shown in Figure 1. The SM often obtains a higher peak performance in most settings than ADMM, however, it can require more communication rounds to do so. This is highlighted in the 'class' imbalance and 'client + class' imbalance settings. The SM algorithm is also relatively stable to the choice of γ, and maintains peak performance across a wide range of values. However, the ADMM algorithm is sensitive to ρ, with performance rapidly decreasing as ρ increases. This suggests that ADMM may require more involved global hyperparameter tuning in practice, but can achieve its peak performance in fewer communication rounds. As hypothesized in Remark 2, we observe that ADMM largely outperforms ADMM-SC due to the additional strongly convex regularization terms. Finally, we also observe that SM and ADMM slightly outperform the best-performing central model in the noisy labels case. This can likely be attributed to our novel ambiguity set's improved uncertainty modeling capability.

The *local hyperparameter* effects are shown in Figure 2. Generally, model performance improves as the radius of the local Wasserstein balls decreases (by increasing β_g). This suggests that performance degrades with larger local Wasserstein balls due to over-conservatism. However, in noisy labels settings, performance of the SM model deteriorates as the local radius decreases. This suggests the need for larger local ambiguity sets to adequately capture label uncertainty. We also observe that the SM model is highly sensitive to the local radius and κ_g in noisy label settings, whereas the ADMM achieves its best performance across a broader range of hyperparameter values. This suggests the need for local hyperparameter fine tuning if SM is used in an application with highly uncertain labels. Moreover, it can be seen that in all other settings, ADMM's performance tends to improve as κ_g increases, which is to be expected, since lower κ_g implies greater anticipation of label uncertainty, and thus over-conservatism. Similar to our observation in the global hyperparameter experiments, we again observe the suboptimality of the ADMM-SC, which underscores the sacrifice in model performance that is associated with enforcing guaranteed convergence.

6.2.2 Benchmarking

Baselines. We utilize the same benchmark models utilized in the UCI data experiment.

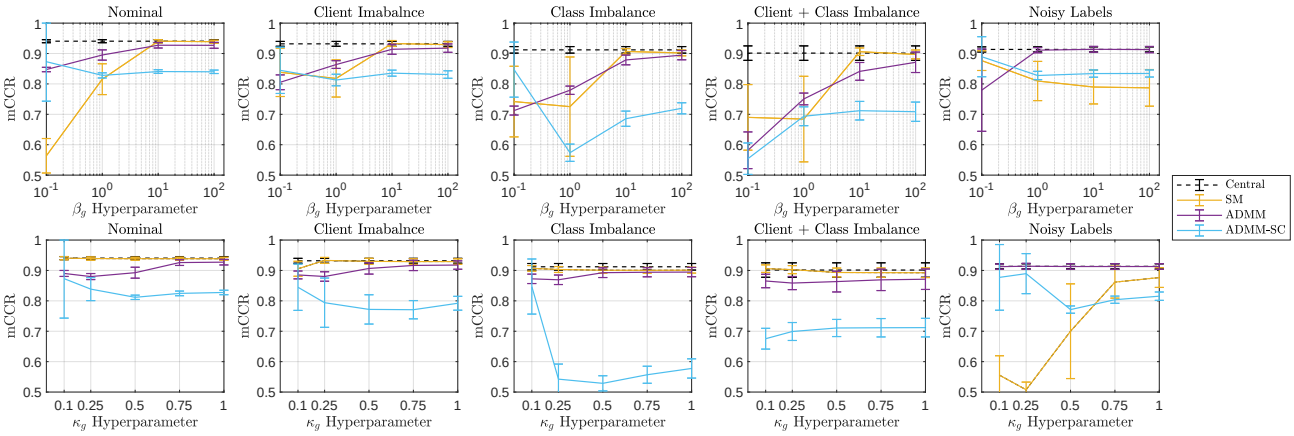

Figure 2: mCCR vs. the Local Hyperparameters, Comparing Our Proposed Methods to the Best-Performing Central Model.

Table 2: F-1 Score Attained by Classification Models on Industrial Dataset in 5 Settings.

Model	Nominal	Client Imbalance	Class Imbalance	Client + Class Imbalance	Noisy Labels
Central (DR-SVM)	$.939 \pm .004$	$\mathbf{.930} \pm .012$	$\mathbf{.903} \pm .012$	$\mathbf{.901} \pm .017$	$\mathbf{.908} \pm .011$
FedSGD (ℓ_2-SVM)	$.886 \pm .008$	$.887 \pm .007$	$.675 \pm .014$	$.685 \pm .030$	$.861 \pm .009$
FedAvg (ℓ_2-SVM)	$.923 \pm .006$	$.919 \pm .010$	$.866 \pm .018$	$.845 \pm .059$	$.894 \pm .017$
FedProx (ℓ_2-SVM)	$.926 \pm .010$	$.919 \pm .011$	$.862 \pm .019$	$.842 \pm .058$	$.894 \pm .019$
FedDRO (KL)	$.913 \pm .007$	$.914 \pm .010$	$.858 \pm .014$	$.835 \pm .052$	$.883 \pm .012$
SM (FDR-SVM)	$\mathbf{.942} \pm .006$	$\mathbf{.930} \pm .010$	$\mathbf{.883} \pm .022$	$\mathbf{.879} \pm .035$	$.894 \pm .015$
ADMM (FDR-SVM)	$.918 \pm .010$	$.910 \pm .028$	$.868 \pm .018$	$.855 \pm .025$	$\mathbf{.903} \pm .011$
ADMM-SC (FDR-SVM)	$.817 \pm .009$	$.819 \pm .011$	$.638 \pm .020$	$.627 \pm .028$	$.806 \pm .013$

Hyperparameters. We tune the same global and local hyperparamaters discussed in the sensitivity analysis of the industrial data experiment. However, we use 5-fold cross-validation for hyperparameter tuning, and we tune both the global and local hyperparameters simultaneously.

Results. Table 2 shows the results of this study, which are averaged over 10 repetitions. As in the UCI data experiment, we observe that one of our methods obtains the best performance out of all federated approaches for all the settings tested. This underscores the practical impact of our proposed model and its solution algorithms in federated classification problems. Unlike the UCI data experiment, however, we observe that the SM algorithm is the peak performer in most settings in this experiment. This suggests that algorithm choice should be influenced by the dataset under study among other factors. Finally, we observe that for this dataset ADMM-SC is largely outperformed by ADMM. This again provides an example where opting for theoretically guaranteed convergence may come at a sacrifice in model accuracy due to redundant regularization.

7 CONCLUSIONS

We propose an FDR-SVM–a classifier that is *distributionally robust* to uncertainty in training data features and labels, and can be trained in a *federated* fashion. To that end, we propose a novel MoWB ambiguity set, extending the classical Wasserstein ball to the federated setting. We also demonstrate that it exhibits desirable out-of-sample guarantees, and that it allows for problem separability. We then rigorously derive two different algorithms to train our proposed model and analyze their convergence behavior. Finally, we evaluate the performance of our proposed model using various datasets, demonstrating that it frequently outperforms existing methods. Future extensions could utilize the MoWB to robustify other FL models, or explore convergence behavior of our algorithms with partial client participation.

Acknowledgements

We thank the referees and the AC for their valuable input, which allowed us to improve the presentation and content of our paper.

Michael Ibrahim, Heraldo Rozas, and Nagi Gebraeel were supported by the National Aeronautics and Space Administration (NASA), Space Technology Research Institute (STRI) Habitats Optimized for Missions of Exploration (HOME) 'SmartHab' Project (grant No. 80NSSC19K1052). Weijun Xie was supported in part by National Science Foundation (NSF) grant No. 2246414 and Office of Naval Research (ONR) grant No. N00014-24-1-2066.

References

Charalambos D. Aliprantis and Kim C. Border. *Convexity*, pages 251–305. Springer-Verlag Berlin, Heidelberg, 2006. ISBN 978-3-540-32696-0. doi: 10.1007/3-540-29587-9. URL https://doi.org/10.1007/3-540-29587-9.

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019. URL https://arxiv.org/abs/1912.00818.

Adil Bagirov, Napsu Karmitsa, and Marko M. Mäkelä. *Subgradient Methods*, pages 295–297. Springer International Publishing, Cham, 2014. ISBN 978-3-319-08114-4. doi: 10.1007/978-3-319-08114-4_10. URL https://doi.org/10.1007/978-3-319-08114-4_10.

Guzin Bayraksan and David Love. *Data-Driven Stochastic Programming Using Phi-Divergences*, pages 1–19. INFORMS, 10 2015. ISBN 978-0-9843378-8-0. doi: 10.1287/educ.2015.0134.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust Optimization*, volume 28 of *Princeton Series in Applied Mathematics*. Princeton University Press, 2009. ISBN 978-1-4008-3105-0.

D. Bertsekas. *Convex Optimization Theory*. Athena Scientific optimization and computation series. Athena Scientific, 2009. ISBN 9781886529311. URL https://books.google.com/books?id=0H1iQwAACAAJ.

Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004. ISSN 0030364X, 15265463. URL http://www.jstor.org/stable/30036559.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Stephen Boyd, Lin Xiao, and Almir Mutapcic. Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter*, 2004(01), 2003. URL https://web.stanford.edu/class/ee364b/lectures/subgrad_method_notes.pdf.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015. ISSN 1935-8237. doi: 10.1561/2200000050. URL http://dx.doi.org/10.1561/2200000050.

Caihua Chen, Bingsheng He, Yinyu Ye, and Xiaoming Yuan. The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, 155(1):57–79, 2016. URL https://link.springer.com/article/10.1007/s10107-014-0826-5.

Ashish Cherukuri and Jorge Cortés. Cooperative data-driven distributionally robust optimization. *IEEE Transactions on Automatic Control*, 65(10):4400–4407, 2020. doi: 10.1109/TAC.2019.2955031. URL https://ieeexplore.ieee.org/document/8910389.

Ilkay Cınar and Murat Koklu. Rice (Cammeo and Osmancik). UCI Machine Learning Repository, 2019. DOI: https://doi.org/10.24432/C5MW4Z.

Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):595–612, 2010. doi: 10.1287/opre.1090.0741. URL https://doi.org/10.1287/opre.1090.0741.

Wei Deng, Ming-Jun Lai, Zhimin Peng, and Wotao Yin. Parallel multi-block ADMM with O (1/k) convergence. *Journal of Scientific Computing*, 71:712–736, 2017. URL https://link.springer.com/article/10.1007/s10915-016-0318-2.

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Distributionally robust federated averaging. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15111–15122. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/ac450d10e166657ec8f93a1b65ca1b14-Paper.pdf.

Jiahao Du, Na Qin, Deqing Huang, Yiming Zhang, and Xinming Jia. An efficient federated learning framework for machinery fault diagnosis with improved model aggregation and local model training. *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*, 35(7):10086–10097, JUL 2024. doi: 10.1109/TNNLS.2023.3238724. URL https://ieeexplore.ieee.org/document/10035917.

Nabanita Dutta, Palanisamy Kaliannan, and Paramasivam Shanmugam. SVM algorithm for vibration fault diagnosis in centrifugal pump. *INTELLIGENT AUTOMATION AND SOFT COMPUTING*, 35(3):2997–3020, 2023. ISSN 1079-8587. doi: 10.32604/iasc.2023.028704. URL https://www.techscience.com/iasc/v35n3/49369.

Matthias Elter. Mammographic Mass. UCI Machine Learning Repository, 2007. DOI: https://doi.org/10.24432/C53K6Z.

Daniel Faccini, Francesca Maggioni, and Florian A. Potra. Robust and distributionally robust optimization models

for linear support vector machine. *Computers & Operations Research*, 147:105930, 2022. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2022.105930. URL https://www.sciencedirect.com/science/article/pii/S0305054822001861.

Rui Gao and Anton Kleywegt. Distributionally robust stochastic optimization with wasserstein distance. *Mathematics of Operations Research*, 48(2):603–655, 2023. doi: 10.1287/moor.2022.1275. URL https://doi.org/10.1287/moor.2022.1275.

Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Convex Analysis and Minimization Algorithms I*. Springer, 1 edition, 1993.

Feihu Huang, Junyi Li, and Heng Huang. Compositional federated learning: Applications in distributionally robust averaging and meta learning. *ArXiv*, abs/2106.11264, 2021. URL https://api.semanticscholar.org/CorpusID:235490601.

Hamdi Kahraman, Ilhami Colak, and Seref Sagiroglu. User Knowledge Modeling. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C5231X.

L. Kantorovitch. On the translocation of masses. *Doklady Akademii Nauk USSR*, 37:227–229, 1942.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 5132–5143. PMLR, 2020. URL http://proceedings.mlr.press/v119/karimireddy20a/karimireddy20a.pdf.

Prashant Khanduri, Chengyin Li, Rafi Ibn Sultan, Yao Qiang, Joerg Kliewer, and Dongxiao Zhu. Feddro: Federated compositional optimization for distributionally robust learning. *ArXiv*, abs/2311.12652, 2023. URL https://api.semanticscholar.org/CorpusID:265309068.

Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence, 2016. URL https://arxiv.org/abs/1610.02527.

Daniel Kuhn, Peyman Mohajerin Esfahani, Viet Anh Nguyen, and Soroosh Shafieezadeh-Abadeh. *Wasserstein Distributionally Robust Optimization: Theory and Applications in Machine Learning*, chapter 6, pages 130–166. INFORMS, 2019. doi: 10.1287/educ.2019.0198. URL https://pubsonline.informs.org/doi/abs/10.1287/educ.2019.0198.

Tim Tsz-Kit Lau and Han Liu. Wasserstein distributionally robust optimization with wasserstein barycenters. *arXiv preprint arXiv:2203.12136*, 2022. URL https://arxiv.org/abs/2203.12136.

Long Tan Le, Tung-Anh Nguyen, Tuan-Dung Nguyen, Nguyen H Tran, Nguyen Binh Truong, Phuong L Vo, Bui Thanh Hung, and Tuan Anh Le. Distributionally robust federated learning for mobile edge networks. *Mobile Networks and Applications*, pages 1–11, 2024. URL https://link.springer.com/article/10.1007/s11036-024-02316-w.

Royson Lee, Minyoung Kim, Da Li, Xinchi Qiu, Timothy Hospedales, Ferenc Huszar, and Nicholas Lane. FedL2P: Federated learning to personalize. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 14818–14836. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/2fb57276bfbaf1b832d7bfcba36bb41c-Paper-Conference.pdf.

Yaguo Lei, Bin Yang, Xinwei Jiang, Feng Jia, Naipeng Li, and Asoke K. Nandi. Applications of machine learning to machine fault diagnosis: A review and roadmap. *MECHANICAL SYSTEMS AND SIGNAL PROCESSING*, 138, APR 2020. ISSN 0888-3270. doi: 10.1016/j.ymssp.2019.106587. URL https://www.sciencedirect.com/science/article/pii/S0888327019308088.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020. URL https://proceedings.mlsys.org/paper_files/paper/2020/file/1f5fe83998a09396ebe6477d9475ba0c-Paper.pdf.

Tian-Yi Lin, Shi-Qian Ma, and Shu-Zhong Zhang. On the sublinear convergence rate of multi-block admm. *Journal of the Operations Research Society of China*, 3:251–274, 2015. URL https://link.springer.com/article/10.1007/s40305-015-0092-0.

Max Little. Parkinsons. UCI Machine Learning Repository, 2008. DOI: https://doi.org/10.24432/C59C74.

Volker Lohweg. Banknote Authentication. UCI Machine Learning Repository, 2013. DOI: https://doi.org/10.24432/C55P57.

Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated multi-task learning under a mixture of distributions. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman

Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 15434–15447. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/82599a4ec94aca066873c99b4c741ed8-Paper.pdf.

Josey Mathew, Chee Khiang Pang, Ming Luo, and Weng Hoe Leong. Classification of imbalanced data by oversampling in kernel space of support vector machines. *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*, 29(9):4065–4076, SEP 2018. ISSN 2162-237X. doi: 10.1109/TNNLS.2017.2751612. URL https://ieeexplore.ieee.org/document/8064210.

Mathworks. Multi-class fault detection using simulated data, n.d. URL https://www.mathworks.com/help/predmaint/ug/multi-class-fault-detection-using-simulated-data.html.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, pages 1273–1282. PMLR, 2017. URL http://proceedings.mlr.press/v54/mcmahan17a/mcmahan17a.pdf.

Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: performance guarantees and tractable reformulations. *Mathematical Programming*, 171:115–166, 2018. ISSN 1-2. URL https://link.springer.com/article/10.1007/s10107-017-1172-1.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Yurii Nesterov and Arkadii Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994. doi: 10.1137/1.9781611970791. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611970791.

Sloan Nietert, Ziv Goldfeld, and Soroosh Shafiee. Outlier-robust wasserstein DRO. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 62792–62820. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/c67b138497305835e76fdedd48dd4e59-Paper-Conference.pdf.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. URL https://www.jmlr.org/papers/v12/pedregosa11a.html.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/pdf?id=ryxGuJrFvS.

Herbert E Scarf, KJ Arrow, and S Karlin. *A min-max solution of an inventory problem*. Rand Corporation Santa Monica, 1957.

Terry Sejnowski and R. Gorman. Connectionist Bench (Sonar, Mines vs. Rocks). UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C5T01Q.

Aras Selvi, Mohammad Reza Belbasi, Martin Haugh, and Wolfram Wiesemann. Wasserstein logistic regression with mixed features. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 16691–16704. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/6a13cffb5ec4128324f64a186785215b-Paper-Conference.pdf.

Soroosh Shafieezadeh Abadeh, Peyman Mohajerin Mohajerin Esfahani, and Daniel Kuhn. Distributionally robust logistic regression. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/cc1aa436277138f61cda703991069eaf-Paper.pdf.

Soroosh Shafieezadeh-Abadeh, Daniel Kuhn, and Peyman Mohajerin Esfahani. Regularization via mass transportation. *Journal of Machine Learning Research*, 20 (103):1–68, 2019. URL https://jmlr.org/papers/volume20/17-633/17-633.pdf.

Alexander Shapiro. Distributionally robust stochastic programming. *SIAM Journal on Optimization*, 27(4):2258–2275, 2017. doi: 10.1137/16M1058297. URL https://epubs.siam.org/doi/10.1137/16M1058297.

Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczynski. *Lectures on Stochastic Programming: Modeling and Theory, Third Edition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2021.

doi: 10.1137/1.9781611976595. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611976595.

Min Tao and Xiaoming Yuan. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM Journal on Optimization*, 21(1):57–81, 2011. URL https://epubs.siam.org/doi/10.1137/100781894.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7611–7623. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/564127c03caab942e503ee6f810f54fd-Paper.pdf.

William Wolberg, Olvi Mangasarian, Nick Street, and W. Street. Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository, 1995. DOI: https://doi.org/10.24432/C5DW2B.

Bingzhe Wu, Zhipeng Liang, Yuxuan Han, Yatao Bian, Peilin Zhao, and Junzhou Huang. DRFLM: Distributionally robust federated learning with inter-client noise via local mixup. *arXiv preprint arXiv:2204.07742*, 2022. URL https://arxiv.org/abs/2204.07742.

Ting Wu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. Modeling the q-diversity in a min-max play game for robust optimization. *arXiv preprint arXiv:2305.12123*, 2023. URL https://arxiv.org/abs/2305.12123.

Jian Xu, Yi Yan, and Shao-Lun Huang. FedPer++: Toward improved personalized federated learning on heterogeneous and imbalanced data. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 01–08, 2022. URL https://ieeexplore.ieee.org/document/9892585.

Shuming Yang, Changlin Xie, Yuqiang Cheng, Biao Wang, Xunyi Ma, and Zinuo Wang. Data-driven fault diagnosis for rolling bearings based on machine learning and multi-sensor information fusion. *IEEE SENSORS JOURNAL*, 25(2):3452–3464, JAN 15 2025. ISSN 1530-437X. doi: 10.1109/JSEN.2024.3499365. URL https://ieeexplore.ieee.org/document/10768937.

Matteo Zecchin, Marios Kountouris, and David Gesbert. Communication-efficient distributionally robust decentralized learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=tnRRHzZPMq.

FDR-SVM: A Federated Distributionally Robust Support Vector Machine via a Mixture of Wasserstein Balls Ambiguity Set (Supplementary Material)

Michael Ibrahim[1] **Heraldo Rozas**[2] **Nagi Gebraeel**[1] **Weijun Xie**[1]

[1] H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia, USA
[2] Department of Electrical Engineering, University of Chile, Santiago, Chile

CONTENTS

A Additional Background on Wasserstein DRO **15**

B Proofs and Supplementary Theoretical Results **15**

B.1 Preliminary Lemmas . . . 15

B.2 Proofs of Theoretical Results . . . 18

B.2.1 Proof of Proposition 1 . . . 18

B.2.2 Proof of Proposition 2 . . . 19

B.2.3 Proof of Proposition 3 . . . 19

B.2.4 Proof of Theorem 1 . . . 20

B.2.5 Proof of Theorem 2 . . . 20

B.2.6 Proof of Proposition 4 . . . 21

B.2.7 Proof of Proposition 5 . . . 21

B.2.8 Proof of Theorem 3 . . . 22

B.2.9 Proof of Theorem 4 . . . 22

B.3 Supplementary Theoretical Results . . . 23

B.3.1 Strongly Convex ADMM Client Update Problem . . . 23

C Further Experimental Details and Supplementary Results **23**

C.1 Software and Hardware Details . . . 23

C.2 Datasets Utilized . . . 23

C.2.1 UCI Data Experiment . . . 23

C.2.2 Industrial Data Experiment . . . 23

C.3 Hyperparameter Details . . . 24

C.4 UCI Data Experiment Statistical Significance . . . 25

C.5 Scalability Experiment . . . 25

A ADDITIONAL BACKGROUND ON WASSERSTEIN DRO

Distributionally robust optimization has been recently popularized as an intermediate approach between stochastic programming (SP) [Shapiro et al., 2021] and robust optimization (RO) [Ben-Tal et al., 2009]. Indeed, it can be viewed as a stochastic programming problem where the true distribution $\mathbb{P}$ governing the data is unknown. Alternatively, it can be seen as a robust optimization problem where worst-case perturbations of the data distribution are modeled rather than those of individual data points. This makes DRO attractive as it is a method of modeling the uncertainty without requiring knowledge of the true distribution $\mathbb{P}$ (like in SP) or potentially being overly conservative (like in RO) [Bertsimas and Sim, 2004]. DRO relies on defining an ambiguity set $\mathcal{A}$ of distributions, and subsequently minimizing the worst-case risk attained by any distribution $\mathbb{Q}$ within the ambiguity set $\mathcal{A}$. There have been various different methods of defining the ambiguity set in the literature. This includes moment-based methods [Delage and Ye, 2010], which use certain moment properties to define the set, and distance-based methods [Bayraksan and Love, 2015, Kuhn et al., 2019], which define the set as a sphere centered at some reference distribution, and whose radius is in the sense of some distance measure. Commonly used measures include ϕ-divergences (such as KL divergence) [Bayraksan and Love, 2015] and the Wasserstein distance [Kuhn et al., 2019]. Moreover, in most Machine Learning problems, the reference distribution is taken to be the empirical distribution $\widehat{\mathbb{P}}_N$ of the N training data samples.

In our work, we focus on ambiguity sets defined via the type-1 Wasserstein distance. This is because Wasserstein DRO offers many desirable advantages over its counterparts, as demonstrated by Kuhn et al. [2019]. For example, the Wasserstein ambiguity set can contain both discrete and continuous distributions regardless of the structure of the empirical distribution, which cannot be achieved by the KL divergence ambiguity set. Moreover, one can derive out-of-sample performance guarantees using concentration inequalities when using a Wasserstein ambiguity set, which cannot be achieved in moment-based approaches. The type-1 Wasserstein $W_{d,1}$ distance [Kantorovitch, 1942] is commonly referred to as optimal transport metric or earth mover's distance. This is because of its interpretation as the minimum cost of transforming a distribution $\mathbb{Q}$ to $\mathbb{Q}'$. Therefore, it utilizes a transportation cost function $d(\boldsymbol{\xi}, \boldsymbol{\xi}')$ to define the transportation cost function per unit mass from point $\boldsymbol{\xi}$ to point $\boldsymbol{\xi}'$. We can express the type-1 Wasserstein distance mathematically as follows.

$$W_{d,1}(\mathbb{Q}, \mathbb{Q}') \coloneqq \inf_{\pi \in \Pi(\mathbb{Q}, \mathbb{Q}')} \int_{\Xi \times \Xi} d\left(\boldsymbol{\xi}, \boldsymbol{\xi}'\right) \pi(\mathrm{d}\boldsymbol{\xi}, \mathrm{d}\boldsymbol{\xi}'),$$

where $d(\boldsymbol{\xi}, \boldsymbol{\xi}')$ denotes the transportation cost function, and $\Pi(\mathbb{Q}, \mathbb{Q}')$ is the set of all joint distributions of $\boldsymbol{\xi}$ and $\boldsymbol{\xi}'$ with marginals $\mathbb{Q}$ and $\mathbb{Q}'$, respectively. Note that the data in our classification problem is comprised of continuous features $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^P$ and categorical labels $y \in \{-1, +1\}$. Therefore, a commonly used transportation cost function for such setting is

$$d(\boldsymbol{\xi}, \boldsymbol{\xi}') \coloneqq ||\boldsymbol{x} - \boldsymbol{x}'|| + \kappa \mathbb{1}_{\{y \neq y'\}},$$

where $|| \cdot ||$ is any norm on $\mathbb{R}^P$, and κ is the label-flipping cost, treated as a user-defined hyperparameter. This cost function allows us to quantify differences in both the features and labels between samples.

B PROOFS AND SUPPLEMENTARY THEORETICAL RESULTS

B.1 PRELIMINARY LEMMAS

Lemma 1. *Any two real scalars $a, a' \in \mathbb{R}$ obey the following*

$$|\max\{0, a\} - \max\{0, a'\}| \leq |a - a'|.$$

Proof. To see this, consider the following cases:

1. $a, a' \geq 0$. In this case one can directly see that

$$|\max\{0, a\} - \max\{0, a'\}| = |a - a'|$$

2. $a \geq 0,\ a' < 0$. In this case, we have the following:

$$|\max\{0, a\} - \max\{0, a'\}| = a < |a| + |a'| = |a - a'|.$$

3. $a < 0,\ a' \geq 0$. This is symmetric to the previous case.

4. $a < 0,\ a' < 0$. In this case we have the following:

$$|\max\{0, a\} - \max\{0, a'\}| = 0 \leq |a - a'|.$$

□

Lemma 2 (SM Objective Function Convexity)**.** *Suppose Assumption 3 holds and let* $f(\boldsymbol{w}) = \sum_{g=1}^{G} \alpha_g \sup_{\mathbb{Q}_g \in \mathcal{A}^{(g)}_{\varepsilon_g,1,d}(\Xi)} \mathbb{E}^{\mathbb{Q}_g}[\ell_H(\boldsymbol{w}; \boldsymbol{\xi})]$*. Then,* $f(\boldsymbol{w})$ *is convex in* $\boldsymbol{w}$*.*

Proof. Since $\ell_H(\boldsymbol{w}, \boldsymbol{\xi})$ is a maximum of linear terms in $\boldsymbol{w}$, then it is convex in $\boldsymbol{w}$. Moreover, sums, scalar multiplication, taking the supremum, and the expectation are all operations that preserve convexity [Boyd and Vandenberghe, 2004]. Thus, $f(\boldsymbol{w})$ is convex in $\boldsymbol{w}$. □

Lemma 3 (SM Objective Function Lipschitz Continuity)**.** *Suppose Assumption 3 holds and let* $f(\boldsymbol{w}) = \sum_{g=1}^{G} \alpha_g \sup_{\mathbb{Q}_g \in \mathcal{A}^{(g)}_{\varepsilon_g,1,d}(\Xi)} \mathbb{E}^{\mathbb{Q}_g}[\ell_H(\boldsymbol{w}; \boldsymbol{\xi})]$*. Then,* $f(\boldsymbol{w})$ *is Lipschitz continuous in* $\boldsymbol{w}$*.*

Proof. As discussed by Shafieezadeh-Abadeh et al. [2019], if assumption 3 holds then one can obtain the discrete distribution described in (6) that attains the worst case risk. Therefore, we have the following:

$$\begin{aligned} f(\boldsymbol{w}) &= \sum_{g=1}^{G} \alpha_g f_g(\boldsymbol{w}) \\ &:= \sum_{g=1}^{G} \alpha_g \left(\frac{1}{N_g} \sum_{n_g=1}^{N_g} {\beta^+_{n_g}}^* \ell_H(\boldsymbol{w}; (\widehat{y}_{n_g}, \widehat{\boldsymbol{z}}^+_{n_g})) + {\beta^-_{n_g}}^* \ell_H(\boldsymbol{w}; (-\widehat{y}_{n_g}, \widehat{\boldsymbol{z}}^-_{n_g})) \right), \end{aligned}$$

Now, suppose we have $\boldsymbol{w}$ and $\boldsymbol{w}'$ which correspond to worst-case distributions characterized by $({\beta^\pm_{n_g}}^*, \widehat{\boldsymbol{z}}^\pm_{N_g})$ and $(\beta^{\pm\prime *}_{n_g}, \widehat{\boldsymbol{z}}^{\pm\prime}_{N_g})$, respectively. Then we can write the following:

$$\begin{aligned} &|f_g(\boldsymbol{w}) - f_g(\boldsymbol{w}')| \\ &= \frac{1}{N_g} \Bigg| \sum_{n_g=1}^{N_g} \left[{\beta^+_{n_g}}^* \ell_H(\boldsymbol{w}; (\widehat{y}_{n_g}, \widehat{\boldsymbol{z}}^+_{n_g})) + {\beta^-_{n_g}}^* \ell_H(\boldsymbol{w}; (-\widehat{y}_{n_g}, \widehat{\boldsymbol{z}}^-_{n_g})) \right] \\ &\qquad - \sum_{n_g=1}^{N_g} \left[\beta^{+\prime *}_{n_g} \ell_H(\boldsymbol{w}'; (\widehat{y}_{n_g}, \widehat{\boldsymbol{z}}^{+\prime}_{n_g})) + \beta^{-\prime *}_{n_g} \ell_H(\boldsymbol{w}'; (-\widehat{y}_{n_g}, \widehat{\boldsymbol{z}}^{-\prime}_{n_g})) \right] \Bigg| && \text{(10a)} \\ &\leq \frac{1}{N_g} \left| \sum_{n_g=1}^{N_g} \left[{\beta^+_{n_g}}^* \ell_H(\boldsymbol{w}; (\widehat{y}_{n_g}, \widehat{\boldsymbol{z}}^+_{n_g})) - \beta^{+\prime *}_{n_g} \ell_H(\boldsymbol{w}'; (\widehat{y}_{n_g}, \widehat{\boldsymbol{z}}^{+\prime}_{n_g})) \right] \right| \\ &\qquad + \left| \sum_{n_g=1}^{N_g} \left[{\beta^-_{n_g}}^* \ell_H(\boldsymbol{w}; (-\widehat{y}_{n_g}, \widehat{\boldsymbol{z}}^-_{n_g})) - \beta^{-\prime *}_{n_g} \ell_H(\boldsymbol{w}'; (-\widehat{y}_{n_g}, \widehat{\boldsymbol{z}}^{-\prime}_{n_g})) \right] \right| && \text{(10b)} \\ &\leq \frac{1}{N_g} \sum_{n_g=1}^{N_g} \Bigg[\left| {\beta^+_{n_g}}^* \ell_H(\boldsymbol{w}; (\widehat{y}_{n_g}, \widehat{\boldsymbol{z}}^+_{n_g})) - \beta^{+\prime *}_{n_g} \ell_H(\boldsymbol{w}'; (\widehat{y}_{n_g}, \widehat{\boldsymbol{z}}^{+\prime}_{n_g})) \right| \\ &\qquad + \left| {\beta^-_{n_g}}^* \ell_H(\boldsymbol{w}; (-\widehat{y}_{n_g}, \widehat{\boldsymbol{z}}^-_{n_g})) - \beta^{-\prime *}_{n_g} \ell_H(\boldsymbol{w}'; (-\widehat{y}_{n_g}, \widehat{\boldsymbol{z}}^{-\prime}_{n_g})) \right| \Bigg] && \text{(10c)} \\ &\leq \frac{1}{N_g} \sum_{n_g=1}^{N_g} \Bigg[\left| {\beta^+_{n_g}}^* \ell_H(\boldsymbol{w}; (\widehat{y}_{n_g}, \widehat{\boldsymbol{z}}^+_{n_g})) - {\beta^+_{n_g}}^* \ell_H(\boldsymbol{w}'; (\widehat{y}_{n_g}, \widehat{\boldsymbol{z}}^+_{n_g})) \right| \\ &\qquad + \left| {\beta^-_{n_g}}^* \ell_H(\boldsymbol{w}; (-\widehat{y}_{n_g}, \widehat{\boldsymbol{z}}^-_{n_g})) - {\beta^-_{n_g}}^* \ell_H(\boldsymbol{w}'; (-\widehat{y}_{n_g}, \widehat{\boldsymbol{z}}^-_{n_g})) \right| \Bigg] && \text{(10d)} \end{aligned}$$

$$\leq \frac{1}{N_g}\sum_{n_g=1}^{N_g}\Bigg[\Big|\beta_{n_g}^{+\,*}\max\{0,1-\widehat{y}_{n_g}\cdot \boldsymbol{w}^{\mathsf{T}}\widehat{\boldsymbol{z}}_{n_g}^{+}\}-\beta_{n_g}^{+\,*}\max\{0,1-\widehat{y}_{n_g}\cdot \boldsymbol{w}'^{\mathsf{T}}\widehat{\boldsymbol{z}}_{n_g}^{+}\}\Big|$$

$$+\Big|\beta_{n_g}^{-\,*}\max\{0,1+\widehat{y}_{n_g}\cdot \boldsymbol{w}^{\mathsf{T}}\widehat{\boldsymbol{z}}_{n_g}^{-}\}-\beta_{n_g}^{-\,*}\max\{0,1+\widehat{y}_{n_g}\cdot \boldsymbol{w}'^{\mathsf{T}}\widehat{\boldsymbol{z}}_{n_g}^{-}\}\Big|\Bigg] \tag{10e}$$

$$=\frac{1}{N_g}\sum_{n_g=1}^{N_g}\left[\Big|(\boldsymbol{w}'-\boldsymbol{w})^{\mathsf{T}}(\beta_{n_g}^{+\,*}\cdot\widehat{y}_{n_g}\cdot\widehat{\boldsymbol{z}}_{n_g}^{+})\Big|+\Big|(\boldsymbol{w}-\boldsymbol{w}')^{\mathsf{T}}(\beta_{n_g}^{-\,*}\cdot\widehat{y}_{n_g}\cdot\widehat{\boldsymbol{z}}_{n_g}^{-})\Big|\right] \tag{10f}$$

$$\leq \frac{1}{N_g}\sum_{n_g=1}^{N_g}||\boldsymbol{w}-\boldsymbol{w}'||\left(\Big|\Big|\beta_{n_g}^{+\,*}\cdot\widehat{y}_{n_g}\cdot\widehat{\boldsymbol{z}}_{n_g}^{+}\Big|\Big|_*+\Big|\Big|\beta_{n_g}^{-\,*}\cdot\widehat{y}_{n_g}\cdot\widehat{\boldsymbol{z}}_{n_g}^{-}\Big|\Big|_*\right) \tag{10g}$$

$$=||\boldsymbol{w}-\boldsymbol{w}'||\left[\frac{1}{N_g}\sum_{n_g=1}^{N_g}\Big|\Big|\beta_{n_g}^{+\,*}\cdot\widehat{y}_{n_g}\cdot\widehat{\boldsymbol{z}}_{n_g}^{+}\Big|\Big|_*+\Big|\Big|\beta_{n_g}^{-\,*}\cdot\widehat{y}_{n_g}\cdot\widehat{\boldsymbol{z}}_{n_g}^{-}\Big|\Big|_*\right], \tag{10h}$$

where (10b) and (10c) follow from the triangle inequality, and (10d) follows by noting that the distribution characterized by $(\beta_{n_g}^{\pm\prime *},\widehat{\boldsymbol{z}}_{N_g}^{\pm\prime})$ maximizes the expected risk with respect to $\boldsymbol{w}'$, thus the distribution characterized by $(\beta_{n_g}^{\pm\,*},\widehat{\boldsymbol{z}}_{N_g}^{\pm})$ will at most attain the same risk with respect to $\boldsymbol{w}'$. Additionally, (10e) follows from the definition of the hinge loss function, (10f) follows from Lemma 1, and (10g) follows from the Cauchy-Schwarz inequality, where $||\cdot||_*$ is the dual norm of $||\cdot||$ used to measure distances in the space of $\boldsymbol{w}$. Given the previous, we can obtain the final result as follows:

$$|f(\boldsymbol{w})-f(\boldsymbol{w}')|=\left|\sum_{g=1}^{G}\alpha_g f_g(\boldsymbol{w})-\sum_{g=1}^{G}\alpha_g f_g(\boldsymbol{w}')\right| \tag{11a}$$

$$\leq \sum_{g=1}^{G}\alpha_g|f_g(\boldsymbol{w})-f_g(\boldsymbol{w}')| \tag{11b}$$

$$\leq ||\boldsymbol{w}-\boldsymbol{w}'||\sum_{g=1}^{G}\alpha_g\,\mathrm{Lip}(f_g(\boldsymbol{w})), \tag{11c}$$

where (11b) follows from the triangle inequality and $\mathrm{Lip}(f_g(\boldsymbol{w}))$ is taken from (10h). □

Lemma 4 (SM Objective Function Coercivity)**.** *Suppose Assumption 3 holds and let* $f(\boldsymbol{w})=\sum_{g=1}^{G}\alpha_g\sup_{\mathbb{Q}_g\in\mathcal{A}_{\varepsilon_g,1,d}^{(g)}(\Xi)}\mathbb{E}^{\mathbb{Q}_g}[\ell_H(\boldsymbol{w};\boldsymbol{\xi})]$*. Then,* $f(\boldsymbol{w})$ *is coercive in* $\boldsymbol{w}$*.*

Proof. We begin out proof by studying each of the individual terms $f_g(\boldsymbol{w})$ as follows

$$f_g(\boldsymbol{w})\coloneqq\sup_{\mathbb{Q}_g\in\mathcal{A}_{\varepsilon_g,1,d}^{(g)}(\Xi)}\mathbb{E}^{\mathbb{Q}_g}[\ell_H(\boldsymbol{w};\boldsymbol{\xi})]$$

$$=\inf_{\lambda_g\geq 0}\lambda_g\varepsilon_g+\frac{1}{N_g}\sum_{n_g=1}^{N_g}\sup_{\boldsymbol{\xi}\in\Xi}\Big\{\ell_H(\boldsymbol{w};\boldsymbol{\xi})-\lambda_g d(\boldsymbol{\xi},\widehat{\boldsymbol{\xi}}_{n_g})\Big\} \tag{12a}$$

$$=\begin{cases}\inf\limits_{\lambda_g\geq 0,s_{n_g}} & \lambda_g\varepsilon_g+\dfrac{1}{N_g}\displaystyle\sum_{n_g=1}^{N_g}s_{n_g}\\ \text{s. t.} & \sup\limits_{\boldsymbol{\xi}\in\Xi}\Big\{\ell_H(\boldsymbol{w};\boldsymbol{\xi})-\lambda_g d(\boldsymbol{\xi},\widehat{\boldsymbol{\xi}}_{n_g})\Big\}\leq s_{n_g}\quad\forall n_g\in[N_g]\end{cases} \tag{12b}$$

$$=\begin{cases}\inf\limits_{\lambda_g\geq 0,s_{n_g}} & \lambda_g\varepsilon_g+\dfrac{1}{N_g}\displaystyle\sum_{n_g=1}^{N_g}s_{n_g}\\ \text{s. t.} & \sup\limits_{\boldsymbol{x}\in\mathcal{X}}\big\{\ell_H(\boldsymbol{w};(\boldsymbol{x},\widehat{y}_{n_g}))-\lambda_g||\boldsymbol{x}-\widehat{\boldsymbol{x}}_{n_g}||\big\}\leq s_{n_g}\qquad\forall n_g\in[N_g]\\ & \sup\limits_{\boldsymbol{x}\in\mathcal{X}}\big\{\ell_H(\boldsymbol{w};(\boldsymbol{x},-\widehat{y}_{n_g}))-\lambda_g||\boldsymbol{x}-\widehat{\boldsymbol{x}}_{n_g}||\big\}-\kappa\lambda_g\leq s_{n_g}\quad\forall n_g\in[N_g]\end{cases} \tag{12c}$$

$$
= \begin{cases} \inf\limits_{\lambda_g, s_{n_g}} & \lambda_g \varepsilon_g + \dfrac{1}{N_g} \displaystyle\sum_{n_g=1}^{N_g} s_{n_g} \\ \text{s. t.} & \ell_H(\boldsymbol{w}; (\widehat{\boldsymbol{x}}_{n_g}, \widehat{y}_{n_g})) \leq s_{n_g} \qquad \forall n_g \in [N_g] \\ & \ell_H(\boldsymbol{w}; (\widehat{\boldsymbol{x}}_{n_g}, -\widehat{y}_{n_g})) - \kappa\lambda_g \leq s_{n_g} \quad \forall n_g \in [N_g] \\ & \lambda_g \geq ||\boldsymbol{w}||_* \end{cases} \tag{12d}
$$

where (12a) follows from the strong duality result presented by Shafieezadeh-Abadeh et al. [2019], Kuhn et al. [2019], (12b) is obtained through the introduction of slack variables and moving the maximization problems to the constraints, and (12c) is obtained through the definition of the separable transportation cost function (3) and by noting that $y \in \{-1, +1\}$, and finally (12d) is obtained by recalling that the hinge loss function $\ell_H(\boldsymbol{w}; \boldsymbol{\xi})$ is convex and Lipschitz continuous in $\boldsymbol{x}$, and therefore it follows from Lemma A.3 in [Shafieezadeh-Abadeh et al., 2019] that

$$
\sup_{\boldsymbol{x} \in \mathcal{X}} \{\ell_H(\boldsymbol{w}; (\boldsymbol{x}, y)) - \lambda_g ||\boldsymbol{x} - \widehat{\boldsymbol{x}}||\} = \begin{cases} \ell_H(\boldsymbol{w}; (\widehat{\boldsymbol{x}}, y)) & \text{if} ||\boldsymbol{w}||_* \leq \lambda_g \\ +\infty & \text{otherwise,} \end{cases}
$$

where $||\cdot||_*$ is the dual to the norm utilized in the definition of the transportation cost function (3). Therefore, as $||\boldsymbol{w}||_* \to \infty$, we get that $\lambda_g \to \infty$. Since λ_g has a positive sign in the objective function of (12d), then $f_g(\boldsymbol{w}) \to +\infty$ as $\lambda_g \to \infty$. This implies that $f(\boldsymbol{w}) = \sum_g^G f_g(\boldsymbol{w})$ is a coercive function of $\boldsymbol{w}$, since $f(\boldsymbol{w}) \to +\infty$ as $||\boldsymbol{w}||_* \to \infty$. □

Lemma 5 (ADMM Objective Properties)**.** *Let $f(\boldsymbol{w}_g) = \sup_{\mathbb{Q}_g \in \mathcal{A}^{(g)}_{\varepsilon_g,1,d}(\Xi)} \mathbb{E}^{\mathbb{Q}_g}[\ell_H(\boldsymbol{w}_g; \boldsymbol{\xi})]$, then $f(\boldsymbol{w}_g)$ is a closed proper convex function in $\boldsymbol{w}_g$.*

Proof. Recall that $\ell_H(\boldsymbol{w}_g, \boldsymbol{\xi})$ is convex in $\boldsymbol{w}_g$, and taking the supremum and expectation are operations that preserve convexity [Boyd and Vandenberghe, 2004], thus $f(\boldsymbol{w}_g)$ is convex in $\boldsymbol{w}_g$. Now, note that

$$
\ell_H(\boldsymbol{w}_g, \boldsymbol{\xi}) \geq 0 \Rightarrow f(\boldsymbol{w}_g) \geq 0 \quad \forall \boldsymbol{w}_g \in \mathbb{R}^P.
$$

Now, observe that $f(\mathbf{0}) = 1$ since $\ell_H(\mathbf{0}, \boldsymbol{\xi}) = 1 \; \forall \boldsymbol{\xi} \in \Xi$. Since $f(\boldsymbol{w}_g) > -\infty$ and it has a nonempty effective domain, then it is proper convex [Aliprantis and Border, 2006]. Finally, since $f(\boldsymbol{w}_g) : \mathbb{R}^P \to (-\infty, \infty]$ is proper convex, then it is continuous by Proposition 1.3.11 in [Bertsekas, 2009]. This implies the closedness of the function. □

B.2 PROOFS OF THEORETICAL RESULTS

B.2.1 Proof of Proposition 1

Proof.

$$
\inf_{\boldsymbol{w}} \sup_{\mathbb{Q} \in \mathcal{A}_G} \mathbb{E}^{\mathbb{Q}}[\ell_H(\boldsymbol{w}; \boldsymbol{\xi})] = \inf_{\boldsymbol{w}} \sup_{\left\{\mathbb{Q}_g \in \mathcal{A}^{(g)}_{\varepsilon_g,1,d}(\Xi)\right\}_{g=1}^G} \mathbb{E}^{\sum_{g=1}^G \alpha_g \mathbb{Q}_g}[\ell_H(\boldsymbol{w}; \boldsymbol{\xi})] \tag{13a}
$$

$$
= \inf_{\boldsymbol{w}} \sup_{\left\{\mathbb{Q}_g \in \mathcal{A}^{(g)}_{\varepsilon_g,1,d}(\Xi)\right\}_{g=1}^G} \sum_{g=1}^G \alpha_g \mathbb{E}^{\mathbb{Q}_g}[\ell_H(\boldsymbol{w}; \boldsymbol{\xi})] \tag{13b}
$$

$$
= \inf_{\boldsymbol{w}} \sum_{g=1}^G \alpha_g \sup_{\mathbb{Q}_g \in \mathcal{A}^{(g)}_{\varepsilon_g,1,d}(\Xi)} \mathbb{E}^{\mathbb{Q}_g}[\ell_H(\boldsymbol{w}; \boldsymbol{\xi})], \tag{13c}
$$

where (13a) follows from the definition of the global ambiguity $\mathcal{A}_G$ set in (4), (13b) follows from the Law of Total Expectation, and (13c) follows by recognizing that the maximization problems are separable due to each decision variable only affecting its corresponding term. □

B.2.2 Proof of Proposition 2

Proof. Suppose the assumptions in the Proposition statement hold. Then, as demonstrated by Kuhn et al. [2019] we have the following

$$\mathbb{P}^{N_g}\{\mathbb{P}_g \in \mathcal{A}^{(g)}_{\varepsilon_g,1,d}(\Xi)\} \geq (1-\eta_g)$$

Therefore, we can obtain the following.

$$\mathbb{P}^N\{\mathbb{P} \in \mathcal{A}_G\} \geq \prod_{g=1}^{G} \mathbb{P}^{N_g}\{\mathbb{P}_g \in \mathcal{A}^{(g)}_{\varepsilon_g,1,d}(\Xi)\} \tag{14a}$$

$$\geq \prod_{g=1}^{G}(1-\eta_g), \tag{14b}$$

where (14a) follows by noting that the local data and Wasserstein balls at all G clients are mutually independent, and that $\mathbb{P} = \sum_{g=1}^{G} \alpha_g \mathbb{P}_g$. Furthermore, note that (14a) contains an inequality instead of an equality as there is no guarantee that $\mathbb{P}$ cannot be constructed as a mixture of distributions from the local Wasserstein balls $\{\mathcal{A}^{(g)}_{\varepsilon_g,1,d}(\Xi)\}_{g=1}^{G}$. Therefore, we have that

$$\mathbb{P}^N\left\{\mathbb{P} \in \mathcal{A}_G \cap \mathbb{P}_g \notin \{\mathcal{A}^{(g)}_{\varepsilon_g,1,d}(\Xi)\}_{g=1}^{G}\right\} \neq 0.$$

□

B.2.3 Proof of Proposition 3

Proof. Firstly let us note the following:

$$\partial \sup_{\mathbb{Q}_g \in \mathcal{A}^{(g)}_{\varepsilon_g,1,d}(\Xi)} \mathbb{E}^{\mathbb{Q}_g}[\ell_H(\boldsymbol{w};\boldsymbol{\xi})] \supseteq \partial \mathbb{E}^{\mathbb{Q}_g^*}[\ell_H(\boldsymbol{w};\boldsymbol{\xi})] \tag{15a}$$

$$= \mathbb{E}^{\mathbb{Q}_g^*}[\partial \ell_H(\boldsymbol{w};\boldsymbol{\xi})], \tag{15b}$$

where 15a follows from Lemma 4.4.1 in [Hiriart-Urruty and Lemaréchal, 1993] by the fact that $\mathbb{Q}_g^*$ is a maximizer of the supremum on the left hand side, and 15a follows from the fact that $\ell_H(\boldsymbol{w};\boldsymbol{\xi})$ is convex and integrable, and $\mathbb{Q}_g^*$ is a discrete distribution. Thus $\mathbb{E}^{\mathbb{Q}_g^*}[\cdot]$ is a weighted sum.

Now, let us introduce the functions $h_1(\boldsymbol{w})$, and $h_2(\boldsymbol{w})$ to simplify notation as follows:

$$\mathbb{E}^{\mathbb{Q}_g^*}[\ell_H(\boldsymbol{w};\boldsymbol{\xi})] = \frac{1}{N_g}\sum_{n_g=1}^{N_g} {\beta^+_{n_g}}^* \ell_H(\boldsymbol{w};(\widehat{\boldsymbol{z}}^+_{n_g}, \widehat{y}_{n_g})) + {\beta^-_{n_g}}^* \ell_H(\boldsymbol{w};(\widehat{\boldsymbol{z}}^-_{n_g}, -\widehat{y}_{n_g})) \tag{16a}$$

$$:= \frac{1}{N_g}\sum_{n_g=1}^{N_g} h_1(\boldsymbol{w}) + h_2(\boldsymbol{w}), \tag{16b}$$

where 16a uses the definition of $\mathbb{Q}_g^*$ from Equation 6, and $\widehat{\boldsymbol{z}}^{\pm}_{n_g} = \widehat{\boldsymbol{x}}_{n_g} - {\boldsymbol{q}^{\pm}_{n_g}}^*/{\beta^{\pm}_{n_g}}^*$. Now, observe that we can write the subdifferentials of $h_1(\boldsymbol{w})$ and $h_2(\boldsymbol{w})$ with respect to $\boldsymbol{w}$ as follows:

$$\partial h_1(\boldsymbol{w}) = \begin{cases} \mathbf{0} & \text{if } 1-\widehat{y}_{n_g}\cdot \boldsymbol{w}^\mathsf{T}\widehat{\boldsymbol{z}}^+_{n_g} < 0 \\ -{\beta^+_{n_g}}^*\widehat{y}_{n_g}\widehat{\boldsymbol{z}}^+_{n_g} & \text{if } 1-\widehat{y}_{n_g}\cdot \boldsymbol{w}^\mathsf{T}\widehat{\boldsymbol{z}}^+_{n_g} > 0 \\ \text{conv}\left(\{\mathbf{0}, -{\beta^+_{n_g}}^*\widehat{y}_{n_g}\widehat{\boldsymbol{z}}^+_{n_g}\}\right) & \text{if } 1-\widehat{y}_{n_g}\cdot \boldsymbol{w}^\mathsf{T}\widehat{\boldsymbol{z}}^+_{n_g} = 0 \end{cases}$$

$$\partial h_2(\boldsymbol{w}) = \begin{cases} \mathbf{0} & \text{if } 1+\widehat{y}_{n_g}\cdot \boldsymbol{w}^\mathsf{T}\widehat{\boldsymbol{z}}^-_{n_g} < 0 \\ {\beta^-_{n_g}}^*\widehat{y}_{n_g}\widehat{\boldsymbol{z}}^-_{n_g} & \text{if } 1+\widehat{y}_{n_g}\cdot \boldsymbol{w}^\mathsf{T}\widehat{\boldsymbol{z}}^-_{n_g} > 0 \\ \text{conv}\left(\{\mathbf{0}, {\beta^-_{n_g}}^*\widehat{y}_{n_g}\widehat{\boldsymbol{z}}^-_{n_g}\}\right) & \text{if } 1+\widehat{y}_{n_g}\cdot \boldsymbol{w}^\mathsf{T}\widehat{\boldsymbol{z}}^-_{n_g} = 0 \end{cases}$$

Therefore, we can use the previous result to obtain the following:

$$\mathbb{E}^{\mathbb{Q}_g^*}[\partial \ell_H(\boldsymbol{w};\boldsymbol{\xi})] = \frac{1}{N_g}\sum_{n_g=1}^{N_g} \partial h_1(\boldsymbol{w}) + \partial h_2(\boldsymbol{w}),$$

where we use the Minkowski sum in the above equation. □

B.2.4 Proof of Theorem 1

Proof. As shown by Nesterov [2013], the subgradient method guarantees convergence assuming the following conditions are met.

1. The objective function is convex.
2. The objective function is Lipschitz continuous.
3. The step-size diminishes at an appropriate rate as stated in the theorem statement.
4. The distance between any optimal solution $\boldsymbol{w}^*$ and any initial solution $\boldsymbol{w}^{(0)}$ is bounded from above. That is $||\boldsymbol{w}^* - \boldsymbol{w}^{(0)}|| \leq C$, where $C \in \mathbb{R}$ need not be known.

Note that we verify the first two conditions in Lemmas 2 and 3, whereas the third condition can be ensured by selecting an appropriately diminishing step-size sequence, as exemplified in the theorem statement. In examining the fourth condition, we note that it is readily satisfied through the coercivity of the objective function, which we prove in Lemma 4. To see this, first note that $f(\mathbf{0}) = 1$, and by definition $\inf_{\boldsymbol{w}} f(\boldsymbol{w}) \leq f(\mathbf{0})$. Suppose we have a set $\mathcal{W} = \{\boldsymbol{w} : f(\boldsymbol{w}) \leq f(\mathbf{0})\}$. We know that for $\boldsymbol{w}^*$ to be a minimizer of $f(\boldsymbol{w})$, it must be that $\boldsymbol{w}^* \in \mathcal{W}$. Suppose further that the set $\mathcal{W}$ contains a sequence $\boldsymbol{w}_i$ such that $||\boldsymbol{w}_i|| \to \infty$. This results in a contradiction, as

$$||\boldsymbol{w}_i|| \to \infty \Rightarrow f(\boldsymbol{w}_i) \to +\infty \Rightarrow \boldsymbol{w}_i \notin \mathcal{W},$$

which follows from the coercivity of $f(\boldsymbol{w})$. Thus, there must exist some constant $R \in \mathbb{R}$ such that

$$\boldsymbol{w} \in \mathcal{W} \Rightarrow ||\boldsymbol{w}|| \leq R.$$

Finally, suppose we choose any finite initializer $\boldsymbol{w}^{(0)}$ for the SM algorithm. Then, by the triangle inequality we have

$$||\boldsymbol{w}^* - \boldsymbol{w}^{(0)}|| \leq R + ||\boldsymbol{w}^{(0)}||,$$

proving that the distance between any initializer $\boldsymbol{w}^{(0)}$ and any optimizer $\boldsymbol{w}^*$ is indeed bounded from above. □

B.2.5 Proof of Theorem 2

Proof. We first examine the time complexity of problem (7) that each client g solves at each iteration t. When the ℓ_∞-norm is used in (7), the problem becomes a Linear Program (LP) with $4N_gP + 2N_g$ decision variables (including slack variables) and $4N_gP + 7N_g$ constraints, where N_g is the number of training samples at the gth client. Solving the problem via the barrier method with the log barrier function and Newton updates requires $\mathcal{O}(\sqrt{(C)}\log(\epsilon_2^{-1}))$ iterations to reach an ϵ_2-solution [Nesterov and Nemirovskii, 1994], where C is the number of constraints. Moreover, each iteration has an arithmetic complexity of $\mathcal{O}(CD^2)$, where D is the number of decision variables. Therefore, the theoretical worst-case time complexity of solving the problem in (7) is:

$$\mathcal{O}([4N_gP + 7N_g]^{1.5}[4N_gP + 2N_g]^2 \log(\epsilon_2^{-1})).$$

By eliminating scalar multipliers and constants, we arrive at the following simplified expression,

$$\mathcal{O}([N_gP]^{3.5}\log(\epsilon_2^{-1})).$$

Since all clients can solve their local problems in parallel, and will have the same number of features. Thus, the client with the largest number of samples N_{g^*} will have the highest time complexity. Furthermore, the central server performs a summation of $G+1$ vectors of dimension P during each iteration t, the time complexity of which is $\mathcal{O}(GP)$. We obtain the final result by noting that the subgradient method converges to a solution with tolerance ϵ_1 in $\mathcal{O}(\epsilon_1^{-2})$ iterations [Bubeck, 2015]. Note that we do not explicitly consider the time complexity of computing the local subgradient at each client since it is lower than that of solving the problem in (7). □

B.2.6 Proof of Proposition 4

In order to obtain updated local model $\boldsymbol{w}_g^*$, each client g must minimize the global Lagrangian with respect to $\boldsymbol{w}_g$. Thus, the updated local model $\boldsymbol{w}_g^*$ can be obtained as the minimizer to the following problem.

$$
\begin{aligned}
J_g(\boldsymbol{w}, \boldsymbol{\mu}_g) &= \inf_{\boldsymbol{w}_g} \mathcal{L}_\rho(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_G, \boldsymbol{w}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_G) \\
&= \inf_{\boldsymbol{w}_g} \mathcal{L}_{\rho_g}(\boldsymbol{w}_g, \boldsymbol{w}, \boldsymbol{\mu}_g) && \text{(17a)} \\
&= \inf_{\boldsymbol{w}_g} \sup_{\mathbb{Q}_g \in \mathcal{A}^{(g)}_{\varepsilon_g,1,d}(\Xi)} \mathbb{E}^{\mathbb{Q}_g}[\ell_H(\boldsymbol{w}_g; \boldsymbol{\xi})] + \frac{\rho}{2}||\boldsymbol{w}_g - \boldsymbol{w} + \boldsymbol{\mu}_g||_2^2 && \text{(17b)} \\
&= \inf_{\boldsymbol{w}_g, \lambda_g \geq 0} \lambda_g \varepsilon_g + \frac{1}{N_g} \sum_{n_g=1}^{N_g} \sup_{\boldsymbol{\xi} \in \Xi} \left\{ \ell_H(\boldsymbol{w}_g; \boldsymbol{\xi}) - \lambda_g d(\boldsymbol{\xi}, \widehat{\boldsymbol{\xi}}_{n_g}) \right\} + \frac{\rho}{2}||\boldsymbol{w}_g - \boldsymbol{w} + \boldsymbol{\mu}_g||_2^2 && \text{(17c)} \\
&= \begin{cases}
\min\limits_{\boldsymbol{w}_g, \lambda_g, s_{n_g}} & \lambda_g \varepsilon_g + \frac{1}{N_g} \sum\limits_{n_g=1}^{N_g} s_{n_g} + \frac{\rho}{2}||\boldsymbol{w}_g - \boldsymbol{w} + \boldsymbol{\mu}_g||_2^2 \\
\text{s. t.} & \ell_H(\boldsymbol{w}_g; (\widehat{\boldsymbol{x}}_{n_g}, \widehat{y}_{n_g})) \leq s_{n_g} \qquad \forall n_g \in [N_g] \\
& \ell_H(\boldsymbol{w}_g; (\widehat{\boldsymbol{x}}_{n_g}, -\widehat{y}_{n_g})) - \kappa \lambda_g \leq s_{n_g} \qquad \forall n_g \in [N_g] \\
& \lambda \geq ||\boldsymbol{w}_g||_*
\end{cases} && \text{(17d)}
\end{aligned}
$$

where 17a follows from the separability of the Augmented Lagrangian, 17b follows by definition of the local Lagrangian, 17c exploits the notable duality result presented by Mohajerin Esfahani and Kuhn [2018], Kuhn et al. [2019] to rewrite the inner maximization problem as a minimization problem, and (17d) follows by introducing slack variables s_{n_g}, recalling that $\ell_H(\boldsymbol{w}; \boldsymbol{\xi})$ is convex and Lipschitz continuous, and utilizing similar arguments to the ones presented in the proof of Theorem 1 in [Shafieezadeh Abadeh et al., 2015].

B.2.7 Proof of Proposition 5

Proof. The central server can obtain updated global parameters $\boldsymbol{w}^*$ by minimizing the global Lagrangian with respect to $\boldsymbol{w}$. This can be done as follows.

$$
\begin{aligned}
\boldsymbol{w}^* &= \arg\min_{\boldsymbol{w}} \mathcal{L}_\rho(\boldsymbol{w}_1, \ldots, \boldsymbol{w}_G, \boldsymbol{w}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_G) && \text{(18a)} \\
&= \arg\min_{\boldsymbol{w}} \sum_{g=1}^{G} \alpha_g \mathcal{L}_{\rho_g}(\boldsymbol{w}_g, \boldsymbol{w}, \boldsymbol{\mu}_g) && \text{(18b)} \\
&= \arg\min_{\boldsymbol{w}} \sum_{g=1}^{G} \alpha_g \frac{\rho}{2}||\boldsymbol{w}_g - \boldsymbol{w} + \boldsymbol{\mu}_g||_2^2, && \text{(18c)}
\end{aligned}
$$

where 18c follows by observing that the norm term is the only term involving the variable $\boldsymbol{w}$. Let us define $f(\boldsymbol{w}) = \sum_{g=1}^{G} \alpha_g \frac{\rho}{2}||\boldsymbol{w} - (\boldsymbol{w}_g + \boldsymbol{\mu}_g)||_2^2$. We note that $f(\boldsymbol{w})$ is strongly convex as it is a sum of strongly convex terms. Thus, it has a unique minimizer. We analyze its partial derivative with respect to $\boldsymbol{w}$ by setting it to 0 to obtain our minimizer as follows.

$$
\begin{aligned}
\frac{\partial f}{\partial \boldsymbol{w}} &= \sum_{g=1}^{G} \alpha_g \frac{\rho}{2} \left[2\boldsymbol{w} - 2(\boldsymbol{w}_g + \boldsymbol{\mu}_g) \right] && \text{(19a)} \\
&= 0 && \text{(19b)}
\end{aligned}
$$

Finally, we derive a closed form solution for $\boldsymbol{w}^*$ as follows:

$$\sum_{g=1}^{G} \alpha_g \frac{\rho}{2} \left[2\boldsymbol{w} - 2(\boldsymbol{w}_g + \boldsymbol{\mu}_g)\right] = 0 \tag{20a}$$

$$\Leftrightarrow \sum_{g=1}^{G} \alpha_g \boldsymbol{w} = \sum_{g=1}^{G} \alpha_g (\boldsymbol{w}_g + \boldsymbol{\mu}_g) \tag{20b}$$

$$\Leftrightarrow \boldsymbol{w} = \sum_{g=1}^{G} \alpha_g (\boldsymbol{w}_g + \boldsymbol{\mu}_g), \tag{20c}$$

where 20c follows by recalling that $\sum_{g=1}^{G} \alpha_g = 1$. □

B.2.8 Proof of Theorem 3

Proof. As mentioned previously, even when the client objective functions are closed proper convex functions as we demonstrate in 5, and strong duality holds as shown by Kuhn et al. [2019], multi-block ADMM is not theoretically guaranteed to converge [Chen et al., 2016]. However, Lin et al. [2015] establish the convergence of multi-block ADMM in the setting where the objective functions of $(B-1)$ of the B blocks are strongly convex with strong convexity parameter σ_b for each block b. They formulate the problem to be solved via ADMM as follows:

$$\begin{aligned} \min \quad & f_1(\boldsymbol{v}_1) + f_2(\boldsymbol{v}_2) + \cdots + f_B(\boldsymbol{v}_B) \\ \text{s.t.} \quad & \boldsymbol{A}_1\boldsymbol{v}_1 + \boldsymbol{A}_2\boldsymbol{v}_2 + \cdots + \boldsymbol{A}_B\boldsymbol{v}_B = c \\ & \boldsymbol{v}_b \in \mathcal{V}_b \quad \forall b \in [B], \end{aligned} \tag{21}$$

where $f_b(\boldsymbol{v}_b)$ is the objective function term and $\boldsymbol{v}_b$ is the decision variable associated with the bth block.

Note that if we were to rewrite our problem from (8) in a similar form, there would be no distinction between the clients and the central server, and the objective function term associated with the central server would remain 0. Thus, we add a strongly convex term $\tau_g ||\boldsymbol{w}_g||_2^2$ to the objective function term associated with each of the clients to meet the requirement that $B-1$ of the blocks must have a strongly convex objective function. During the server aggregation step, each $\tau_g ||\boldsymbol{w}_g||_2^2$ term will be multiplied by its respective weight α_g. Therefore, the strong convexity parameter associated with client g would be $2\alpha_g\tau_g$.

To rewrite problem (8) in the form of problem (21), the $\boldsymbol{A}$ matrix associated with client g would be a block matrix of $P \times P$ matrices stacked vertically in G blocks. The gth block from the top would be the identity matrix, whereas all the other blocks would be zero. Similarly, the matrix associated with the central server would be a block matrix of similar structure but where all the blocks are the negative of the identity matrix. Incorporating this insight into the condition on ρ described in Theorem 3.3 in [Lin et al., 2015] allows us to obtain the final result. □

B.2.9 Proof of Theorem 4

Proof. This proof follows a very similar strategy to that of Theorem 2. We begin by noting that the strongly convex variant of the local model problem in (9) equipped with the ℓ_1-norm can be written as a quadratically constrained quadratic problem (QCQP) with $N_g + 2P + 3$ decision variables (including slack variables) and $2N_g + 2P + 3$ constraints. When solved via the barrier method equipped with the log barrier function and Newton updates [Nesterov and Nemirovskii, 1994], this problem would have the following worst-case time complexity

$$\mathcal{O}([N_g + P]^{3.5} \log(\epsilon_2^{-1})).$$

Similar to the previous algorithm, all clients can solve their local problems in parallel and will have the same number of features. Thus the client with the greatest number of samples N_{g^*} will have the problem with the greatest time complexity. Furthermore, we note that the central server aggregates $2G$ vectors of dimension P in each iteration, the time complexity of which is $\mathcal{O}(GP)$. Therefore, we obtain the final result by noting that ADMM converges to an ϵ_1-solution in $\mathcal{O}(\epsilon_1^{-1})$ iterations assuming the strong convexity of the objective function and that the upper bound on ρ is satisfied [Lin et al., 2015]. While each client g also performs the update of the local scaled Lagrange multipliers $\boldsymbol{\mu}_g$ during each iteration, this process has a much lower complexity than solving the local problem and is, therefore, not explicitly considered in this analysis. □

B.3 SUPPLEMENTARY THEORETICAL RESULTS

B.3.1 Strongly Convex ADMM Client Update Problem

In the main body of the paper we presented the optimization problem to be solved locally by each client during each round of our proposed ADMM algorithm 2. Below, we present the strongly convex version of this problem $J_g^{\text{SC}}(\boldsymbol{w}, \boldsymbol{\mu}_g)$, which theoretically guarantees the convergence of the algorithm. This is the version utilized by the ADMM-SC algorithm. Please note that the proof for this formulation is exactly the same as that of Proposition 4.

$$J_g^{\text{SC}}(\boldsymbol{w}, \boldsymbol{\mu}_g) := \begin{cases} \min\limits_{\boldsymbol{w}_g, \lambda_g, s_{n_g}} & \lambda_g \varepsilon_g + \dfrac{1}{N_g} \displaystyle\sum_{n_g=1}^{N_g} s_{n_g} + \dfrac{\rho}{2} ||\boldsymbol{w}_g - \boldsymbol{w} + \boldsymbol{\mu}_g||_2^2 + \tau_g ||\boldsymbol{w}_g||_2^2 \\ \text{s. t.} & \ell_H(\boldsymbol{w}_g; (\widehat{\boldsymbol{x}}_{n_g}, \widehat{y}_{n_g})) \leq s_{n_g} \qquad \forall n_g \in [N_g] \\ & \ell_H(\boldsymbol{w}_g; (\widehat{\boldsymbol{x}}_{n_g}, -\widehat{y}_{n_g})) - \kappa \lambda_g \leq s_{n_g} \qquad \forall n_g \in [N_g] \\ & \lambda \geq ||\boldsymbol{w}_g||_*, \end{cases}$$

where $|| \cdot ||_*$ is the dual of the norm used in 3.

C FURTHER EXPERIMENTAL DETAILS AND SUPPLEMENTARY RESULTS

In this section we provide all the details of all the experiments presented in this paper, as well as the results of a **scalability experiment**. Please note that the all the code and instructions associated with all the experiments are available at this link.

C.1 SOFTWARE AND HARDWARE DETAILS

All the experiments presented in this work were executed on Intel Xeon Gold 6226 CPUs @ 2.7 GHz (using 4 cores) with 120 Gb of DDR4-2993 MHz DRAM. Table 3 provides more detail on all the software used in the paper.

Table 3: Details on All the Software Used in the Numerical Experiments.

Software	Version	License
Gurobi	10.0.1	Academic license
MATLAB	R2021B	Academic license
Python	3.10.9	Open source license
Scikit-Learn	1.2.1	Open source license
Numpy	1.23.5	Open source license
Scipy	1.10.0	Open source license
UCIMLRepo	0.0.3	Open source license

C.2 DATASETS UTILIZED

C.2.1 UCI Data Experiment

We provide details on the datasets used in the experiment described in Section 6.1. Note that Parkinson's exhibited very high levels of class imbalance (75% from one class and 25% from the other), which suggests that the SM algorithm is more successful with data that exhibits such levels of imbalance. Moreover, note that the "Very Low" and "Low" classes in the UKM dataset were combined into one class, whereas "Middle" and "High" were combined into another.

C.2.2 Industrial Data Experiment

The data used in the experiment described in Section 6.2 is a simulation dataset that uses a physics-driven Simulink model to simulate the healthy and faulty operation of a reciprocating pump [Mathworks, n.d.]. The generated simulation data belongs

Table 4: Details on Datasets Utilized for UCI Experiments.

Dataset	Abbreviation	License
Banknote Authentication [Lohweg, 2013]	Banknote	CC BY 4.0
Breast Cancer Wisconsin (Diagnostic) [Wolberg et al., 1995]	BCW	CC BY 4.0
Connectionist Bench (Sonar) [Sejnowski and Gorman, 1988]	CB	CC BY 4.0
Mammographic Mass [Elter, 2007]	MM	CC BY 4.0
Parkinson's [Little, 2008]	Parkinson's	CC BY 4.0
Rice (Cammeo and Osmancik) [Cınar and Koklu, 2019]	Rice	CC BY 4.0
User Knowledge Modeling [Kahraman et al., 2013]	UKM	CC BY 4.0

to two classes: healthy pump and leak fault. We focus on the binary classification problem since binary classification models can directly extend to multiclass problems via a one-vs-all framework as mentioned previously. Therefore, performance in the binary setting is indicative of that in the multiclass setting. However, data is generated to simulate different severities of the leak fault, where each client has a different severity to simulate data heterogeneity across clients. Note that leak fault severity is controlled via a `leak_area_set_factor` variable in the MATLAB script. The four values used in our experiments are $[1e-3, 4e-3, 7e-3, 1e-2]$. Features extracted from the generated time series data (such as kurtosis and skewness) are used for classification.

C.3 HYPERPARAMETER DETAILS

In all of our implementations of the SM algorithm we utilize a step-size that diminishes according to $\gamma(t) = \frac{\gamma}{t}$, where we treat γ as a model hyperparameter. This step-size obeys the conditions required for algorithm convergence stated in Theorem 1. Next, we provide details on the hyperparameter values used in the UCI Data Experiment and the Industrial Data Experiment in Sections 6.1 and 6.2, respectively.

UCI Data Experiment. For the centralized baseline, we tune $\varepsilon \in \{1 \times 10^b\}_{b=-5}^{-1}$ and $\kappa \in \{0.1, 0.25, 0.5, 0.75, 1\}$. For the federated baselines we use diminishing step-size of $\gamma(t) = \frac{\gamma(0)}{t}$, where $\gamma(0)$ is treated as a tuning hyperparameter and takes values $\gamma(t) \in \{1e-3, 1e-2, 1e-1, 1e0\}$, and a local regularization penalty of $\frac{1}{10N_g}$ at each client. For `FedAvg` and `FedProx`, we utilize a local batch size of 20% of the available training data, and $E = 5$ local SGD epochs where appropriate. We also use a $\mu = 1$ for `FedProx`. For our proposed methods we fix $\kappa_g = 1$ and $\varepsilon_g = \frac{1}{10N_g}$, and we tune $\rho \in \{1e-3, 1e-2, 1e-1, 1e0\}$ and $\gamma \in \{1e0, 1e1, 1e2, 1e3\}$. Finally, for all federated methods (including baselines and ours) we use $G = 4$ with equal client weights. Finally, for ADMM, ADMM-SC and federated baselines, we tune $T \in \{5, 10, 20, 60, 100, 140, 180, 220\}$, whereas for SM we tune $T \in \{100, 140, 180, 220\}$. All tuning is done via 5-fold cross-validation.

Industrial Data Experiment - Sensitivity Analysis. In the global hyperparameters experiment we evaluate the performance of our proposed federated algorithms for $T \in \{5, 10, 20, 60, 100, 140, 180, 220\}$ and $\rho, \gamma \in \{1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2, 1e3\}$. We fix $\varepsilon_g = \frac{1}{10N_g}$ and $\kappa_g = 0.5$ for each client g. While such values of ε_g and κ_g may not be optimal, we use them to demonstrate that our proposed model can perform well when compared to the central baseline.

In the local hyperparameters testing, We evaluate the performance of both the our federated algorithms for $\varepsilon_g = \frac{1}{\beta N_g}$ where $\beta \in \{0.1, 1, 10, 100\}$ and for $\kappa_g = \kappa \in \{0.1, 0.25, 0.5, 0.75, 1\}$. We fix $T = 220$ and $\gamma = 1 \times 10^2$ and $T = 100$ and $\rho = 1 \times 10^{-3}$ for the SM and ADMM algorithms, respectively. In all settings we evaluate the performance of the baseline central model for $\varepsilon \in \{1 \times 10^b\}_{b=-5}^{-1}$ and $\kappa \in \{0.1, 0.25, 0.5, 0.75, 1\}$, and we only report the peak performance achieved.

In all settings we utilize $\tau_g = 18\rho$ for the ADMM-SC algorithm, which is the minimum value τ_g can take while maintaining guaranteed convergence as shown in Theorem 3. We do this as increasing τ increases the strength of the redundant regularization, thereby impacting the performance.

Industrial Data Experiment - Benchmarking. In this portion we utilize 5-fold cross-validation to tune the same hyperparameters discussed in the previous paragraph. Namely, we fix $T = 220$, and we tune $\rho \in \{1e-3, 1e-2, 1e-1\}$ or $\gamma \in \{1e1, 1e2, 1e3\}$, $\kappa \in \{0.1, 0.5, 1\}$, and $\beta \in \{10, 100\}$ for all our methods. Tuning is done via 5-fold cross-validation.

Model Parameter Initialization. In all of our experiments, we use initial model parameters $\boldsymbol{w}^{(0)} = \mathbf{0}$ (i.e., a vector of zeros), and initial scaled Lagrange multipliers $\boldsymbol{\mu}_g^{(0)} = \mathbf{1}$ (i.e., a vector of ones).

C.4 UCI DATA EXPERIMENT STATISTICAL SIGNIFICANCE

In order to evaluate the statistical significance of the results presented in Table 1, we perform a one-sided Wilcoxon signed-rank test. The test compares the performance of the best performing version of our model to that attained by each of the benchmarks in a pairwise fashion. The null H_0 and alternative H_1 hypotheses of this test are defined next.

- H_0: The distribution of the differences in performance between our model and each benchmark has median zero. That is, there is no systematic increase or decrease between the pairs.
- H_1: The median of the differences is greater than 0. That is, our approach is statistically better.

The results of this test are presented in Table 5, utilizing a significance level of $\alpha = 0.05$. The table indicates whether the null hypothesis H_0 is rejected or not. We observe from the table that the performance improvement offered by our model algorithm is indeed statistically significant for most datasets and most benchmark models. This is because we "Reject" the null hypothesis H_0 in most settings. This underscores the practical impact and performance improvements offered by our proposed model.

Table 5: Results of One-Sided Wilcoxon Signed-Rank Test Performed on Results of Benchmarking Experiments on 7 UCI Datasets.

Model	Banknote	BCW	CB	MM	Parkinson's	Rice	UKM
FedSGD (ℓ_2-SVM)	Fail to reject	Reject	Reject	Reject	Reject	Reject	Reject
FedAvg (ℓ_2-SVM)	Fail to reject	Reject	Fail to reject	Reject	Reject	Fail to reject	Fail to reject
FedProx (ℓ_2-SVM)	fail to reject	Reject	Fail to reject	Reject	Reject	Reject	Fail to reject
FedDRO (KL)	Reject	Reject	Reject	Reject	Reject	Reject	Reject

C.5 SCALABILITY EXPERIMENT

The purpose of this experiment is to examine the scalability of our proposed algorithms as the total number of samples N, the number of clients G, and the number of features P grow. We measure performance in runtime required to achieve peak mCCR. We do this since the subgradient method lacks a practically implementable stopping criterion [Bagirov et al., 2014], and similarly, no stopping criterion is provided for multi-block ADMM by Lin et al. [2015]. Moreover, it was already established in the experiment in Section 6.2 that the SM algorithm requires more rounds of communication to attain peak performance. This experiment is more focused on computational effort required to achieve this performance. We examine the the following settings:

1. *Increasing clients [fixed training samples]*: $N = 1000$, $P = 4$, $G \in \{10, 20, 30, 40, 50\}$.
2. *Increasing clients [increasing training samples]*: $N = 100G$, $P = 4$, $G \in \{10, 20, 30, 40, 50\}$.
3. *Increasing training samples*: $G = 10$, $P = 4$, $N \in \{1000, 1500, 2000, 2500, 3000\}$.
4. *Increasing features*: $N = 4$, $G = 10$, $P \in \{4, 6, 8, 10, 12\}$.

Dataset. This experiment uses simulation data that is generated using the `make_classification` module of the Scikit-Learn Python package [Pedregosa et al., 2011]. The data generated belongs to two classes, each of which contains data sampled from a standard Gaussian distribution with means located at vertices of a P-dimensional hypercube with sides of length 2.4 centered at the origin. The data is distributed equally across all clients and both classes, and no labels are altered.

Baseline. We utilize the centralized DR-SVM by Shafieezadeh-Abadeh et al. [2019] as a baseline in this experiment.

Hyperparameters. For the SM algorithm, we test performance for $T \in \{140, 180, 220\}$ and $\gamma \in \{1e1, 1e2, 1e3\}$. For the ADMM and ADMM-SC algorithms, we test performance for $T \in \{10, 20, 30\}$ and $\rho \in \{1e-3, 1e-2, 1e-1\}$. Across all algorithms, we fix $\varepsilon_g = \frac{1}{10N_g}$ and $\kappa = 0.25$. The central model's hyperparameters are varied in the same way as in the Sensitivity Analysis portoin of the experiment in Section 6.2, and the runtime that is reported reflects the time taken to solve the optimization problem.

Results. The results of this study are reported in Figure 3. We observe a rough trend of increasing runtime as N and P increase for all models due to the increasing complexity of the local client problems. However, the trend is clearer with the

SM algorithm, whereas it is noisy with all versions of the ADMM algorithm, and is hardly observable with the central model. This could be attributed to the fact that the SM algorithm requires a much longer time to reach peak mCCR, making the effect of random computer system variations minimal on the reported time. On the contrary, all versions of the ADMM and the central model reach peak mCCR in a very short time, making the reported time highly susceptible to system variations. These results highlight the fact that any performance gains achieved by using SM come at the cost of a much longer runtime. However, the runtime of ADMM and ADMM-SC is much closer to that of the central approach. Additionally, we observe that the runtime remains roughly constant for all federated algorithms as G increases if N is fixed. This is because a fixed N makes the local problem at each client increasingly simpler and faster to solve as G increases. In contrast, when both G and N are increasing we observe that all algorithms exhibit a trend of increasing runtime with the number of clients.

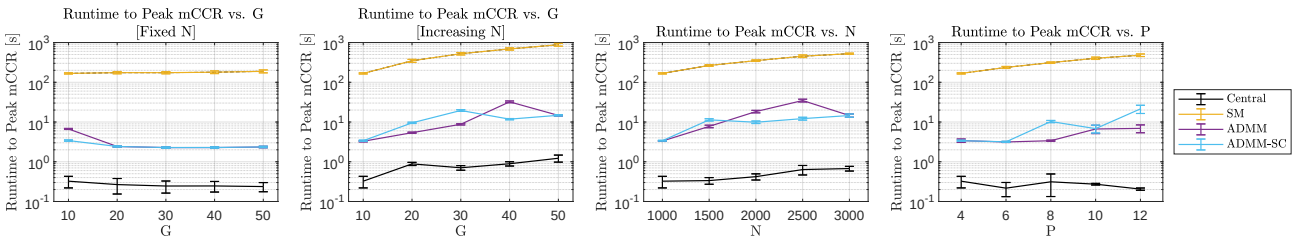

Figure 3: Plots of Runtime to Reach Peak mCCR vs. the Number of Clients G with Fixed and Increasing N, the Number of Features P, and the Number of Training Samples N for All Methods Tested.