Federated Online Prediction from Experts with Differential Privacy: Separations and Regret Speed-ups

Fengyu Gao, Ruiquan Huang, Jing Yang

Abstract

We study the problems of differentially private federated online prediction from experts against both *stochastic adversaries* and *oblivious adversaries*. We aim to minimize the average regret on m clients working in parallel over time horizon T with explicit differential privacy (DP) guarantees. With stochastic adversaries, we propose an algorithm that achieves $\sqrt{m}$-fold speed-up of the per-client regret compared to the single-player counterparts under both pure DP and approximate DP constraints, while maintaining logarithmic communication costs. With oblivious adversaries, we establish non-trivial lower bounds indicating that *collaboration among clients does not lead to regret speed-up with general oblivious adversaries*. However, when there exists a low-loss expert, we show that a new algorithm can achieve an m-fold regret speed-up under both pure DP and approximate DP constraints over the single-player counterparts. Our lower bound indicates that Fed-SVT is nearly optimal up to logarithmic factors. To the best of our knowledge, this is the first work examining the online prediction from experts problem with DP guarantees in the federated setting.

1 Introduction

Federated Learning (FL) (McMahan et al., 2017) is a distributed machine learning framework, where numerous clients collaboratively train a model by exchanging model update through a server. Owing to its advantage in protecting the privacy of local data and reducing communication overheads, FL is gaining increased attention in the research community, particularly in the online learning framework (Mitra et al., 2021; Park et al., 2022; Kwon et al., 2023; Gauthier et al., 2023). Noticeable advancements include various algorithms in federated multi-armed bandits (Shi et al., 2021; Huang et al., 2021; Li and Wang, 2022; Yi and Vojnovic, 2022, 2023), federated online convex optimization (Patel et al., 2023; Kwon et al., 2023; Gauthier et al., 2023), etc.

Meanwhile, differential privacy (DP) has been integrated into online learning, pioneered by Dwork et al. (2010). In the single-client setting, Asi et al. (2022) studied different types of adversaries, developing some of the best existing algorithms and establishing lower bounds. Within the federated framework, although differentially private algorithms have been proposed for stochastic bandits (Li et al., 2020; Zhu et al., 2021; Dubey and Pentland, 2022, 2020; Li et al., 2022; Zhou and Chowdhury, 2023; Huang et al., 2023), to the best of our knowledge, federated online learning algorithms in the adversarial setting with explicit DP considerations remain largely unexplored.

In this work, we focus on federated online prediction from experts (OPE) with rigorous differential privacy (DP) guarantees. OPE (Arora et al., 2012) is a classical online learning problem under which, a player chooses one out of a set of experts at each time slot and an adversary chooses a loss function. The player incurs a loss based on its choice and observes the loss function. With all previous observations, the player needs to decide which expert to select each time to minimize the cumulative expected loss. We consider two types of adversaries in the context of OPE. The first type, *stochastic adversary*, chooses a

School of EECS, The Pennsylvania State University, University Park, PA, USA. Correspondence to: Jing Yang <yangjing@psu.edu>.

distribution over loss functions and samples a loss function independently and identically distributed (IID) from this distribution at each time step. The second type, *oblivious adversary*, chooses a sequence of loss functions in advance. We aim to answer the following question: *Can we design differentially private federated OPE algorithms to achieve regret speed-up against both stochastic and oblivious adversaries?*

2 Results, Techniques, and Discussion

2.1 Differentially private federated online prediction from experts (OPE)

Federated OPE consists of a central server, m clients and an interactive T-round game between an adversary and an algorithm. At time step t, each client $i \in [m]$ first selects an expert $x_{i,t} \in [d]$, and then, the adversary releases a loss function $l_{i,t}$. *Stochastic adversaries* choose a distribution over loss functions and sample a sequence of loss functions $l_{1,1}, \ldots, l_{m,T}$ in an IID fashion from this distribution, while *oblivious adversaries* choose a sequence of loss functions $l_{1,1}, \ldots, l_{m,T}$ at the beginning of the game.

For federated OPE, the utility of primary interest is the expected cumulative regret among all clients defined as:

$$\operatorname{Reg}(T,m) = \frac{1}{m}\left[\sum_{i=1}^{m}\sum_{t=1}^{T} l_{i,t}(x_{i,t}) - \min_{x^\star \in [d]} \sum_{i=1}^{m}\sum_{t=1}^{T} l_{i,t}(x^\star)\right].$$

Differential privacy. We define differential privacy in the online setting following (Dwork et al., 2010). If an adversary chooses a loss sequence $\mathcal{S} = \left(l_{1,1}, \ldots, l_{m,T}\right)$, we denote $\mathcal{A}(\mathcal{S}) = (x_{1,1}, \ldots, x_{m,T})$ the output of the interaction between the federated online algorithm $\mathcal{A}$ and the adversary. We say $\mathcal{S} = \left(l_{1,1}, \ldots, l_{m,T}\right)$ and $\mathcal{S}' = \left(l'_{1,1}, \ldots, l'_{m,T}\right)$ are neighboring datasets if $\mathcal{S}$ and $\mathcal{S}'$ differ in a one element.

Definition 2.1 ((ε, δ)-DP)**.** A randomized federated algorithm $\mathcal{A}$ is (ε, δ)-differentially private against an adversary if, for all neighboring datasets $\mathcal{S}$ and $\mathcal{S}'$ and for all events $\mathcal{O}$ in the output space of $\mathcal{A}$, we have

$$\mathbb{P}[\mathcal{A}(\mathcal{S}) \in \mathcal{O}] \leq e^{\varepsilon}\mathbb{P}\left[\mathcal{A}\left(\mathcal{S}'\right) \in \mathcal{O}\right] + \delta.$$

Communication model. Our setup involves a central server facilitating periodic communication with zero latency with all clients. Specifically, the clients can send “local model updates” to the central server, which then aggregates and broadcasts the updated “global model” to the clients. We assume full synchronization between clients and the server (McMahan et al., 2017).

2.2 Speed-up for stochastic adversaries.

For stochastic adversaries, we develop a communication-efficient algorithm Fed-DP-OPE-Stoch with DP guarantees. The algorithm features the following elements in its design: 1) *Local loss function gradient estimation for global expert determination.* To reduce communication cost, we propose to estimate the gradient of each client's previous loss functions locally, and only communicate these estimates to the server instead of all previous loss functions. 2) *Local privatization process.* To satisfy the DP constraint, we add noise to the local gradient estimate on the client side, which essentially builds a local differentially private algorithm. The performance of Fed-DP-OPE-Stoch is summarized in the following theorem.

Theorem 2.2. *Assume that the loss function $l_{i,t}(\cdot)$ is convex, α-Lipschitz, β-smooth w.r.t. $\|\cdot\|_1$. Fed-DP-OPE-Stoch (i) satisfies ε-DP and (ii) achieves the per-client regret of*

$$\mathit{Reg}(T,m) = O\left((\alpha+\beta)\log T\sqrt{\frac{T\log d}{m}} + \frac{\sqrt{\alpha\beta}\sqrt{T}\log T\log d}{m^{\frac{1}{4}}\sqrt{\varepsilon}}\right),$$

with (iii) a communication cost of $O\left(m^{5/4}d\sqrt{\frac{T\varepsilon\beta}{\alpha\log d}}\log T\right)$.

Notably, the regret for the single-player counterpart Asi et al. (2022) scales in $O\left(\log T\sqrt{T\log d} + \frac{\log T \log d}{\varepsilon}\right)$. This result is obtained under the small-β regime, i.e. $\beta = O(\frac{1}{T\varepsilon})$. Let $\beta = O(\frac{1}{T\varepsilon})$, Theorem 2.2 reduces to the following corollary, which states that Fed-DP-OPE-Stoch achieves $m^{1/4}$-fold speed-ups compared with the single-client setting. The detailed implementation of Fed-DP-OPE-Stoch can be found in Appendix A.

Corollary 2.3. *If $\beta = O(\frac{1}{T\varepsilon})$, then, Fed-DP-OPE-Stoch (i) satisfies ε-DP and (ii) achieves the per-client regret of*

$$Reg(T, m) = O\left(\alpha \log T\sqrt{\frac{T\log d}{m}} + \frac{\sqrt{\alpha}\log T\log d}{m^{\frac{1}{4}}\varepsilon}\right),$$

with (iii) a communication cost of $O\left(m^{5/4} d \log T\right)$.

Finally, we remark that Fed-DP-OPE-Stoch can be slightly modified into a central differentially private algorithm and achieves $\sqrt{m}$-fold speed-ups. Specifically, we change the local privatization process to a global privatization process and add a Laplacian noise on the server side. This mechanism results in less noise added and thus better utility performance.

2.3 Lower bounds for oblivious adversaries.

We establish new lower bounds for federated OPE with oblivious adversaries, applicable to both non-private and private scenarios.

Theorem 2.4. *For any federated OPE algorithm against oblivious adversaries, the* **per-client** *regret is lower bounded by $\Omega(\sqrt{T\log d})$. Let $\varepsilon \in (0, 1]$ and $\delta = o(1/T)$, for any (ε, δ)-DP federated OPE algorithm, the* **per-client** *regret is lower bounded by $\Omega\left(\min\left(\frac{\log d}{\varepsilon}, T\right)\right)$.*

Theorem 2.4 states that with oblivious adversaries, the per-client regret under any federated OPE is fundamentally independent of the number of clients m. In other words, collaboration among clients does **not** lead to regret speed-up in this context, which in sharp contrast to the stochastic adversariy framework. Theorem 2.4 also emphasizes the influence of the DP guarantees. Our lower bounds represent the first non-trivial impossibility results for the federated OPE problem to the best of our knowledge.

We provide some insights to the reason why oblivious adversaries make FL lose its power in the following. FL can potentially speed up the learning process by collecting more data at the same time to gain better information to future predictions. In the stochastic setting, the advantage of collaboration lies in the ability to collect more observations from the same distribution, which leads to variance reduction. However, when facing oblivious adversaries, the problem changes fundamentally. Oblivious adversaries can select loss functions arbitrarily, meaning that having more data does not necessarily help with predicting their future selections.

2.4 Speed-up for oblivious adversaries under realizability assumption.

While Section 2.3 suggests that federated learning in oblivious adversaries fails to enjoy benefits in the worst case, we show that *realizability assumption* entirely flips the result. First, we introduce the definition of realizability.

Definition 2.5 (Realizability)**.** A federated OPE problem is *realizable* if there exists a feasible solution $x^\star \in [d]$ such that $\sum_{t=1}^{T} l_{i,t}(x^\star) = 0$, $\forall i \in [m]$. If the best expert achieves small loss $L^\star \ll T$, i.e., there exists $x^\star \in [d]$ such that $\sum_{t=1}^{T} l_{i,t}(x^\star) \leq L^\star$, $\forall i \in [m]$, the problem is near-realizable.

Intuitively, collaboration is notably advantageous in this context, as all clients share the same goal of reaching the zero-loss solution $x^\star$. As more clients participate, the shared knowledge pool expands, making the identification of the optimal solution more efficient. Following this intuition, we propose Fed-SVT, a new federated algorithm with the following theoretical guarantee.

Theorem 2.6 (Upper bound with realizability)**.** *Let $l_{i,t} \in [0,1]^d$ be chosen by an oblivious adversary under near-realizability assumption. Then, Fed-SVT is ε-DP, the communication cost scales in $O\left(mdT/N\right)$, and with probability at least $1-\rho$, the pre-client regret is*

$$\mathit{Reg}(T,m) = O\left(\frac{\log^2(d) + \log\left(\frac{T^2}{N^2\rho}\right)\log\left(\frac{d}{\rho}\right)}{m\varepsilon} + (N+L^\star)\log\left(\frac{d}{\rho}\right)\right).$$

Moreover, by changing the parameter setting, Fed-SVT is (ε,δ)-DP, the communication cost scales in $O\left(mdT/N\right)$, and with probability at least $1-\rho$, the pre-client regret is

$$\mathit{Reg}(T,m) = O\left(\frac{\log^{\frac{3}{2}}(d)\sqrt{\log(\frac{1}{\delta})} + \log\left(\frac{T^2}{N^2\rho}\right)\log\left(\frac{d}{\rho}\right)}{m\varepsilon} + (N+L^\star)\log\left(\frac{d}{\rho}\right)\right)$$

Compared to the best upper bound $O\left(\frac{\log^2 d + \log T \log d}{\varepsilon}\right)$ and $O\left(\frac{\log T \log d + \log^{3/2} d\sqrt{\log(1/\delta)}}{\varepsilon}\right)$ for the single-player scenario (Asi et al., 2023), our results indicate an m-fold regret speed-up when $N + L^\star = O\left(\frac{\log T}{m\varepsilon}\right)$. Note that even if $N + L^\star = \Omega\left(\frac{\log T}{m\varepsilon}\right)$, the coefficient $N + L^*$ is independent with the privacy budget ε. Therefore, federated learning in large m regime still presents significant benefit over single-client setting. The detailed implementaion of Fed-SVT can be found in Appendix B. We further develop a lower bound under the realizable oblivious adversary framework, which is presented as follows.

Theorem 2.7 (Lower bound with realizability)**.** *Let $\varepsilon \le 1$ and $\delta \le \varepsilon/d$. For any (ε,δ)-DP federated OPE algorithm against oblivious adversaries in the realizable setting, the per-client regret is lower bounded by $\Omega\left(\frac{\log(d)}{m\varepsilon}\right)$.*

Note that our lower bound (Theorem 2.7) scales in $\Omega\left(\frac{\log d}{m\varepsilon}\right)$, which matches with the upper bound in terms of m and ϵ, and is nearly optimal up to logarithmic factors.

Finally, a summary of our main results and how they compare with the state of the art is shown in Table 1.

Table 1: Comparisons for Online Prediction from Experts under DP Constraints

Adversaries	Algorithm (Reference)	Model	DP	Regret	Communication cost
Stochastic	Limited Updates (Asi et al., 2022)	SING	ε-DP, (ε,δ)-DP	$O\left(\sqrt{T\log d} + \frac{\log d \log T}{\varepsilon}\right)^*$	-
	Fed-DP-OPE-Stoch (**Theorem 2.2**)	FED	ε-DP	$O\left(\sqrt{\frac{T\log d}{m}} + \frac{\log d \log T}{\sqrt{m}\varepsilon}\right)^*$	$O(md\log T)$
Oblivious (Realizable)	Sparse-Vector (Asi et al., 2023)	SING	ε-DP	$O\left(\frac{\log T\log d + \log^2 d}{\varepsilon}\right)$	-
	Fed-SVT (**Theorem 2.6**)	FED	ε-DP	$O\left(\frac{\log T\log d + \log^2 d}{m\varepsilon}\right)$	$O\left(m^2dT\varepsilon\right)$
	Sparse-Vector (Asi et al., 2023)	SING	(ε,δ)-DP	$O\left(\frac{\log T\log d + \log^{3/2} d}{\varepsilon}\right)$	-
	Fed-SVT (**Theorem 2.6**)	FED	(ε,δ)-DP	$O\left(\frac{\log T\log d + \log^{3/2} d}{m\varepsilon}\right)$	$O\left(m^2dT\varepsilon\right)$
	Lower bound (**Theorem 2.7**)	FED	ε-DP, (ε,δ)-DP	$\Omega\left(\frac{\log d}{m\varepsilon}\right)$	-
Oblivious	Private SD (Asi et al., 2022)	SING	ε-DP	$O\left(\frac{\sqrt{T}\log d}{\varepsilon}\right)$	-
	Private SD (Asi et al., 2022)	SING	(ε,δ)-DP	$O\left(\sqrt{T\log d} + \frac{T^{1/3}\log d}{\varepsilon}\right)$	-
	Lower bound (**Theorem 2.4**)	FED	ε-DP, (ε,δ)-DP	$\Omega\left(\min\left(\frac{\log d}{\varepsilon}, T\right)\right)$	-

m: number of clients; T: time horizon; d: number of experts; ε, δ: DP parameters; SING and FED stand for single-client and federated settings, respectively; $*$: the results are contingent upon the smoothness constraints of the loss functions. Detailed outcomes under varying constraints are provided in Theorem 2.2.

References

Arora, S., Hazan, E., and Kale, S. (2012). The multiplicative weights update method: a meta-algorithm and applications. *Theory of computing*, 8(1):121–164.

Asi, H., Feldman, V., Koren, T., and Talwar, K. (2021). Private stochastic convex optimization: Optimal rates in l1 geometry. In *International Conference on Machine Learning*, pages 393–403. PMLR.

Asi, H., Feldman, V., Koren, T., and Talwar, K. (2022). Private online prediction from experts: Separations and faster rates. *arXiv preprint arXiv:2210.13537*.

Asi, H., Feldman, V., Koren, T., and Talwar, K. (2023). Near-optimal algorithms for private online optimization in the realizable regime. *arXiv preprint arXiv:2302.14154*.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge university press.

Dubey, A. and Pentland, A. (2020). Differentially-private federated linear bandits. *Advances in Neural Information Processing Systems*, 33:6003–6014.

Dubey, A. and Pentland, A. (2022). Private and byzantine-proof cooperative decision-making. *arXiv preprint arXiv:2205.14174*.

Dwork, C., Naor, M., Pitassi, T., and Rothblum, G. N. (2010). Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 715–724.

Dwork, C., Roth, A., et al. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407.

Gauthier, F., Gogineni, V. C., Werner, S., Huang, Y.-F., and Kuh, A. (2023). Asynchronous online federated learning with reduced communication requirements. *arXiv preprint arXiv:2303.15226*.

Huang, R., Wu, W., Yang, J., and Shen, C. (2021). Federated linear contextual bandits. *Advances in neural information processing systems*, 34:27057–27068.

Huang, R., Zhang, H., Melis, L., Shen, M., Hejazinia, M., and Yang, J. (2023). Federated linear contextual bandits with user-level differential privacy. In *International Conference on Machine Learning*, pages 14060–14095. PMLR.

Jain, P., Raskhodnikova, S., Sivakumar, S., and Smith, A. (2023). The price of differential privacy under continual observation. In *International Conference on Machine Learning*, pages 14654–14678. PMLR.

Kwon, D., Park, J., and Hong, S. (2023). Tighter regret analysis and optimization of online federated learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Li, C. and Wang, H. (2022). Asynchronous upper confidence bound algorithms for federated linear bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 6529–6553. PMLR.

Li, F., Zhou, X., and Ji, B. (2022). Differentially private linear bandits with partial distributed feedback. In *2022 20th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt)*, pages 41–48. IEEE.

Li, T., Song, L., and Fragouli, C. (2020). Federated recommendation system via differential privacy. In *2020 IEEE international symposium on information theory (ISIT)*, pages 2592–2597. IEEE.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

Mitra, A., Hassani, H., and Pappas, G. J. (2021). Online federated learning. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 4083–4090. IEEE.

Park, J., Kwon, D., et al. (2022). Ofedqit: Communication-efficient online federated learning via quantization and intermittent transmission. *arXiv preprint arXiv:2205.06491*.

Patel, K. K., Wang, L., Saha, A., and Srebro, N. (2023). Federated online and bandit convex optimization.

Shi, C., Shen, C., and Yang, J. (2021). Federated multi-armed bandits with personalization. In *International conference on artificial intelligence and statistics*, pages 2917–2925. PMLR.

Srebro, N., Sridharan, K., and Tewari, A. (2010). Smoothness, low noise and fast rates. *Advances in neural information processing systems*, 23.

Yi, J. and Vojnovic, M. (2022). On regret-optimal cooperative nonstochastic multi-armed bandits. *arXiv preprint arXiv:2211.17154*.

Yi, J. and Vojnovic, M. (2023). Doubly adversarial federated bandits. In *International Conference on Machine Learning*, pages 39951–39967. PMLR.

Zhou, X. and Chowdhury, S. R. (2023). On differentially private federated linear contextual bandits. *arXiv preprint arXiv:2302.13945*.

Zhu, Z., Zhu, J., Liu, J., and Liu, Y. (2021). Federated bandit: A gossiping approach. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 5(1):1–29.

Contents

A Federated OPE with stochastic adversaries

In this section, we present the detail algorithm design of Fed-DP-OPE with stochastic adversaries that achieves regret speed-up compared to the single-player setting under DP constraints with low communication cost. We consider the loss functions $l_{1,1}(\cdot), \dots, l_{m,T}(\cdot)$ to be convex, α-Lipschitz and β-smooth w.r.t. $\|\cdot\|_1$ in this section.

A.1 Intuition behind our algorithm design

To gain a better understanding of our algorithm design, we first elaborate the difficulties encountered when extending prevalent OPE models to the FL setting under DP constraints. It is worth noting that all current OPE models with stochastic adversaries rely on gradient-based optimization methods. The central task in designing OPE models with stochastic adversaries lies in leveraging past loss functions to guide the generation of expert predictions. Specifically, we focus on the prominent challenges associated with the widely adopted Frank-Wolfe-based methods (Asi et al., 2022). This algorithm iteratively moves the expert selection x_t towards a point that minimizes the gradient estimate derived from the past loss functions $l_1, \dots, l_{t-1}$ over the decision space $\mathcal{X}$, where $\mathcal{X} = \Delta_d = \left\{x \in \mathbb{R}^d : x_i \geq 0, \sum_{i=1}^d x_i = 1\right\}$, and each $x \in \mathcal{X}$ symbolizes a probability distribution over d experts. With DP constraints, a tree-based method is used for private aggregation of the gradients of loss functions (Asi et al., 2021).

In the federated setting, it is infeasible for the central server to have full access to past loss functions due to the high communication cost. To overcome this, we employ a design of *local loss function gradient estimation for global expert determination.* Our solution involves locally estimating the gradient of each client's loss functions, and then communicating these estimates to the server, which globally generates a new prediction. This strategy bypasses the need for full access to all loss functions, reducing the communication overhead while maintaining efficient expert selection.

To further enhance the privacy of the federated system, we implement an *enhanced two-step privatization process.* The first step involves client-server privacy-enhanced communication. When communication is triggered, clients send "local estimates" of the gradients of their loss functions to the server. These local estimates include strategically added noise, adhering to DP principles. The second step is private global prediction at the server, where the server aggregates the noisy local estimates and updates "global prediction" for the new expert selection privately. The two-step privatization process is crucial as it extends beyond privatizing the selection of experts; it ensures the privacy of all information exchanged between the central server and clients within the FL framework.

A.2 Algorithm design

To address the aforementioned challenges, we propose the Fed-DP-OPE-Stoch algorithm. The Fed-DP-OPE-Stoch algorithm works in phases. In total, it has P phases, and each phase $p \in [P]$ contains 2^{p-1} time indices. Fed-DP-OPE-Stoch contains a client-side subroutine (Algorithm 1) and a server-side subroutine (Algorithm 2). The framework of our algorithm is outlined as follows.

At the initialization phase, the server selects an arbitrary point $z \in \mathcal{X}$ and broadcasts to all clients. Subsequently, each client initializes its expert selection $x_{i,1} = z$, pays cost $l_{i,1}(x_{i,1})$ and observes the loss function.

Starting at phase $p = 2$, each client uses loss functions from the last phase to update its local loss function gradient estimation, and then coordinates with the server to update its expert selection. After that, it sticks with its decision throughout the current phase and observes the loss functions. We elaborate this procedure as follows.

Private Local Loss Function Gradient Estimation: At the beginning of phase p, each client privately estimates the gradient using local loss functions from the last phase $\mathcal{B}_{i,p} = \{l_{i,2^{p-2}}, \dots, l_{i,2^{p-1}-1}\}$. We employ the tree mechanism for the private aggregation at each client, as in the DP-FW algorithm from Asi et al. (2021). Roughly speaking, DP-FW involves constructing binary trees and allocating sets of loss functions to each vertex. The gradient at each vertex is then estimated using the loss functions of that vertex and the gradients along the path to the root. Specifically, we run a DP-FW subroutine at each client i, with sample set $\mathcal{B}_{i,p}$, parameter T_1 and batch size b. In DP-FW, each vertex s in the binary tree j corresponds to a gradient estimate $v_{i,j,s}$. DP-FW iteratively updates gradient estimates by visiting the vertices of binary trees. The details of DP-FW can be found in Algorithm 3.

Intuitively, in the DP-FW subroutine, reaching a leaf vertex marks the completion of gradient refinement for a specific tree path. At these critical points, we initiate communication between the central server and individual clients. Specifically, as the DP-FW subroutine reaches a leaf vertex s of tree j, each client sends the server a set of noisy inner products for each decision set vertex c_n with their gradient estimate $v_{i,j,s}$. In other words, each client communicates with the server by sending $\{\langle c_n, v_{i,j,s}\rangle + \xi_{i,n}\}_{n\in[d]}$, where $c_1, \ldots, c_d$ represents d vertices of decision set $\mathcal{X} = \Delta_d$, $\xi_{i,n} \sim \mathrm{Lap}(\lambda_{i,j,s})$, and $\lambda_{i,j,s} = \frac{4\alpha 2^j}{b\varepsilon}$.

Private Global Expert Prediction: After receiving $\{\langle c_n, v_{i,j,s}\rangle + \xi_{i,n}\}_{n\in[d]}$ from all clients, the central server privately predicts a new expert:

$$\bar{w}_{j,s} = \underset{c_n : 1\le n\le d}{\arg\min} \left[\frac{1}{m} \sum_{i=1}^{m} \left(\langle c_n, v_{i,j,s}\rangle + \xi_{i,n} \right) \right]. \tag{1}$$

Subsequently, the server broadcasts the “global prediction" $\bar{w}_{j,s}$ to all clients.

Local Expert Selection Updating: Denote the index of the leaf s of tree j as k. Then, upon receiving the global expert selection $\bar{w}_{j,s}$, each client i updates its expert prediction for leaf $k+1$, denoted as $x_{i,p,k+1} \in \mathcal{X}$, as follows:

$$x_{i,p,k+1} = (1-\eta_{i,p,k}) x_{i,p,k} + \eta_{i,p,k} \bar{w}_{j,s}, \tag{2}$$

where $\eta_{i,p,k} = \frac{2}{k+1}$.

After updating all leaf vertices of the trees, the client obtains $x_{i,p,K}$, the final state of the expert prediction in phase p. Then, each client i sticks with expert selection $x_{i,p,K}$ throughout phase p and collects loss functions $l_{i,2^{p-1}}, \ldots, l_{i,2^p-1}$.

The key difference between our Fed-DP-OPE-Stoch algorithm and non-federated algorithms (Asi et al., 2022) lies in our innovative approach to centralized coordination and communication efficiency. Unlike the direct application of DP-FW in each phase in Asi et al. (2022), our algorithm employs a more nuanced strategy in the federated setting. Our innovative components of *local loss function gradient estimation*, *global expert prediction* and *local expert selection updating* enable a significant speed-up in the learning process. Additionally, our strategic communication protocol, where clients communicate with the server only when DP-FW subroutine reaches leaf vertices, significantly reduces communication costs. Moreover, the integration of DP at both local and global levels in our algorithm distinguishes it from non-federated approaches. These features underscore our contribution to the FL setting, improving collaborative decision-making while maintaining privacy and reducing communication overhead.

A.3 DP-FW

In Fed-DP-OPE-SToch, we run a DP-FW subroutine (Asi et al., 2021) at each client i in each phase p. DP-FW maintains T_1 binary trees indexed by $1 \le j \le T_1$, each with depth j, where T_1 is a predetermined parameter. An example of the tree structure is shown below. We introduce the notation $s \in \{0,1\}^{\le j}$ to denote vertices within binary tree j. $\emptyset$ signifies the tree's root. For any $s, s' \in \{0,1\}^{\le j}$, if $s = s'0$, then s denotes the left child of s'. Conversely, when $s = s'1$, s is attributed as the right child of s'. For each client i, each vertex s in the binary tree j corresponds to a parameter $x_{i,j,s}$ and a gradient estimate $v_{i,j,s}$. Each client iteratively updates parameters and gradient estimates by visiting the vertices of a binary tree according to the Depth-First Search (DFS) order: as it visits a left child vertex s, the algorithm maintains the parameter $x_{i,j,s}$ and the gradient estimate $v_{i,j,s}$ identical to those of its parent vertex s',

Algorithm 1 Fed-DP-OPE-Stoch: Client i

```
1: Input: Phases $P$, trees $T_1$, decision set $\mathcal{X} = \Delta_d$ with vertices $\{c_1, \ldots, c_d\}$, batch size $b$.
2: Initialize: Set $x_{i,1} = z \in \mathcal{X}$ and pay cost $l_{i,1}(x_{i,1})$.
3: for $p = 2$ to $P$ do
4:   Set $\mathcal{B}_{i,p} = \{l_{i,2^{p-2}}, \ldots, l_{i,2^{p-1}-1}\}$
5:   Set $k = 1$ and $x_{i,p,1} = x_{i,p-1,K}$
6:   $\{v_{i,j,s}\}_{j\in[T_1], s\in\{0,1\}^{\le j}} = \text{DP-FW}\left(\mathcal{B}_{i,p}, T_1, b\right)$
7:   for all leaf vertices $s$ reached in DP-FW do
8:     Communicate to server: $\{\langle c_n, v_{i,j,s}\rangle + \xi_{i,n}\}_{n\in[d]}$, where $\xi_{i,n} \sim \text{Lap}(\lambda_{i,j,s})$
9:     Receive from server: $\bar{w}_{j,s}$
10:    Update $x_{i,p,k}$ according to Equation (2)
11:    Update $k = k + 1$
12:  end for
13:  Final iterate outputs $x_{i,p,K}$
14:  for $t = 2^{p-1}$ to $2^p - 1$ do
15:    Receive loss $l_{i,t} : \mathcal{X} \to \mathbb{R}$ and pay cost $l_{i,t}(x_{i,p,K})$
16:  end for
17: end for
```

Algorithm 2 Fed-DP-OPE-Stoch: Central server

```
1: Input: Phases $P$, number of clients $m$, decision set $\mathcal{X} = \Delta_d$ with vertices $\{c_1, \ldots, c_d\}$.
2: Initialize: Pick any $z \in \mathcal{X}$ and broadcast to clients.
3: for $p = 2$ to $P$ do
4:   Receive from clients: $\{\langle c_n, v_{i,j,s}\rangle + \xi_{i,n}\}_{n\in[d]}$
5:   Update $\bar{w}_{j,s}$ according to Equation (1)
6:   Communicate to clients: $\bar{w}_{j,s}$
7: end for
```

i.e. $v_{i,j,s} = v_{i,j,s'}$ and $x_{i,j,s} = x_{i,j,s'}$. As it proceeds to a right child vertex, we uniformly select $2^{-|s|}b$ loss functions from set $\mathcal{B}_{i,p} = \{l_{i,2^{p-2}}, \ldots, l_{i,2^{p-1}-1}\}$ to subset $\mathcal{B}_{i,j,s}$ without replacement, where b denoting the predetermined batch size and $|s|$ representing the depth of the vertex s. Then the algorithm improves the gradient estimate $v_{i,j,s}$ at the current vertex s of tree j using the estimate $v_{i,j,s'}$ at the parent vertex s', i.e.

$$v_{i,j,s} = v_{i,j,s'} + \nabla l(x_{i,j,s}; \mathcal{B}_{i,j,s}) - \nabla l(x_{i,j,s'}; \mathcal{B}_{i,j,s}). \tag{3}$$

where $\nabla l(\cdot; \mathcal{B}_{i,j,s}) = \frac{1}{|\mathcal{B}_{i,j,s}|}\sum_{l\in\mathcal{B}_{i,j,s}} \nabla l(\cdot)$, and $\mathcal{B}_{i,j,s}$ is the subset of loss functions at vertex s in the binary tree j for client i. The full algorithm is shown in Algorithm 3.

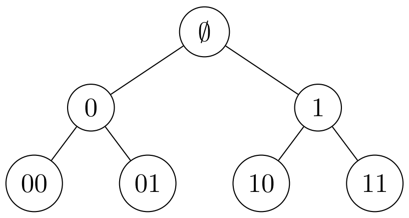

Figure 1: Binary tree with depth $j = 2$ in Algorithm 3.

B Federated OPE with oblivious adversaries: Realizable setting

In this section, we provide the details of Fed-SVT in the near-realizable and realizable settings.

Algorithm 3 DP-FW at client i (Asi et al., 2021)

```
1: Input: Sample set $\mathcal{B}$, number of trees $T_1$, batch size $b$.
2: for $j = 1$ to $T_1$ do
3:   Set $x_{i,j,\emptyset} = x_{i,j-1,L_{j-1}}$
4:   Uniformly select $b$ samples to $\mathcal{B}_{i,j,\emptyset}$
5:   $v_{i,j,\emptyset} = \nabla l_{i,t}(x_{i,j,\emptyset}; \mathcal{B}_{i,j,\emptyset})$
6:   for $s \in \mathrm{DFS}[j]$ do
7:     Let $s = s'a$ where $a \in \{0,1\}$
8:     if $a == 0$ then
9:       $v_{i,j,s} = v_{i,j,s'}$; $x_{i,j,s} = x_{i,j,s'}$
10:    else
11:      Uniformly select $2^{-|s|}b$ samples to $\mathcal{B}_{i,j,s}$
12:      Update $v_{i,j,s}$ according to Equation (3)
13:    end if
14:  end for
15: end for
16: Return $\{v_{i,j,s}\}_{j\in[T_1],s\in\{0,1\}^{\leq j}}$
```

B.1 Algorithm design

We start with the intuition behind the design of our algorithm. Note that all current OPE models in the realizable setting employ limited switching methods (Srebro et al., 2010). In the FL setting, a significant challenge is balancing between communication efficiency and precise expert selection. Limited switching methods, while enhancing privacy, may cause delays in adapting to adversaries' strategies, potentially leading to prolonged selection of sub-optimal experts and thus increased regret. To address this, we focus on optimizing communication intervals, ensuring they are adequately frequent for timely response to adversaries, yet sufficiently infrequent to keep communication costs low.

We develop an algorithm termed Fed-SVT, inspired by the foundational principles of the sparse-vector-zero-loss algorithm (Asi et al., 2023) for the single-player setting. Our algorithm operates as follows:

Periodic communication: Our approach in the federated setting contrasts with non-federated algorithms (Asi et al., 2023) by adopting a structured communication protocol. An intuitive attempt is to implement event-trigger communication methods for adaptive expert switching, where clients signal the central server to switch experts. Since adaptive expert switching schedules can risk privacy leakage if they rely on local non-private data, local privatization of switching is essential. However, unlike fixed schedules where IID noise aggregation from clients reduces variance, event-trigger methods only track the maximum loss among clients, not the total aggregated loss, missing out on this variance reduction benefit.

To overcome this, we adopt a fixed communication schedule in our federated setting, splitting the time horizon T into T/N phases, each with length N. In Fed-SVT, every client selects the same expert, i.e., $x_{i,t} = x_t$ at each time step t. Initially, each client starts with a randomly chosen expert x_1. At the beginning of each phase n, each client sends the accumulated loss of the last phase $\sum_{t'=(n-1)N}^{nN-1} l_{i,t'}(x)$ to the central server.

Global expert selection: The server, upon receiving $\sum_{t'=(n-1)N}^{nN-1} l_{i,t'}(x)$ from all clients, decides whether to continue with the current expert or switch to a new one. This decision is grounded in the sparse-vector-zero-loss algorithm (Asi et al., 2023), where the accumulated loss from all clients over a phase is treated as a single loss instance in the sparse-vector-zero-loss algorithm. Based on the server's expert decision, clients update their experts accordingly. The full algorithm is provided in Algorithm 4 (client-side subroutine) and Algorithm 5 (server-side subroutine).

Algorithm 4 Fed-SVT: Client i

```
1: **Input:** Number of Iterations $T$
2: **Initialize:** Set current expert $x_0 = \text{Unif}[d]$.
3: **for** $t = 1$ to $T$ **do**
4:   **if** $t == nN$ for some integer $n \geq 1$ **then**
5:     **Communicate to server:** $\sum_{t'=t-N}^{t-1} l_{i,t'}(x)$
6:     **Receive from server:** $x_t$
7:   **else**
8:     Set $x_t = x_{t-1}$
9:   **end if**
10:  Each client receives local loss $l_{i,t} : [d] \to [0,1]$ and pays cost $l_{i,t}(x_t)$
11: **end for**
```

Algorithm 5 Fed-SVT: Central server

```
1: **Input:** Number of Iterations $T$, number of clients $m$, optimal loss $L^\star$, switching budget $\kappa$, sampling
   parameter $\eta > 0$, threshold parameter $L$, failure probability $\rho$, privacy parameters $\varepsilon$
2: **Initialize:** Set $k = 0$, $\tau = 0$ and $\hat{L} = L + \text{Lap}\left(\frac{4}{\varepsilon}\right)$
3: **while** not reaching the time horizon $T$ **do**
4:   **if** $t == nN$ for some integer $n \geq 1$ **then**
5:     **Receive from clients:** $\sum_{t'=t-N}^{t-1} l_{i,t'}(x)$
6:     **if** $k < \kappa$ **then**
7:       Server defines a new query $q_t = \sum_{i=1}^m \sum_{t'=\tau}^{t-1} l_{i,t'}(x_{t'})$
8:       Let $\gamma_t = \text{Lap}\left(\frac{8}{\varepsilon}\right)$
9:       **if** $q_t + \gamma_t \leq \hat{L}$ **then**
10:        **Communicate to clients:** $x_t = x_{t-1}$
11:      **else**
12:        Sample $x_t$ with scores $s_t(x) = \max\left(\sum_{i=1}^m \sum_{t'=1}^{t-1} l_{i,t'}(x_{t'}), mL^\star\right)$ for $x \in [d]$: $\mathbb{P}(x_t = x) \propto$
           $e^{-\eta s_t(x)/2}$
13:        **Communicate to clients:** $x_t$
14:        Set $k = k + 1$, $\tau = t$ and $\hat{L} = L + \text{Lap}\left(\frac{4}{\varepsilon}\right)$
15:      **end if**
16:    **else**
17:      Server broadcasts $x_t = x_{t-1}$ to all the clients
18:    **end if**
19:  **end if**
20: **end while**
```

C Proof of lower bound for general oblivious adversaries

Theorem C.1 (Restatement of Theorem 2.4)**.** *For any federated OPE algorithm against oblivious adversaries, the per-client regret is lower bounded by $\Omega(\sqrt{T \log d})$. Let $\varepsilon \in (0,1]$ and $\delta = o(1/T)$, for any (ε, δ)-DP federated OPE algorithm, the per-client regret is lower bounded by $\Omega\left(\min\left(\frac{\log d}{\varepsilon}, T\right)\right)$.*

Proof. Consider the case when all clients receive the same loss function from the oblivious adversary at each time step, i.e. $l_{i,t} = l'_t$. Then we define the average policy among all clients $p'_t(k) = \frac{1}{m}\sum_{i=1}^m p_{i,t}(k), \forall k \in [d]$.

Now the regret is

$$\begin{aligned}\mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m}\sum_{t=1}^{T} l_{i,t}(x_{i,t})\right] - \frac{1}{m}\sum_{i=1}^{m}\sum_{t=1}^{T} l_{i,t}(x^\star) &= \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m}\sum_{t=1}^{T} l'_t(x_{i,t})\right] - \sum_{t=1}^{T} l'_t(x^\star)\\ &= \frac{1}{m}\sum_{i=1}^{m}\sum_{t=1}^{T}\sum_{k=1}^{d} p_{i,t}(k)\cdot l'_t(k) - \sum_{t=1}^{T} l'_t(x^\star)\\ &= \sum_{t=1}^{T}\sum_{k=1}^{d}\left(\frac{1}{m}\sum_{i=1}^{m} p_{i,t}(k)\right)\cdot l'_t(k) - \sum_{t=1}^{T} l'_t(x^\star)\\ &= \sum_{t=1}^{T}\sum_{k=1}^{d} p'_t(k)\cdot l'_t(k) - \sum_{t=1}^{T} l'_t(x^\star).\end{aligned}$$

Note that $p'_t(k)$ is defined by $p_{1,t}(k), \ldots, p_{m,t}(k)$, which in turn are determined by $l_{1,1}, \ldots, l_{m,t-1}$. According to our choice of $l_{i,t} = l'_t$, $p'_t(k)$ is determined by $l'_1, l'_2, \ldots, l'_{t-1}$. Therefore $p'_1, p'_2, \ldots, p'_t$ are generated by a legitimate algorithm for online learning with expert advice problems.

There exists a sequence of losses $l'_1, l'_2, \ldots, l'_t$ such that for any algorithm for online learning with expert advice problem, the expected regret satisfies (Cesa-Bianchi and Lugosi (2006), Theorem 3.7)

$$\sum_{t=1}^{T}\sum_{k=1}^{d} p'_t(k)\cdot l'_t(k) - \sum_{t=1}^{T} l'_t(x^\star) \geq \Omega(\sqrt{T\log d}).$$

Therefore, we have

$$\mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m}\sum_{t=1}^{T} l_{i,t}(x_{i,t})\right] - \frac{1}{m}\sum_{i=1}^{m}\sum_{t=1}^{T} l_{i,t}(x^\star) \geq \Omega(\sqrt{T\log d}).$$

From Lemma C.2, if $\varepsilon \in (0,1]$ and $\delta = o(1/T)$, then there exists a sequence of losses $l'_1, l'_2, \ldots, l'_t$ such that for any (ε,δ)-DP algorithm for online learning with expert advice problem against oblivious adversaries, the expected regret satisfies

$$\sum_{t=1}^{T}\sum_{k=1}^{d} p'_t(k)\cdot l'_t(k) - \sum_{t=1}^{T} l'_t(x^\star) \geq \Omega\left(\min\left(\frac{\log d}{\varepsilon}, T\right)\right).$$

Therefore, we have for any (ε,δ)-DP algorithm,

$$\mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m}\sum_{t=1}^{T} l_{i,t}(x_{i,t})\right] - \frac{1}{m}\sum_{i=1}^{m}\sum_{t=1}^{T} l_{i,t}(x^\star) \geq \Omega\left(\min\left(\frac{\log d}{\varepsilon}, T\right)\right).$$

□

Lemma C.2. *For any $\varepsilon \in (0,1], \delta = o(1/T), T, d \in \mathbb{N}$ such that $d > 1$, any (ε,δ)-DP algorithm $\mathcal{A}$ for DP-OPE against oblivious adversaries satisfies*

$$\sum_{t=1}^{T} l_t(x_t) - \min_{x^\star\in[d]}\sum_{t=1}^{T} l_t(x^\star) \geq \Omega\left(\min\left(\frac{\log d}{\varepsilon}, T\right)\right).$$

Proof. Let $n, d \in \mathbb{N}$. Define $\mathbf{y} \in \mathcal{Y}^n$ containing n records, where $\mathcal{Y} = \{0,1\}^d$. The function 1-Select$_d$: $\mathcal{Y}^n \to [d]$ corresponds to selecting a coordinate $b \in [d]$ in the batch model.

Then we define the regret of 1-Select$_d$. For a batched algorithm $\mathcal{M}$ with input dataset $\mathbf{y}$ and output

$b \in [d]$, define

$$\mathrm{Reg}_{\text{1-Select}_d}(\mathcal{M}(\mathbf{y})) = \frac{1}{n}\left[\sum_{i=1}^{n} y_i(b) - \min_{x^\star \in [d]}\left(\sum_{i=1}^{n} y_i(x^\star)\right)\right].$$

Let $\mathcal{A}$ be an (ε, δ)-DP algorithm $\left(\{0,1\}^d\right)^T \to ([d])^T$ for DP-OPE against oblivious adversaries with regret $\sum_{t=1}^T l_t(x_t) - \min_{x^\star \in [d]} \sum_{t=1}^T l_t(x^\star) \le \alpha$. We can use $\mathcal{A}$ to construct an (ε, δ)-DP algorithm $\mathcal{M}$ for 1-Select$_d$ in the batch model. The details of the algorithm appear in Algorithm 6.

Algorithm 6 Batch algorithm $\mathcal{M}$ for 1-Select (Jain et al. (2023), Algorithm 2 with $k = 1$)

```
1: Input: $\mathbf{y} = (y_1, \dots, y_n) \in \mathcal{Y}^n$, where $\mathcal{Y} = \{0,1\}^d$, and black-box access to a DP-OPE algorithm $\mathcal{A}$
   for oblivious adversaries.
2: Construct a stream $\mathbf{z} \leftarrow \mathbf{y}$ with $n$ records.
3: for $t = 1$ to $n$ do
4:     Send the record $z_t$ to $\mathcal{A}$ and get the corresponding output $x_t$.
5: end for
6: Output $b = x_n$.
```

Let $\varepsilon > 0, \alpha \in \mathbb{R}^+$, and $T, d, n \in \mathbb{N}$, where $T = n$. If a DP-OPE algorithm $\mathcal{A}\colon \left(\{0,1\}^d\right)^T \to ([d])^T$ for oblivious adversaries is (ε, δ)-DP and the regret is upper bounded by α, i.e. $\sum_{t=1}^T l_t(x_t) - \min_{x^\star \in [d]} \sum_{t=1}^T l_t(x^\star) \le \alpha$, then by Lemma C.3, we have the batch algorithm $\mathcal{M}$ for 1-Select$_d$ is (ε, δ)-DP and $\mathrm{Reg}_{\text{1-Select}_d}(\mathcal{M}) \le \frac{\alpha}{n}$.

If $\delta = o(1/T)$, then $n = \Omega\left(\frac{n \log d}{\varepsilon \alpha}\right)$ (Lemma C.4). We have $\alpha = \min\left(\Omega\left(\frac{\log d}{\varepsilon}\right), n\right) = \min\left(\Omega\left(\frac{\log d}{\varepsilon}\right), T\right)$. So $\alpha \ge \Omega\left(\min\left(\frac{\log d}{\varepsilon}, T\right)\right)$. Therefore, if an algorithm for DP-OPE against oblivious adversaries is (ε, δ)-differentially private and $\sum_{t=1}^T l_t(x_t) - \min_{x^\star \in [d]} \sum_{t=1}^T l_t(x^\star) \le \alpha$ holds, then $\alpha \ge \Omega\left(\min\left(\frac{\log d}{\varepsilon}, T\right)\right)$. This means that $\sum_{t=1}^T l_t(x_t) - \min_{x^\star \in [d]} \sum_{t=1}^T l_t(x^\star) \ge \Omega\left(\min\left(\frac{\log d}{\varepsilon}, T\right)\right)$. □

Lemma C.3. *Let $\mathcal{M}$ be the batch algorithm for 1-Select$_d$. For all $\varepsilon > 0, \delta \ge 0, \alpha \in \mathbb{R}^+$, and $T, d, n \in \mathbb{N}$, where $T = n$, if a DP-OPE algorithm $\mathcal{A}\colon \left(\{0,1\}^d\right)^T \to ([d])^T$ for oblivious adversaries is (ε, δ)-differentially private and the regret is upper bounded by α, i.e. $\sum_{t=1}^T l_t(x_t) - \min_{x^\star \in [kd]} \sum_{t=1}^T l_t(x^\star) \le \alpha$, then batch algorithm $\mathcal{M}$ for 1-Select$_d$ is (ε, δ)-differentially private and $\mathrm{Reg}_{\text{1-Select}_d}(\mathcal{M}) \le \frac{\alpha}{n}$.*

Proof. **DP guarantee:** Fix neighboring datasets $\mathbf{y}$ and $\mathbf{y}'$ that are inputs to algorithm $\mathcal{M}$. According to the algorithm design for 1-Select$_d$, we stream $\mathbf{y}$ and $\mathbf{y}'$ to a DP-OPE algorithm $\mathcal{A}$. Since $\mathcal{A}$ is (ε, δ)-DP, and $\mathcal{M}$ only post-processes the outputs received from $\mathcal{A}$, therefore $\mathcal{M}$ is (ε, δ)-DP.

Regret upper bound: Fix a dataset $\mathbf{y}$. Note that if $\alpha \ge n$, the accuracy guarantee for $\mathcal{M}$ is vacuous. Now assume $\alpha < n$. Let γ be the regret of $\mathcal{M}$, that is,

$$\begin{aligned}\gamma &= \mathrm{Reg}_{\text{1-Select}_d}(\mathcal{M}) \\ &= \frac{1}{n}\left[\sum_{i=1}^{n} y_i(b) - \min_{x^\star \in [d]}\left(\sum_{i=1}^{n} y_i(x^\star)\right)\right],\end{aligned}$$

where b is the output of $\mathcal{M}$.

The main observation in the regret analysis is that if α is small, so is γ. Specifically, the regret of $\mathcal{M}$ is at most $\frac{\alpha}{n}$. Therefore, $\gamma \le \frac{\alpha}{n}$. □

Lemma C.4 (Jain et al. (2023), Lemma 4.2)**.** *For all $d, n \in \mathbb{N}, \varepsilon \in (0,1], \delta = o(1/n), \gamma \in \left[0, \frac{1}{20}\right]$, and (ε, δ)-DP 1-Select$_d$ algorithms $\mathcal{M}\colon \left(\{0,1\}^d\right)^n \to [d]$ with $\mathrm{Reg}_{\text{1-Select}_d}(\mathcal{M}) \le \gamma$, we have $n = \Omega\left(\frac{\log d}{\varepsilon \gamma}\right)$.*

C.1 Proof of Theorem 2.2 (for pure DP)

Theorem C.5 (Restatement statement of Theorem 2.2)**.** *Assume that loss function $l_{i,t}(\cdot)$ is convex, α-Lipschitz, β-smooth w.r.t. $\|\cdot\|_1$. Setting $\lambda_{i,j,s} = \frac{4\alpha 2^j}{b\varepsilon}$, $\mu_{j,s} = \frac{4\alpha 2^j}{bm\varepsilon}$, $b = \frac{2^{p-1}}{(p-1)^2}$ and $T_1 = \frac{1}{2}\log\left(\frac{b\varepsilon\beta\sqrt{m}}{\alpha\log d}\right)$, Fed-DP-OPE-Stoch (i) satisfies ε-DP and (ii) achieves the per-client regret of*

$$O\left((\alpha+\beta)\log T\sqrt{\frac{T\log d}{m}} + \frac{\sqrt{\alpha\beta}\sqrt{T}\log T\log d}{m^{\frac{1}{4}}\sqrt{\varepsilon}}\right),$$

with (iii) a communication cost of $O\left(m^{\frac{5}{4}} d\sqrt{\frac{T\varepsilon\beta}{\alpha\log d}}\right)$.

We define population loss as $L_t(x) = \mathbb{E}[\frac{1}{m}\sum_{i=1}^m l_{i,t}(x_{i,t})]$. The per-client regret can be expressed as $\mathbb{E}\left[\sum_{t=1}^T L_t(x_{i,t}) - \min_{x^\star\in\mathcal{X}}\sum_{t=1}^T L_t(x^\star)\right]$. We use $v_{i,p,k}$, $w_{i,p,k}$, $x_{i,p,k}$ and $\eta_{i,p,k}$ to denote quantities corresponding to phase p, iteration k, and client i. Also we introduce some average quantities $\bar{v}_{p,k} = \frac{1}{m}\sum_{i=1}^m v_{i,p,k}$ and $\bar{w}_{p,k} = \frac{1}{m}\sum_{i=1}^m w_{i,p,k}$.

To prove the theorem, we start with Lemma C.6 that gives pure privacy guarantees.

Lemma C.6. *Assume that $2^{T_1} \le b$. Setting $\lambda_{i,j,s} = \frac{4\alpha 2^j}{b\varepsilon}$ and $\mu_{j,s} = \frac{4\alpha 2^j}{bm\varepsilon}$, Fed-DP-OPE-Stoch is ε-DP.*

Proof. Let $\mathcal{S}$ and $\mathcal{S}'$ be two neighboring datasets and assume that they differ in sample l_{i_1,t_1} or l'_{i_1,t_1}, where $2^{p_1-1} \le t_1 < 2^{p_1}$. We denote V as the set encompassing $\langle c_n, v_{i,p,k}\rangle + \xi_{i,n}$ where $\xi_{i,n} \sim \mathrm{Lap}(\lambda_{i,j,s})$ for $1 \le n \le d, 1 \le i \le m, 1 \le k \le K$, and $p \in [P]$. Also we denote W as the set containing $\bar{w}_{p,k}$ for $1 \le k \le K$ and $p \in [P]$. The algorithm is ε-DP if we have $(x_{1,1,K},\dots,x_{m,P,K}) \approx_{(\varepsilon,0)} (x'_{1,1,K},\dots,x'_{m,P,K})$, $V \approx_{(\varepsilon,0)} V'$, and $W \approx_{(\varepsilon,0)} W'$.

Let $\mathcal{B}_{i_1,j,s}$ be the set that contains l_{i_1,t_1} or l'_{i_1,t_1}. Recall that $|\mathcal{B}_{i_1,j,s}| = 2^{-|s|}b$. The key point is that this set is used in the calculation of $v_{i_1,p_1,k}$ for at most $2^{j-|s|}$ iterates, i.e. the leaves that are descendants of the vertex. Let k_0 and k_1 be the first and last iterate such that $\mathcal{B}_{i_1,j,s}$ is used for the calculation of $v_{i_1,p_1,k}$, hence $k_1 - k_0 + 1 \le 2^{j-|s|}$. Now we summarize our proof for privacy guarantees in the following 3 steps. For a sequence $a_i,\dots,a_j$, we use the shorthand $a_{i:j} = \{a_i,\dots,a_j\}$.

Step 1: $\mathbf{x_{1:m,p_1,k_0:k_1}} \approx_{(\varepsilon,0)} \mathbf{x'_{1:m,p_1,k_0:k_1}}$ and $\bar{\mathbf{w}}_{\mathbf{p_1,k_0:k_1}} \approx_{(\varepsilon,0)} \bar{\mathbf{w}}'_{\mathbf{p_1,k_0:k_1}}$ by basic composition, post-processing and report noisy max

We will show that $(x_{1,p_1,k_0},\dots,x_{m,p_1,k_1})$ and $(x'_{1,p_1,k_0},\dots,x'_{m,p_1,k_1})$ are ε-indistinguishable. Since $\mathcal{B}_{i_1,j,s}$ is used to calculate $v_{i_1,p_1,k}$ for at most $2^{j-|s|}$ iterates, i.e. $k_1 - k_0 + 1 \le 2^{j-|s|}$, it is enough to show that $\bar{w}_{p_1,k} \approx_{(\frac{\varepsilon}{2^{j-|s|}},0)} \bar{w}'_{p_1,k}$ for $k_0 \le k \le k_1$ and then apply basic composition and post-processing. Note that for every $k_0 \le k \le k_1$, the sensitivity $|\langle c_n, v_{i_1,p_1,k} - v'_{i_1,p_1,k}\rangle| \le \frac{4\alpha}{2^{-|s|}b}$, therefore for fixed $\xi_{i,n}$, $|\frac{1}{m}\sum_{i=1}^m\left(\langle c_n, v_{i,p_1,k}\rangle + \xi_{i,n}\right) - \frac{1}{m}\sum_{i=1}^m\left(\langle c_n, v'_{i,p_1,k}\rangle + \xi_{i,n}\right)| \le \frac{4\alpha}{2^{-|s|}mb}$. Using privacy guarantees of report noisy max (Dwork et al. (2014), Claim 3.9), we have $\bar{w}_{p_1,k} \approx_{(\frac{\varepsilon}{2^{j-|s|}},0)} \bar{w}'_{p_1,k}$ for $k_0 \le k \le k_1$ with $\mu_{j,s} = \frac{4\alpha 2^j}{bm\varepsilon}$.

Step 2: $\mathbf{x_{1:m,1:P,K}} \approx_{(\varepsilon,0)} \mathbf{x'_{1:m,1:P,K}}$ and $\mathbf{W} \approx_{(\varepsilon,0)} \mathbf{W'}$ by post-processing

In order to show $(x_{1,1,K},\dots,x_{m,P,K}) \approx_{(\varepsilon,0)} (x'_{1,1,K},\dots,x'_{m,P,K})$, we only need to prove that $(x_{1,p_1,K},\dots,x_{m,p_1,K}) \approx_{(\varepsilon,0)} (x'_{1,p_1,K},\dots,x'_{m,p_1,K})$ and apply post-processing. It is enough to show that iterates $(x_{1,p_1,1},\dots,x_{m,p_1,K})$ and $(x'_{1,p_1,1},\dots,x'_{m,p_1,K})$ is ε-indistinguishable.

The iterates $x_{1:m,p_1,1:k_0-1}$ and $x'_{1:m,p_1,1:k_0-1}$ do not depend on l_{i_1,t_1} or l'_{i_1,t_1}, hence 0-indistinguishable. Moreover, given that $(x_{1,p_1,k_0},\dots,x_{m,p_1,k_1})$ and $(x'_{1,p_1,k_0},\dots,x'_{m,p_1,k_1})$ are ε-indistinguishable, it is clear that $(x_{1,p_1,k_1+1},\dots,x_{m,p_1,K})$ and $(x'_{1,p_1,k_1+1},\dots,x'_{m,p_1,K})$ are ε-indistinguishable by post-processing. Similarly we have $W \approx_{(\varepsilon,0)} W'$.

Step 3: $\mathbf{V} \approx_{(\varepsilon,0)} \mathbf{V'}$ by Laplace mechanism

For our purpose, it suffices to establish that

$$(\langle c_n, v_{i_1,p_1,k_0}\rangle,\dots,\langle c_n, v_{i_1,p_1,k_1}\rangle) \approx_{(\varepsilon,0)} (\langle c_n, v'_{i_1,p_1,k_0}\rangle,\dots,\langle c_n, v'_{i_1,p_1,k_1}\rangle)$$

holds for $1 \leq n \leq d$, since l_{i_1,t_1} or l'_{i_1,t_1} is only used in client i_1, at phase p_1 and iteration $k_0, \ldots, k_1$. Therefore it is enough to show that $\langle c_n, v_{i_1,p_1,k} \rangle \approx_{(\frac{\varepsilon}{2^{j-|s|}},0)} \langle c_n, v'_{i_1,p_1,k} \rangle$ holds for $k_0 \leq k \leq k_1$ and $1 \leq n \leq d$, because $k_1 - k_0 + 1 \leq 2^{j-|s|}$. Note that $|\langle c_n, v_{i_1,p_1,k} - v'_{i_1,p_1,k} \rangle| \leq \frac{4\alpha}{2^{-|s|}b}$. Setting $\lambda_{i,j,s} = \frac{4\alpha 2^j}{b\varepsilon}$ and applying standard results of Laplace mechanism (Dwork et al. (2014)) lead to our intended results. $\square$

To help prove the upper bound of the regret, we introduce some lemmas first.

Lemma C.7. *Let t be the time-step, p be the index of phases, i be the index of the clients, and (j, s) be a vertex. For every index $1 \leq k \leq d$ of the vectors, we have*

$$\mathbb{E}\left[\exp\left\{c(\bar{v}_{p,j,s,k} - \nabla L_{t,k}(x_{i,p,j,s}))\right\}\right] \leq \exp\left(\frac{O(1)c^2(\alpha^2 + \beta^2)}{bm}\right),$$

where $\bar{v}_{p,j,s} = \frac{1}{m}\sum_{i=1}^{m} v_{i,p,j,s}$.

Proof. Let us fix p, t, k and i for simplicity and let $A_{j,s} = \bar{v}_{p,j,s,k} - \nabla L_{t,k}(x_{i,p,j,s})$. We prove the lemma by induction on the depth of the vertex, i.e., $|s|$. If $|s| = 0$, then $v_{i,p,j,\emptyset} = \nabla l(x_{i,p,j,\emptyset}; \mathcal{B}_{i,p,j,\emptyset})$ where $\mathcal{B}_{i,p,j,\emptyset}$ is a sample set of size b. Therefore we have

$$\begin{aligned}
\mathbb{E}[\exp cA_{j,s}] &= \mathbb{E}[\exp c(\bar{v}_{p,j,\emptyset,k} - \nabla L_{t,k}(x_{i,p,j,\emptyset})] \\
&= \mathbb{E}\left[\exp c\left(\frac{1}{mb}\sum_{i=1}^{m}\sum_{s\in\mathcal{B}_{i,p,j,\emptyset}} \nabla l_k(x_{i,p,j,\emptyset}; s) - \nabla L_{t,k}(x_{i,p,j,\emptyset})\right)\right] \\
&= \prod_{s\in\mathcal{B}_{i,p,j,\emptyset}}\prod_{i\in[m]} \mathbb{E}\left[\exp\frac{c}{bm}(\nabla l_k(x_{i,p,j,\emptyset}; s) - \nabla L_{t,k}(x_{i,p,j,\emptyset}))\right] \\
&\leq \exp\left(\frac{c^2\alpha^2}{2bm}\right),
\end{aligned}$$

where the last inequality holds because for a random variable $X \in [-\alpha, \alpha]$, we have $\mathbb{E}[\exp c(X - \mathbb{E}[X])] \leq \exp\left(\frac{c^2\alpha^2}{2}\right)$.

Assume the depth of the vertex $|s| \geq 0$ and let $s = s'a$ where $a \in \{0, 1\}$. If $a = 0$, clearly the lemma holds. If $a = 1$, recall that $v_{i,p,j,s} = v_{i,p,j,s'} + \nabla l(x_{i,p,j,s}; \mathcal{B}_{i,p,j,s}) - \nabla l(x_{i,p,j,s'}; \mathcal{B}_{i,p,j,s})$, then

$$\begin{aligned}
A_{j,s} &= \bar{v}_{p,j,s,k} - \nabla L_{t,k}(x_{i,p,j,s}) \\
&= A_{j,s'} + \frac{1}{m}\sum_{i=1}^{m}\nabla l_k(x_{i,p,j,s}; \mathcal{B}_{i,p,j,s}) - \frac{1}{m}\sum_{i=1}^{m}\nabla l_k(x_{i,p,j,s'}; \mathcal{B}_{i,p,j,s}) \\
&\quad - \nabla L_{t,k}(x_{i,p,j,s}) + \nabla L_{t,k}(x_{i,p,j,s'})
\end{aligned}$$

Let $\mathcal{B}_{i,p,<(j,s)} = \cup_{(j_1,s_1)<(j,s)}\mathcal{B}_{i,p,j_1,s_1}$ be the set containing all the samples used up to vertex (j, s) in phase p at client i. We have

$$\begin{aligned}
&\mathbb{E}\left[\exp cA_{j,s}\right]\\
&=\mathbb{E}\Bigg[\exp\Bigg(c(A_{j,s'}+\frac{1}{m}\sum_{i=1}^{m}\nabla l_k(x_{i,p,j,s};\mathcal{B}_{i,p,j,s})-\frac{1}{m}\sum_{i=1}^{m}\nabla l_k(x_{i,p,j,s'};\mathcal{B}_{i,p,j,s})\Bigg)\\
&\quad\times\exp\Bigg(-\nabla L_{t,k}(x_{i,p,j,s})+\nabla L_{t,k}(x_{i,p,j,s'})\Bigg)\Bigg]\\
&=\mathbb{E}\Bigg[\mathbb{E}\Bigg[\exp\Bigg(c(A_{j,s'}+\frac{1}{m}\sum_{i=1}^{m}\nabla l_k(x_{i,p,j,s};\mathcal{B}_{i,p,j,s})-\frac{1}{m}\sum_{i=1}^{m}\nabla l_k(x_{i,p,j,s'};\mathcal{B}_{i,p,j,s})\\
&\quad-\nabla L_{t,k}(x_{i,p,j,s})+\nabla L_{t,k}(x_{i,p,j,s'})\Bigg)\Bigg|\mathcal{B}_{i,p,<(j,s)}\Bigg]\Bigg]\\
&=\mathbb{E}\Bigg[\mathbb{E}\left[\exp\left(cA_{j,s'}\right)|\mathcal{B}_{i,p,<(j,s)}\right]\\
&\quad\times\mathbb{E}[\exp\Bigg(c(\frac{1}{m}\sum_{i=1}^{m}\nabla l_k(x_{i,p,j,s};\mathcal{B}_{i,p,j,s})-\frac{1}{m}\sum_{i=1}^{m}\nabla l_k(x_{i,p,j,s'};\mathcal{B}_{i,p,j,s})\\
&\quad-\nabla L_{t,k}(x_{i,p,j,s})+\nabla L_{t,k}(x_{i,p,j,s'})\Bigg)\Bigg|\mathcal{B}_{i,p,<(j,s)}\Bigg]\Bigg].
\end{aligned}$$

Since $l_{i,t}(\cdot;s)$ is β-smooth w.r.t. $\|\cdot\|_1$, we have

$$|\nabla l_k(x_{i,p,j,s};\mathcal{B}_{i,p,j,s})-\nabla l_k(x_{i,p,j,s'};\mathcal{B}_{i,p,j,s})|\le\beta\|x_{i,p,j,s}-x_{i,p,j,s'}\|_1.$$

Since vertex (j,s) is the right son of vertex (j,s'), the number of updates between $x_{i,p,j,s}$ and $x_{i,p,j,s'}$ is at most the number of leafs visited between these two vertices i.e. $2^{j-|s|}$. Therefore we have

$$\|x_{i,p,j,s}-x_{i,p,j,s'}\|_1\le\eta_{i,p,j,s'}2^{j-|s|}\le 2^{2-|s|}.$$

By using similar arguments to the case $|s|=0$, we can get

$$\begin{aligned}
&\mathbb{E}\Bigg[\exp\Bigg(c(\frac{1}{m}\sum_{i=1}^{m}\nabla l_k(x_{i,p,j,s};\mathcal{B}_{i,p,j,s})-\frac{1}{m}\sum_{i=1}^{m}\nabla l_k(x_{i,p,j,s'};\mathcal{B}_{i,p,j,s})\Bigg)\\
&\quad\times\exp\Bigg(-\nabla L_{t,k}(x_{i,p,j,s})+\nabla L_{t,k}(x_{i,p,j,s'})\Bigg)\Bigg|\mathcal{B}_{i,p,<(j,s)}\Bigg]\\
&\le\exp\left(\frac{O(1)c^2\beta^2 2^{-2|s|}}{m|\mathcal{B}_{i,p,j,s}|}\right)\\
&\le\exp\left(\frac{O(1)c^2\beta^2 2^{-|s|}}{bm}\right).
\end{aligned}$$

Then we get

$$\mathbb{E}[\exp cA_{j,s}]\le\mathbb{E}[\exp cA_{j,s'}]\exp\left(\frac{O(1)c^2\beta^2 2^{-|s|}}{bm}\right).$$

Apply this inductively, we have for every index $1\le k\le d$

$$\mathbb{E}[\exp cA_{j,s}]\le\exp\left(\frac{O(1)c^2(\alpha^2+\beta^2)}{bm}\right).$$

□

Lemma C.8 upper bounds the variance of the average gradient.

Lemma C.8. *At phase p, for each vertex (j,s) and $2^{p-1} \leq t < 2^p - 1$, we have*

$$\mathbb{E}\left[\|\bar{v}_{p,j,s} - \nabla L_t(x_{i,p,j,s})\|_\infty\right] \leq (\alpha+\beta) O\left(\sqrt{\frac{\log d}{bm}}\right),$$

where $\bar{v}_{p,j,s} = \frac{1}{m}\sum_{i=1}^m v_{i,p,j,s}$.

Proof. Lemma C.7 implies that $\bar{v}_{p,j,s,k} - \nabla L_{t,k}(x_{i,p,j,s})$ is $O(\frac{\alpha^2+\beta^2}{bm})$-sub-Gaussian for every index $1 \leq k \leq d$ of the vectors. Applying standard results of the maximum of d sub-Gaussian random variables, we get

$$\mathbb{E}[\|\bar{v}_{p,j,s} - \nabla L_t(x_{i,p,j,s})\|_\infty] \leq O\left(\sqrt{\frac{\alpha^2+\beta^2}{bm}}\right)\sqrt{\log d}.$$

□

Lemma C.9 gives the tail bound of the sum of i.i.d. random variables following Laplace distribution.

Lemma C.9. *Let $\xi_{i,n}$ be IID random variables following the distribution Lap$(\lambda_{i,j,s})$. Then we have*

$$\mathbb{E}\left[\max_{n:1\leq n\leq d}\left|\sum_{i=1}^m \xi_{i,n}\right|\right] \leq O(\sqrt{m}\lambda_{i,j,s}\ln d).$$

Proof. $\xi_{i,n}$'s are md IID random variables following the distribution Lap$(\lambda_{i,j,s})$. We note that

$$\mathbb{E}\left[\exp(u\xi_{i,n})\right] = \frac{1}{1-\lambda_{i,j,s}{}^2u^2}, |u| \leq \frac{1}{\lambda_{i,j,s}}.$$

Since $\frac{1}{1-\lambda_{i,j,s}{}^2u^2} \leq 1 + 2\lambda_{i,j,s}{}^2u^2 \leq \exp(2\lambda_{i,j,s}{}^2u^2)$, when $|u| \leq \frac{1}{2\lambda_{i,j,s}}$, $\xi_{i,n}$ is sub-exponential with parameter $(4\lambda_{i,j,s}{}^2, 2\lambda_{i,j,s})$. Applying standard results of linear combination of sub-exponential random variables, we can conclude that $\sum_{i=1}^m \xi_{i,n}$, denoted as Y_n, is sub-exponential with parameter $(4m\lambda_{i,j,s}{}^2, 2\lambda_{i,j,s})$. From standard results of tail bounds of sub-exponential random variables, we have

$$\mathbb{P}\left(|Y_n| \geq c\right) \leq 2\exp\left(-\frac{c^2}{8m\lambda_{i,j,s}{}^2}\right), \text{ if } 0 \leq c \leq 2m\lambda_{i,j,s},$$

$$\mathbb{P}(|Y_n| \geq c) \leq 2\exp\left(-\frac{c}{4\lambda_{i,j,s}}\right), \text{ if } c \geq 2m\lambda_{i,j,s}.$$

Since $\mathbb{P}\left(\max_{n:1\leq n\leq d}|Y_n| \geq c\right) \leq \sum_{n:1\leq n\leq d}\mathbb{P}\left(|Y_n| \geq c\right)$, we have

$$\mathbb{P}(\max_{n:1\leq n\leq d}|Y_n| \geq c) \leq 2d\exp\left(-\frac{c^2}{8m\lambda_{i,j,s}{}^2}\right), \text{ if } 0 \leq c \leq 2m\lambda_{i,j,s},$$

$$\mathbb{P}(\max_{n:1\leq n\leq d}|Y_n| \geq c) \leq 2d\exp\left(-\frac{c}{4\lambda_{i,j,s}}\right), \text{ if } c \geq 2m\lambda_{i,j,s}.$$

Then we have

$$
\begin{aligned}
\mathbb{E}\left[\max_{n:1\le n\le d}|Y_n|\right] &= \int_0^\infty \mathbb{P}\left(\max_{n:1\le n\le d}|Y_n| \ge c\right) dc \\
&= \sqrt{8m\ln d}\lambda_{i,j,s} + \int_{\sqrt{8m\ln d}\lambda_{i,j,s}}^{2m\lambda_{i,j,s}} \mathbb{P}\left(\max_{n:1\le n\le d}|Y_n| \ge c\right) dc \\
&\quad + \int_{2m\lambda_{i,j,s}}^{\infty} \mathbb{P}\left(\max_{n:1\le n\le d}|Y_n| \ge c\right) dc \\
&\le \sqrt{8m\ln d}\lambda_{i,j,s} + \int_{\sqrt{8m\ln d}\lambda_{i,j,s}}^{2m\lambda_{i,j,s}} 2\exp\left(\ln d - \frac{c^2}{8m\lambda_{i,j,s}{}^2}\right) dc \\
&\quad + \int_{2m\lambda_{i,j,s}}^{\infty} 2\exp\left(\ln d - \frac{c}{4\lambda_{i,j,s}}\right) dc \\
&\le \sqrt{8m\ln d}\lambda_{i,j,s} + \sqrt{8m}\lambda_{i,j,s}\int_{-\infty}^{+\infty} \exp\left(-u^2\right) du \\
&\quad + 8\lambda_{i,j,s}\int_{2/m-\ln d}^{\infty} \exp(-u) du \\
&\le \sqrt{8m\ln d}\lambda_{i,j,s} + \sqrt{8m}\lambda_{i,j,s}\int_{-\infty}^{+\infty} \exp\left(-u^2\right) du \\
&\quad + 8\lambda_{i,j,s}\ln d + 8\lambda_{i,j,s}\int_{0}^{\infty} \exp(-u) du \\
&= \sqrt{8m\ln d}\lambda_{i,j,s} + \sqrt{8m\pi}\lambda_{i,j,s} + 8\lambda_{i,j,s}\ln d + 8\lambda_{i,j,s} \\
&= O(\sqrt{m}\lambda_{i,j,s}\ln d).
\end{aligned}
$$

□

Lemma C.10. *Setting $\lambda_{i,j,s} = \frac{4\alpha 2^j}{b\varepsilon}$, we have*

$$
\mathbb{E}[\langle \bar{v}_{p,k}, \bar{w}_{p,k}\rangle] \le \mathbb{E}\left[\min_{w\in\mathcal{X}}\langle \bar{v}_{p,k}, w\rangle\right] + O\left(\frac{\alpha 2^j \ln d}{b\varepsilon\sqrt{m}}\right).
$$

Proof. Since $\bar{w}_{p,k} = \arg\min_{c_n:1\le n\le d}\left[\frac{1}{m}\sum_{i=1}^m \left(\langle c_n, v_{i,p,k}\rangle + \xi_{i,n}\right)\right]$, where $\xi_{i,n} \sim \mathrm{Lap}(\lambda_{i,j,s})$. We denote $\bar{w}_{p,k}$ as $c_{n^\star}$ and we have

$$
\begin{aligned}
\langle \bar{w}_{p,k}, \bar{v}_{p,k}\rangle &= \langle c_{n^\star}, \bar{v}_{p,k}\rangle \\
&= \min_{n:1\le n\le d}\left(\langle c_n, \bar{v}_{p,k}\rangle + \frac{1}{m}\sum_{i=1}^m \xi_{i,n}\right) - \frac{1}{m}\sum_{i=1}^m \xi_{i,n^\star} \\
&\le \min_{n:1\le n\le d}\langle c_n, \bar{v}_{p,k}\rangle + \frac{1}{m}\max_{n:1\le n\le d}\sum_{i=1}^m \xi_{i,n} - \frac{1}{m}\min_{n:1\le n\le d}\sum_{i=1}^m \xi_{i,n} \\
&\le \min_{n:1\le n\le d}\langle c_n, \bar{v}_{p,k}\rangle + \frac{2}{m}\max_{n:1\le n\le d}\left|\sum_{i=1}^m \xi_{i,n}\right|.
\end{aligned}
$$

Applying Lemma C.9, we get

$$
\mathbb{E}[\langle \bar{v}_{p,k}, \bar{w}_{p,k}\rangle] \le \mathbb{E}\left[\min_{w\in\mathcal{X}}\langle \bar{v}_{p,k}, w\rangle\right] + O\left(\frac{\lambda_{i,j,s}\ln d}{\sqrt{m}}\right).
$$

□

With the lemmas above, we are ready to prove Theorem 2.2.

Proof. Lemma C.6 implies the claim about privacy. We proceed to prove the regret upper bound.

$$\begin{aligned}
L_t(x_{i,p,k+1}) &\overset{(a)}{\leq} L_t(x_{i,p,k}) + \langle \nabla L_t(x_{i,p,k}), x_{i,p,k+1} - x_{i,p,k}\rangle + \beta\frac{\|x_{i,p,k+1} - x_{i,p,k}\|_1^2}{2} \\
&\overset{(b)}{\leq} L_t(x_{i,p,k}) + \eta_{i,p,k}\langle \nabla L_t(x_{i,p,k}), \bar{w}_{p,k} - x_{i,p,k}\rangle + \frac{1}{2}\beta{\eta_{i,p,k}}^2 \\
&= L_t(x_{i,p,k}) + \eta_{i,p,k}\langle \nabla L_t(x_{i,p,k}), x^\star - x_{i,p,k}\rangle + \eta_{i,p,k}\langle \nabla L_t(x_{i,p,k}) - \bar{v}_{p,k}, \bar{w}_{p,k} - x^\star\rangle \\
&\quad + \eta_{i,p,k}\langle \bar{v}_{p,k}, \bar{w}_{p,k} - x^\star\rangle + \frac{1}{2}\beta{\eta_{i,p,k}}^2 \\
&\overset{(c)}{\leq} L_t(x_{i,p,k}) + \eta_{i,p,k}[L_t(x^\star) - L_t(x_{i,p,k})] + \eta_{i,p,k}\|\nabla L_t(x_{i,p,k}) - \bar{v}_{p,k}\|_\infty + \eta_{i,p,k}(\langle \bar{v}_{p,k}, \bar{w}_{p,k}\rangle \\
&\quad - \min_{w\in\mathcal{X}}\langle \bar{v}_{p,k}, w\rangle) + \frac{1}{2}\beta{\eta_{i,p,k}}^2,
\end{aligned}$$

where (a) is due to β-smoothness, (b) is because of the updating rule of $x_{i,p,k}$, and (c) follows from the convexity of the loss function and Hölder's inequality.

Subtracting $L_t(x^\star)$ from each side and taking expectations, we have

$$\begin{aligned}
\mathbb{E}[L_t(x_{i,p,k+1}) - L_t(x^\star)] \leq (1-\eta_{i,p,k})\mathbb{E}[L_t(x_{i,p,k}) - L_t(x^\star)] + \eta_{i,p,k}\mathbb{E}[\|\nabla L_t(x_{i,p,k}) - \bar{v}_{p,k}\|_\infty] \\
+ \eta_{i,p,k}\mathbb{E}[\langle \bar{v}_{p,k}, \bar{w}_{p,k}\rangle - \min_{w\in\mathcal{X}}\langle \bar{v}_{p,k}, w\rangle] + \frac{1}{2}\beta{\eta_{i,p,k}}^2.
\end{aligned}$$

Applying Lemma C.10 and Lemma C.8, we have

$$\begin{aligned}
\mathbb{E}[L_t(x_{i,p,k+1}) - L_t(x^\star)] \leq (1-\eta_{i,p,k})\mathbb{E}[L_t(x_{i,p,k}) - L_t(x^\star)] + \eta_{i,p,k}(\alpha+\beta)O\left(\sqrt{\frac{\log d}{bm}}\right) \\
+ \frac{\eta_{i,p,k}\alpha 2^j}{b\varepsilon}O\left(\frac{\ln d}{\sqrt{m}}\right) + \frac{1}{2}\beta{\eta_{i,p,k}}^2.
\end{aligned}$$

Let $\alpha_k = \eta_{i,p,k}(\alpha+\beta)O\left(\sqrt{\frac{\log d}{bm}}\right) + \frac{\eta_{i,p,k}\alpha 2^j}{b\varepsilon}O\left(\frac{\ln d}{\sqrt{m}}\right) + \frac{1}{2}\beta{\eta_{i,p,k}}^2$. We simplify the notion of $\eta_{i,p,k}$ to η_k. Then we have

$$\begin{aligned}
\mathbb{E}[L_t(x_{i,p,k}) - L_t(x^\star)] &\leq \sum_{k'=1}^{K}\alpha_k\prod_{i>k}^{K-1}(1-\eta_{k'}) \\
&= \sum_{k=1}^{K}\alpha_k\frac{(k+1)k}{K(K-1)} \\
&\leq \sum_{k=1}^{K}\alpha_k\frac{(k+1)^2}{(K-1)^2},
\end{aligned}$$

where $\eta_k = \frac{2}{k+1}$.

Since $K = 2^{T_1}$, simply algebra implies that

$$\mathbb{E}[L_t(x_{i,p,K}) - L_t(x^\star)] \leq O\Bigg((\alpha+\beta)\sqrt{\frac{\log d}{bm}} + \frac{\alpha 2^{T_1}\ln d}{b\varepsilon\sqrt{m}} + \frac{\beta}{2^{T_1}}\Bigg). \tag{4}$$

At iteration $2^{p-1} \leq t < 2^p$, setting $b = \frac{2^{p-1}}{(p-1)^2}$ and $T_1 = \frac{1}{2}\log\left(\frac{b\varepsilon\beta\sqrt{m}}{\alpha\log d}\right)$ in Equation (4), we have

$$\mathbb{E}[L_t(x_{i,p,K}) - L_t(x^\star)] \leq O\Bigg((\alpha+\beta)p\sqrt{\frac{\log d}{2^p m}} + \frac{p\sqrt{\alpha\beta}\log d}{m^{\frac{1}{4}}\sqrt{2^p\varepsilon}}\Bigg).$$

Therefore, the total regret from time step 2^p to $2^{p+1}-1$ is at most

$$\mathbb{E}\left[\sum_{t=2^p}^{2^{p+1}-1} L_t(x_{i,t}) - \min_{u\in\mathcal{X}} \sum_{t=2^p}^{2^{p+1}-1} L_t(u)\right] \le O\left((\alpha+\beta)p\sqrt{\frac{2^p\log d}{m}} + \frac{p\sqrt{\alpha\beta}\log d\sqrt{2^p}}{m^{\frac{1}{4}}\sqrt{\varepsilon}}\right).$$

Summing over p, we can get

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T} L_t(x_{i,t}) - \min_{u\in\mathcal{X}} \sum_{t=1}^{T} L_t(u)\right] &\le \sum_{p=1}^{\log T} O\left((\alpha+\beta)p\sqrt{\frac{2^p\log d}{m}} + \frac{p\sqrt{\alpha\beta}\log d\sqrt{2^p}}{m^{\frac{1}{4}}\sqrt{\varepsilon}}\right)\\ &\le O\left((\alpha+\beta)\sum_{p=1}^{\log T} n\sqrt{\frac{2^p\log d}{m}} + \frac{\sqrt{\alpha\beta}\log d}{m^{\frac{1}{4}}\sqrt{\varepsilon}}\sum_{p=1}^{\log T} n\sqrt{2^p}\right)\\ &\le O\left((\alpha+\beta)\log T\sqrt{\frac{T\log d}{m}} + \frac{\sqrt{\alpha\beta}\log d\sqrt{T}\log T}{m^{\frac{1}{4}}\sqrt{\varepsilon}}\right).\end{aligned}$$

Now we turn our focus to communication cost. Since there are $\log T$ phases, and within each phase, there are 2^{T_1} leaf vertices where communication is initiated, the communication frequency scales in $O(\sum_p 2^{T_1})$. Therefore, the communication cost scales in $O(m^{5/4}d\sqrt{T\varepsilon\beta/(\alpha\log d)}\log T)$. □

D Proof of Theorem 2.6

Theorem D.1 (Restatement of Theorem 2.6)**.** *Let $l_{i,t}\in[0,1]^d$ be chosen by an oblivious adversary under near-realizability assumption. Set $0<\rho<1/2$, $\kappa=O(\log(d/\rho))$, $L=mL^\star+\frac{8\log(2T^2/(N^2\rho))}{\varepsilon}+4/\eta$, and $\eta=\varepsilon/2\kappa$. Then the algorithm is ε-DP, the communication cost scales in $O\left(mdT/N\right)$, and with probability at least $1-O(\rho)$, the pre-client regret is upper bounded by $O\left(\frac{\log^2(d)+\log\left(\frac{T^2}{N^2\rho}\right)\log\left(\frac{d}{\rho}\right)}{m\varepsilon}+(N+L^\star)\log\left(\frac{d}{\rho}\right)\right)$.*

Moreover, setting $\eta=\varepsilon/\sqrt{\kappa\log(1/\delta)}$, we have the algorithm is (ε,δ)-DP, the communication cost scales in $O\left(mdT/N\right)$, and with probability at least $1-O(\rho)$, the pre-client regret is upper bounded by $O\left(\frac{\log^{\frac{3}{2}}(d)\sqrt{\log(\frac{1}{\delta})}+\log\left(\frac{T^2}{N^2\rho}\right)\log\left(\frac{d}{\rho}\right)}{m\varepsilon}+(N+L^\star)\log\left(\frac{d}{\rho}\right)\right)$.

Proof. **DP guarantee:** There are κ applications of exponential mechanism with privacy parameter η. Moreover, sparse vector technique is applied over each sample once, hence the κ applications of sparse-vector are $\varepsilon/2$-DP. Overall, the algorithm is $(\varepsilon/2+\kappa\eta)$-DP and $(\varepsilon/2+\sqrt{2\kappa\log(1/\delta)}\eta+\kappa\eta(e^\eta-1),\delta)$-DP (Advanced composition). Setting $\eta=\varepsilon/2\kappa$ results in ε-DP and $\eta=\varepsilon/\sqrt{\kappa\log(1/\delta)}$ results in (ε,δ)-DP.

Communication cost: The number of communication between the central server and clients scales in $O(mT/N)$. Moreover, within each communication, the number of scalars exchanged scales in $O(d)$. Therefore the communication cost is $O(mdT/N)$.

Regret upper bound: We define a potential at phase $n\in[T/N]$:

$$\phi_n=\sum_{x\in[d]} e^{-\eta L_n(x)/2}$$

where $L_n(x)=\max\left(\sum_{i=1}^m\sum_{t'=1}^{nN-1} l_{i,t'}(x), mL^\star\right)$. Note that $\phi_0=de^{-\eta mL^\star/2}$ and $\phi_n\ge e^{-\eta mL^\star/2}$ for all $n\in[T/N]$ since there is $x\in[d]$ such that $\sum_{i=1}^m\sum_{t=1}^T l_{i,t}(x)\le mL^\star$. We split the iterates to $s=\lceil\log d\rceil$ rounds $n_0N, n_1N,\ldots,n_sN$ where n_p is the largest $n\in[T/N]$ such that $\phi_{n_p}\ge\phi_0/2^p$. Let Z_p be the number of switches in $\left[n_pN,(n_{p+1}-1)N\right]$ (number of times the exponential mechanism is used to pick x_t). Let $Z=\sum_{p=0}^{s-1}Z_p$ be the total number of switches. Note that $Z\le 3s+\sum_{p=0}^{s-1}\max\left(Z_p-3,0\right)$

and Lemma D.2 implies $\max\left(Z_p - 3, 0\right)$ is upper bounded by a geometric random variable with success probability $1/3$. Therefore, using concentration of geometric random variables (Lemma D.3), we get that

$$P(Z \geq 3s + 24\log(1/\rho)) \leq \rho.$$

Since $K \geq 3s + 24\log(1/\rho)$, the algorithm does not reach the switching budget with probability $1 - O(\rho)$. So the total number of switching scales as $O(\log(d/\rho))$. Now we analyze the regret. Define $T_1N, \dots T_CN$ as the switching time steps with $T_CN = T$, where $C = O(\log(d/\rho))$. Theorem 3.24 in Dwork et al. (2014) implies that with probability at least $1 - \rho$,

$$\begin{aligned}
\sum_{t=1}^{T}\sum_{i=1}^{m} l_{i,t}(x_{i,t}) &= \sum_{c=1}^{C}\sum_{t=T_{c-1}N+1}^{T_cN}\sum_{i=1}^{m} l_{i,t}(x_{i,t}) \\
&= \sum_{c=1}^{C}\left(\sum_{t=T_{c-1}N+1}^{T_cN-N}\sum_{i=1}^{m} l_{i,t}(x_{i,t}) + \sum_{t=T_cN-N+1}^{T_cN}\sum_{i=1}^{m} l_{i,T_n}(x_t)\right) \\
&\leq \sum_{c=1}^{C}\left(L + \frac{8\log(2T^2/(N^2\rho))}{\epsilon} + mN\right) \\
&= \sum_{c=1}^{C}\left(mL^\star + \frac{16\log(2T^2/(N^2\rho))}{\epsilon} + 4/\eta + mN\right) \\
&= O\left(mL^\star\log(d/\rho) + \frac{\log(T^2/(N^2\rho))\log(d/\rho)}{\varepsilon} + \frac{4\log(d/\rho)}{\eta} + mN\log(d/\rho)\right). \quad (5)
\end{aligned}$$

Case 1: Setting $\eta = \varepsilon/2\kappa$ in Equation (5), we have

$$\sum_{i=1}^{m}\sum_{t=1}^{T} l_{i,t}(x_{i,t}) \leq O\left(mL^\star\log d + \frac{\log^2 d + \log(T^2/(N^2\rho))\log(d/\rho)}{\varepsilon} + mN\log(d/\rho)\right).$$

Case 2: Setting $\eta = \varepsilon/\sqrt{\kappa\log(1/\delta)}$ in Equation (5), we have

$$\sum_{i=1}^{m}\sum_{t=1}^{T} l_{i,t}(x_{i,t}) \leq O\left(mL^\star\log d + \frac{\log^{3/2} d\sqrt{\log(1/\delta)} + \log(T^2/(N^2\rho))\log(d/\rho)}{\varepsilon} + mN\log(d/\rho)\right).$$

Lemma D.2. *Fix $0 \leq p \leq s-1$. Then for any $1 \leq k \leq T/N$, it holds that*

$$P\left(Z_p = k+3\right) \leq (2/3)^{k+2} < (2/3)^{k-1}(1/3).$$

Proof. Let $n_pN \leq nN \leq n_{p+1}N$ be a time-step when a switch happens (exponential mechanism is used to pick x_t). Note that $\phi_{n_{p+1}} \geq \phi_n/2$. We prove that the probability that x_t is switched between nN and $n_{p+1}N$ is at most $2/3$. To this end, note that if x_t is switched before $n_{p+1}N$ then $\sum_{i=1}^{m}\sum_{t'=nN}^{n_{p+1}N-1} l_{i,t'}(x_{t'}) \geq L - \frac{8\log(2T^2/(N^2\rho))}{\varepsilon}$, therefore $L_{n_{p+1}}(x) - L_n(x) \geq L - \frac{8\log(2T^2/(N^2\rho))}{\varepsilon} \geq 4/\eta$. Thus we have that

$$\begin{aligned}
P\left(x_t \text{ is switched before } n_{p+1}N\right) &\leq \sum_{x\in[d]} P\left(x_t = x\right) 1\left\{L_{n_{p+1}}(x) - L_n(x) \geq 4/\eta\right\} \\
&= \sum_{x\in[d]} \frac{e^{-\eta L_n(x)/2}}{\phi_n} \cdot 1\left\{L_{n_{p+1}}(x) - L_n(x) \geq 4/\eta\right\}
\end{aligned}$$

$$\begin{aligned}
&\leq \sum_{x\in[d]} \frac{e^{-\eta L_n(x)/2}}{\phi_n} \cdot \frac{1-e^{-\eta\left(L_{n_{p+1}}(x)-L_n(x)\right)/2}}{1-e^{-2}} \\
&\leq 4/3\left(1-\phi_{n_{p+1}}/\phi_n\right) \\
&\leq 2/3.
\end{aligned}$$

where the second inequality follows the fact that $1\{a \geq b\} \leq \frac{1-e^{-\eta a}}{1-e^{-\eta b}}$ for $a, b, \eta \geq 0$, and the last inequality since $\phi_{n_{p+1}}/\phi_n \geq 1/2$. This argument shows that after the first switch inside the range $\left[n_p N, n_{p+1}N - 1\right]$, each additional switch happens with probability at most 2/3. So we have

$$P\left(Z_p = k+3\right) \leq (2/3)^{k+2} < (2/3)^{k-1}(1/3).$$

$\square$

Lemma D.3 (Asi et al. (2023), Lemma A.2)**.** *Let $W_1, \ldots, W_n$ be i.i.d. geometric random variables with success probability p. Let $W = \sum_{i=1}^n W_i$. Then for any $k \geq n$*

$$\mathbb{P}(W > 2k/p) \leq \exp(-k/4).$$

$\square$