

Byzantine-Resilient Federated Principal Subspace Estimation

Ankit Pratap Singh

Department of Electrical and Computer Engineering
Iowa State University
Ames, IA, USA
Email: sankit@iastate.edu

Namrata Vaswani

Department of Electrical and Computer Engineering
Iowa State University
Ames, IA, USA
Email: namrata@iastate.edu

Abstract—THIS PAPER IS ELIGIBLE FOR THE STUDENT PAPER AWARD. This work studies the problem of reliably estimating a subspace in a federated setting, when some nodes' outputs can be compromised by Byzantine attacks. Typically, the subspace of interest is the principal subspace of an unknown symmetric matrix. Each node has access to data that can be used to estimate this matrix and its principal subspace. This meta-problem occurs in various applications; two important examples are federated PCA, and the spectral initialization step of iterative solutions to various low-rank (LR) matrix recovery problems in federated settings. We introduce a novel solution framework called Subspace-Median to solve this problem in a provably Byzantine-resilient, communication-efficient, and private fashion.

I. INTRODUCTION

Federated learning is a setting where multiple entities/nodes collaborate to solve a machine learning problem. Each node stores its data locally and can communicate only with the central server or service provider. Nodes share summaries of their data with the central server which aggregates these and broadcasts the aggregate to all the nodes [1]. One of the challenges in this setup is adversarial attacks on the clients. In this work we consider Byzantine attacks, defined using their most general definition [2]. There has been a large amount of recent work on Byzantine-resilient learning [2]–[5], [5]–[7], [7]–[18]. Much of these come with either no guarantees or asymptotic guarantees. Almost all of these works study the gradient descent (GD) or stochastic GD algorithms. Typical solutions involve replacing the sum/mean of the gradients from the different nodes by a different robust statistic, such as geometric median (of means) [2], trimmed mean, coordinate-wise mean [4] or Krum [3].

A. Problem setting

The goal is to reliably estimate an r -dimensional subspace of \mathbb{R}^n in a federated setting, when some nodes can be Byzantine. Typically, this subspace is the span of top r eigenvectors (principal subspace) of a symmetric matrix Φ^* . Denote the $n \times r$ matrix containing basis vectors for this subspace by U^* . We assume that there are L total nodes and each node observes a data matrix, D_ℓ of size $n \times q_\ell$, that allows it to estimate this matrix as $\Phi_\ell = D_\ell D_\ell^\top / q_\ell$, followed by computing its top r eigenvectors. Some nodes may send Byzantine outputs.

This means that these nodes can create adversarial outputs in order to arbitrarily corrupt the estimates of the algorithm being implemented¹. The federated setting requires the algorithms to be communication-efficient and private. Here *private* means that the center should not have access to the nodes' raw data.

Here and below, all U matrices denote the subspaces spanned by their columns. For two $n \times r$ matrices U_1, U_2 with orthonormal columns, we use $SD_F(U_1, U_2) := \|(I - U_1 U_1^\top)U_2\|_F$ to quantify the Subspace Distance (SD) between their column spans. This is the ℓ_2 norm if the sine of the r principal angles between the two subspaces.

Assumption 1.1 (Number of Byzantine nodes). *We assume that at most τL nodes are Byzantine, with $\tau \leq 0.4$. Denote the set of good (non-Byzantine) nodes by \mathcal{J}_{good} . Equivalently, this means that $|\mathcal{J}_{good}| > (1 - \tau)L$.*

B. Contribution and Novelty

This work provably solves the Byzantine-resilient federated subspace estimation problem in a communication-efficient and private fashion. We introduce a novel approach called Subspace-Median for solving it, and provide a simple guarantee for it to work. We also explain why the most obvious solution, a geometric median based modification of the power method, does not work. Numerical experiments are used to corroborate our theoretical guarantees and discussion.

The federated subspace estimation meta-problem occurs in various applications; two important examples are federated PCA, and the spectral initialization step of iterative solutions to various low-rank (LR) matrix recovery problems in federated settings. One component that is missing in most existing work on Byzantine resilient optimization is how to develop a resilient algorithm initialization that enables the algorithm (typically gradient descent or stochastic GD) to converge to a specific minimizer (usually the signal of interest). Most works either consider strongly convex cost functions (which have only a unique minimizer), or prove convergence to a local minimizer. But, good initialization is a critical component for correctly solving a large number of structured signal and

¹A Byzantine node can observe outputs of all the nodes and knows the algorithm being implemented and its parameters. Also the various Byzantine nodes can collude.

matrix recovery problems and many mildly (e.g., element-wise) nonlinear problems. Some examples includes low rank (LR) column-wise sensing [19], LR matrix completion, robust PCA, and standard, sparse, or LR phase retrieval [20]. All iterative, e.g. GD-based or AltMin-based, solutions to these problems are initialized using a spectral initialization, which involves subspace estimation. To our best knowledge, there is no existing provable solution for Byzantine resilient federated PCA or subspace tracking. Thus, our algorithm and proof techniques are of independent interest for many of these problems as well.

In the long version of this work, we have used subspace median for Byzantine-resilient federated PCA and federated low rank (LR) column-wise sensing.

Novelty. Subspaces do not lie in a vector space whereas almost all existing work on Byzantine resiliency is in the context of gradient descent w.r.t. vector space variables. A commonly studied approach in this area is the geometric median (GM) [2]. The GM is defined using the Euclidean distance (ℓ_2 norm for vectors and Frobenius norm for matrices). But this is not a valid measure of distance between subspace basis matrices, e.g., \mathbf{U} , $-\mathbf{U}$ specify the same subspace even though $\|\mathbf{U} - (-\mathbf{U})\|_F = 2\sqrt{r} \neq 0$. Our work provides a novel approach to solve this problem. Our proposed algorithm, “Subspace-Median”, first computes the projection matrices for the subspace estimates obtained from different nodes, vectorizes them, and combines these by computing their GM. The output is then obtained by finding the subspace estimate whose vectorized projection matrix is closest to the GM.

C. Existing Work

One of the first non-asymptotic results for Byzantine attacks is [2]. This used the geometric median (GM) of means to replace the regular mean/sum of the partial gradients from each node. Under strong convexity and standard assumptions, it provided an exponentially decaying bound on the distance between the estimate at the t -th iteration and the unique global minimizer. In follow-up work [4], the authors studied the coordinate-wise mean and the trimmed-mean estimators and developed guarantees for both convex and non-convex problems. Because this work used coordinate-wise estimators, these new results needed smoothness and convexity along each dimension. Another interesting series of works [6], [8] provides non-asymptotic guarantees for Byzantine resilient stochastic GD. These last two works assume that the set of Byzantine nodes is the same for all GD iterations.

The above works considered the homogeneous data setting (the data that is i.i.d. (independent and identically distributed) across all nodes). More recent work has focused on heterogeneous distributions and proved results under upper bounds on the amount of heterogeneity [12]–[15]. Other works rely on detection methods to handle heterogeneous gradients [5], [7], [16]–[18]. These assume the existence of a trustworthy root/validation dataset at the central server and that is used for detecting the adversarial gradients.

Algorithm 1 Subspace Median

Input Subspace estimates $\hat{\mathbf{U}}_\ell$, $\ell \in [L]$.

Parameters T_{gm}

- 1: Orthonormalize: $\mathbf{U}_\ell \leftarrow QR(\hat{\mathbf{U}}_\ell)$, $\ell \in [L]$
- 2: Compute $\mathcal{P}_{\mathbf{U}_\ell} \leftarrow \mathbf{U}_\ell \mathbf{U}_\ell^\top$, $\ell \in [L]$
- 3: Compute GM: $\mathcal{P}_{gm} \leftarrow \text{approxGM}\{\mathcal{P}_{\mathbf{U}_\ell}, \ell \in [L]\}$
(Use [26, Algorithm 1] with parameter T_{gm}).
- 4: Find $\ell_{best} = \arg \min_\ell \|\mathcal{P}_{\mathbf{U}_\ell} - \mathcal{P}_{gm}\|_F$
- 5: Output $\mathbf{U}_{out} = \mathbf{U}_{\ell_{best}}$

We should mention that the federated robust Subspace Tracking work [21], [22] deal with robustness to row-wise and column-wise *sparse* outliers in the data matrix. This means only a few entries of each data vector can be corrupted. Byzantine attack on L_{byz} nodes means that $q \cdot (L_{byz}/L)$ of the data vectors may be entirely corrupted; while the rest of the $q - q(L_{byz}/L)$ data vectors are clean. The two problems are thus quite different and need different techniques. Furthermore, existing literature on federated PCA, including works by [23]–[25] does not address the susceptibility to Byzantine attacks, nor does it consider any form of malicious attacks targeting the output of an entire node.

II. RESILIENT FEDERATED SUBSPACE ESTIMATION

We develop a solution approach that relies on the geometric median (GM). The GM is one well-known approach to compute a reliable estimate of a vector-valued quantity using multiple individual estimates of it when some of these estimates may be corrupted by outliers [2], [27]. For L data vectors $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L$, this is defined as $\mathbf{z}_{gm} = \min_{\mathbf{z}} \sum_{\ell=1}^L \|\mathbf{z}_\ell - \mathbf{z}\|$. Here and below, $\|\cdot\|$ without a subscript denotes the ℓ_2 norm. The GM cannot be computed in closed form. When we say \mathbf{z}_{gm} is a $(1 + \epsilon_0)$ approximate GM we mean that $\sum_{\ell=1}^L \|\mathbf{z}_{gm} - \mathbf{z}_\ell\| \leq (1 + \epsilon_0) \min_{\mathbf{z} \in \mathbb{R}^n} \sum_{\ell=1}^L \|\mathbf{z} - \mathbf{z}_\ell\|$. There are two popular iterative solutions for computing this. The most commonly used one in practice, Weiszfeld’s algorithm [28], [29], does not come with a useful iteration complexity guarantee. The work of [26] introduced a nearly linear-time algorithm with the following guarantee. We state this below. Our theoretical result assumes that this is used.

Claim 2.2 (Theorem 1 [26]). *Consider [26, Algorithm 1]. With constant probability, $1 - c_0$ (with $c_0 < 1$ being a numerical constant), it obtains a $(1 + \epsilon_{gm})$ -approximate GM in order $T_{gm} = C \log(\frac{L}{\epsilon_{gm}})$ iterations. Its per iteration complexity is order $Ln \log^2(\frac{L}{\epsilon_{gm}})$. Thus, its total time complexity is order $Ln \log^3(\frac{L}{\epsilon_{gm}})$.*

Here and below, we reuse the letters c, C to denote different numerical constants in each use with the convention that $c < 1$ and $C \geq 1$.

In Appendix A, we explain how to analyze the (approximate) GM for robust estimation.

A. Proposed solution: Subspace Median

Notice from above that the GM is defined for quantities whose distance can be measured using the vector l_2 norm (equivalently, matrix Frobenius norm). Our solution adapts the GM to use it for subspaces by using the fact that the Frobenius norm between the projection matrices of two subspaces is another measure of subspace distance: $\|\mathcal{P}_U - \mathcal{P}_{U^*}\|_F = \sqrt{2}SD_F(U, U^*)$ [30, Lemma 2.5]. Here $\mathcal{P}_U := UU^\top$ is the projection matrix for subspace U (assumes U has orthonormal columns).

Our approach, which we refer to as “Subspace Median” proceeds as follows. Each node computes a subspace estimate, denoted \hat{U}_ℓ , and sends it to the center. If node ℓ is good, then \hat{U}_ℓ already has orthonormal columns; however if the node is Byzantine, then it is not. The center first orthonormalizes the columns of all the received \hat{U}_ℓ : $U_\ell = QR(\hat{U}_\ell)$ for all $\ell \in [L]$. It then computes the projection matrices $\mathcal{P}_{U_\ell} := U_\ell U_\ell^\top$, $\ell \in [L]$, followed by vectorizing them, computing their GM, and then converting the GM into a matrix. Denote this by \mathcal{P}_{gm} . Finally, the center finds the ℓ for which \mathcal{P}_{U_ℓ} is closest to \mathcal{P}_{gm} in Frobenius norm and outputs the corresponding U_ℓ . We summarize the complete algorithm in Algorithm 1. We can show the following for it.

Theorem 2.3 (Subspace-Median). *For a $\delta > 0$, consider Algorithm 1 with $T_{gm} = C \log(\frac{Lr}{\delta})$. Assume that Assumption 1.1 holds. If for all $\ell \in \mathcal{J}_{good}$,*

$$\Pr(SD_F(U^*, U_\ell) \leq \delta) \geq 1 - p$$

then, w.p. at least $1 - c_0 - \exp(-L\psi(0.4 - \tau, p))$,

$$SD_F(U^*, U_{out}) \leq 23\delta.$$

Here $\psi(a, b) := (1 - a) \log \frac{1-a}{1-b} + a \log \frac{a}{b}$ for $0 < a, b < 1$ is the binary KL divergence.

B. Federated subspace estimation: two obvious solutions vs Subspace Median

Let $\sigma_1^* \geq \dots \geq \sigma_n^*$ denote the singular values of Φ^* . Recall that our goal is to estimate its principal subspace U^* . We first explain why the two obvious solutions to this problem fail, and then show how Subspace Median provides an efficient solution. In a centralized setting, the obvious solution would be to compute the GM of the vectorized matrices Φ_ℓ followed by obtaining the principal subspace (r -SVD) of the GM matrix; this was studied in [27]. However, in a federated setting, this is communication inefficient because it requires each node to share an $n \times n$ matrix. For the same reason it is not private either. We refer to this as “Share Raw Data”.

For a communication-efficient and private solution, in the no-attack federated setting, one would use the distributed power method [31], [32]. A direct modification of this to deal with attacks is to use its GM based modification: at each iteration, instead of summing the $n \times r$ matrices, $\tilde{U}_\ell := (\Phi_\ell U)$ received from each node, we compute the GM of their vectorized versions. We refer to this as *Resilient Power Method* (*ResPowMeth*), and summarize it in Algorithm 2.

Algorithm 2 Resilient Power Method - a bad solution

Parameters T_{pow}, T_{gm}, ω

- 1: **Central Server** Randomly Initialize U_{rand} with i.i.d standard Gaussian entries. Set $U_0 = U_{rand}$.
- 2: **for** $t \in T_{pow}$ **do**
- 3: **Nodes** $\ell = 1, \dots, L$
- 4: Compute $\Phi_\ell U_{t-1}$
- 5: **Central Server**
- 6: $GM \leftarrow \text{approxGMthresh}\{\Phi_\ell U_{t-1}, \ell \in [L]\}$
(Use [26, Algorithm 1] with T_{gm} iterations on $\{\Phi_\ell U_{t-1}, \ell \in [L]\} \setminus \{\ell : \|\Phi_\ell U_{t-1}\| > \omega\}$)
- 7: Orthonormalize: using QR $GM \stackrel{QR}{=} \hat{U}R$
- 8: Return $U_t \leftarrow \hat{U}$
- 9: **end for**
- 10: Output $U_{out} \leftarrow U_{T_{pow}}$

However, this works with high probability (w.h.p.) only if all the Φ_ℓ 's are extremely accurate estimates of Φ^* . The reason it needs this is because it computes the GM of the node outputs $\Phi_\ell U$ at each iteration including the first one. At the first iteration, U_0 is a randomly generated matrix and thus, w.h.p., this is a bad approximation of the desired subspace $\text{span}(U^*)$. Consequently, the same is true for the column span of $\tilde{U}_\ell = \Phi_\ell U_0$'s. Thus, unless all the Φ_ℓ 's are very similar, the column spans of the different \tilde{U}_ℓ 's are not very close. As a result, the GM of their projection matrices is unable to distinguish between the good and Byzantine ones, and, there is a good chance it approximates the Byzantine one(s). This then means that the updated U is also a bad approximation of $\text{span}(U^*)$. The same idea repeats at the second iteration. Thus, with significant probability, the subspace estimates do not improve over iterations. This intuition is captured in the guarantee provided next.

Theorem 2.4 (Resilient Power Method (ResPowMeth) guarantee). *Assume that Assumption 1.1 holds. Assume also that $\sigma_r^* - \sigma_{r+1}^* \geq \Delta$ for a $\Delta > 0$. Consider ResPowMeth (Algorithm 2) with $T_{pow} = C \frac{\sigma_r^*}{\Delta} \log(\frac{n}{\epsilon})$, $T_{gm} = \log(\frac{Lnr}{\epsilon})$, and $\omega = 1.1\sigma_1^* \sqrt{r}$. If, for all $\ell \in \mathcal{J}_{good}$,*

$$\Pr\left\{\|\Phi_\ell - \Phi^*\| \leq \frac{1}{70} \min\left(\frac{\epsilon}{\sqrt{r}}, \frac{1}{2nr}\right) \Delta\right\} \geq 1 - p$$

then w.p. at least $1 - c_0 - Lp - \exp(-L\psi(0.4 - \tau, p)) - \frac{2}{\sqrt{n}}$, $SD_F(U_{out}, U^) \leq \epsilon$.*

The communication cost is $nrT_{pow} = Cnr \frac{\sigma_r^}{\Delta} \log(\frac{n}{\epsilon})$ per node. The computational cost at the center is $nrLT_{gm} \cdot T_{pow} = nrL \frac{\sigma_r^*}{\Delta} \log^3(\frac{Lnr}{\epsilon}) \log(\frac{n}{\epsilon})$. The computational cost at any node is $nq\ell rT_{pow} = nq\ell r \frac{\sigma_r^*}{\Delta} \log(\frac{n}{\epsilon})$.*

Observe that this result needs $\|\Phi_\ell - \Phi^*\| \lesssim \min(\epsilon/\sqrt{r}, 1/(nr)) \cdot \Delta$. This is a very stringent requirement, e.g., even to get an $\epsilon = 0.1$ accurate subspace estimate, we need $\|\Phi_\ell - \Phi^*\| \lesssim \Delta/nr$. On the other hand, by combining Theorem 2.3 with Davis-Kahan sin Θ theorem [33], we can

show the following for Subspace Median. This only needs $\|\Phi_\ell - \Phi^*\| \lesssim (\epsilon/\sqrt{r})\Delta$.

Theorem 2.5 (Subspace-Median guarantee). *Assume that Assumption 1.1 holds. Assume that $\sigma_r^* - \sigma_{r+1}^* \geq \Delta$ for a $\Delta > 0$. Consider Algorithm 1 with $T_{gm} = C \log(\frac{Lr}{\epsilon})$. Assume that, for all $\ell \in \mathcal{J}_{good}$,*

$$\Pr\left\{\|\Phi_\ell - \Phi^*\| \leq \frac{\epsilon}{92\sqrt{r}}\Delta\right\} \geq 1 - p.$$

Then, w.p. at least $1 - c_0 - \exp(-L\psi(0.4 - \tau, p + \frac{1}{n^{10}}))$, $SD_F(U_{out}, U^*) \leq \epsilon$.

The communication cost is nr per node. The computational cost at the center is order $n^2 L \log^3(\frac{Lr}{\epsilon})$.

If we assume that each node uses the power method with T_{pow} iterations to compute the singular vectors, then we need $T_{pow} = C \frac{\sigma_r^*}{\Delta} \log(\frac{n}{\epsilon})$ to guarantee that the above conclusion holds w.p. at least $1 - c_0 - \exp(-L\psi(0.4 - \tau, p + \frac{1}{n^{10}}))$. With this, the computational cost at any node is order $nq\ell r T_{pow}$.

C. Discussion

The ‘‘Share Raw data’’ approach has very high communication cost and is not private. ResPowMeth is communication-efficient and private, but needs a very tight bound on $\max_\ell \|\Phi_\ell - \Phi^*\|$. Subspace Median needs a much looser bound on this quantity. Moreover, it is even more communication-efficient than ResPowMeth (nodes only need to share the final U_ℓ).

In terms of computation cost, both ResPowMeth and Subspace-Median have the same cost at the nodes. But at the center, the per iteration cost is higher for Subspace Median by a factor of n/r roughly (because it needs to compute the GM of n^2 length vectors). However, this needs to be done only once. In case of ResPowMeth, the GM of nr length vectors needs to be computed $T_{pow} = C \log(n/\epsilon)$ times. In many practical applications, the nodes are power limited, and hence their computation cost, and communication cost, needs to be low. The computational cost at the center is a lesser concern.

An open question for future work is, is it possible to reduce the computational cost of the GM step of Subspace Median by trying to compute the GM of $U_\ell U_\ell^T Q$'s for some appropriately chosen $n \times r$ matrix Q . The approach needs to be carefully designed to avoid the shortcomings of ResPowMeth.

III. PROOFS

Proof of Theorem 2.3. The proof uses the GM lemma given in Lemma 1.8 Appendix A and careful linear algebra. Since $SD_F(U_\ell, U^*) = (1/\sqrt{2})\|\mathcal{P}_{U_\ell} - \mathcal{P}_{U^*}\|_F$ [30, Lemma 2.5], thus, the theorem assumption implies that $\max_{\ell \in \mathcal{J}_{good}} \|\mathcal{P}_{U_\ell} - \mathcal{P}_{U^*}\|_F \leq \sqrt{2}\delta$.

Observe that $\|\mathcal{P}_U\|_F \leq \sqrt{r}$ for any matrix U with orthonormal columns. Thus $\|\mathcal{P}_{U_\ell}\| \leq \sqrt{r}$ for all ℓ including the Byzantine ones (recall that we orthonormalize the received \hat{U}_ℓ 's using QR at the center before computing \mathcal{P}_{U_ℓ}). Hence, using GM Lemma 1.8, we have w.p. at least $1 - c_0 - \exp(-L\psi(0.4 - \tau, p))$

$$\|\mathcal{P}_{gm} - \mathcal{P}_{U^*}\|_F \leq 6\sqrt{2}\delta + 5\epsilon_{gm}\sqrt{r} \quad (1)$$

Here $\mathcal{P}_{gm} = GM\{\mathcal{P}_{U_\ell}, \ell \in [L]\}$. Thus,

$$\begin{aligned} & \max_{\ell \in \mathcal{J}_{good}} \|\mathcal{P}_{U_\ell} - \mathcal{P}_{gm}\|_F \\ & \leq \max_{\ell \in \mathcal{J}_{good}} \|\mathcal{P}_{U_\ell} - \mathcal{P}_{U^*}\|_F + \|\mathcal{P}_{gm} - \mathcal{P}_{U^*}\|_F \\ & \leq \sqrt{2}\delta + 6\sqrt{2}\delta + 5\epsilon_{gm}\sqrt{r} = 7\sqrt{2}\delta + 5\epsilon_{gm}\sqrt{r} \end{aligned}$$

w.p. at least $1 - c_0 - \exp(-L\psi(0.4 - \tau, p))$. Next we bound the SD between \mathcal{P}_{gm} and the node closest to it. This is denoted ℓ_{best} in the algorithm.

$$\begin{aligned} \|\mathcal{P}_{U_{\ell_{best}}} - \mathcal{P}_{gm}\|_F &= \min_\ell \|\mathcal{P}_{U_\ell} - \mathcal{P}_{gm}\|_F \\ &\leq \min_{\ell \in \mathcal{J}_{good}} \|\mathcal{P}_{U_\ell} - \mathcal{P}_{gm}\|_F \\ &\leq \max_{\ell \in \mathcal{J}_{good}} \|\mathcal{P}_{U_\ell} - \mathcal{P}_{gm}\|_F \\ &\leq 7\sqrt{2}\delta + 5\epsilon_{gm}\sqrt{r} \end{aligned}$$

In this we used $\mathcal{J}_{good} \subseteq [L]$ and hence the minimum value over all L is smaller than that over all $\ell \in \mathcal{J}_{good}$. We use this to bound the SD between $U_{\ell_{best}}$ and U^* .

$$\begin{aligned} & \|\mathcal{P}_{U_{\ell_{best}}} - \mathcal{P}_{U^*}\|_F \\ & \leq \|\mathcal{P}_{U_{\ell_{best}}} - GM\|_F + \|GM - \mathcal{P}_{U^*}\|_F \\ & \leq 7\sqrt{2}\delta + 5\epsilon_{gm}\sqrt{r} + 6\sqrt{2}\delta + 5\epsilon_{gm}\sqrt{r} \\ & \leq 13\sqrt{2}\delta + 10\epsilon_{gm}\sqrt{r} \end{aligned} \quad (2)$$

Set $\epsilon_{gm} = \delta\sqrt{2}/\sqrt{r}$. Thus, we have that, w.p. at least $1 - c_0 - \exp(-L\psi(0.4 - \tau, p))$,

$$\|\mathcal{P}_{U_{\ell_{best}}} - \mathcal{P}_{U^*}\|_F \leq 23\sqrt{2}\delta$$

This then implies that $SD_F(U_{out}, U^*) = SD_F(U_{\ell_{best}}, U^*) \leq 23\delta$ since $U_{out} = U_{\ell_{best}}$.

Note: It is possible that ℓ_{best} is not a good node (we cannot prove that it is). This is why the above steps are needed to bound $\|\mathcal{P}_{U_{\ell_{best}}} - \mathcal{P}_{U^*}\|_F$. \square

Proof of Theorem 2.4. Use the GM lemma for unbounded vectors, Lemma 1.9 Appendix A, to bound $\|GM\{\Phi_\ell U\}_{\ell=1}^L - \Phi^* U\|_F$ followed by interpreting ResPowMeth as an instance of the noisy power method studied in [34]. Apply [34, Corollary 1] with $G_t = \Phi^* U - GM\{\Phi_\ell U\}_{\ell=1}^L$. See Appendix B-B. \square

Proof of Theorem 2.5. The first part is a direct consequence of Theorem 2.3 and the Davis-Khan sin Θ theorem stated next [33]. This bounds the distance between the principal subspaces of two symmetric matrices. The version of Davis-Kahan sin Θ theorem [33] stated next is taken from [30, Corollary 2.8].

The second part uses guarantee for the power method from [34] and is proved in Appendix B-A.

Claim 3.6 (Davis-Kahan sin Θ theorem [33], [30]). *Let Φ^*, Φ be $n \times n$ symmetric matrices with $U^* \in \mathbb{R}^{n \times r}$, $U \in \mathbb{R}^{n \times r}$ being the matrices of top r singular/eigen vectors of Φ^*, Φ*

Attacks	Methods	rank-($r+1$)	full rank
Alternating	SubsMed	0.091(0.689)	0.348(0.680)
	ResPowMeth	0.898(0.497)	0.972(0.475)
Ones	SubsMed	0.091(0.669)	0.349(0.704)
	ResPowMeth	0.952(0.477)	0.990(0.513)
Orthogonal	SubsMed	0.091(0.672)	0.348(0.689)
	ResPowMeth	0.208(0.475)	0.366(0.500)
No Attack	PowMeth (Baseline)	0.050(0.505)	0.182(0.529)

TABLE I: $n = 1000$, $L = 3$, $L_{byz} = 1$, $r = 60$, $q = 1800$, $T_{pow} = 10$.

respectively. Let $\sigma_1^* \geq \dots \geq \sigma_n^*$ be the eigenvalues of Φ^* . If $\sigma_r^* - \sigma_{r+1}^* > 0$ and $\|\Phi - \Phi^*\| \leq \left(1 - \frac{1}{\sqrt{2}}\right)(\sigma_r^* - \sigma_{r+1}^*)$ then

$$SD_F(U, U^*) \leq \frac{2\sqrt{r}\|\Phi - \Phi^*\|}{\sigma_r^* - \sigma_{r+1}^*}$$

Suppose that, for all $\ell \in \mathcal{J}_{good}$,

$$\Pr\{\|\Phi_\ell - \Phi^*\| \leq b_0\} \geq 1 - p$$

Using Claim 3.6, if $b_0 < (1 - 1/\sqrt{2})\Delta$, this implies that, for all $\ell \in \mathcal{J}_{good}$, w.p. at least $1 - p$,

$$SD_F(U_\ell, U^*) \leq \frac{2\sqrt{r}b_0}{\Delta}$$

Using Theorem 2.3 with $\delta \equiv \frac{2\sqrt{r}b_0}{\Delta}$, this then implies that, w.p. at least $1 - c_0 - \exp(-L\psi(0.4 - \tau, p))$,

$$SD_F(U_{out}, U^*) \leq 23 \frac{2\sqrt{r}b_0}{\Delta} = 46\sqrt{r} \frac{b_0}{\Delta}$$

To get the right hand side $\leq \epsilon$ we need $b_0 \leq \frac{\epsilon}{46\sqrt{r}}\Delta$. \square

IV. SIMULATION EXPERIMENTS

All numerical experiments were performed using MATLAB on Intel(R)Xeon(R) CPU E3-1240 v5 @ 3.50GHz processor with 32.0 GB RAM.

Data generation. We generated $\Phi^* = U_{full}^* S_{full} U_{full}^{*\top}$, with U_{full}^* generated by orthogonalizing an $n \times n$ standard Gaussian matrix; S_{full} is a diagonal matrix of eigenvalues. This was generated once. The model parameters n , r , q , L , $L_{byz} = \tau L$, and diagonal entries of S_{full} are set as described below. In all experiments, we averaged over 1000 Monte Carlo runs. In each run, we sampled q data vectors from the Gaussian distribution, $\mathcal{N}(\mathbf{0}, \Phi^*)$ to form the data matrix D . This is split into L column sub-matrices, D_1, D_2, \dots, D_L with each containing $\tilde{q} = q/L$ columns. q, L are set so that q/L is an integer. Each run also generated a new U_{rand} to initialize the power methods used by the nodes in case of SubsMed and used by the center in case of ResPowMeth. Let $L_{byz} = \tau L$.

Attacks. Since federated subspace estimation problem has not been studied for Byzantine resiliency, there are no known difficult attacks for it. It is impossible to simulate the most general Byzantine attack. We focused on three types of attacks. Motivated by reverse gradient (rev) attack [35], we generated

the first one by colluding with other nodes to set $U_{corrupt}$ as a matrix in the subspace orthogonal to that spanned by $\sum_\ell \hat{U}_\ell$ at each iteration. This is generated as follows. Let $U = \sum_\ell \hat{U}_\ell$ (in case of SubsMed) and $U = \sum_\ell \Phi_\ell U_t$ (for ResPowMeth). Orthonormalize it $\tilde{U} = orth(U)$ and let $\tilde{M} = I - \tilde{U}\tilde{U}^\top$, obtain its QR decomposition $\tilde{M}^{QR} = U_{perp} R$ and set $U_{corrupt} = (\omega/\sqrt{r})U_{perp}(:, 1 : r)$. We call this *Orthogonal attack*. Since SubsMed runs all its iterations locally, this is generated once for SubsMed, but it is generated at each iteration for ResPowMeth. The second attack that we call the *ones attack* consists of an $n \times r$ matrix of -1 multiplied by a large constant C_{attack} . The third attack that we call the *Alternating attack* is an $n \times r$ matrix of alternating $+1, -1$ multiplied by a large $C_{attack} > 0$. We set $C_{attack} = 0.9\omega/\sqrt{nr}$.

Algorithm Parameters. For all GM computations, we used Weiszfeld's algorithm initialized using the average of the input data points. We set $T_{gm} = 10$, $T_{pow} = 10$.

Experiments. We report mean SD_F (mean time) in Table I. Here mean SD_F is mean over all 1000 Monte Carlo runs. In our first experiment, we let $n = 1000$, $r = 60$, $q = 1800$, $L = 3$, $L_{byz} = 1$, and we let S_{full} be a full rank diagonal matrix with first r entries set to 15, the $(r+1)$ -th entry to 1, and the others generated as $1 - (1/n), 1 - (2/n), \dots$. In our second one, we simulated an approximately low rank Σ^* by setting its first r entries set to 15, the $(r+1)$ -th entry to 1, and the other entries to zero. We report results for both these experiments in Table I. Observe that in both experiments, the error of SubsMed is much lower or lower than that of ResPowMeth. When going from full rank to approximately low rank Σ^* , the error reduces for both, but the reduction is higher for SubsMed.

REFERENCES

- [1] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al., “Advances and open problems in federated learning,” *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.
- [2] Yudong Chen, Lili Su, and Jiaming Xu, “Distributed statistical machine learning in adversarial settings: Byzantine gradient descent,” *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 2, pp. 1–25, 2017.
- [3] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer, “Machine learning with adversaries: Byzantine tolerant gradient descent,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [4] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett, “Byzantine-robust distributed learning: Towards optimal statistical rates,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 5650–5659.
- [5] Cong Xie, Sanmi Koyejo, and Indranil Gupta, “Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 6893–6901.
- [6] Dan Alistarh, Zeyuan Allen-Zhu, and Jerry Li, “Byzantine stochastic gradient descent,” *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [7] Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong, “Fltrust: Byzantine-robust federated learning via trust bootstrapping,” *arXiv preprint arXiv:2012.13995*, 2020.
- [8] Zeyuan Allen-Zhu, Faeze Ebrahimi, Jerry Li, and Dan Alistarh, “Byzantine-resilient non-convex stochastic gradient descent,” *arXiv preprint arXiv:2012.14368*, 2020.
- [9] Zhaoxian Wu, Qing Ling, Tianyi Chen, and Georgios B Giannakis, “Federated variance-reduced stochastic gradient descent with robustness to byzantine attacks,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 4583–4596, 2020.
- [10] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien, “SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives,” *Advances in neural information processing systems*, vol. 27, 2014.
- [11] Anish Acharya, Abolfazl Hashemi, Prateek Jain, Sujay Sanghavi, Inderjit S Dhillon, and Ufuk Topcu, “Robust training in high dimensions via block coordinate geometric median descent,” in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 11145–11168.
- [12] Krishna Pillutla, Sham M Kakade, and Zaid Harchaoui, “Robust aggregation for federated learning,” *arXiv preprint arXiv:1912.13445*, 2019.
- [13] Deepesh Data and Suhas Diggavi, “Byzantine-resilient SGD in high dimensions on heterogeneous data,” in *2021 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2021, pp. 2310–2315.
- [14] Liping Li, Wei Xu, Tianyi Chen, Georgios B Giannakis, and Qing Ling, “RSA: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 1544–1551.
- [15] Avishek Ghosh, Justin Hong, Dong Yin, and Kannan Ramchandran, “Robust federated learning in a heterogeneous environment,” *arXiv preprint arXiv:1906.06629*, 2019.
- [16] Jayanth Regatti, Hao Chen, and Abhishek Gupta, “Byzantine Resilience With Reputation Scores,” in *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2022, pp. 1–8.
- [17] Shiwei Lu, Ruihu Li, Xuan Chen, and Yuena Ma, “Defense against local model poisoning attacks to byzantine-robust federated learning,” *Frontiers of Computer Science*, vol. 16, no. 6, pp. 166337, 2022.
- [18] Xinyang Cao and Lifeng Lai, “Distributed gradient descent algorithm robust to an arbitrary number of byzantine attackers,” *IEEE Transactions on Signal Processing*, vol. 67, no. 22, pp. 5850–5864, 2019.
- [19] S. Nayer and N. Vaswani, “Fast and sample-efficient federated low rank matrix recovery from column-wise linear and quadratic projections,” Feb. 2023.
- [20] S. Nayer and N. Vaswani, “Sample-efficient low rank phase retrieval,” 2021.
- [21] Praneeth Narayanamurthy, Namrata Vaswani, and Aditya Ramamoorthy, “Federated over-air subspace tracking from incomplete and corrupted data,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 3906–3920, 2022.
- [22] Aref Miri Rekavandi, Abd-Krim Seghouane, Karim Abed-Meraim, et al., “Robust subspace tracking with contamination mitigation via α -divergence,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [23] Andreas Grammenos, Rodrigo Mendoza Smith, Jon Crowcroft, and Cecilia Mascolo, “Federated principal component analysis,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 6453–6464, 2020.
- [24] Santiago Silva, Boris A Gutman, Eduardo Romero, Paul M Thompson, Andre Altmann, and Marco Lorenzi, “Federated learning in distributed medical databases: Meta-analysis of large-scale subcortical brain data,” in *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)*. IEEE, 2019, pp. 270–274.
- [25] Zezhong Zhang, Guangxu Zhu, Rui Wang, Vincent KN Lau, and Kaibin Huang, “Turning channel noise into an accelerator for over-the-air principal component analysis,” *IEEE Transactions on Wireless Communications*, vol. 21, no. 10, pp. 7926–7941, 2022.
- [26] Michael B Cohen, Yin Tat Lee, Gary Miller, Jakub Pachocki, and Aaron Sidford, “Geometric median in nearly linear time,” in *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, 2016, pp. 9–21.
- [27] Stanislav Minsker, “Geometric median and robust estimation in banach spaces,” 2015.
- [28] Endre Weiszfeld, “Sur le point pour lequel la somme des distances de n points donnés est minimum,” *Tohoku Mathematical Journal, First Series*, vol. 43, pp. 355–386, 1937.
- [29] Amir Beck and Shoham Sabach, “Weiszfeld’s method: Old and new results,” *Journal of Optimization Theory and Applications*, vol. 164, no. 1, pp. 1–40, 2015.
- [30] Yuxin Chen, Yuejie Chi, Jianqing Fan, Cong Ma, et al., “Spectral methods for data science: A statistical perspective,” *Foundations and Trends textregistered in Machine Learning*, vol. 14, no. 5, pp. 566–806, 2021.
- [31] Gene H Golub and Charles F Van Loan, “Matrix computations,” *The Johns Hopkins University Press, Baltimore, USA*, 1989.
- [32] Sissi Xiaoxiao Wu, Hoi-To Wai, Lin Li, and Anna Scaglione, “A review of distributed algorithms for principal component analysis,” *Proceedings of the IEEE*, vol. 106, no. 8, pp. 1321–1340, 2018.
- [33] Yi Yu, Tengyao Wang, and Richard J Samworth, “A useful variant of the Davis–Kahan theorem for statisticians,” *Biometrika*, vol. 102, no. 2, pp. 315–323, 2015.
- [34] Moritz Hardt and Eric Price, “The noisy power method: A meta algorithm with applications,” *Advances in neural information processing systems*, vol. 27, 2014.
- [35] Shashank Rajput, Hongyi Wang, Zachary Charles, and Dimitris Papailiopoulos, “DETOX: A redundancy-based framework for faster and more robust gradient aggregation,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.

The following 2 pages are the optional appendix provided by the author(s).

It may contain further details, proofs, etc.

It will not be published in the proceedings.

Reviewers are not required to take it into account with their review, but may find it helpful.

APPENDIX A USING THE GEOMETRIC MEDIAN (GM) FOR ROBUST ESTIMATION

In robust estimation, the goal is to get a good estimate of a vector quantity \tilde{z} using L individual estimates of it, denoted z_ℓ , when most of the estimates are good, but a few can be arbitrarily corrupted or modified by Byzantine attackers. A good approach to do this is to use the GM. The following lemma, borrowed from [2], studies this.

Lemma 1.7. *Let $\mathcal{A} = \{z_1, \dots, z_L\}$ with each $z_\ell \subseteq \mathbb{R}^n$ and let z_{gm} denote their $(1 + \epsilon_{gm})$ approximate GM estimate. Fix an $\alpha \in (0, 1/2)$. Suppose that the following holds for estimates from at least $(1 - \alpha)L$ nodes: $\|z_\ell - \tilde{z}\| \leq \epsilon \|\tilde{z}\|$. Then, w.p. at least $1 - c_0$,*

$$\begin{aligned} \|z_{gm} - \tilde{z}\| &\leq C_\alpha \epsilon \|\tilde{z}\| + \epsilon_{gm} \frac{\sum_{\ell=1}^L \|z_\ell^* - z_\ell\|}{(1 - 2\alpha)L} \\ &\leq C_\alpha \epsilon \|\tilde{z}\| + \epsilon_{gm} \frac{\max_{\ell \in [L]} \|z_\ell\|}{1 - 2\alpha} \end{aligned}$$

where $C_\alpha := \frac{2(1-\alpha)}{1-2\alpha}$.

To understand this lemma simply, fix the value α to 0.4. Then $C_\alpha = 6$. We can also fix $\epsilon_{gm} = \epsilon$. Then, it says the following. If at least 60% of the L estimates are ϵ close to \tilde{z} , then, the $(1 + \epsilon)$ approximate GM, z_{gm} , is $11\epsilon \max(\|\tilde{z}\|, \max_\ell \|z_\ell\|)$ close to \tilde{z} . The next lemma follows using the above lemma and is a minor modification of [2, Lemma 3.5]. It fixes $\alpha = 0.4$ and considers the case when most estimates are good with high probability (whp).

Lemma 1.8. *Let $\mathcal{A} = \{z_1, \dots, z_L\}$ with each $z_\ell \subseteq \mathbb{R}^n$, and let z_{gm} denote a $(1 + \epsilon_{gm})$ approximate GM. For a $\tau < 0.4$, suppose that, for at least $(1 - \tau)L$ z_ℓ 's,*

$$\Pr\{\|z_\ell - \tilde{z}\| \leq \epsilon \|\tilde{z}\|\} \geq 1 - p$$

Then, w.p. at least $1 - c_0 - \exp(-L\psi(0.4 - \tau, p))$,

$$\|z_{gm} - \tilde{z}\| \leq 6\epsilon \|\tilde{z}\| + 5\epsilon_{gm} \max_{1 \leq i \leq L} \|z_i\|$$

where $\psi(a, b) = (1 - a) \log \frac{1-a}{1-b} + a \log \frac{a}{b}$.

Furthermore, if $\|z_\ell\| \leq \|\tilde{z}\|$, then $\|z_{gm} - \tilde{z}\| \leq 11\epsilon \|\tilde{z}\|$ with above probability. The number of iterations needed is $T_{gm} = C \log(\frac{L}{\epsilon_{gm}})$, and the time complexity is $O(nL \log^3(\frac{L}{\epsilon_{gm}}))$.

Suppose that, for a $\tau < 0.4$, at least $(1 - \tau)L$ z_ℓ s are “good” (are ϵ close to \tilde{z}) whp. Let $\epsilon_{gm} = \epsilon$ and suppose that all z_ℓ 's, including the corrupted ones, are bounded in 2-norm by $\|\tilde{z}\|$. Then, the $(1 + \epsilon)$ -approximate GM is about $11\epsilon \|\tilde{z}\|$ close to \tilde{z} with at least constant probability. If the GM is approximated with probability 1, i.e., if $c_0 = 0$, then, the above result says that, for p small enough and large L , the reliability of the GM is actually higher than that of the individual good estimates. For example, for a $p < 0.01$, the probability is at least $1 - p^{L(0.4-\tau)}$. The increase depends on $(0.4 - \tau)$ and L , e.g., if $\tau \geq 0.2$ and $L \geq 10$, then, the probability is at least $1 - p^{0.2L} \geq 1 - p^2$.

Using the above result for unbounded z_ℓ s. In settings where all z_ℓ 's are bounded, the above result is directly applicable. When they are not bounded, we need an extra thresholding step. Observe that, from the lemma assumption, w.p. at least $1 - Lp$, the good z_ℓ 's are bounded by $(1 + \epsilon)\|\tilde{z}\|$. Thus, to get a bounded set of z_ℓ 's while not eliminating any of the good ones, we can create a new set \mathcal{A}_{thresh} that only contains z_ℓ s with norm smaller than threshold $\omega = (1 + \epsilon)\|\tilde{z}\|$, i.e., we use $\mathcal{A}_{thresh} = \{z_1, \dots, z_L\} \setminus \{z_\ell : \|z_\ell\| > (1 + \epsilon)\|\tilde{z}\|\}$ as the input to the GM computation algorithm [26, Algorithm 1]. We have the following corollary of Lemma 1.9 for this setting.

Lemma 1.9. *Let z_{gm} denote a $(1 + \epsilon_{gm})$ approximate GM of $\{z_1, \dots, z_L\} \setminus \{z_\ell : \|z_\ell\| > \omega\}$, all vectors are in \mathbb{R}^n . Set $\omega = (1 + \epsilon)\|\tilde{z}\|$. For a $\tau < 0.4$, suppose that, for at least $(1 - \tau)L$ z_ℓ 's,*

$$\Pr\{\|z_\ell - \tilde{z}\| \leq \epsilon \|\tilde{z}\|\} \geq 1 - p$$

Then, w.p. at least $1 - c_0 - Lp - \exp(-L\psi(0.4 - \tau, p))$,

$$\|z_{gm} - \tilde{z}\| \leq 6\epsilon \|\tilde{z}\| + 5\epsilon_{gm}(1 + \epsilon)\|\tilde{z}\| < 14 \max(\epsilon, \epsilon_{gm})\|\tilde{z}\|$$

The number of iterations needed is $T_{gm} = C \log(\frac{L}{\epsilon_{gm}})$, and the time complexity is $O(nL \log^3(\frac{L}{\epsilon_{gm}}))$.

APPENDIX B RESILIENT FEDERATED SUBSPACE ESTIMATION APPENDIX A. Proof of Theorem 2.5, second part (SVD at nodes computed using power method)

This uses the guarantee for the power method summarized in Claim 2.10 Appendix B-B [34].

Suppose that, for all $\ell \in \mathcal{J}_{good}$,

$$\Pr\{\|\Phi_\ell - \Phi^*\| \leq b_0\} \geq 1 - p$$

Using Claim 3.6, if $b_0 < (1 - 1/\sqrt{2})\Delta$, this implies that, for all $\ell \in \mathcal{J}_{good}$, w.p. at least $1 - p$,

$$SD_F(U_\ell, U^*) \leq \frac{2\sqrt{r}b_0}{\Delta} \quad (3)$$

Suppose that \hat{U}_ℓ is an estimate of U_ℓ computed using the power method. Next we use Claim 2.10 to help guarantee that $SD_F(\hat{U}_\ell, U_\ell)$ is also bounded by $2\sqrt{r}b_0/\Delta$. Using Claim 2.10 with $\Phi = \Phi_\ell$, $U = U_\ell$, $G_\tau = 0$ for all τ , $\epsilon_{pow} = \frac{2b_0}{\Delta}$, and $\gamma = n^{10}$, we can conclude that if $T_{pow} > C \frac{\sigma_r(\Phi_\ell)}{\sigma_r(\Phi_\ell) - \sigma_{r+1}(\Phi_\ell)} \log(\frac{n \cdot n^{10}}{\epsilon_{pow}})$, then $SD_2(\hat{U}_\ell, U_\ell) \leq \epsilon_{pow} = \frac{2b_0}{\Delta}$ w.p. at least $1 - p - 1/n^{10}$. Here $\sigma_i = \sigma_i(\Phi_\ell)$. Using $\|\Phi_\ell - \Phi^*\| \leq b_0$ and Weyl's inequality, $\sigma_r - \sigma_{r+1} \geq \Delta - 2b_0$ and $\sigma_r < \sigma_r^* + b_0$. Thus, if

$$T_{pow} \geq C \frac{\sigma_r^* + b_0}{\Delta - 2b_0} \log(n \frac{\Delta}{b_0})$$

then

$$SD_2(\hat{U}_\ell, U_\ell) \leq \epsilon_{pow} = \frac{2b_0}{\Delta}$$

w.p. at least $1 - p - 1/n^{10}$. This then implies that $SD_F(\hat{U}_\ell, U_\ell) \leq \frac{2b_0\sqrt{r}}{\Delta}$.

Combining this bound with the Davis-Kahan bound from (3), we can conclude that, w.p. at least $1 - p - 1/n^{10}$,

$$SD_F(\hat{U}_\ell, U^*) \leq 2 \frac{2\sqrt{r}b_0}{\Delta} = 4\sqrt{r} \frac{b_0}{\Delta} \quad (4)$$

Applying Theorem 2.3 with $\delta \equiv 4\sqrt{r} \frac{b_0}{\Delta}$, this then implies that, w.p. at least $1 - \exp(-L\psi(0.4 - \tau, p + 1/n^{10}))$,

$$SD_F(U_{out}, U^*) \leq 23 \cdot 4\sqrt{r} \frac{b_0}{\Delta} = 92\sqrt{r} \frac{b_0}{\Delta} \quad (5)$$

If we want the RHS of the above to be $\leq \epsilon$, we need

$$b_0 = \frac{\epsilon}{92\sqrt{r}} \Delta$$

and we need $T_{pow} \geq C \frac{\sigma_r^* + b_0}{\Delta - 2b_0} \log(n \frac{\Delta}{b_0})$ with this choice of b_0 . By substituting for b_0 in the above expression, and upper bounding to simplify it, we get the following as one valid choice of T_{pow}

$$T_{pow} = C(1 + 6\epsilon) \frac{\sigma_r^*}{\Delta} \log(n \frac{92\sqrt{r}}{\epsilon})$$

This used $(1 + \epsilon)(1 - 2\epsilon)^{-1} < (1 + \epsilon)(1 + 4\epsilon) < 1 + 6\epsilon$ for $\epsilon < 1$. Since we are using C to include all constants, and using $\epsilon < 1$, this further simplifies to $T_{pow} = C \frac{\sigma_r^*}{\Delta} \log(\frac{n\gamma}{\epsilon})$

B. Proof of Theorem 2.4

Lemma 2.10 [34] describes the convergence behavior of power method that is perturbed by a "noise"/perturbation G_t in each iteration t .

Claim 2.10. [Noisy Power Method [34]] Let U^* ($n \times r$) denote top r singular vectors of a symmetric $n \times n$ matrix Φ^* , and let σ_i denote its i -th singular value. Consider the following algorithm (noisy PM).

- 1) Let U_{rand} be an $n \times r$ matrix with i.i.d. standard Gaussian entries. Set $U_{t=0} = U_{rand}$.
- 2) For $t = 1$ to T_{pow} do,
 - a) $\hat{U}_t \leftarrow \Phi^* U_{t-1} + G_t$
 - b) $\hat{U}_t \leftarrow QR(\hat{U}_t)$

If at every step of this algorithm, we have

$$\begin{aligned} 5\|G_t\| &\leq \epsilon_{pow}(\sigma_r^* - \sigma_{r+1}^*), \\ 5\|U^{*\top} G_t\| &\leq (\sigma_r^* - \sigma_{r+1}^*) \frac{\sqrt{r} - \sqrt{r-1}}{\gamma\sqrt{n}} \end{aligned}$$

for some fixed parameter γ and $\epsilon_{pow} < 1/2$. Then w.p. at least $1 - \gamma^{-C_1} - \exp^{-C_2 n}$, there exists a $T_{pow} \geq C \frac{\sigma_r^*}{\sigma_r^* - \sigma_{r+1}^*} \log(\frac{n\gamma}{\epsilon_{pow}})$ so that after T_{pow} steps we have that

$$\|(I - U_{T_{pow}} U_{T_{pow}}^\top) U^*\| \leq \epsilon_{pow}$$

We state below a lower bound on $\sqrt{r} - \sqrt{r-1}$ based on Bernoulli's inequality.

Fact 2.11. Writing $\sqrt{r} - \sqrt{r-1} = \sqrt{r} \left(1 - \sqrt{1 - \frac{1}{r}}\right)$ and using Bernoulli's inequality $(1+x)^x \leq 1 + xr$ for every real number $0 \leq x \leq 1$ and $r \geq -1$ we have $\frac{1}{2\sqrt{r}} < \sqrt{r} - \sqrt{r-1}$

We use Claim 2.10 with $G_t = \Phi^* U - GM\{\Phi_\ell U\}_{\ell=1}^L$ and output $U_{T_{pow}} \in \mathbb{R}^{n \times r}$. To apply it, we need $\|G_t\|$ to satisfy the two bounds given in the claim. We use Lemma 1.9 to bound it.

Suppose that, for at least $(1 - \tau)L$, Φ_ℓ 's,

$$\Pr\{\|\Phi_\ell - \Phi^*\| \leq b_0 \sigma_1^*\} \geq 1 - p$$

Since $\|U\|_F = \sqrt{r}$, this implies

$$\Pr\{\|\Phi_\ell U - \Phi^* U\|_F \leq b_0 \sqrt{r} \sigma_1^*\} \geq 1 - p.$$

We use this and apply Lemma 1.9 with $z_\ell \equiv \text{vec}(\Phi_\ell U)$ and $\tilde{z} \equiv \text{vec}(\Phi^* U)$ so that $\|\tilde{z}\| = \|\Phi^* U\|_F \leq \sigma_1^* \sqrt{r}$. Setting $\epsilon_{gm} = b_0$ and applying the lemma, we have w.p. at least $1 - c_0 - Lp - \exp(-L\psi(0.4 - \tau, p))$

$$\|G_t\| \leq \|G_t\|_F = \|GM\{\Phi_\ell U\}_{\ell=1}^L - \Phi^* U\|_F \leq 14b_0 \sqrt{r} \sigma_1^*$$

Recall that $\sigma_r^* - \sigma_{r+1}^* \geq \Delta$. We thus need $5\|G_t\| \leq \epsilon \Delta$ to hold. This will hold with high probability if $b_0 \sqrt{r} \sigma_1^* \leq \frac{\epsilon \Delta}{70}$. Using Fact 2.11 and $\gamma = \sqrt{n}$, for the second condition of Claim 2.10 to hold, we need $\|G_t\| \leq \Delta \frac{1}{10n\sqrt{r}}$. This then implies that we need $b_0 \sqrt{r} \sigma_1^* \leq \frac{\Delta}{140n\sqrt{r}}$.

$$\text{Thus we can set } b_0 = \min\left(\frac{\epsilon}{70\sqrt{r}}, \frac{1}{140nr}\right) \frac{\Delta}{\sigma_1^*}.$$

We also need $T_{pow} > C \frac{\sigma_r^*}{\sigma_r^* - \sigma_{r+1}^*} \log(\frac{n\gamma}{\epsilon})$. This holds if we set $T_{pow} = C \frac{\sigma_r^*}{\Delta} \log(\frac{n\sqrt{n}}{\epsilon})$.

Hence w.p. at least $1 - Lp - \exp(-L\psi(0.4 - \tau, p)) - 1/\sqrt{n} - e^{-C_2 n} \geq 1 - Lp - \exp(-L\psi(0.4 - \tau, p)) - 2/\sqrt{n}$

$$SD_F(U_{out}, U^*) \leq \epsilon$$