PEARL-Prox: Proximal Algorithm for Resolving Player Drift in Multiplayer Federated Learning

TaeHo Yoon TYOON7@JHU.EDU
Nicolas Loizou NLOIZOU@JHU.EDU
Department of Applied Mathematics & Statistics, Johns Hopkins University
Mathematical Institute for Data Science (MINDS), Johns Hopkins University

Abstract

Recently, Yoon et al. [25] introduced multiplayer federated learning (MpFL), a novel federated learning framework capable of formulating the strategically behaving, rational clients. In MpFL, the clients are modeled as players of a multiplayer game with individual objectives, aiming to seek an equilibrium. While Per-Player Local Stochastic Gradient Descent (PEARL-SGD) algorithm has been proposed as a counterpart of Local SGD in the MpFL setup, it exhibits the *player drift* phenomenon— excessive local updates by individual players lead to divergence of the global dynamics. In this work, we formalize the concept of player drift and propose the *Per-Player Local Proximal Algorithm* (PEARL-Prox) to resolve it. PEARL-Prox lets each player optimize a regularized objective with high accuracy, ensuring convergence to the equilibrium while enabling the players to exploit their local compute budgets. Consequently, PEARL-Prox offers a significantly improved communication complexity of $\mathcal{O}\left(\log \epsilon^{-1}\right)$ compared to the $\Omega\left(\epsilon^{-1/2}\right)$ complexity of PEARL-SGD under the same theoretical assumptions.

1. Introduction

In classical federated learning (FL), multiple clients cooperatively train a model without explicitly revealing their local data [10]. This assumes that all clients' interests are aligned toward optimizing a shared global objective. However, strategic clients may have distinct objectives, potentially involving components that are competing with others' interests, which can be formulated in the language of game theory [4, 16]. Recently, in [25], *multiplayer federated learning (MpFL)* has been proposed as a federated learning framework that formulates clients behaving as players in a game, with the collective goal of reaching the equilibrium.

One of the primary concerns in designing FL training algorithms is the frequency of *communication* (also called synchronization) — the process by which the central server collects information from, and distributes updates to, each client [11]. In FedAvg (Local SGD), the cornerstone of FL algorithms, each client locally performs SGD updates on their local data, which are aggregated by the server once in a while [17]. Due to its costly nature the communication occurs infrequently, which allows clients to perform a large number of local computations between communications. However, letting each client fully exploit their computational capabilities to minimize (run a local optimization algorithm for too many steps) their local objectives may lead to suboptimal results [17, 19]. In classical FL, this phenomenon is called *client drift*, and occurs when the global model fails to converge to the minimizer of the global objective due to data heterogeneity [12, 29]. In recent years, multiple works have proposed algorithmic adjustments to mitigate client drift [5, 9, 12, 18, 27].

The analogous phenomenon to the client drift, in the setup of MpFL, is called the *player drift* [25]—greedy local optimization by each player causes the game dynamics to diverge away from the

equilibrium. However, as we explain in this work the player drift arises not only from heterogeneity in data (like classical FL) but also due to the conceptually different setting of the game dynamics appear in MpFL—it is attributed to the nature of multiplayer games where the players have possibly competing objectives. The first MpFL algorithm Per-Player Local SGD (PEARL-SGD), proposed in [25], is prone to the player drift phenomenon.

In this work, we propose PEARL-Prox, an algorithm for MpFL where each player locally optimizes their regularized objective. This allows each player to leverage their local computing power to reach the optimum, while ensuring the global game dynamics to converge to the equilibrium.

2. MpFL: Formulation and notations

MpFL is a recently introduced federated learning framework capable of handling a broad scope of setups expressible using game-theoretic formulation. We consider the multiplayer game setup where there are n players, indexed by $i = 1, \ldots, n$. The player i has an unconstrained and continuous space of actions $x_i \in \mathbb{R}^{d_i}$. We denote the joint action vector of all players by $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^D = \mathbb{R}^{d_1+\cdots+d_n}$. Each player i has their objective function $f_i(x_1, \ldots, x_n) \colon \mathbb{R}^{d_1+\cdots+d_n} \to \mathbb{R}$ which they prefer to minimize with respect to x_i. The goal of an n player (multiplayer) game is to find an *equilibrium* $\mathbf{x}^\star = (x_1^\star, \ldots, x_n^\star) \in \mathbb{R}^D$, which formally expressed as

$$\operatorname*{find}_{\mathbf{x}^\star=(x_1^\star,\ldots,x_n^\star)\in\mathbb{R}^D} \quad f_i(x_i^\star; x_{-i}^\star) \leq f_i(x_i; x_{-i}^\star), \quad \forall x_i \in \mathbb{R}^{d_i}, \quad \forall i \in [n], \tag{1}$$

where $x_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) \in \mathbb{R}^{D-d_i}$ denotes the vector of actions from all players other than the player i, and $f_i(x_i; x_{-i}) = f_i(x_1, \ldots, x_n)$.

MpFL considers the setup where f_i are given as $f_i(x_1, \ldots, x_n) = \mathbb{E}_{\xi_i \sim \mathcal{D}_i}\left[f_{i,\xi_i}(x_1, \ldots, x_n)\right]$ where $\mathcal{D}_i$ is the data distribution for the player i, ξ_i are data samples and $f_{i,\xi_i}(\cdot)$ is the corresponding loss function. Given fixed x_{-i}, each player can compute the stochastic gradients $\nabla_{x_i} f_{i,\xi_i}(\cdot; x_{-i})$ of f_i. Based on these computations, in each round (before communication occurs), each player locally adjusts their action by the amount Δx_i. Then at the communication step, the server collects these updated local actions $(x_1 + \Delta x_1, \ldots, x_n + \Delta x_n)$ and distributes them back to all players.

3. Player drift in MpFL

Algorithm 1: PEARL-SGD

Input: Step sizes $\{\gamma_k^p\}_{k=0}^{\tau-1} > 0$, Synchronization interval $\tau \geq 1$, Number of synchronization/local update rounds $R \geq 1$

for $p = 0, \ldots, R-1$ **do**

  Master server collects x_i^p from players $i = 1, \ldots, n$ and forms $\mathbf{x}^p = (x_1^p, \ldots, x_n^p)$;

  Master server distributes $\mathbf{x}^p$ back to players $i = 1, \ldots, n$;

  for $i = 1, \ldots, n$ **do**

    $x_i^{p+1} \leftarrow \mathtt{SGD}\left(f_i(\cdot; x_{-i}^p), x_i^p, \tau, \{\gamma_k^p\}_{k=0}^{\tau-1}\right)$;

  end

end

Output: $\mathbf{x}^R \in \mathbb{R}^D$

In this section, we define and discuss player drift. Let us first consider the PEARL-SGD (Algorithm 1) from [25], designed for the MpFL setup. In PEARL-SGD, at each communication round $p = 0, \ldots, R-1$, each player i is given with $\mathbf{x}^p = (x_1^p, \ldots, x_n^p)$. Then they run an inner loop of SGD with $f_i(\cdot; x_{-i}^p)$ as the objective function, x_i^p as the initial point, and perform τ steps of stochastic gradient descent using a step size sequence $\{\gamma_k^p\}_{k=0}^{\tau-1}$. The output of this SGD subroutine x_i^{p+1} is then sent to the server for synchronization.

Algorithm 2: $\texttt{SGD}(h, x^0, T, \{\gamma_k\}_{k=0}^{T-1})$

Input: Objective function $h(x) = \mathbb{E}_{\xi\sim\mathcal{D}}[h_\xi(x)]$, initial point x^0, total iteration number T, step sizes $\{\gamma_k\}_{k=0}^{T-1}$

for $k = 0, \ldots, T-1$ **do**

  Sample $\xi_k \sim \mathcal{D}$;

  $x^{k+1} \leftarrow x^k - \gamma_k \nabla h_{\xi_k}(x^k)$;

end

Output: x^T

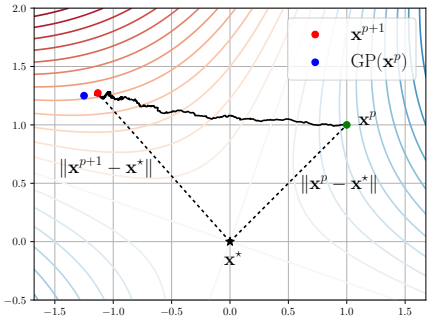


Figure 1: PEARL-SGD behaving similarly to GP dynamics for minimax game with $f(x_1, x_2) = \frac{\mu}{2}x_1^2 + x_1x_2 - \frac{\mu}{2}x_2^2$, $\mu \in (0, 1)$.

Greedy player dynamics. Suppose that the MpFL network has low communication frequency and players have sufficient computation budget for local optimization (i.e., τ is large enough). If each player acts selfishly, they can select an appropriate sequence of step sizes to minimize their local cost function f_i. In other words, $\lim_{\tau\to\infty} x_i^{p+1}(\tau; \mathbf{x}^p) = \operatorname{argmin}_{x_i\in\mathbb{R}^{d_i}} f_i(\cdot; x_{-i}^p) := \tilde{x}_i^\star\left(x_{-i}^p\right)$, where we hide the dependency on γ_k^p, which are assumed to be optimally tuned. Then the resulting global dynamics between adjacent synchronization steps can be expressed as

$$\mathbf{x} \mapsto \mathrm{GP}(\mathbf{x}) := (\tilde{x}_1^\star(x_{-1}), \ldots, \tilde{x}_n^\star(x_{-n}))$$

which we call the *greedy player (GP)* dynamics.

However, even under favorable assumptions such as strong monotonicity of F, the greedy player dynamics may lead to divergence away from the equilibrium, i.e., one can have $\|\mathrm{GP}(\mathbf{x})-\mathbf{x}^\star\|^2/\|\mathbf{x}-\mathbf{x}^\star\|^2 > 1$ (Fig. 1). We say that *player drift* occurs if excessive local compute in an MpFL algorithm results in a similar divergent behavior of the GP dynamics, i.e., $\lim_{\tau\to\infty} \left\|\mathbf{x}^{p+1}(\tau;\mathbf{x}^p)-\mathbf{x}^\star\right\|^2/\|\mathbf{x}^p-\mathbf{x}^\star\|^2 > 1$.

In particular, in [25] it was shown that PEARL-SGD converges to $\mathbf{x}^\star$, but only if the players use the step size $\gamma = \mathcal{O}(1/\tau)$. This indicates that the amount of total progression allowed within each local computation round between synchronization steps should be limited. In this regard, PEARL-SGD suffers from player drift, as the local compute power of players (clients) cannot be fully exploited.

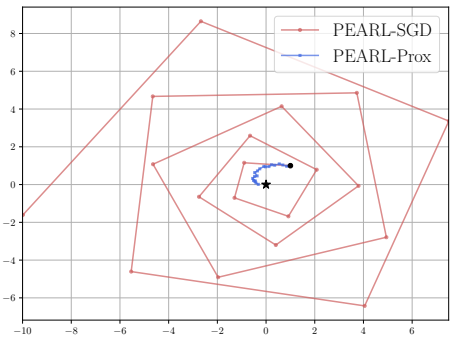


Figure 2: Trajectory plot for the minimax game with $f(x_1, x_2) = \frac{\mu}{2}x_1^2 + x_1x_2 - \frac{\mu}{2}x_2^2$, $\mu = 0.8$. While PEARL-SGD diverges with $\gamma_k^p \equiv \gamma = 0.1$, $\tau = 25$, PEARL-Prox converges (with $\lambda = 10$).

Algorithm 3: Per-Player Local Proximal Algorithm (PEARL-Prox)

Input: Regularization parameter $\lambda > 0$, Number of rounds $R \geq 1$
Initialize $x^0 = (x_1^0, x_2^0, \ldots, x_n^0)$;
for $p = 0, \ldots, R-1$ **do**
  Master server collects x_i^p from players $i = 1, \ldots, n$ and forms $\mathbf{x}^p = (x_1^p, \ldots, x_n^p)$;
  Master server distributes $\mathbf{x}^p$ back to players $i = 1, \ldots, n$;
  for $i = 1, \ldots, n$ **do**
    Approximately compute $x_i^{p+1} \approx \operatorname{argmin}_{x_i} f_i(x_i; x_{-i}^p) + \frac{\lambda}{2} \|x_i - x_i^p\|^2$;
  end
end
Output: $x^R \in \mathbb{R}^D$

4. PEARL-Prox: Algorithm design and analysis

In this section, we propose the new algorithm PEARL-Prox for handling the player drift in MpFL. In each communication round of PEARL-Prox, players minimize the regularized local objectives $f_i(\cdot; x_{-i}^p) + \frac{\lambda}{2} \|\cdot - x_i^p\|^2$ with hyperparameter $\lambda > 0$, instead of $f_i(\cdot; x_{-i}^p)$.

When each player has sufficient compute power, at each round, they will attain the solution

$$x_i^{p+1} = \operatorname{Prox}_{\frac{1}{\lambda} f_i(\cdot; x_{-i}^p)}(x_i^p) = \operatorname*{argmin}_{x_i \in \mathbb{R}^{d_i}} f_i(\cdot; x_{-i}^p) + \frac{\lambda}{2} \|x_i - x_i^p\|^2 . \tag{2}$$

With λ chosen appropriately, this identifies an update direction which points toward the equilibrium $\mathbf{x}^\star$ but is not excessive (as the objective enforces x_i^{p+1} to be close to x_i^p), and thus, convergence is ensured. Importantly, one does not have to specify a subroutine used by each player to minimize the regularized objectives, including the choice of algorithm, number of iterations or algorithm parameters such as step sizes. Algorithm 3 emphasizes this aspect with the intentional ambiguity.

By algorithm design, PEARL-Prox takes full advantage of local steps and resolves player drift. Unlike PEARL-SGD where minimizing $f_i(\cdot; x_{-i}^p)$ can cause divergence, in PEARL-Prox accurate minimization is rather beneficial and enables convergence (Fig. 2, Theorem 4). Due to this property, under stochasticity, PEARL-Prox provides a much larger communication gain than PEARL-SGD.

4.1. Theoretical assumptions

We present the theoretical assumptions, which are the same as in [25] proposing MpFL, required for convergence analyses. We define the *joint gradient operator* $F \colon \mathbb{R}^D \to \mathbb{R}^D$ by $F(\mathbf{x}) = F(x_1, \ldots, x_n) = (\nabla_{x_1} f_1(x_1; x_{-1}), \ldots, \nabla_{x_n} f_n(x_n; x_{-n}))$. Note that given Assumption 1, the equilibrium condition (1) is equivalent to $F(\mathbf{x}^\star) = 0$.

Assumption 1 *For $i = 1, \ldots, n$, for any $x_{-i} \in \mathbb{R}^{D-d_i}$, the function $f_i(\cdot; x_{-i}) \colon \mathbb{R}^{d_i} \to \mathbb{R}$ is convex and L_i-smooth, i.e., $\nabla_{x_i} f_i(\cdot; x_{-i})$ is L_i-Lipschitz continuous.*

Assumption 2 *The joint gradient operator F is μ-quasi-strongly monotone, i.e., there exists an equilibrium $\mathbf{x}^\star \in \mathbb{R}^D$ and $\mu > 0$ such that $\forall \mathbf{x} \in \mathbb{R}^D$, $\langle F(\mathbf{x}), \mathbf{x} - \mathbf{x}^\star \rangle \geq \mu \|\mathbf{x} - \mathbf{x}^\star\|^2$. Further, F is ℓ star-cocoercive, i.e., there exists $\ell > 0$ such that $\forall \mathbf{x} \in \mathbb{R}^D$, $\langle F(\mathbf{x}), \mathbf{x} - \mathbf{x}^\star \rangle \geq \ell \|F(\mathbf{x})\|^2$.*

Assumption 3 *For $i = 1, \ldots, n$, there exists $\sigma_i \geq 0$ such that for any $x_i \in \mathbb{R}^{d_i}, x_{-i} \in \mathbb{R}^{D-d_i}$, we have $\mathbb{E}_{\xi_i \sim \mathcal{D}_i} \left[\|\nabla_{x_i} f_{i,\xi_i}(x_i; x_{-i}) - \nabla_{x_i} f_i(x_i; x_{-i})\|^2 \right] \leq \sigma_i^2$.*

4.2. Convergence of exact PEARL-Prox

To illustrate the game dynamics expected as a result of minimizing the regularized objectives, we first analyze the idealized scenario where we assume that each player is capable of computing (2) exactly. In this case, PEARL-Prox converges linearly to $\mathbf{x}^\star$, as deterministic (full-batch) version of PEARL-SGD does ([25, Theorem 3.3]).

Theorem 4 *Suppose that Assumptions 1 and 2 hold. Let $L_{\max} = \max\{L_1, \ldots, L_n\}$, and let $\kappa = \ell/\mu$ be the condition number of the game. Suppose $\lambda > \frac{1}{2}(\ell + 2L_{\max}\sqrt{\kappa})$. Then PEARL-Prox with exact proximal operator computation (2) converges with the rate $\left\|\mathbf{x}^R - \mathbf{x}^\star\right\|^2 \leq \left(1 - \frac{2\mu\zeta}{\lambda}\right)^R \left\|\mathbf{x}^0 - \mathbf{x}^\star\right\|^2$, where $\zeta = 1 - \frac{\ell + 2L_{\max}\sqrt{\kappa}}{2\lambda} > 0$.*

Theorem 4 requires λ to be sufficiently large, which is intuitively clear as exact PEARL-Prox will converge to the GP dynamics when $\lambda \to 0$. In other words, if λ is small, the effect of algorithmic modification of adding the quadratic regularizer will not be strong enough to fix player drift.

4.3. Convergence of PEARL-Prox with inexactness

We have observed that PEARL-Prox converges with exact proximal operator computation, if λ is chosen appropriately. In practice, however, players have finite compute limits and will be only able to compute (2) with some error (inexactness). Below, we provide a convergence guarantee for this scenario. For that purpose, we start with quantifying the magnitude of errors.

Assumption 5 *(Magnitude of errors) At each round $p = 0, \ldots, R-1$, each player $i = 1, \ldots, n$ computes $x_i^{p+1} = \mathrm{Prox}_{\frac{1}{\lambda} f_i(\cdot; x_{-i}^p)}(x_i^p) + v_i^p$ for some random error vector $v_i^p \in \mathbb{R}^{d_i}$ satisfying*

$$\mathbb{E}\left[\left\|v_i^p\right\|^2 \middle| x_i^p\right] \leq \delta_i^p \left\|\nabla_{x_i} f_i(x_i^p; x_{-i}^p)\right\|^2 + \epsilon_i^p \tag{3}$$

for some $\delta_i^p, \epsilon_i^p > 0$, where the conditional expectation is taken over all randomness within the process of computing x_i^{p+1} given x_i^p.

We consider the error bound of the form (3) because it is consistent with the most common form of convergence guarantee for algorithms minimizing strongly convex functions. For details, please see Appendix B.1.

Theorem 6 *Suppose that Assumptions 1 and 2 hold. Suppose that $\lambda > 2\left(\ell + L_{\max}\sqrt{\kappa}\right)$ ($L_{\max}, \kappa$ defined as in Theorem 4) and PEARL-Prox satisfies Assumption 5 with $\delta_i^p \leq \frac{1}{4\lambda\ell}$ for $i = 1, \ldots, n$ and $p = 0, \ldots, R-1$. Then PEARL-Prox exhibits the rate*

$$\mathbb{E}\left[\left\|\mathbf{x}^R - \mathbf{x}^\star\right\|^2\right] \leq \left(1 - \frac{\mu\zeta}{\lambda}\right)^R \left\|\mathbf{x}^0 - \mathbf{x}^\star\right\|^2 + \left(2 + \frac{2\lambda}{\mu}\right) \sum_{p=0}^{R-1} \left(1 - \frac{\mu\zeta}{\lambda}\right)^{R-p-1} \sum_{i=1}^n \epsilon_i^p$$

where $\zeta = 1 - \frac{2\left(\ell + L_{\max}\sqrt{\kappa}\right)}{\lambda} > 0$.

Theorem 6 implies that if each player minimizes $f_i(\cdot; x_{-i}^p) + \frac{\lambda}{2}\left\|\cdot - x_i^p\right\|^2$ with a certain degree of accuracy, then PEARL-Prox linearly converges to a correspondingly sized neighborhood of the equilibrium. The technical condition $\delta_i^p \leq \frac{1}{4\lambda\ell}$ can be fulfilled easily if each player uses a moderately large local iteration number, as δ_i^p corresponds to a_T in the general guarantee (5) which decays exponentially in T for strongly convex objectives.

Remark. Setting $\delta_i^p = \epsilon_i^p = 0$ in Theorem 6 (indicating exact prox computation) gives a linear convergence result as in Theorem 4. However, the condition on λ and the linear convergence factor become more conservative due to some technical steps introduced to handle stochasticity.

4.4. Convergence and communication efficiency of PEARL-Prox with SGD inner loop

In principle, players can use any algorithm to compute the approximate minimizer of the function $f_i(\cdot; x_{-i}^p) + \frac{\lambda}{2}\|\cdot - x_i^p\|^2$. Among them, in this section, we analyze Algorithm 4 (pseudocode provided in Appendix C), a version of PEARL-Prox where all players use SGD as their minimization subroutine. This allows us to make a direct comparison to PEARL-SGD in terms of communication complexity. Let us denote the number of SGD steps taken by each player i at round p by τ_i^p, and the corresponding step size schedule by $\gamma_{i,t}^p$ ($t = 0, \ldots, \tau_i^p - 1$). Then we have the following result.

Corollary 7 *Suppose that Assumptions 1, 2 and 3 hold and* $\lambda > 2(\ell + L_{\max}\sqrt{\kappa})$. *Let* $\zeta = 1 - \frac{2(\ell + L_{\max}\sqrt{\kappa})}{\lambda} > 0$, $\tau \geq \max\left\{\sqrt{\frac{4\ell}{\lambda}}, 16\left(1 + \frac{L_{\max}}{\lambda}\right)^2\right\}$ *and* $\sigma^2 = \sum_{i=1}^n \sigma_i^2$. *Then Algorithm 4 with* $\tau_i^p \equiv \tau$ *and* $\gamma_{i,t}^p \equiv \frac{2\log\tau}{\lambda\tau}$ *converges with the rate*

$$\mathbb{E}\left[\left\|\mathbf{x}^R - \mathbf{x}^\star\right\|^2\right] \leq \left(1 - \frac{\mu\zeta}{\lambda}\right)^R \left\|\mathbf{x}^0 - \mathbf{x}^\star\right\|^2 + \left(2 + \frac{2\lambda}{\mu}\right)\frac{2\sigma^2\log\tau}{\mu\zeta\lambda\tau} \tag{4}$$

Note that as we use SGD as a subroutine for computing the proximal operator, we are interested in moderately large τ, sufficient for attaining certain level of accuracy. This is why we have a lower bound on τ in Corollary 7. Nevertheless, due to the condition $\lambda > 2(\ell + L_{\max}\sqrt{\kappa})$ this lower bound is not too restrictive; e.g., with $\lambda = 4(\ell + L_{\max}\sqrt{\kappa})$ we have $\max\left\{\sqrt{\frac{4\ell}{\lambda}}, 16\left(1 + \frac{L_{\max}}{\lambda}\right)^2\right\} \leq 25$.

Communication efficiency. Note that (4) indicates linear convergence to a neighborhood with size $\propto \frac{\log\tau}{\tau}$, which is reduced as τ grows larger. Importantly, this does not depend on the number of communication R and clearly indicates the advantage of large τ. In particular, if τ is sufficiently large ($\tau = \tilde{\Omega}\left(\epsilon^{-1}\right)$), we have $\mathbb{E}\left[\left\|\mathbf{x}^R - \mathbf{x}^\star\right\|^2\right] \leq \epsilon$ if $R \gtrapprox \frac{\lambda}{\mu\zeta}\log\frac{\left\|\mathbf{x}^0 - \mathbf{x}^\star\right\|^2}{\epsilon} = \Omega\left(\frac{\ell + L_{\max}\sqrt{\kappa}}{\mu}\log\frac{1}{\epsilon}\right)$. That is, even in the stochastic setup, PEARL-Prox can take full advantage of large local computation budget of the players to compute proximal operator with high accuracy, and reach the ϵ-neighborhood of the equilibrium using only logarithmically many communications, which is comparable to the complexity from deterministic (full-batch) scenario. This is in sharp contrast with the case of PEARL-SGD, which requires at least $\Omega\left(\epsilon^{-1/2}\right)$ communication rounds in the stochastic setup, regardless of how large τ is. For a more detailed argument on this, we refer the readers to Appendix D.

5. Conclusion

We provide a formal characterization of player drift in multiplayer federated learning, which indicates the divergence of game dynamics due to excessive local computations. As a resolution to player drift, we introduce the novel PEARL-Prox algorithm. We present theoretical convergence guarantees for PEARL-Prox, demonstrating that it can take the full advantage of large local computes, leading to a significant improvement in communication efficiency compared to PEARL-SGD, which suffers from player drift. We present the numerical simulations that verify our theoretical results, demonstrating the effectiveness and practicality of the proposed PEARL-Prox, in Appendix E.

References

[1] Maruan Al-Shedivat, Trapit Bansal, Yura Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. Continuous adaptation via meta-learning in nonstationary and competitive environments. *International Conference on Learning Representations*, 2018.

[2] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv:2003.13461*, 2020.

[3] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Neural Information Processing Systems*, 2020.

[4] Drew Fudenberg and Jean Tirole. *Game theory*. MIT press, 1991.

[5] Eduard Gorbunov, Filip Hanzely, and Peter Richtarik. Local SGD: Unified theory and new efficient methods. *International Conference on Artificial Intelligence and Statistics*, 2021.

[6] Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv:2002.05516*, 2020.

[7] Filip Hanzely, Slavomír Hanzely, Samuel Horváth, and Peter Richtarik. Lower bounds and optimal algorithms for personalized federated learning. *Neural Information Processing Systems*, 2020.

[8] Filip Hanzely, Boxin Zhao, and mladen kolar. Personalized federated learning: A unified framework and universal optimization techniques. *Transactions on Machine Learning Research*, 2023.

[9] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning. *International Conference on Machine Learning*, 2020.

[10] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv:1610.02527*, 2016.

[11] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *NeurIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[12] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[13] Yang Liu, Xinwei Zhang, Yan Kang, Liping Li, Tianjian Chen, Mingyi Hong, and Qiang Yang. FedBCD: A communication-efficient collaborative learning framework for distributed features. *IEEE Transactions on Signal Processing*, 70:4277–4290, 2022.

[14] Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. Vertical federated learning: Concepts, advances, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3615–3634, 2024.

[15] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Neural Information Processing Systems*, 30, 2017.

[16] Michael Maschler, Shmuel Zamir, and Eilon Solan. *Game theory*. Cambridge University Press, 2020.

[17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. *International Conference on Artificial Intelligence and Statistics*, 2017.

[18] Konstantin Mishchenko, Grigory Malinovsky, Sebastian Stich, and Peter Richtarik. ProxSkip: Yes! Local gradient steps provably lead to communication acceleration! Finally! *International Conference on Machine Learning*, 2022.

[19] Aritra Mitra, Rayana Jaafar, George J. Pappas, and Hamed Hassani. Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients. *Neural Information Processing Systems*, 2021.

[20] Jiaju Qi, Qihao Zhou, Lei Lei, and Kan Zheng. Federated reinforcement learning: techniques, applications, and open challenges. *Intelligence & Robotics*, 1(1):18–57, 2021.

[21] Ronald L Rivest, Len Adleman, Michael L Dertouzos, et al. On data banks and privacy homomorphisms. *Foundations of secure computation*, 4(11):169–180, 1978.

[22] Sebastian U Stich. Unified optimal analysis of the (stochastic) gradient method. *arXiv:1907.04232*, 2019.

[23] Canh T. Dinh, Nguyen Tran, and Josh Nguyen. Personalized federated learning with moreau envelopes. *Neural Information Processing Systems*, 2020.

[24] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2), 2019.

[25] TaeHo Yoon, Sayantan Choudhury, and Nicolas Loizou. Multiplayer federated learning: Reaching equilibrium with less communication. *arXiv:2501.08263*, 2025.

[26] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. *Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms*, pages 321–384. Springer International Publishing, 2021.

[27] Siqi Zhang, Sayantan Choudhury, Sebastian U Stich, and Nicolas Loizou. Communication-efficient gradient descent-accent methods for distributed variational inequalities: Unified analysis and local updates. *International Conference on Learning Representations*, 2024.

[28] Jingang Zhao. The equilibria of a multiple objective game. *International Journal of Game Theory*, 20(2):171–182, 1991.

[29] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv:1806.00582*, 2018.

Appendix A. Related work and connection to other FL frameworks

Below, we discuss connections between MpFL and personalized FL, vertical FL and federated/multiagent reinforcement learning.

Personalized federated learning. In personalized federated learning (PFL), clients cooperate as in FL, but with the goal of training local models x^i, performing well on their local dataset $\mathcal{D}_i$ (rather than producing a single global model) [6, 23]. Multiple approaches such as mixing the local and global models [2, 6], model-agnostic meta-learning (MAML) based formulation [3] or bilevel optimization formulation using Moreau envelopes [23] have been proposed and gained popularity. Among these, consider the following characterization of PFL which does not require to explicitly keep a separate global model apart from local models [6–8]:

$$\underset{\mathbf{x}=(x^1,\dots,x^n)\in\mathbb{R}^{nd}}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^n h_i(x_i) + \frac{\alpha}{2n}\sum_{i=1}^n \|x_i - \overline{x}\|^2,$$

where $x_1,\dots,x_n \in \mathbb{R}^d$ are local models of each player, $h_1,\dots,h_n$ are the local losses following each player's distributions, $\overline{x} = \frac{1}{n}\sum_{i=1}^n x_i$ is the average model and $\alpha > 0$ is a hyperparameter. As the first-order optimality condition to the above problem is given by $x_i^\star - \overline{x}^\star + \frac{1}{\alpha}\nabla h_i(x_i^\star) = 0$ [7], when each h_i is convex, this is equivalent to finding an equilibrium of the n-player game where each player has the objective function $f_i(x_i; x_{-i}) = h_i(x_i) + \frac{\alpha}{2}\sum_{i=1}^n \|x_i - \overline{x}\|^2$. That is, one of the popular formulations for PFL can be viewed as an instance of an MpFL problem.

Vertical federated learning. Classical FL implicitly assumes the "horizontal" FL setting, where each client holds data of distinct users but with identical features [24]. In the vertical federated learning (VFL) setting, multiple clients rather possess data of distinct features for the identical set of users; in this scenario, the clients require a coordinated collaboration both in the training phase and in the inference phase [14]. We formally describe the formulated provided in [13]: the n clients have the dataset with N data $\mathcal{D} = \{\phi_j, y_j\}_{j=1}^N$, consisting of feature vectors $\phi_j = (\phi_{j,1},\dots,\phi_{j,n}) \in \mathbb{R}^{m_1+\cdots+m_n}$ and the corresponding classification label y_j. Client i has access to the i-th 'block feature' $\phi_{j,i} \in \mathbb{R}^{m_i}$ for each data $j = 1,\dots,N$, i.e., has the dataset $\mathcal{D}_i = \{\phi_{j,i}\}_{j=1}^N$. Each client has a model parameter $x_i \in \mathbb{R}^{d_i}$, and they collaborate to solve the problem

$$\underset{(x_1,\dots,x_n)\in\mathbb{R}^D}{\text{minimize}} \quad \frac{1}{N}\sum_{j=1}^N \ell\left(x_1,\dots,x_n;\phi_j,y_j\right) + \alpha\sum_{i=1}^n g(x_i)$$

where ℓ is a loss function (depending on model parameters of all clients), $g\colon \mathbb{R}^{d_i} \to \mathbb{R}$ are client-wise regularizers, $\alpha > 0$ is a hyperparameter and $D = d_1 + \cdots + d_n$. This scenario can be viewed as a multiplayer game where each player $i \in [n]$ has the objective function $f_i(x_i; x_{-i}) = \frac{1}{N}\sum_{j=1}^N \ell\left(x_1,\dots,x_n;\phi_j,y_j\right) + \alpha g(x_i)$. One can verify that PEARL-SGD applied to this setup (viewing the setting as MpFL) gives the update rule identical to that of the FedBCD algorithm in [13]. However, we note that VFL applications generally require additional systematic components, e.g. homomorphic encryption [21] to keep x_i private, while in MpFL, communication efficiency is of primary interest and privacy is considered an auxiliary (an orthogonal) issue.

Multiagent reinforcement learning. In multiagent reinforcement learning (MARL), n agents interact within a common environment with the set of states $\mathcal{S}$. Each agent i has its action space $\mathcal{A}_i$,

and tries to optimize a policy $\pi_i \colon \mathcal{S} \to \Delta(\mathcal{A}_i)$. Given a state s and the joint actions $\mathbf{a} = (a_1, \ldots, a_n)$ of all players, the environment yields a next state s_{t+1} according to a distribution $P(\cdot|s_t, a_t)$, and rewards $R_i(s_t, \mathbf{a}_t, s_{t+1})$ for each agent. The value function of the agent i is defined as

$$V_{\pi_i, \pi_{-i}}(s) = \mathbb{E}_{s_{t+1} \sim P(\cdot|s_t, a_t), a_{-i} \sim \pi_{-i}(\cdot|s_t)} \left[\sum_{t \geq 0} \gamma^t R_i(s_t, \mathbf{a}_t, s_{t+1}) \,\middle|\, a_{i,t} \sim \pi_i(\cdot|s_t), s_0 = s \right]$$

The concept of Nash equilibrium among agents is characterized by $V_{\pi_i^\star, \pi_{-i}^\star}(s) \geq V_{\pi_i, \pi_{-i}^\star}(s), \forall s \in \mathcal{S}$ [26] for any policy π_i, $i = 1, \ldots, n$. The significant challenge of solving this problem arises from both its game-theoretic structure and non-stationarity of the environment from the perspective of each agent [1, 15, 20]. One interpretation of the problem is that it seeks Nash equilibria for $|\mathcal{S}|$ distinct games, one for each state $s \in \mathcal{S}$. MpFL does not immediately subsume this formulation in its basic form, but there exists a close connection which could be made precise by extending the framework of MpFL using multiobjective games [28], where π_i being the decision variable (action) of the player i and objective functions are value functions with one component per each $s \in \mathcal{S}$.

Appendix B. Missing proofs for Section 4

B.1. Discussion on Assumption 5 and a key lemma

In Assumption 5, we consider the error bound of the form (3) because most optimization algorithms minimizing a strongly convex function $h(z)$ exhibit a convergence guarantee of the form

$$\mathbb{E}\left[\left\|z^T - z^\star\right\|^2\right] \leq a_T \left\|z^0 - z^\star\right\|^2 + b_T, \tag{5}$$

where T is the total iteration number, z^0 is the initial point and $z^\star = \operatorname{argmin}_z h(z)$. In our context, each player i, at round p, takes $f_i(\cdot; x_{-i}^p) + \frac{\lambda}{2}\|\cdot - x_i^p\|^2$ as the objective function (whose minimum is $\operatorname{Prox}_{\frac{1}{\lambda} f_i(\cdot; x_{-i}^p)}(x_i^p)$), x_i^p as the initial point, and execute a minimization subroutine of their choice to output the approximate minimizer x_i^{p+1}. In this setting, the left hand side of (5) will be $\mathbb{E}\left[\|v_i^p\|^2 \middle| x_i^p\right]$. The $\left\|z^0 - z^\star\right\|^2 = \left\|x_i^p - \operatorname{Prox}_{\frac{1}{\lambda} f_i(\cdot; x_{-i}^p)}(x_i^p)\right\|^2$ term in the right hand side can be upper bounded using the following Lemma 8. Combined, (5) can be rewritten into the form (3).

Lemma 8 *Suppose $f_i(\cdot; x_{-i}^p)\colon \mathbb{R}^{d_i} \to \mathbb{R}$ is convex and let $\lambda > 0$. Then*

$$\left\|x_i^p - \operatorname{Prox}_{\frac{1}{\lambda} f_i(\cdot; x_{-i}^p)}(x_i^p)\right\| \leq \frac{1}{\lambda}\left\|\nabla_{x_i} f_i(x_i^p; x_{-i}^p)\right\|.$$

Proof Note that $\operatorname{Prox}_{\frac{1}{\lambda} f_i(\cdot; x_{-i}^p)}(x_i^p) = \left(\mathrm{Id} + \frac{1}{\lambda}\nabla_{x_i} f_i(\cdot; x_{-i}^p)\right)^{-1}(x_i^p)$ and

$$\nabla_{x_i} f_i(\cdot; x_{-i}^p)\colon \mathbb{R}^{d_i} \to \mathbb{R}^{d_i}$$

is a monotone operator, so $\operatorname{Prox}_{\frac{1}{\lambda} f_i(\cdot; x_{-i}^p)}\colon \mathbb{R}^{d_i} \to \mathbb{R}^{d_i}$ is nonexpansive. Next, we have

$$x_i^p = \operatorname{Prox}_{\frac{1}{\lambda} f_i(\cdot; x_{-i}^p)}\left(x_i^p + \frac{1}{\lambda}\nabla_{x_i} f_i(x_i^p; x_{-i}^p)\right)$$

and therefore,

$$\begin{aligned}
\left\|x_i^p - \operatorname{Prox}_{\frac{1}{\lambda} f_i(\cdot; x_{-i}^p)}(x_i^p)\right\| &= \left\|\operatorname{Prox}_{\frac{1}{\lambda} f_i(\cdot; x_{-i}^p)}\left(x_i^p + \frac{1}{\lambda}\nabla_{x_i} f_i(x_i^p; x_{-i}^p)\right) - \operatorname{Prox}_{\frac{1}{\lambda} f_i(\cdot; x_{-i}^p)}(x_i^p)\right\| \\
&\leq \left\|\left(x_i^p + \frac{1}{\lambda}\nabla_{x_i} f_i(x_i^p; x_{-i}^p)\right) - x_i^p\right\| \\
&= \frac{1}{\lambda}\left\|\nabla_{x_i} f_i(x_i^p; x_{-i}^p)\right\|.
\end{aligned}$$

■

B.2. Proof of Theorem 4

As each player computes proximal operator exactly, we have

$$x_i^{p+1} = \operatorname{Prox}_{\frac{1}{\lambda} f_i(\cdot; x_{-i}^p)}(x_i^p)$$

for $i = 1, \ldots, n$ and therefore, by Lemma 8,

$$\left\|x_i^p - x_i^{p+1}\right\| \leq \frac{1}{\lambda} \left\|\nabla_{x_i} f_i(x_i^p; x_{-i}^p)\right\|.$$

Now observe that for each $p = 0, \ldots, R-1$,

$$\begin{aligned}
\left\|\mathbf{x}^{p+1} - \mathbf{x}^\star\right\|^2 &= \left\|\mathbf{x}^p - \mathbf{x}^\star - (\mathbf{x}^p - \mathbf{x}^{p+1})\right\|^2 \\
&= \left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 - 2\left\langle \mathbf{x}^p - \mathbf{x}^{p+1}, \mathbf{x}^p - \mathbf{x}^\star \right\rangle + \left\|\mathbf{x}^p - \mathbf{x}^{p+1}\right\|^2 \\
&= \left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 - 2\sum_{i=1}^n \left\langle x_i^p - x_i^{p+1}, x_i^p - x_i^\star \right\rangle + \sum_{i=1}^n \left\|x_i^p - x_i^{p+1}\right\|^2. \qquad (6)
\end{aligned}$$

Because $x_i^p - x_i^{p+1} = \frac{1}{\lambda}\nabla_{x_i} f_i(x_i^{p+1}; x_{-i}^p)$, we can write

$$x_i^p - x_i^{p+1} = \frac{1}{\lambda}\nabla_{x_i} f_i(x_i^p; x_{-i}^p) + \frac{1}{\lambda}\underbrace{\left(\nabla_{x_i} f_i(x_i^{p+1}; x_{-i}^p) - \nabla_{x_i} f_i(x_i^p; x_{-i}^p)\right)}_{:=\delta_i^p}$$

where $\|\delta_i^p\| \leq L_i \left\|x_i^{p+1} - x_i^p\right\|$. Plugging this into (6) we obtain

$$\begin{aligned}
\left\|\mathbf{x}^{p+1} - \mathbf{x}^\star\right\|^2 &= \left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 - \frac{2}{\lambda}\sum_{i=1}^n \left\langle \nabla_{x_i} f_i(x_i^p; x_{-i}^p), x_i^p - x_i^\star \right\rangle \\
&\quad - \frac{2}{\lambda}\sum_{i=1}^n \langle \delta_i^p, x_i^p - x_i^\star \rangle + \sum_{i=1}^n \left\|x_i^p - x_i^{p+1}\right\|^2 \\
&\leq \left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 - \frac{2}{\lambda}\left\langle F(\mathbf{x}^p), \mathbf{x}^p - \mathbf{x}^\star \right\rangle \\
&\quad + \frac{1}{\lambda}\sum_{i=1}^n \left(\alpha \left\|x_i^p - x_i^\star\right\|^2 + \frac{1}{\alpha}\left\|\delta_i^p\right\|^2\right) + \sum_{i=1}^n \left\|x_i^p - x_i^{p+1}\right\|^2 \\
&\leq \left(1 + \frac{\alpha}{\lambda}\right)\left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 - \frac{2}{\lambda}\left\langle F(\mathbf{x}^p), \mathbf{x}^p - \mathbf{x}^\star \right\rangle + \sum_{i=1}^n \left(1 + \frac{L_i^2}{\lambda\alpha}\right)\left\|x_i^p - x_i^{p+1}\right\|^2 \\
&\leq \left(1 + \frac{\alpha}{\lambda}\right)\left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 - \frac{2}{\lambda}\left\langle F(\mathbf{x}^p), \mathbf{x}^p - \mathbf{x}^\star \right\rangle + \left(1 + \frac{L_{\max}^2}{\lambda\alpha}\right)\sum_{i=1}^n \frac{1}{\lambda^2}\left\|\nabla_{x_i} f_i(x_i^p; x_{-i}^p)\right\|^2 \\
&= \left(1 + \frac{\alpha}{\lambda}\right)\left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 - \frac{2}{\lambda}\left\langle F(\mathbf{x}^p), \mathbf{x}^p - \mathbf{x}^\star \right\rangle + \left(1 + \frac{L_{\max}^2}{\lambda\alpha}\right)\frac{1}{\lambda^2}\left\|F(\mathbf{x}^p)\right\|^2
\end{aligned}$$

where for the first inequality, we use Young's inequality (with $\alpha > 0$ to be determined later), for the second inequality we use the identity $\left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 = \sum_{i=1}^n \left\|x_i^p - x_i^\star\right\|^2$ and $\|\delta_i^p\| \leq L_i \left\|x_i^{p+1} - x_i^p\right\|$, and for the last inequality, we use Lemma 8 with $L_i \leq L_{\max}$.

Now, using star-cocoercivity of F we have

$$\begin{aligned}
\left\|\mathbf{x}^{p+1} - \mathbf{x}^\star\right\|^2 &\leq \left(1 + \frac{\alpha}{\lambda}\right)\left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 - \frac{2}{\lambda}\left\langle F(\mathbf{x}^p), \mathbf{x}^p - \mathbf{x}^\star \right\rangle + \left(1 + \frac{L_{\max}^2}{\lambda\alpha}\right)\frac{\ell}{\lambda^2}\left\langle F(\mathbf{x}^p), \mathbf{x}^p - \mathbf{x}^\star \right\rangle \\
&\leq \left(1 + \frac{\alpha}{\lambda}\right)\left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 - \underbrace{\left(\frac{2}{\lambda} - \left(1 + \frac{L_{\max}^2}{\lambda\alpha}\right)\frac{\ell}{\lambda^2}\right)}_{:=C}\left\langle F(\mathbf{x}^p), \mathbf{x}^p - \mathbf{x}^\star \right\rangle.
\end{aligned}$$

Provided that $C > 0$, we can further use μ-quasi-strong monotonicity of F to obtain

$$\begin{aligned}\left\|\mathbf{x}^{p+1} - \mathbf{x}^\star\right\|^2 &\leq \left(1 + \frac{\alpha}{\lambda}\right)\left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 - C\mu\left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 \\ &= \left[1 + \frac{\alpha}{\lambda} - \mu\left(\frac{2}{\lambda} - \frac{\ell}{\lambda^2}\left(1 + \frac{L_{\max}^2}{\lambda\alpha}\right)\right)\right]\left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2.\end{aligned}$$

Now to minimize the last factor

$$1 + \frac{\alpha}{\lambda} - \mu\left(\frac{2}{\lambda} - \frac{\ell}{\lambda^2}\left(1 + \frac{L_{\max}^2}{\lambda\alpha}\right)\right) = 1 - \frac{2\mu}{\lambda} + \frac{\mu\ell}{\lambda^2} + \frac{\alpha}{\lambda} + \frac{\mu\ell L_{\max}^2}{\lambda^3\alpha}$$

with respect to α, we choose $\alpha = \frac{L_{\max}\sqrt{\mu\ell}}{\lambda}$, which gives

$$1 - \frac{2\mu}{\lambda} + \frac{\mu\ell}{\lambda^2} + \frac{2L_{\max}\sqrt{\mu\ell}}{\lambda^2} = 1 - \frac{2\mu}{\lambda}\left(1 - \frac{\ell}{2\lambda} - \frac{L_{\max}\sqrt{\ell/\mu}}{\lambda}\right) = 1 - \frac{2\mu\zeta}{\lambda}$$

where $\zeta = 1 - \frac{\ell + 2L_{\max}\sqrt{\kappa}}{2\lambda} > 0$ by the choice of λ. Finally, we verify that with our choice of α,

$$C = \frac{2}{\lambda}\left(1 - \frac{\ell + L_{\max}\sqrt{\kappa}}{2\lambda}\right) > \frac{2}{\lambda}\left(\zeta + \frac{L_{\max}\sqrt{\kappa}}{2\lambda}\right) > 0.$$

B.3. Proof of Theorem 6

Denote $\tilde{x}_i^{p+1} = \mathrm{Prox}_{\frac{1}{\lambda}f_i(\cdot;x_{-i}^p)}(x_i^p)$, so that $x_i^{p+1} = \tilde{x}_i^{p+1} + v_i^p$ where $v_i^p \in \mathbb{R}^{d_i}$ is a random vector such that and $\mathbb{E}\left[\left\|v_i^p\right\|^2 \middle| x_i^p\right] \leq \delta_i^p\left\|\nabla_{x_i} f_i(x_i^p; x_{-i}^p)\right\|^2 + \epsilon_i^p$. Then for each $i = 1, \ldots, n$, we have

$$\begin{aligned}\mathbb{E}\left[\left\|x_i^p - x_i^{p+1}\right\|^2 \middle| x_i^p\right] &= \mathbb{E}\left[\left\|x_i^p - \tilde{x}_i^{p+1} - v_i^p\right\|^2 \middle| x_i^p\right] \\ &\leq \mathbb{E}\left[2\left\|x_i^p - \tilde{x}_i^{p+1}\right\|^2 + 2\left\|v_i^p\right\|^2 \middle| x_i^p\right] \\ &= 2\left\|x_i^p - \tilde{x}_i^{p+1}\right\|^2 + 2\left(\delta_i^p\left\|\nabla_{x_i} f_i(x_i^p; x_{-i}^p)\right\|^2 + \epsilon_i^p\right) \qquad (7)\end{aligned}$$

where in the third line, we use the fact that $\tilde{x}_i^{p+1}$ is a non-random quantity given x_i^p. Now observe that for each $p = 0, \ldots, R-1$,

$$\begin{aligned}\left\|\mathbf{x}^{p+1} - \mathbf{x}^\star\right\|^2 &= \left\|\mathbf{x}^p - \mathbf{x}^\star - (\mathbf{x}^p - \mathbf{x}^{p+1})\right\|^2 \\ &= \left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 - 2\left\langle \mathbf{x}^p - \mathbf{x}^{p+1}, \mathbf{x}^p - \mathbf{x}^\star\right\rangle + \left\|\mathbf{x}^p - \mathbf{x}^{p+1}\right\|^2 \\ &= \left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 - 2\sum_{i=1}^n\left\langle x_i^p - x_i^{p+1}, x_i^p - x_i^\star\right\rangle + \sum_{i=1}^n\left\|x_i^p - x_i^{p+1}\right\|^2. \qquad (8)\end{aligned}$$

Because $x_i^p - x_i^{p+1} = x_i^p - \tilde{x}_i^{p+1} - v_i^p = \frac{1}{\lambda}\nabla_{x_i} f_i(\tilde{x}_i^{p+1}; x_{-i}^p) - v_i^p$, we can write

$$x_i^p - x_i^{p+1} = \frac{1}{\lambda}\nabla_{x_i} f_i(x_i^p; x_{-i}^p) + \frac{1}{\lambda}\left(\nabla_{x_i} f_i(\tilde{x}_i^{p+1}; x_{-i}^p) - \nabla_{x_i} f_i(x_i^p; x_{-i}^p)\right) - v_i^p.$$

Plugging this into the inner product term of (8), we obtain

$$\begin{aligned}
\left\|\mathbf{x}^{p+1}-\mathbf{x}^\star\right\|^2 &= \|\mathbf{x}^p-\mathbf{x}^\star\|^2 - \frac{2}{\lambda}\sum_{i=1}^n \left\langle \nabla_{x_i} f_i(x_i^p; x_{-i}^p), x_i^p - x_i^\star\right\rangle \\
&\quad - 2\sum_{i=1}^n \left\langle \frac{1}{\lambda}\left(\nabla_{x_i} f_i(\tilde{x}_i^{p+1}; x_{-i}^p) - \nabla_{x_i} f_i(x_i^p; x_{-i}^p)\right) - v_i^p, x_i^p - x_i^\star\right\rangle + \sum_{i=1}^n \left\|x_i^p - x_i^{p+1}\right\|^2 \\
&\le \|\mathbf{x}^p-\mathbf{x}^\star\|^2 - \frac{2}{\lambda}\left\langle F(\mathbf{x}^p), \mathbf{x}^p-\mathbf{x}^\star\right\rangle \\
&\quad + \frac{1}{\lambda}\sum_{i=1}^n \left(\alpha\left\|x_i^p - x_i^\star\right\|^2 + \frac{1}{\alpha}\left\|\nabla_{x_i} f_i(\tilde{x}_i^{p+1}; x_{-i}^p) - \nabla_{x_i} f_i(x_i^p; x_{-i}^p)\right\|^2\right) \\
&\quad + \sum_{i=1}^n \left(\beta\left\|x_i^p - x_i^\star\right\|^2 + \frac{1}{\beta}\left\|v_i^p\right\|^2\right) + \sum_{i=1}^n \left\|x_i^p - x_i^{p+1}\right\|^2 \\
&\le \left(1+\frac{\alpha}{\lambda}+\beta\right)\|\mathbf{x}^p-\mathbf{x}^\star\|^2 - \frac{2}{\lambda}\left\langle F(\mathbf{x}^p), \mathbf{x}^p-\mathbf{x}^\star\right\rangle + \sum_{i=1}^n \frac{L_i^2}{\lambda\alpha}\left\|\tilde{x}_i^{p+1} - x_i^p\right\|^2 \\
&\quad + \sum_{i=1}^n \left\|x_i^p - x_i^{p+1}\right\|^2 + \frac{1}{\beta}\sum_{i=1}^n \left\|v_i^p\right\|^2
\end{aligned}$$

where the last inequality uses L_i-smoothness of $f_i(\cdot; x_{-i}^p)$ and the constants $\alpha, \beta > 0$ arising from the application of Young's inequality are to be determined later. Taking the expectation of both sides (conditioned on $\mathbf{x}^p$) and using (7), we obtain

$$\begin{aligned}
\mathbb{E}\left[\left\|\mathbf{x}^{p+1}-\mathbf{x}^\star\right\|^2 \middle| \mathbf{x}^p\right] &\le \left(1+\frac{\alpha}{\lambda}+\beta\right)\|\mathbf{x}^p-\mathbf{x}^\star\|^2 - \frac{2}{\lambda}\left\langle F(\mathbf{x}^p), \mathbf{x}^p-\mathbf{x}^\star\right\rangle \\
&\quad + \sum_{i=1}^n \left(2+\frac{L_i^2}{\lambda\alpha}\right)\left\|x_i^p - \tilde{x}_i^{p+1}\right\|^2 + \sum_{i=1}^n 2\delta_i^p \left\|\nabla_{x_i} f_i(x_i^p; x_{-i}^p)\right\|^2 + \left(2+\frac{1}{\beta}\right)\sum_{i=1}^p \epsilon_i^p.
\end{aligned}$$

Now using Lemma 8 to bound $\left\|x_i^p - \tilde{x}_i^{p+1}\right\|^2 \leq \frac{1}{\lambda^2}\left\|\nabla_{x_i} f_i(x_i^p; x_{-i}^p)\right\|^2$ and denoting $\delta_{\max} = \max_{\substack{i=1,\ldots,n \\ p=0,\ldots,R-1}} \delta_i^p$ we obtain

$$\begin{aligned}
\mathbb{E}\left[\left\|\mathbf{x}^{p+1} - \mathbf{x}^\star\right\|^2 \middle| \mathbf{x}^p\right] &\leq \left(1 + \frac{\alpha}{\lambda} + \beta\right)\left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 - \frac{2}{\lambda}\left\langle F(\mathbf{x}^p), \mathbf{x}^p - \mathbf{x}^\star\right\rangle \\
&\quad + \sum_{i=1}^n \left(\frac{1}{\lambda^2}\left(2 + \frac{L_i^2}{\lambda\alpha}\right) + 2\delta_i^p\right)\left\|\nabla_{x_i} f_i(x_i^p; x_{-i}^p)\right\|^2 + \left(2 + \frac{1}{\beta}\right)\sum_{i=1}^p \epsilon_i^p \\
&\leq \left(1 + \frac{\alpha}{\lambda} + \beta\right)\left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 - \frac{2}{\lambda}\left\langle F(\mathbf{x}^p), \mathbf{x}^p - \mathbf{x}^\star\right\rangle \\
&\quad + \left[\frac{1}{\lambda^2}\left(2 + \frac{L_{\max}^2}{\lambda\alpha}\right) + 2\delta_{\max}\right]\left\|F(\mathbf{x}^p)\right\|^2 + \left(2 + \frac{1}{\beta}\right)\sum_{i=1}^n \epsilon_i^p \\
&\leq \left(1 + \frac{\alpha}{\lambda} + \beta\right)\left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 - \left[\frac{2}{\lambda} - \ell\left(\frac{1}{\lambda^2}\left(2 + \frac{L_{\max}^2}{\lambda\alpha}\right) + 2\delta_{\max}\right)\right]\left\langle F(\mathbf{x}^p), \mathbf{x}^p - \mathbf{x}^\star\right\rangle \\
&\quad + \left(2 + \frac{1}{\beta}\right)\sum_{i=1}^n \epsilon_i^p
\end{aligned}$$

where the last inequality uses star-cocoercivity $\left\|F(\mathbf{x}^p)\right\|^2 \leq \ell\left\langle F(\mathbf{x}^p), \mathbf{x}^p - \mathbf{x}^\star\right\rangle$. Next, assuming $C = \frac{2}{\lambda} - \frac{1}{\lambda^2}\left(2 + \frac{L_{\max}^2}{\lambda\alpha}\right) - 2\delta_{\max} > 0$ (we verify this below with a particular choice of α) and using quasi-strong monotonicity $\left\langle F(\mathbf{x}^p), \mathbf{x}^p - \mathbf{x}^\star\right\rangle \geq \mu\left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2$, we have

$$\begin{aligned}
&\mathbb{E}\left[\left\|\mathbf{x}^{p+1} - \mathbf{x}^\star\right\|^2 \middle| \mathbf{x}^p\right] \\
&\leq \left[1 + \frac{\alpha}{\lambda} + \beta - \frac{2\mu}{\lambda} + \mu\ell\left(\frac{1}{\lambda^2}\left(2 + \frac{L_{\max}^2}{\lambda\alpha}\right) + 2\delta_{\max}\right)\right]\left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 + \left(2 + \frac{1}{\beta}\right)\sum_{i=1}^n \epsilon_i^p
\end{aligned} \tag{9}$$

Now we take $\beta = \frac{\mu}{2\lambda}$ and use the condition $\delta_{\max} \leq \frac{1}{4\lambda\ell} \implies 2\mu\ell\delta_{\max} \leq \frac{\mu}{2\lambda}$ to bound the coefficient of the $\left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2$ term as

$$1 + \frac{\alpha}{\lambda} + \beta - \frac{2\mu}{\lambda} + \mu\ell\left(\frac{1}{\lambda^2}\left(2 + \frac{L_{\max}^2}{\lambda\alpha}\right) + 2\delta_{\max}\right) \leq 1 - \frac{\mu}{\lambda} + \frac{2\mu\ell}{\lambda^2} + \frac{\alpha}{\lambda} + \frac{\mu\ell L_{\max}^2}{\lambda^3\alpha}.$$

Now taking $\alpha = \frac{L_{\max}\sqrt{\mu\ell}}{\lambda}$ to optimize the right hand side, (9) becomes

$$\begin{aligned}
\mathbb{E}\left[\left\|\mathbf{x}^{p+1} - \mathbf{x}^\star\right\|^2 \middle| \mathbf{x}^p\right] &\leq \left(1 - \frac{\mu}{\lambda} + \frac{2\mu\ell + 2\sqrt{\mu\ell}L_{\max}}{\lambda^2}\right)\left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 + \left(2 + \frac{2\lambda}{\mu}\right)\sum_{i=1}^n \epsilon_i^p \\
&\leq \left(1 - \frac{\mu\zeta}{\lambda}\right)\left\|\mathbf{x}^p - \mathbf{x}^\star\right\|^2 + \left(2 + \frac{2\lambda}{\mu}\right)\sum_{i=1}^n \epsilon_i^p
\end{aligned}$$

where $\zeta = 1 - \frac{2\ell + 2L_{\max}\sqrt{\kappa}}{\lambda} > 0$. Note that with this choice of α, we have $1 + \frac{\alpha}{\lambda} + \beta - C\mu = 1 - \frac{\mu\zeta}{\lambda} < 1$, which implies that $C > 0$, so the step deriving (9) is valid. Unrolling the recursion and taking the total expectation, we conclude that

$$\mathbb{E}\left[\left\|\mathbf{x}^R - \mathbf{x}^\star\right\|^2\right] \leq \left(1 - \frac{\mu\zeta}{\lambda}\right)^R\left\|\mathbf{x}^0 - \mathbf{x}^\star\right\|^2 + \left(2 + \frac{2\lambda}{\mu}\right)\sum_{p=0}^{R-1}\left(1 - \frac{\mu\zeta}{\lambda}\right)^{R-p-1}\sum_{i=1}^n \epsilon_i^p.$$

B.4. Proof of Corollary 7

We first state a general convergence result for SGD.

Lemma 9 *Let $h(\cdot) = \mathbb{E}_{\xi\sim\mathcal{D}}[h_\xi(\cdot)]\colon \mathbb{R}^d \to \mathbb{R}$ be μ-strongly convex and L-smooth, and assume that there exists $\sigma > 0$ such that $\mathbb{E}\left[\|\nabla h_\xi(x) - \nabla h(x)\|^2\right] \le \sigma^2$ for all $x \in \mathbb{R}^d$. Then SGD with constant step size $\gamma \in (0, \frac{1}{2L}]$ satisfies*

$$\left\|x^T - x^\star\right\|^2 \le (1-\gamma\mu)^T \left\|x^0 - x^\star\right\|^2 + \frac{\gamma\sigma^2}{\mu}.$$

Proof The proof uses standard arguments and can be found in, e.g., [22] (see Lemma 1 and the discussion thereafter). ■

Now, we apply Lemma 9 to the regularized objectives $h(\cdot) = f_i(\cdot; x^p_{-i}) + \frac{\lambda}{2}\|\cdot - x_i^p\|^2$ (which are λ-strongly convex, $(\lambda + L_i)$-smooth) with unique minimum $\mathrm{Prox}_{\frac{1}{\lambda} f_i(\cdot;x^p_{-i})}(x_i^p)$, and $\nabla_{x_i} f_{i,\xi_i}(\cdot; x^p_{-i})$ which are unbiased estimators of $\nabla_{x_i} f_i(\cdot; x^p_{-i})$ with bounded variance $\le \sigma_i^2$ (by Assumption 3). Then, provided that $\gamma \le \frac{1}{2(\lambda + L_{\max})}$, Algorithm 4 with $\tau_i^p \equiv \tau$ using constant step size γ satisfies

$$\begin{aligned}
\left\|x_i^{p+1} - \mathrm{Prox}_{\frac{1}{\lambda} f_i(\cdot;x^p_{-i})}(x_i^p)\right\|^2 &\le (1-\gamma\lambda)^\tau \left\|x_i^p - \mathrm{Prox}_{\frac{1}{\lambda} f_i(\cdot;x^p_{-i})}(x_i^p)\right\|^2 + \frac{\gamma\sigma_i^2}{\lambda} \\
&\le (1-\gamma\lambda)^\tau \frac{1}{\lambda^2}\left\|\nabla_{x_i} f_i(x_i^p; x^p_{-i})\right\|^2 + \frac{\gamma\sigma_i^2}{\lambda} \\
&\le \frac{e^{-\gamma\lambda\tau}}{\lambda^2}\left\|\nabla_{x_i} f_i(x_i^p; x^p_{-i})\right\|^2 + \frac{\gamma\sigma_i^2}{\lambda}.
\end{aligned}$$

where in the second line we use Lemma 8 and the third line uses $1 + t \le e^t$ for $t \in \mathbb{R}$. Now taking $\gamma = \frac{2\log\tau}{\lambda\tau}$ the above bound becomes

$$\left\|x_i^{p+1} - \mathrm{Prox}_{\frac{1}{\lambda} f_i(\cdot;x^p_{-i})}(x_i^p)\right\|^2 \le \frac{1}{\lambda^2\tau^2}\left\|\nabla_{x_i} f_i(x_i^p; x^p_{-i})\right\|^2 + \frac{2\sigma_i^2\log\tau}{\lambda^2\tau}$$

Hence, Algorithm 4 satisfies Assumption 5 with $\delta_i^p = \frac{1}{\lambda^2\tau^2}$ and $\epsilon_i^p = \frac{2\sigma_i^2\log\tau}{\lambda^2\tau}$. Assuming that $\frac{1}{\lambda^2\tau^2} \le \frac{1}{4\lambda\ell}$ we can apply Theorem 6, which implies

$$\begin{aligned}
\mathbb{E}\left[\left\|\mathbf{x}^R - \mathbf{x}^\star\right\|^2\right] &\le \left(1 - \frac{\mu\zeta}{\lambda}\right)^R \left\|\mathbf{x}^0 - \mathbf{x}^\star\right\|^2 + \left(2 + \frac{2\lambda}{\mu}\right)\sum_{p=0}^{R-1}\left(1 - \frac{\mu\zeta}{\lambda}\right)^{R-p-1}\sum_{i=1}^n \epsilon_i^p \\
&\le \left(1 - \frac{\mu\zeta}{\lambda}\right)^R \left\|\mathbf{x}^0 - \mathbf{x}^\star\right\|^2 + \left(2 + \frac{2\lambda}{\mu}\right)\frac{2\log\tau}{\lambda^2\tau}\sum_{q=0}^{\infty}\left(1 - \frac{\mu\zeta}{\lambda}\right)^q \sum_{i=1}^n \sigma_i^2 \\
&= \left(1 - \frac{\mu\zeta}{\lambda}\right)^R \left\|\mathbf{x}^0 - \mathbf{x}^\star\right\|^2 + \left(2 + \frac{2\lambda}{\mu}\right)\frac{2\log\tau}{\lambda^2\tau}\sum_{q=0}^{\infty}\left(1 - \frac{\mu\zeta}{\lambda}\right)^q \sum_{i=1}^n \sigma_i^2 \\
&= \left(1 - \frac{\mu\zeta}{\lambda}\right)^R \left\|\mathbf{x}^0 - \mathbf{x}^\star\right\|^2 + \left(2 + \frac{2\lambda}{\mu}\right)\frac{2\sigma^2\log\tau}{\mu\zeta\lambda\tau}.
\end{aligned}$$

Finally, note that for the above to hold true we need to verify the two conditions on τ: first, $\delta_{\max} = \frac{1}{\lambda^2\tau^2} \leq \frac{1}{4\lambda\ell} \iff \tau \geq \sqrt{\frac{4\ell}{\lambda}}$ and $\gamma = \frac{2\log\tau}{\lambda\tau} \leq \frac{1}{2(\lambda+L_{\max})}$. The second condition is satisfied if $\tau \geq 16\left(1+\frac{L_{\max}}{\lambda}\right)^2$, as this implies

$$\frac{2\log\tau}{\lambda\tau} \leq \frac{2}{\lambda\sqrt{\tau}} \leq \frac{2}{4\lambda\left(1+\frac{L_{\max}}{\lambda}\right)} = \frac{1}{2(\lambda+L_{\max})}$$

where we use $\log\tau \leq \sqrt{\tau}$ for $\tau \geq 1$.

Appendix C. Algorithm specification

In this section, we provide the precise pseudocode for PEARL-Prox where each player uses the SGD subroutine to perform local (approximate) proximal computation, as following.

Algorithm 4: PEARL-Prox with SGD subroutine

Input: Regularization parameter $\lambda > 0$, Number of rounds $R \geq 1$, SGD parameters $\tau_i^p, \{\gamma_{i,t}^p\}_{t=0}^{\tau_i^p}$

Initialize $x^0 = (x_1^0, x_2^0, \ldots, x_n^0)$; **for** $p = 0, \ldots, R-1$ **do**

  Master server collects x_i^p from players $i \in [n]$;

  Master server distributes $\mathbf{x}^p$ back to players;

  for $i = 1, \ldots, n$ **do**

    $x_i^{p+1} \leftarrow \mathtt{SGD}\big(f_i(\cdot; x_{-i}^p) + \frac{\lambda}{2}\|\cdot - x_i^p\|^2, x_i^p, \tau_i^p, \{\gamma_{i,t}^p\}_{t=0}^{\tau_i^p}\big)$;

  end

end

Output: $x^R \in \mathbb{R}^D$

Appendix D. Discussion on communication complexity

In [25, Corollary 3.5, Theorem 3.6], it is shown that PEARL-SGD (with properly tuned hyperparameters) converges with the rate of

$$\mathbb{E}\left[\left\|\mathbf{x}^T-\mathbf{x}^\star\right\|^2\right]=\tilde{\mathcal{O}}\left(\frac{\left\|\mathbf{x}^0-\mathbf{x}^\star\right\|^2}{T^2}+\frac{\sigma^2}{\mu^2 T}+\frac{\tau^2\sigma^2}{\mu^3 T^2}\right) \tag{10}$$

where τ is the number of local SGD steps performed at each round, R is the number of communication rounds and $T=\tau R$ is the total number of iterations. (Note that here the iterates are numbered differently from our notation; their index corresponds to the cumulative number of gradient computations performed to produce it starting from $\mathbf{x}^0$, while we only keep track of the number of communications performed.) The bound (10) implies that with $\tau=\Theta(\sqrt{T})$, one can achieve a convergence rate that is asymptotically not slower than the fully communicating case $\tau=1$, while requiring a reduced number of communications $R=T/\tau=\Theta(\sqrt{T})$. In this case, the number of communications required to achieve $\mathbb{E}\left[\left\|\mathbf{x}^T-\mathbf{x}^\star\right\|^2\right]\leq\epsilon$ is $\Theta(\epsilon^{-1/2})$. The communication cost cannot be reduced further by increasing τ beyond $\Theta(\sqrt{T})$ because then the last term in (10), proportional to $\tau^2/T^2=1/R^2$, becomes dominant, and one will still require $R=\Omega(\epsilon^{-1/2})$ to attain $\mathbb{E}\left[\left\|\mathbf{x}^T-\mathbf{x}^\star\right\|^2\right]\leq\epsilon$.

In our Corollary 7 we show

$$\mathbb{E}\left[\left\|\mathbf{x}^R-\mathbf{x}^\star\right\|^2\right]=\mathcal{O}\left(e^{-\mu\zeta R/\lambda}\left\|\mathbf{x}^0-\mathbf{x}^\star\right\|^2+\frac{\sigma^2\log\tau}{\mu^2\tau}\right),$$

which implies that the dominant term (proportional to $\log\tau/\tau$) is independent of R and decreases in τ, unlike (10). This is why PEARL-Prox can take full advantage of large τ (the convergence bound only gets improved in the limit $\tau\to\infty$), and in principle, can achieve $\mathbb{E}\left[\left\|\mathbf{x}^R-\mathbf{x}^\star\right\|^2\right]\leq\epsilon$ using $R=\Theta(\log\epsilon^{-1})$ communications, provided that τ is large enough so that $\frac{\log\tau}{\tau}=\mathcal{O}\left(\frac{\mu^2\epsilon}{\sigma^2}\right)$.

Appendix E. Numerical experiments

In this section, we present numerical experiments, with the focus of validating our theoretical predictions. We additionally illustrate the advantages of PEARL-Prox in terms of the player drift.

Following the setup in [25], we consider an n-player game where $d_1 = \cdots = d_n = d$ and

$$f_i(x_i; x_{-i}) = \frac{1}{M}\sum_{m=1}^{M} \frac{1}{2} x_i^\mathsf{T} A_{i,m} x_i + \sum_{\substack{1\le j\le n \\ j\neq i}} x_i^\mathsf{T} B_{i,j,m} x_j + c_{i,m}^\mathsf{T} x_i \tag{11}$$

where $A_{i,m}, B_{i,j,m}$ are $d \times d$ symmetric matrices and $c_{i,m} \in \mathbb{R}^d$ for $i, j = 1, \ldots, n$, $j \neq i$, generated randomly for $m = 1, \ldots, M$. Here the randomness $\xi_i \sim \mathcal{D}_i$ is implemented by mini-batching from the finite sum. To ensure that (11) fulfills our theoretical assumptions, we generate random $A_{i,m}, B_{i,j,m}$ whose spectra are respectively within $[\mu_A, L_A]$ ($0 < \mu_A \le L_A$) and $[0, L_B]$ ($L_B > 0$), and take $B_{j,i,m} = -B_{i,j,m}^\mathsf{T}$ for $i \neq j$ and $m = 1, \ldots, M$. We use $n = 5, d = 10, \mu_A = 10^{-2}, L_A = 1$ and use minibatches of size 10 for stochastic gradients. We use $M = 1000, L_B = 5$ for the first experiment and $M = 100, L_B = 10$ for the rest.

Communication efficiency of PEARL-Prox. We first demonstrate the communication efficiency of PEARL-Prox (using SGD subroutine) predicted by our theory. Corollary 7 implies that with $\gamma \equiv \frac{2\log\tau}{\lambda\tau}$, if τ is large enough, the performance of Algorithm 4 will be close to linear convergence. To verify this prediction, we run Algorithm 4 with theoretical parameter choice in Corollary 7 and plot the relative distance to the equilibrium $\frac{\|\mathbf{x}^p - \mathbf{x}^\star\|^2}{\|\mathbf{x}^0 - \mathbf{x}^\star\|^2}$ versus the number of communication p (Fig. 3), for varying τ. We compare the cases $\tau \in \{17, 50, 100, 200\}$ (where $\tau = 17$ is the smallest possible value allowed under our parameter choice) with the case of exact proximal computation (dashed black line) where the convergence is linear (see Theorem 4). We indeed observe that with larger τ, the performance plot becomes closer to that of exact PEARL-Prox, despite stochasticity.

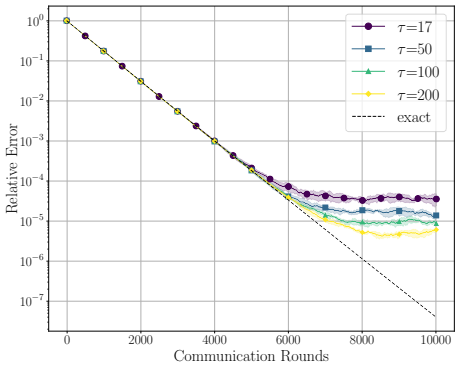
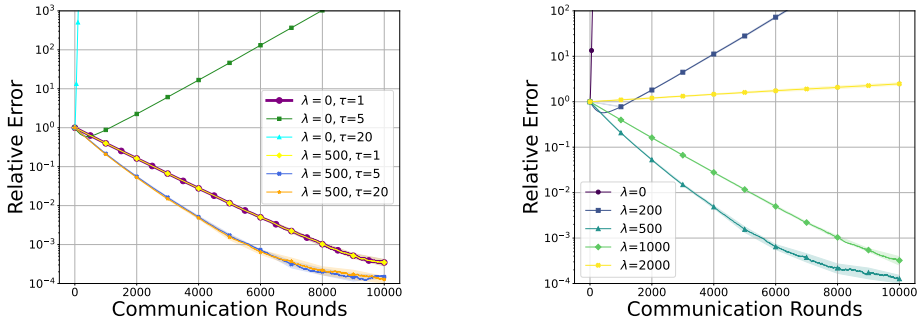


Figure 3: Plots of $\frac{\|\mathbf{x}^p - \mathbf{x}^\star\|^2}{\|\mathbf{x}^0 - \mathbf{x}^\star\|^2}$ for Algorithm 4 (PEARL-Prox with SGD) using theoretical parameters $\lambda = 4(\ell + L_{\max}\sqrt{\kappa})$, $\tau_i^p \equiv \tau$ and $\gamma_{i,t}^p \equiv \frac{2\log\tau}{\lambda\tau}$, for different τ.

Player drift (effect of increasing τ). In the following experiment we illustrate that player drift occurs for PEARL-SGD but not for PEARL-Prox. We run Algorithm 4 with the fixed step size $\gamma_{i,t}^p \equiv \gamma = 10^{-3}$ and compare PEARL-SGD (recovered as the case $\lambda = 0$ from Algorithm 4) against PEARL-Prox with $\lambda = 500$ for varying values of $\tau_i^p \equiv \tau \in \{1, 5, 20\}$ (Fig. 4(*a*)). We observe that PEARL-SGD only converges when $\tau = 1$ and diverges away to infinity with larger τ, more quickly as τ increases. This indicates that the performance of PEARL-SGD is sensitive to τ (excessive local updates are not allowed), i.e., the player drift occurs, as predicted by the prior work [25] which suggested that γ has to scale inversely proportionally with τ. On the other hand, for PEARL-Prox, the proximal computation becomes more accurate as τ increases from 1 to 5, rather accelerating convergence. The convergence pattern does not change significantly as τ increases from 5 to 20 (or even larger) because the SGD subroutine already converges with $\tau = 5$ due to the large strong convexity parameter of the regularized objective ($\lambda = 500$), making it easy to optimize.



(*a*) Effect of increasing τ (*b*) Effect of changing λ

Figure 4: Plots of $\frac{\|\mathbf{x}^p-\mathbf{x}^\star\|^2}{\|\mathbf{x}^0-\mathbf{x}^\star\|^2}$ for Algorithm 4 (PEARL-Prox with SGD) using fixed $\tau_i^p \equiv \tau$ and $\gamma_{i,t}^p \equiv \gamma$ for different values of λ. **(Left)** We fix $\gamma = 10^{-3}$ and test the effect of increasing τ with $\tau \in \{1, 5, 20\}$, for $\lambda = 0$ (PEARL-SGD) and $\lambda = 500$. **(Right)** We fix $\tau = 20$, $\gamma = 10^{-3}$ and test the effect of changing λ with $\lambda \in \{0, 200, 500, 1000, 2000\}$.

λ as a hyperparameter for convergence. Here, we fix $\tau = 100$ and $\gamma = 10^{-3}$, and run Algorithm 4 with different $\lambda \in \{0, 200, 500, 1000, 2000\}$ (Fig. 4(*b*)), demonstrating the usage of λ as a hyperparameter capable of ensuring convergence in the setups where PEARL-SGD does not converge. PEARL-SGD ($\lambda = 0$) rapidly diverges because τ is too large for the given choice of γ. Divergence slows down as λ increases, and at the sweet spot $\lambda = 500$ we observe the fastest convergence. However, increasing beyond $\lambda = 1000$ is suboptimal because then the curvature of the regularized objective grows beyond γ^{-1} and the SGD subroutine may fail to converge (so one should keep $\lambda = \mathcal{O}(\gamma^{-1})$).