Decentralized Differentially Private Power Method

Andrew Campbell [1] **Anna Scaglione** [1] **Sean Peisert** [2]

Abstract

We propose a novel adaptation of the Decentralized Power Method (D-PM) to perform Differentially Private (DP) Principal Component Analysis (PCA) in networked settings. Unlike typical decentralized PCA approaches—where each agent can access the full n-dimensional sample space—we focus on the more challenging case where each agent observes only a subset of the dimensions. To address this complexity, we introduce a Decentralized (ϵ, δ)-DP Power Method (D-DP-PM) that ensures individual data privacy while collaboratively estimating the global eigenvectors across the network. We prove that our method satisfies the prescribed (ϵ, δ)-DP guarantee and derive a corresponding bound on δ. Additionally, we establish the convergence rate of our approach, including the impact of network topology. Experiments on a synthetic dataset confirm the efficacy of our method.

1. Introduction

The goal of computing the singular vectors of decentralized data is a critical data analysis task used in both research and commercial settings. For example, the eigen vector estimation of the covariance matrix of a dataset $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ can be used for learning distance embeddings or for dimensionality reduction. However, as the big data paradigm becomes more popular there is a growing need for incorporating privacy into data analysis pipelines. While distributed PCA methods (Scaglione et al., 2008; Wang et al., 2023; Chai et al., 2022; Froelicher et al., 2023; Grammenos et al., 2020; Qu et al., 2002; Liang et al., 2014) provide some degree of privacy, Differentially Privacy (DP) is required to make formal privacy guarantees. In this paper we focus on the Decentralized Power Method (D-PM) (Scaglione et al., 2008) and its adaptions to the DP setting.

[1]Department of Electrical and Computer Engineering, Cornell University, Cornell Tech, New York, 10044 NY, USA [2]Lawrence Berkeley National Laboratory, CA USA. Correspondence to: Andrew Campbell <ac2458@cornell.edu>.

TPDP Workshop 2025, Mountain View, USA.

In this work, we aim to estimate the eigenvectors of the covariance matrix $\boldsymbol{X}\boldsymbol{X}^\top$ in a decentralized manner (or left singular vectors of $\boldsymbol{X}$), given a data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$. Here, n represents the number of data samples, each of dimension d. We assume that our dataset is partitioned row-wise among m agents in a network. Specifically, each agent i holds a data matrix $\boldsymbol{X}_i \in \mathbb{R}^{n_i \times d}$, $\sum_{i=1}^m n_i = n$, such that:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_1 \\ \vdots \\ \boldsymbol{X}_m \end{bmatrix}. \tag{1}$$

This is a different set up from the prior art which assumes two different scenarios. The most common one distributes the data column wise, i.e each agent has $\boldsymbol{X}_i \in \mathbb{R}^{n \times d_i}$ where

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_1 & \dots & \boldsymbol{X}_m \end{bmatrix}. \tag{2}$$

A second case considered in the literature assume each agent has a portion of $\boldsymbol{X}\boldsymbol{X}^\top$, denoted $\boldsymbol{A}_i$ where

$$\boldsymbol{X}\boldsymbol{X}^\top = \sum_{i=1}^m \boldsymbol{A}_i. \tag{3}$$

This is not possible in our case, since the agents only have have access to the block diagonal elements of $\boldsymbol{X}\boldsymbol{X}^\top$ ($\boldsymbol{X}_i\boldsymbol{X}_i^\top$). Below we describe the threat model given our setup.

Threat Model: We assume that each agent behaves collaboratively—it will not inject false data, ignore received information, collude, etc.—but remains curious and thus potentially attempts to infer the data held by other agents. That is, each agent should be prevented from reconstructing what specific data were in the batch used for the principal components computation. Given that our setup assumes no trustworthy centralized node it falls under the Local Differentially Privacy (LDP) setting. That is, we are interested in providing guarantees on the privacy leakage of node i with respect to (w.r.t) node j. Furthermore, we define adjacent datasets by a single row change such that $\|\boldsymbol{X} - \boldsymbol{X}'\| \leq 1$.

Prior work by (Balcan et al., 2017; Hardt & Price, 2014; Balcan et al., 2016) has established analytical convergence guarantees for the Noisy Power Method, thereby providing foundational convergence results for the Differentially

Private Power Method (DP-PM). More recently (Nicolas et al., 2024; Guo et al., 2021; Wang & Xu, 2020) have proposed federated approaches for the DP-PM algorithm. (Nicolas et al., 2024; Balcan et al., 2017; Guo et al., 2021) assume a data partition matching (3) while (Qu et al., 2002; Grammenos et al., 2020; Wang & Chang, 2018; Ge et al., 2018) utilize a data partitioning consistent with (2). Interestingly, none of the above papers have developed a fully decentralized [1] DP-PM, and thus the DP-PM method has not been extended to multi-agent networks. To the best of our knowledge we are the first to propose the D-DP-PM algorithm in the multi-agent network setting and provide convergence and DP guarantees. Additionally, to the best of our knowledge, we are the first to consider the data partitioning given by (1) for the D-DP-PM algorithm. Below are our contributions.

Contributions:

- We are the first in proposing a D-DP-PM algorithm for estimating sample covariance eigen vectors when the data matrix is split according to (1).
- We analyze the algorithm providing a proof for: 1) the DP guarantee along with an asymptotic DP bound; 2) the convergence of the proposed D-DP-PM algorithm over connected communication topologies.

2. Proposed Method

At a high level, our algorithm is a noisy variant of the D-PM algorithm (Scaglione et al., 2008). In this section we provide the complete algorithm and describe the key steps.

Centralized Power Iteration: Let us consider only the principal eigen vector, i.e. $l = 1$, and omit the index. The (centralized) PM is an iterative algorithm that computes the dominant eigenvector of a matrix using the following update:

$$\boldsymbol{q}^{(t+1)} = \frac{\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{q}^{(t)}}{\left\|\boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{q}^{(t)}\right\|}, \tag{4}$$

where $\boldsymbol{q}^{(0)}$ is a random vector. Running (4) for T iterations generates a vector that approximates the principal eigen vector $\boldsymbol{X}\boldsymbol{X}^\top$. In order to generate the whole eigen subspace, we remove the contribution of $\boldsymbol{q}^{(T)}$ from the $\boldsymbol{X}\boldsymbol{X}^\top$ by

$$(\boldsymbol{X}\boldsymbol{X}^\top)_{\text{new}} = \boldsymbol{X}\boldsymbol{X}^\top - \lambda_1 \boldsymbol{q}\boldsymbol{q}^\top, \tag{5}$$

and repeat (4).

Decentralized Power Iteration: The centralized power method was extended to the decentralized setting (Scaglione et al., 2008) where the observation was made that inner-products could be computed in a distributed manner using the average consensus protocol. That is, for mixing matrix $\boldsymbol{W}$ and for consensus steps c, the inner product at node i is:

$$\boldsymbol{X}^\top \boldsymbol{q} + \mathcal{O}\left(\lambda_2^c\left(\boldsymbol{W}\right)\right) = m \sum_{j=1}^{m} \left(\boldsymbol{W}^c\right)_{ij} \boldsymbol{X}_j^\top \boldsymbol{q}_j. \tag{6}$$

The upshot of (6) is that, via consensus aggregation, agents can approximate the projection over the other agents data. Our contributions lie in the incorporation of a DP mechanism into the network updates and the analysis of the resulting D-DP-PM algorithm.

Decentralized DP-Power Iteration: Notice that in the decentralized PM each agent only shares the projected vector $\boldsymbol{z}_i := \boldsymbol{X}_i^\top \boldsymbol{q}_i^{(t)}$ (step 6 Algorithm 1) via consensus. Crucially, since each agent communicates only its own segment of the projected eigenvector, no agent ever tracks the full vector $\boldsymbol{q}$. This fact, together with the independent and random initialization of each agent's vector $\boldsymbol{q}_i^{(0)}$, inherently provides additional privacy that we deliberately utilize. Moreover, at each iteration of the algorithm we add $\boldsymbol{p}^{(t)} \sim \mathcal{N}(\left(\boldsymbol{0}, \sigma_{\boldsymbol{p}}^2 \boldsymbol{I}_d\right)$ prior to the consensus step (step 5 Algorithm 1). Finally, rather than normalizing $\boldsymbol{q}^{(t+1)}$ by its norm, we use a predetermined scalar α, thus preserving Gaussianity. This is critical for the tractability of the analysis since it allows us to stack the data releases of every agent into a single multi-variate Gaussian observation. To complete the eigenvector computation we must normalize each $\boldsymbol{q}_i$ by the total norm $\boldsymbol{q}$ and therefore must add noise $\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{0}, \sigma_{\boldsymbol{u}}^2 \boldsymbol{I}_{n_i})$ according to the sensitivity of $\boldsymbol{q}$ (step 9 Algorithm 1). Finally, in order to compute the remaining eigenvectors we sequentially remove the contribution from the dominant eigen vector and repeat the algorithm. Concretely, where $\boldsymbol{z}_i^{(t+1/2)} := m \sum_j^n \left(\boldsymbol{W}^c\right)_{ij} \boldsymbol{z}_j^{(t)}$ (step 6 Algorithm 1), each agent updates its local dataset by

$$\boldsymbol{X}_i = \boldsymbol{X}_i - \boldsymbol{q}_{l-1} \boldsymbol{z}_i^{(T-1/2)}$$

which projects the data orthogonally to the directions of the leading singular vector. See Algorithm 1 for the full details.

3. Analysis

We introduce the following assumptions:

Assumption 1 (The mixing matrix)**.** *The mixing matrix $\boldsymbol{W}$ satisfies the following conditions:*

i. $w_{ij} > 0$ *if and only if there exists an edge between nodes i and j.*
ii. The underlying graph is undirected.
iii. $\boldsymbol{W}$ *is doubly-stochastic.*
iv. The agent network, $\mathcal{G}(\{1, \cdots, m\}, E)$ is strongly connected.

A consequence of Assumption 1 is that $0 \leq \lambda_2(\boldsymbol{W}) < 1$.

[1] Prior art assumes there is a central processing node.

Algorithm 1 D-DP-PM

1: **Init:** $\boldsymbol{X}_i$, rank r, c consensus steps, scaling factor α, $\boldsymbol{q}_{il}^{(0)} \sim \mathcal{N}\left(\boldsymbol{0}, \sigma_q^2 \boldsymbol{I}\right)$, $\boldsymbol{p}^{(t)} \sim \mathcal{N}\left(\boldsymbol{0}, \sigma_p^2 \boldsymbol{I}\right)$, $\boldsymbol{u} \sim \mathcal{N}\left(\boldsymbol{0}, \sigma_u^2 \boldsymbol{I}\right)$
2: **for** $l \in [1, r]$ **do**
3: $\quad \boldsymbol{X}_i = \boldsymbol{X}_i - \boldsymbol{q}_{l-1} \boldsymbol{z}^{(T-1/2)}$
4: $\quad$ **for** $t < T-1$ **do**
5: $\quad\quad \boldsymbol{z}_i^{(t)} = \boldsymbol{X}_i^\top \boldsymbol{q}_{i,l}^{(t)} + \boldsymbol{p}^{(t)}$
6: $\quad\quad \boldsymbol{z}_i^{(t+1/2)} = m \sum_j^n \left(\boldsymbol{W}^c\right)_{ij} \boldsymbol{z}_j^{(t)}$
7: $\quad\quad \boldsymbol{q}_{il}^{(t+1)} = \alpha \boldsymbol{X}_i \boldsymbol{z}_i^{(t+1/2)}$
8: $\quad$ **for** $t = T$ **do**
9: $\quad\quad \boldsymbol{q}_{il}^{(T-1)} = \boldsymbol{q}_{il}^{(T-1)} + \boldsymbol{u}$
10: $\quad\quad$ Share $\boldsymbol{q}_{il}^{(T-1)}$, Receive $\boldsymbol{q}_{jl}^{(T-1)}$
11: $\quad\quad \boldsymbol{q}_{il}^{(T)} = \dfrac{\boldsymbol{q}_{il}^{(T)-1}}{\left\|\boldsymbol{q}_l^{(T-1)}\right\|}$
12: return $\hat{\boldsymbol{U}}^{(r)} = \left[\boldsymbol{q}_1^{(T)} \ldots \boldsymbol{q}_r^{(T)}\right]$

Next, we define

$$\boldsymbol{\xi} := \left(\boldsymbol{W}^c \otimes \boldsymbol{I}_d\right) \mathrm{Diag}^\top\left(\boldsymbol{X}\right) - \left(\frac{\boldsymbol{1}\boldsymbol{1}^\top}{m} \otimes \boldsymbol{I}_d\right) \mathrm{Diag}^\top\left(\boldsymbol{X}\right), \tag{7}$$

where

$$\mathrm{Diag}\left(\boldsymbol{X}\right) := \begin{bmatrix} \boldsymbol{X}_1 & \ldots & \boldsymbol{0} \\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \ldots & \boldsymbol{X}_m \end{bmatrix}.$$

Using this definition of $\boldsymbol{\xi}$ we have the following assumption.

Assumption 2. *We assume that α is chosen such that*

$$\rho := \alpha \left\|\boldsymbol{X}\boldsymbol{X}^\top + m\,\mathrm{Diag}(\boldsymbol{X})\boldsymbol{\xi}\right\| < 1. \tag{8}$$

3.1. Differential Privacy

In this section, we introduce Theorem 1, which provides conditions on the algorithm's parameters to ensure it satisfies (ϵ, δ)-DP. The proof of the bound relies on a Chernoff bound and the composition theorems for differential privacy. The key fact, which we noted in the previous section, is that the initial noise distribution and the release of each $\boldsymbol{z}_i$ are both Gaussian. That is,

$$\boldsymbol{Z}^{(T-1)} := \begin{bmatrix} \boldsymbol{z}_1^{(1)} & \ldots & \boldsymbol{z}_m^{(1)} & \boldsymbol{z}_1^{(2)} & \ldots & \boldsymbol{z}_m^{(T-1)} \end{bmatrix}^\top, \tag{9}$$

is a Gaussian matrix that contains the releases of the entire network for all iterations. Let $\boldsymbol{Z}_i^{(T-1)}$ refer to agent i's releases. Then we define

$$\boldsymbol{B}_i := \mathbb{E}\left[\boldsymbol{Z}_i^{(T-1)}\right] \tag{10}$$

$$\boldsymbol{C}_i := \boldsymbol{E}\left[\left(\boldsymbol{Z}_i^{(T-1)} - \boldsymbol{B}_i\right)\left(\boldsymbol{Z}_i^{(T-1)} - \boldsymbol{B}_i\right)^\top\right], \tag{11}$$

as the mean and covariance matrix at agent i after $T-1$ iterations. We denote the adjacent means and covariances by $\boldsymbol{B}_i'$ and $\boldsymbol{C}_i'$. Furthermore, where $\Delta_{\boldsymbol{q}}$ is the sensitivity of $\boldsymbol{q}$ as defined in (Nicolas et al., 2024), we define $\sigma_{\boldsymbol{u}} := \Delta_{\boldsymbol{q}} \sqrt{2\ln(1.25/\delta_1)} \epsilon_1^{-1}$ where $\sigma_{\boldsymbol{u}}^2$ is the variance of the noise distribution in step 9 Alg. 1. Then we have the following theorem.

Theorem 1. *Let $\boldsymbol{U}_i \boldsymbol{\Gamma}_i \boldsymbol{U}_i^\top = \boldsymbol{C}_i^{1/2} \boldsymbol{C}_i'^{-1} \boldsymbol{C}_i$ and $\boldsymbol{\mu}_i = \boldsymbol{U}_i^\top \boldsymbol{C}_i^{-1/2} \left(\boldsymbol{B}_i - \boldsymbol{B}_i'\right)$. Then $\forall s > \max\left(1, \lambda_1\left(\boldsymbol{\Gamma}_i\right)\right)$ the first $T-1$ steps of Algorithm 1, for agent i are (ϵ_i, δ_i)-DP with the following bound*

$$\delta_i \leq \frac{(s-1)^{\frac{d}{2}}}{\det\left(\boldsymbol{\Gamma}_i\right)^{\frac{1}{2(s-1)}} \sqrt{\det\left(s\boldsymbol{I} - \boldsymbol{\Gamma}_i\right)}} \times \exp\left\{-\frac{\epsilon_i}{s-1} + \frac{s}{2(s-1)}\left(\boldsymbol{\mu}_i^\top \left(s\boldsymbol{I} - \boldsymbol{\Gamma}_i\right)^{-1} \boldsymbol{\Gamma}_i \boldsymbol{\mu}_i\right)\right\}. \tag{12}$$

Therefore, the entire algorithm is (ϵ, δ)−DP where $\epsilon = \max_i\left(\epsilon_i\right) + \epsilon_1$ and $\delta = \max_i\left(\delta_i\right) + \delta_1$.

However, to understand the asymptotic behavior we introduce the following corollary.

Corollary 1 (Corollary 2 of (Ramakrishna et al., 2023))**.** *Assuming that $s \gg \max\left(1, \lambda_1(\boldsymbol{\Gamma}_i)\right)$, then*

$$\delta_i \leq \exp\left\{1/2s\left(\boldsymbol{\mu}_i^\top \boldsymbol{\Gamma}_i \boldsymbol{\mu}_i - \ln\det \boldsymbol{\Gamma}_i\right)\right\} e^{-\frac{\epsilon}{s}}. \tag{13}$$

The proof for Theorem 1 and Corollary 1 can be found in (Ramakrishna et al., 2023), however, we provide an explicit formulation of the Gaussian $\boldsymbol{Z}^{(T-1)}$ in terms of the parameters, α ,$\boldsymbol{W}$, and $\boldsymbol{X}$, which can be found in Appendix A.2.1. Corollary 1 indicates that δ_i increases monotonically with respect to an increase in $\boldsymbol{\mu}_i^\top \boldsymbol{\Gamma}_i \boldsymbol{\mu}_i - \ln\det \boldsymbol{\Gamma}_i$. On the other hand, from Theorem 1 we see that the choice of (ϵ_1) and δ_1 is critical. Since we are able to roll up all the $T-1$ iterations into $\max_i(\epsilon_i, \delta_i)$, and since the last iteration a is noisy release of $\boldsymbol{q}_i^{(T)}$, a higher accuracy can be achieved by giving majority of the ϵ to ϵ_1.

3.2. Convergence

Theorem 2. *Let $\boldsymbol{X} \in \mathbb{R}^{n \times d}$ be the global data matrix and suppose α and $\boldsymbol{W}$ are chosen to satisfy Assumption 2. Let $\lambda_1 > \ldots > \lambda_n$ be the eigen values of $\boldsymbol{X}\boldsymbol{X}^\top$, $\boldsymbol{v}$ the principal eigen vector, and let $\boldsymbol{q}$ denote the principal eigen*

vector from Algorithm 1. Furthermore, for some sufficiently large constant $\kappa > 0$, let

$$\eta = \mathcal{O}\left(\frac{1}{\kappa n}\right). \tag{14}$$

Then if

$$T = \Theta\left(\frac{\lambda_1}{\lambda_1 - \lambda_2}\log\left(\frac{\kappa n}{\eta}\right)\right), \tag{15}$$

we have, with probability $1 - \kappa^{-\Omega(1)} - e^{-\Omega(n)}$ that

$$\left\|\boldsymbol{v} - \boldsymbol{q}\boldsymbol{q}^\top \boldsymbol{v}\right\| \leq \eta. \tag{16}$$

Then, let the privacy parameters ϵ, δ, where $\epsilon > 0$, $\delta \in (0,1)$, and $\delta \leq \exp\left(-\frac{\epsilon}{4}\right)$ Finally, by letting $\beta = 1 + \frac{\alpha}{1-\rho}$, we have

$$\eta = \mathcal{O}\left(\frac{\left(\max_{k,t}|\boldsymbol{q}_k^{(t)}|\beta + \rho\sigma_{\boldsymbol{q}}\right)\sqrt{nT\log(T)\log(1/\delta)}}{\epsilon(\lambda_1 - \lambda_2)}\right) \tag{17}$$

The proof of Theorem 2 is based on the works by (Nicolas et al., 2024; Balcan et al., 2016; Hardt & Price, 2014) and the full proof is in Appendix A.1. We highlight that our proof enhances the existing state of the art as it formally incorporates consensus aggregation errors and it does not require unit norm assumption on $\boldsymbol{q}^{(t)}$.

Theorem 2 indicates that an appropriate choice for α is required for a sufficiently tight convergence, however, there is a trade-off with the number of iterations T, and the choice is dependent on the error induced by the mixing matrix. The appropriate choice for α is therefore

$$\alpha \approx \frac{1}{\|\boldsymbol{X}\boldsymbol{X}^\top\| + b}, \tag{18}$$

where b is a positive constant that keeps ρ from growing too large. Additionally Theorem 2 indicates that a smaller $\sigma_{\boldsymbol{q}}$ will lead to tighter convergence, but this must be balanced with the desired T and the privacy constraints.

4. Numerical Results

For a numerical evaluation we generate a graph via a stochastic block model with 400 nodes and define $d = 3$ graph signals $\{\boldsymbol{g}[d]\}_{d=1}^3$ through the diffusion-dynamic graph filter in (Ramakrishna et al., 2020):

$$\boldsymbol{g}[d] = (\mathbf{I} + 0.1\mathbf{S})^{-1}\boldsymbol{x}[t] + \mathbf{n}[d], \forall d, \tag{19}$$

where $\mathbf{S}$ is the graph Laplacian matrix, $\boldsymbol{x}[t]$ is the excitation signal with i.i.d. entries uniformly distributed over $[-1, 1]$, and $\boldsymbol{n}[t]$ representing the Gaussian measurement noise with entries drawn from $\mathcal{N}(0, 0.01)$. We then randomly split this dataset among 4 agents, $\boldsymbol{X}_i \in \mathbb{R}^{100\times 3}$ and assume that the agent network forms a ring topology. We then run Algorithm 1 over varying ϵ. For the high privacy experiments, $\epsilon \leq 5$, we set $\delta \leq .26$ and for $5 < \epsilon \leq 10$, we set $\delta = .1$ while for all the other experiments $\delta < .011$. We compare our results to the naive LDP method. In this setup, each agent individually adds noise to its dataset and then shares the noisy matrix $\boldsymbol{X}_i$ with all other agents. Subsequently, each agent performs SVD on these noisy datasets to produce the eigenvector estimate $\boldsymbol{q}$. The results are presented in Fig 1. The experiment indicates that in the moderate privacy setting ($\epsilon \in [10, 50]$) our method on average reduces the error by .16. Concretely our method can achieve an accuracy of .8 with $\epsilon \leq 50$ while the LDP method requires $\epsilon > 100$.

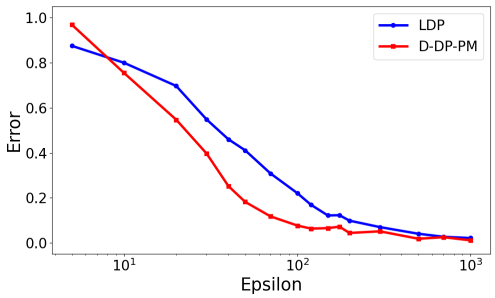

Figure 1. The error curves of the proposed D-DP-PM algorithm and the naive LDP approach. We report the projection error, $\left\|\boldsymbol{v} - \boldsymbol{q}\boldsymbol{q}^\top\boldsymbol{v}\right\|$, as a function of ϵ.

5. Conclusion

In this paper we present the D-DP-PM algorithm, the first Decentralized DP power method for estimating the sample eigen space when datasets are split row wise. Algorithm 1 leverages the fact that each agent only needs to share its local embedding of the current eigen vector $\boldsymbol{z}_i^{(t)}$, and that there is initial randomness, and thus privacy, from $\boldsymbol{q}_i^0$. Using these two facts, we analytically prove that Algorithm 1 satisfies the DP constraint, prove that our algorithm, in the multi-agent setting, converges to the true eigen vector, and provide a numerical validation on a stochastic block model.

6. Acknowledgments

This work was supported in part by the DoD-ARO under Grant W911NF2210228; and in part by the Director, Cybersecurity, Energy Security, and Emergency Response (CESER) Office of U.S. Department of Energy via the Privacy-Preserving, Collective Cyberattack Defense of DERs Project DEAC02-05CH11231.

References

Balcan, M.-F., Du, S. S., Wang, Y., and Yu, A. W. An improved gap-dependency analysis of the noisy power method. In Feldman, V., Rakhlin, A., and Shamir, O. (eds.), *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pp. 284–309. PMLR, 23–26 Jun 2016.

Balcan, M.-F., Dick, T., Liang, Y., Mou, W., and Zhang, H. Differentially private clustering in high-dimensional euclidean spaces. In *International Conference on Machine Learning*, pp. 322–331. PMLR, 2017.

Chai, D., Wang, L., Zhang, J., Yang, L., Cai, S., Chen, K., and Yang, Q. Practical lossless federated singular vector decomposition over billion-scale data. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 46–55, 2022.

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

Froelicher, D., Cho, H., Edupalli, M., Sousa, J. S., Bossuat, J.-P., Pyrgelis, A., Troncoso-Pastoriza, J. R., Berger, B., and Hubaux, J.-P. Scalable and privacy-preserving federated principal component analysis. In *2023 IEEE Symposium on Security and Privacy (SP)*, pp. 1908–1925. IEEE, 2023.

Ge, J., Wang, Z., Wang, M., and Liu, H. Minimax-optimal privacy-preserving sparse pca in distributed systems. In *International Conference on Artificial Intelligence and Statistics*, pp. 1589–1598. PMLR, 2018.

Grammenos, A., Mendoza Smith, R., Crowcroft, J., and Mascolo, C. Federated principal component analysis. *Advances in neural information processing systems*, 33: 6453–6464, 2020.

Guo, X., Li, X., Chang, X., Wang, S., and Zhang, Z. Fedpower: Privacy-preserving distributed eigenspace estimation. *arXiv preprint arXiv:2103.00704*, 2021.

Hardt, M. and Price, E. The noisy power method: A meta algorithm with applications. *Advances in neural information processing systems*, 27, 2014.

Liang, Y., Balcan, M.-F. F., Kanchanapally, V., and Woodruff, D. Improved distributed principal component analysis. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Nicolas, J., Sabater, C., Maouche, M., Mokhtar, S. B., and Coates, M. Differentially private and decentralized randomized power method. *arXiv preprint arXiv:2411.01931*, 2024.

Qu, Y., Ostrouchov, G., Samatova, N., and Geist, A. Principal component analysis for dimension reduction in massive distributed data sets. In *Proceedings of IEEE International Conference on Data Mining (ICDM)*, volume 1318, pp. 1788, 2002.

Ramakrishna, R., Wai, H.-T., and Scaglione, A. A user guide to low-pass graph signal processing and its applications: Tools and applications. *IEEE Signal Process. Mag.*, 37 (6):74–85, Nov. 2020. doi: 10.1109/MSP.2020.3014590.

Ramakrishna, R., Scaglione, A., Wu, T., Ravi, N., and Peisert, S. Differential privacy for class-based data: A practical gaussian mechanism. *IEEE Transactions on Information Forensics and Security*, 18:5096–5108, 2023.

Scaglione, A., Pagliari, R., and Krim, H. The decentralized estimation of the sample covariance. In *2008 42nd Asilomar Conference on Signals, Systems and Computers*, pp. 1722–1726. IEEE, 2008.

Wang, D. and Xu, J. Principal component analysis in the local differential privacy model. *Theoretical computer science*, 809:296–312, 2020.

Wang, S. and Chang, J. M. Differentially private principal component analysis over horizontally partitioned data. In *2018 IEEE Conference on Dependable and Secure Computing (DSC)*, pp. 1–8. IEEE, 2018.

Wang, X., Jiao, Y., Wai, H.-T., and Gu, Y. Incremental aggregated riemannian gradient method for distributed pca. In *International Conference on Artificial Intelligence and Statistics*, pp. 7492–7510. PMLR, 2023.

A. Appendix

A.1. Convergence

Recall that the consensus algorithm for an arbitrary input vector $\boldsymbol{x} \in \mathbb{R}^n$ and for a doubly stochastic mixing matrix $\boldsymbol{W}$ We have the following properties. Let $\bar{\boldsymbol{x}} := \dfrac{\mathbf{1}\mathbf{1}^\top}{n}\boldsymbol{x}$ and $\boldsymbol{e} := \boldsymbol{x} - \bar{\boldsymbol{x}}$, then by the fact that $\boldsymbol{W}^r\bar{\boldsymbol{x}} = \bar{\boldsymbol{x}}$

$$\left\|\boldsymbol{W}^r\boldsymbol{x} - \bar{\boldsymbol{x}}\right\| = \left\|\boldsymbol{W}^r\left(\bar{\boldsymbol{x}} - \boldsymbol{e}\right) - \bar{\boldsymbol{x}}\right\| = \left\|\boldsymbol{W}^r\boldsymbol{e}\right\|. \tag{20}$$

And since

$$\mathbf{1}^\top\boldsymbol{e} = \bar{\boldsymbol{x}} - \bar{\boldsymbol{x}} = 0, \tag{21}$$

we know the dominant eigen vector of $\boldsymbol{W}$ does not contribute anything so

$$\left\|\boldsymbol{W}^r\boldsymbol{e}\right\| \leq \lambda_2^r(\boldsymbol{W})\left\|\boldsymbol{x} - \bar{\boldsymbol{x}}\right\| \tag{22}$$

$$\implies \mathcal{O}\left(\left\|\boldsymbol{W}^r\boldsymbol{x} - \bar{\boldsymbol{x}}\right\|\right) = \mathcal{O}\left(\lambda_2^r(\boldsymbol{W})\right). \tag{23}$$

Let

$$\operatorname{Diag}\left(\boldsymbol{X}_i\right) := \begin{bmatrix} \boldsymbol{X}_1 & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & \boldsymbol{X}_m \end{bmatrix}. \tag{24}$$

Now recall that our update operation, over the entire network is performed by

$$\boldsymbol{q}^{(t+1)} = \alpha\operatorname{Diag}\left(\boldsymbol{X}\right)\left(m\left(\boldsymbol{W}^r \otimes \boldsymbol{I}_d\right)\operatorname{Diag}^\top\left(\boldsymbol{X}\right)\boldsymbol{q}^{(t)} + \boldsymbol{p}^{(t)}\right). \tag{25}$$

However if we let $\boldsymbol{\xi} := \left(\boldsymbol{W}^c \otimes \boldsymbol{I}_d\right)\operatorname{Diag}^\top\left(\boldsymbol{X}\right) - \left(\dfrac{\mathbf{1}\mathbf{1}^\top}{m} \otimes \boldsymbol{I}_d\right)\operatorname{Diag}^\top\left(\boldsymbol{X}\right)$, we then have

$$\boldsymbol{q}^{(t+1)} = \alpha\left(\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{q}^{(t)} + \operatorname{Diag}\left(\boldsymbol{X}\right)\left(m\boldsymbol{\xi}\boldsymbol{q}^{(t)} + \boldsymbol{p}^{(t)}\right)\right). \tag{26}$$

Notice that if we let $\hat{\boldsymbol{p}}^{(t)} := \xi\boldsymbol{q}^{(t)} + \boldsymbol{p}^{(t)}$, then

$$\boldsymbol{g}^{(t)} := \operatorname{Diag}(\boldsymbol{X})\hat{\boldsymbol{p}}^{(t)} \sim \mathcal{N}\left(\mathbf{0}, m^2\operatorname{Diag}(\boldsymbol{X})\boldsymbol{\xi}\Sigma_{\boldsymbol{q}^{(t)}}\boldsymbol{\xi}^\top\operatorname{Diag}(\boldsymbol{X}^\top) + \sigma_p^2\operatorname{Diag}\left(\boldsymbol{X}\boldsymbol{X}^\top\right)\right). \tag{27}$$

Therefore our algorithm is equivalent to the centralized power method given by

$$\boldsymbol{q}^{(t+1)} = \alpha\left(\boldsymbol{X}\boldsymbol{X}^\top\boldsymbol{q}^{(t)} + \boldsymbol{g}^{(t)}\right). \tag{28}$$

Notice that since $\boldsymbol{q}^{(0)} \sim \mathcal{N}\left(0, \sigma_{\boldsymbol{q}}^2\boldsymbol{I}_n\right)$ we can say that with probability $1 - e^{-\Omega(n)}$ that $\left\|\boldsymbol{q}^{(0)}\right\| \lesssim \sigma_{\boldsymbol{q}}\sqrt{n}$ Furthermore we know that, because $\boldsymbol{p}^{(t)} \sim \mathcal{N}(\mathbf{0}, \sigma_{\boldsymbol{p}}^2\boldsymbol{I})$ that with probability $p = 99/100$ that

$$\max_{\forall t}\left\|\boldsymbol{p}^{(t)}\right\| \leq \sigma_{\boldsymbol{p}}\sqrt{n\log(T)}. \tag{29}$$

Before continuing the analysis we define $\boldsymbol{q}^{(t)}$ in terms of $\boldsymbol{q}^{(0)}$ via

$$\boldsymbol{q}^{(t)} = \alpha^t(\boldsymbol{X}\boldsymbol{X}^\top + m\operatorname{Diag}(\boldsymbol{X})\boldsymbol{\xi})^t\boldsymbol{q}^{(0)} + \sum_{k=0}^{t-1}\alpha^{t-1-k}\left(\boldsymbol{X}\boldsymbol{X}^\top + m\operatorname{Diag}(\boldsymbol{X})\boldsymbol{\xi}\right)^{t-1-k}\alpha\operatorname{Diag}(\boldsymbol{X})\boldsymbol{p}^{(t)}. \tag{30}$$

The goal is to with, high probability, bound $\left\|\boldsymbol{q}^{(t)}\right\|$, we start by splitting the term into the homogeneous and non homogeneous part. Then we have that

$$\left\|\alpha^{(t)}\left(\boldsymbol{X}\boldsymbol{X}^\top + m\operatorname{Diag}(\boldsymbol{X})\boldsymbol{\xi}\right)^{(t)}\boldsymbol{q}^{(0)}\right\| \leq \left\|\alpha^t\left(\boldsymbol{X}\boldsymbol{X}^\top + m\operatorname{Diag}(\boldsymbol{X})\boldsymbol{\xi}\right)\right\|\left\|\boldsymbol{q}^0\right\| \tag{31}$$

$$\lesssim \left\|\alpha\left(\boldsymbol{X}\boldsymbol{X}^\top + m\operatorname{Diag}(\boldsymbol{X})\boldsymbol{\xi}\right)\right\|\sigma_{\boldsymbol{q}}\sqrt{n} \tag{32}$$

$$\text{(by Assumption 2)} = \rho\sigma_{\boldsymbol{q}}\sqrt{n}. \tag{33}$$

Now we bound the non-homogeneous term.

$$\left\| \sum_{k=0}^{t-1} \alpha^{t-1-k} \left(\boldsymbol{X}\boldsymbol{X}^\top + m \operatorname{Diag}(\boldsymbol{X})\boldsymbol{\xi} \right)^{t-1-k} \alpha \operatorname{Diag}(\boldsymbol{X})\boldsymbol{p}^{(t)} \right\| \tag{34}$$

$$\leq \sum_{k=0}^{t-1} \left\| \alpha \left(\boldsymbol{X}\boldsymbol{X}^\top + m \operatorname{Diag}(\boldsymbol{X})\boldsymbol{\xi} \right) \right\|^{t-1-k} \left\| \alpha \boldsymbol{p}^{(k)} \right\|. \tag{35}$$

By Assumption 2 we know that $\left\| \alpha \left(\boldsymbol{X}\boldsymbol{X}^\top + m \operatorname{Diag}(\boldsymbol{X})\boldsymbol{\xi} \right) \right\|^\tau$ has a geometric decay with τ. Therefore

$$\sum_{k=0}^{t-1} \left\| \alpha \left(\boldsymbol{X}\boldsymbol{X}^\top + m \operatorname{Diag}(\boldsymbol{X})\boldsymbol{\xi} \right) \right\|^k \leq \frac{1}{1 - \left\| \alpha \left(\boldsymbol{X}\boldsymbol{X}^\top + m \operatorname{Diag}(\boldsymbol{X})\boldsymbol{\xi} \right) \right\|} \leq \frac{1}{1-\rho}. \tag{36}$$

Which implies that

$$\left\| \sum_{k=0}^{t-1} \alpha^{t-1-k} \left(\boldsymbol{X}\boldsymbol{X}^\top + m \operatorname{Diag}(\boldsymbol{X})\boldsymbol{\xi} \right)^{t-1-k} \alpha \operatorname{Diag}(\boldsymbol{X})\boldsymbol{p}^{(t)} \right\| \leq \frac{\alpha}{1-\rho} \max_{k<t} \left\{ \left\| \boldsymbol{p}^{(t)} \right\| \right\} \tag{37}$$

$$\leq \frac{\alpha}{1-\rho} \sigma_{\boldsymbol{p}} \sqrt{n \log(T)}. \tag{38}$$

Combing this all together we can therefore say with probability $1 - e^{\Omega(n)} - \frac{1}{100}$

$$\max_{\forall t} \left\| \boldsymbol{q}^{(t)} \right\| \lesssim \left(\rho \sigma_{\boldsymbol{q}} + \frac{\alpha \sigma_{\boldsymbol{p}}}{1-\rho} \right) \sqrt{n \log(T)} \tag{39}$$

Therefore, and by the fact that $\boldsymbol{g}^{(t)}$ is a Gaussian vector, we can utilize (Hardt & Price, 2014) and we have that with probability $1 - e^{\Omega(n)} - \frac{1}{100}$

$$\max_{\forall t} \left\| \boldsymbol{g}^{(t)} \right\| \leq \sigma_p \sqrt{n \log(T)} + \left(\rho + \frac{\alpha \sigma_{\boldsymbol{p}}}{1-\rho} \right) \sqrt{n \log(T)} \leq \left(\sigma_{\boldsymbol{p}} \left(1 + \frac{\alpha \sigma_{\boldsymbol{p}}}{1-\rho} \right) + \rho \right) \sqrt{n \log(T)}, \tag{40}$$

which implies that, for principal eigen vector $\boldsymbol{v}$

$$\max_{\forall t} \left\| \boldsymbol{v}^\top \boldsymbol{g}^{(t)} \right\| \leq \left(\sigma_{\boldsymbol{p}} \left(1 + \frac{\alpha \sigma_{\boldsymbol{p}}}{1-\rho} \right) + \rho \sigma_{\boldsymbol{q}} \right) \sqrt{n \log(T)}. \tag{41}$$

We define the order of η, for some sufficiently large $\kappa > 0$, by

$$\eta := \mathcal{O}\left(\frac{1}{\log(\kappa n)} \right). \tag{42}$$

If we can then show that (by (Balcan et al., 2016)), $\forall t$

$$\left\| \boldsymbol{g}^{(t)} \right\| = \mathcal{O}\left(\eta(\lambda_1 - \lambda_2) \right) \tag{43}$$

and

$$\left\| \boldsymbol{v}^\top \boldsymbol{g} \right\| = \mathcal{O}\left(\eta(\lambda_1 - \lambda_2) \frac{1}{\kappa \sqrt{n}} \right), \tag{44}$$

and if we let T be defined by

$$T := \Theta\left(\frac{\lambda_1}{\lambda_1 - \lambda_2} \log\left(\frac{\kappa n}{\eta} \right) \right). \tag{45}$$

then with probability $1 - \kappa^{-\Omega(1)} - e^{-\Omega(n)}$ we have that

$$\left\| \boldsymbol{v} - \boldsymbol{q}\boldsymbol{q}^\top \boldsymbol{v} \right\| \leq \eta. \tag{46}$$

Notice that we can satisfy (43) and (44) by setting

$$\eta = \frac{\left(\sigma_{\boldsymbol{p}}\left(1+\frac{\alpha}{1-\rho}\right)+\rho\sigma_{\boldsymbol{q}}\right)\sqrt{n\log(T)}}{\lambda_1-\lambda_2}, \tag{47}$$

which therefore gives us that

$$\left\|\boldsymbol{v}-\boldsymbol{q}\boldsymbol{q}^\top\boldsymbol{v}\right\| \leq \frac{\left(\max_{k,t}|\boldsymbol{q}_k^{(t)}|\left(1+\frac{\alpha}{1-\rho}\right)+\rho\sigma_{\boldsymbol{q}}\right)\sqrt{nT\log(T)\log(1/\delta)}}{\epsilon(\lambda_1-\lambda_2)} \tag{48}$$

A.2. DP

A.2.1. CHERNOFF BOUND

Recall that $\boldsymbol{X}_i \in \mathbb{R}^{n_i\times d}$ where $\sum_{i=1}^m n_i = n$. For iteration t of Algorithm 1 every agent releases $\boldsymbol{z}_i^{(t)} \in \mathbb{R}^d = \boldsymbol{X}_i^\top \boldsymbol{q}_i^{(t)} + \boldsymbol{p}_i$ where $\boldsymbol{p}_i \sim \mathcal{N}(0, \Sigma_p)$. If we concatenate all the releases we have

$$\boldsymbol{z}^{(t)} \in \mathbb{R}^{md} := \begin{bmatrix}\boldsymbol{X}_1^\top\boldsymbol{q}_1^{(t)}\\ \vdots \\ \boldsymbol{X}_m^\top\boldsymbol{q}_m^{(t)}\end{bmatrix} = \mathrm{Diag}(\boldsymbol{X}_i^\top)\boldsymbol{q}^{(t)} \qquad \text{where } \mathrm{Diag}(\boldsymbol{X}_i^\top) := \begin{bmatrix}\boldsymbol{X}_1^\top & \dots & \boldsymbol{0}\\ \vdots & \ddots & \vdots \\ \boldsymbol{0} & \dots & \boldsymbol{X}_m^\top\end{bmatrix}. \tag{49}$$

Using the above we can rewrite $\boldsymbol{z}^{(t)}$ in terms of the mixing matrix $\boldsymbol{W}$ and the previous vector $\boldsymbol{z}^{(t)}$. Let α denote the per iteration normalization constant. Additionally for readability we define

$$\Psi^{(r)} := \mathrm{Diag}(\boldsymbol{X}_i)\left((\alpha m)\boldsymbol{W}^{(r)}\otimes \boldsymbol{I}_d\right). \tag{50}$$

Then we have that

$$\boldsymbol{z}^{(t)} = \mathrm{Diag}(\boldsymbol{X}_i^\top)\boldsymbol{q}^{(t)} + \boldsymbol{p}^{(t)} \tag{51}$$

$$\boldsymbol{q}^{(t+1)} = \Psi^{(r)}\boldsymbol{z}^{(t)}. \tag{52}$$

Furthermore we define

$$\boldsymbol{Z}^{(T)} := \begin{bmatrix}\boldsymbol{z}^{(1)}\\ \vdots \\ \boldsymbol{z}^{(T)}\end{bmatrix} \in \mathbb{R}^{Tmd}. \tag{53}$$

Now we want to describe the distribution of this stacked $\boldsymbol{Z}^t$ vector. First notice that

$$\boldsymbol{q}^{(t+1)} = \Psi^{(r)}\,\mathrm{Diag}(\boldsymbol{X}_i^\top)\boldsymbol{q}^{(t)} + \Psi^{(r)}\boldsymbol{p}^{(t)} \tag{54}$$

$$\text{and} \tag{55}$$

$$\boldsymbol{z}^{(t+1)} = \mathrm{Diag}\left(\boldsymbol{X}^\top\right)\boldsymbol{q}^{(t+1)} + \boldsymbol{p}^{(t+1)} \tag{56}$$

$$\implies \tag{57}$$

$$\boldsymbol{z}^{(t+1)} = \mathrm{Diag}\left(\boldsymbol{X}_i^\top\right)\Psi^{(r)}\,\mathrm{Diag}(\boldsymbol{X}_i^\top)\boldsymbol{q}^{(t)} + \mathrm{Diag}(\boldsymbol{X}_i^\top)\Psi^{(r)}\boldsymbol{p}^{(t)} + \boldsymbol{p}^{(t+1)}. \tag{58}$$

If we continue the recursion and write $\boldsymbol{z}^{(t)}$ in terms of $\boldsymbol{q}^{(0)}$ we have

$$\boldsymbol{z}^{(t)} = \mathrm{Diag}\left(\boldsymbol{X}_i^\top\right)\left(\Psi^{(r)}\,\mathrm{Diag}\left(\boldsymbol{X}_i^\top\right)\right)^{t-1}\boldsymbol{q}^{(0)} + \sum_{\tau=1}^{t-1}\mathrm{Diag}\left(\boldsymbol{X}_i^\top\right)\left(\Psi^{(r)}\,\mathrm{Diag}\left(\boldsymbol{X}_i^\top\right)\right)^{t-1-\tau}\Psi^{(r)}\boldsymbol{p}^{(\tau)} + \boldsymbol{p}^{(t)}. \tag{59}$$

Therefore, if we define $\boldsymbol{L} \in \mathbb{R}^{mdT \times mdT}$ as a lower triangular matrix and $\boldsymbol{M} \in \mathbb{R}^{(dmT \times n)}$ as the stacked matrix of each $\operatorname{Diag}\left(\boldsymbol{X}_i^\top\right)\left(\Psi^{(r)} \operatorname{Diag}\left(\boldsymbol{X}_i^\top\right)\right)^{t-1}$ term, we get the following equation

$$\boldsymbol{Z}^{(T)} = \boldsymbol{M}\boldsymbol{q}^{(0)} + \boldsymbol{L}\begin{bmatrix}\boldsymbol{p}^{(1)}\\ \vdots \\ \boldsymbol{p}^{(T)}\end{bmatrix}. \tag{60}$$

For clarity the $t-th$ block row of $\boldsymbol{M}$ is $\operatorname{Diag}\left(\boldsymbol{X}_i^\top\right)\left(\Psi^{(r)} \operatorname{Diag}\left(\boldsymbol{X}_i^\top\right)\right)^{t-1}$. For the $\boldsymbol{L}$ matrix we define the $t-th$ block row as

$$[\boldsymbol{L}]_{t,\tau} = \begin{cases}\operatorname{Diag}\left(\boldsymbol{X}_i^\top\right)\left(\Psi^{(r)} \operatorname{Diag}\left(\boldsymbol{X}_i^\top\right)\right)^{t-1-\tau}\Psi^{(r)}, & \text{if } 1 \le \tau < t,\\ \boldsymbol{I}_{md}, & \text{if } \tau = t,\\ 0, & \text{if } \tau > t\end{cases} \tag{61}$$

which has the following structure

$$\boldsymbol{L} = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0}\\ \operatorname{Diag}\left(\boldsymbol{X}_i^\top\right)\Psi^{(r)} & \boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0}\\ \operatorname{Diag}\left(\boldsymbol{X}_i^\top\right)\left(\Psi^{(r)} \operatorname{Diag}\left(\boldsymbol{X}_i^\top\right)\right)\Psi^{(r)} & \operatorname{Diag}\left(\boldsymbol{X}_i^\top\right)\Psi^{(r)} & \boldsymbol{I} & \cdots & \boldsymbol{0}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ \operatorname{Diag}\left(\boldsymbol{X}_i^\top\right)\left(\Psi^{(r)} \operatorname{Diag}\left(\boldsymbol{X}_i^\top\right)\right)^{T-2}\Psi^{(r)} & \operatorname{Diag}\left(\boldsymbol{X}_i^\top\right)\left(\Psi^{(r)} \operatorname{Diag}\left(\boldsymbol{X}_i^\top\right)\right)^{T-3}\Psi^{(r)} & \cdots & \operatorname{Diag}\left(\boldsymbol{X}_i^\top\right)\Psi^{(r)} & \boldsymbol{I} \end{pmatrix}. \tag{62}$$

Furthermore, we let $\boldsymbol{P} \in \mathbb{R}^{Tmd}$ be the stacked Guassian vector, then we have that

$$\boldsymbol{P} \sim \mathcal{N}\left(0, \boldsymbol{I}_T \otimes \Sigma_p\right), \tag{63}$$

which implies that

$$\boldsymbol{Z}^{(T)} | \boldsymbol{q}^0 \sim \mathcal{N}(\boldsymbol{M}\boldsymbol{q}^{(0)}, \boldsymbol{L}(\boldsymbol{I}_T \otimes \Sigma_p)\boldsymbol{L}^\top). \tag{64}$$

However, because we eventually want to look at the privacy leakage of node j w.r.t node i (node i's ability to infer information about node j) we need to separate $\boldsymbol{Z}_i^{(T)}$. We accomplish this by replacing the $\operatorname{Diag}\left(\boldsymbol{X}_i^\top\right)$ with $\boldsymbol{T}_i := \boldsymbol{X}_i^\top\left(\boldsymbol{e}_i^\top \otimes \boldsymbol{I}_{n_i}\right)$. That is, the $t-$th row of $\boldsymbol{M}$ becomes $[\boldsymbol{M}_i]_t := \boldsymbol{T}_i\left(\Psi^{(r)} \operatorname{Diag}\left(\boldsymbol{X}_i^\top\right)\right)^{t-1}$ with $\boldsymbol{M}_i \in \mathbb{R}^{dT}$.

Furthermore we replace $\boldsymbol{L}$ with $\boldsymbol{L}_i \in \mathbb{R}^{Td \times Tmd}$ where the $t-th$ row block of $\boldsymbol{L}_i$ is defined by

$$[\boldsymbol{L}_i]_{t,\tau} = \begin{cases}\boldsymbol{T}_i\left(\Psi^{(r)} \operatorname{Diag}\left(\boldsymbol{X}_i^\top\right)\right)^{t-1-\tau}\Psi^{(r)}, & \text{if } 1 \le \tau < t,\\ \boldsymbol{e}_i^\top \otimes \boldsymbol{I}_d, & \text{if } \tau = t,\\ \boldsymbol{0}, & \text{if } \tau > t\end{cases}. \tag{65}$$

Therefore the distribution at agent i for the entire duration of the algorithm is

$$\boldsymbol{Z}_i^{(T)} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{M}_i \Sigma_{\boldsymbol{q}} \boldsymbol{M}_i^\top + \boldsymbol{L}_i\left(\boldsymbol{I}_T \otimes \Sigma_{\boldsymbol{p}}\right)\boldsymbol{L}_i^\top\right). \tag{66}$$

Now we need to split this distribution into what node i knows and what node i receives from its neighbors. Let $\boldsymbol{M}_i^u$ denote the part of $\boldsymbol{M}$ that interacts with $\boldsymbol{q}_i^{(0)}$ and $\boldsymbol{M}_i^{-u}$ as the part that interacts with the rest of the $\boldsymbol{q}^{(0)}$. Formally $\boldsymbol{M}_i^u$ is the $i-th$ column vector of $\boldsymbol{M}_i$ and $\boldsymbol{M}_i^{-u}$ contains the other columns. We do a similar notation for $\boldsymbol{L}_i$ letting $\boldsymbol{L}_i^u$ denote the part interacting with each $\boldsymbol{p}_i^{(t)}$ and $\boldsymbol{L}_i^{-u}$ for the remaining $\boldsymbol{p}_{j\neq i}^{(t)}$. Formally, $\boldsymbol{L}_i^u$ is the positive integer multiples of the $i-th$ columns of $\boldsymbol{L}_i$ with $\boldsymbol{L}_i^{-u}$ containing the leftovers. Therefore we can write

$$\boldsymbol{Z}_i^{(T)} \sim \mathcal{N}\left(0, \boldsymbol{M}_i^u [\Sigma_{\boldsymbol{q}^0}]_i \left(\boldsymbol{M}_i^u\right)^\top + \boldsymbol{M}_i^{-u}[\Sigma_{\boldsymbol{q}^0}]_{j\neq i}\left(\boldsymbol{M}_i^{-u}\right)^\top + \boldsymbol{L}_i^u(\boldsymbol{I}_T \otimes [\Sigma_p]_i)\left(\boldsymbol{L}_i^{-u}\right)^\top + \boldsymbol{L}_i^{-u}(\boldsymbol{I}_T \otimes [\Sigma_p]_{j\neq i})\left(\boldsymbol{L}_i^{-u}\right)^\top\right), \tag{67}$$

and therefore

$$\boldsymbol{Z}^{(T)}|\boldsymbol{q}_i^{(0)}, \boldsymbol{p}_i^{(0)}, \dots \boldsymbol{p}_i^{(T)} \sim \mathcal{N}\left(\boldsymbol{M}_i^u \boldsymbol{q}_i^{(0)} + \boldsymbol{L}_i^u \boldsymbol{P}, \quad \boldsymbol{M}_i^{-u}[\Sigma_{\boldsymbol{q}^0}]_{j\neq i}\left(\boldsymbol{M}_i^{-u}\right)^\top + \boldsymbol{L}_i^{-u}(\boldsymbol{I}_T \otimes [\Sigma_p]_{j\neq i})\left(\boldsymbol{L}_i^{-u}\right)^\top\right). \quad (68)$$

Let $\boldsymbol{A}_i := \boldsymbol{M}_i^{-u}[\Sigma_{\boldsymbol{q}^0}]_{j\neq i}\left(\boldsymbol{M}_i^{-u}\right)^\top + \boldsymbol{L}_i^{-u}(\boldsymbol{I}_T \otimes [\Sigma_p]_{j\neq i})\left(\boldsymbol{L}_i^{-u}\right)^\top$ and let $\boldsymbol{A}_i'$ denote the covariance matrices by replacing $\boldsymbol{X}_j$ with $\boldsymbol{X}_j'$ where $\boldsymbol{X}_j$ and $\boldsymbol{X}_j'$ differ by a single data sample. Similarly we define $\boldsymbol{B}_i := \boldsymbol{M}_i^u \boldsymbol{q}_i^{(0)} + \boldsymbol{L}_i^u \boldsymbol{P}$. By applying the Chernoff bound and letting

$$\begin{aligned} \boldsymbol{U}\boldsymbol{\Gamma}\boldsymbol{U}^\top &= (\boldsymbol{A}_i)^{1/2}\left(\boldsymbol{A}_i'\right)^{-1}(\boldsymbol{A}_i)^{1/2} \\ \boldsymbol{\mu} &= \boldsymbol{U}^\top(\boldsymbol{A}_i)^{-1/2}\left((\boldsymbol{B}_i{}' - \boldsymbol{B}_i)\boldsymbol{q}_i^{(0)}\right) \end{aligned} \quad (69)$$

we have by (Ramakrishna et al., 2023)

$$\Pr\left(L_{\boldsymbol{X}\boldsymbol{X}'}(\boldsymbol{z}) > \epsilon\right) \leq \exp\left(\frac{1}{2s}\left[\boldsymbol{\mu}^\top\boldsymbol{\Gamma}\boldsymbol{\mu} - \ln|\boldsymbol{\Gamma}|\right]\right)\exp(-\epsilon/s). \quad (70)$$

To prove Theorem 1 we apply (70) with the DP-composition method from (Dwork et al., 2014).