ICML 2023 #1187 Rebuttal

Anonymous Authors

March 2023

1 Generic Response

We express our gratitude to the reviewers for providing us with valuable feedback regarding the technical aspects of our submission and constructive criticism that will enhance the overall quality of the paper. We are pleased that most reviewers found the paper's narrative to be clear, the problem well-motivated, formulated, and that our proposed Flag Aggregator (FA) to be a mathematically well-defined approach for gradient aggregation during training in noisy distributed settings.

We believe that due to minor conceptual and specific novelty concerns, the overall score was not high. We have tried to address all of the specific concerns raised by the reviewers and will integrate them into the final manuscript. We hope that this will result in more enthusiastic support for our submission. Below we answer high level questions common across 1-2 reviews.

- **Why does decreasing λ towards zero in Figures 8 and 9 gives the best performance? (R2-jupZ) It seems like the regularization is hurting performance instead. Could you please elaborate on this? (R3-3Cod)**

  Answer: Our goal is to show that choosing λ even approximately can result in similar performance, that is, our estimator is robust to the choice of λ. Reviewer will see that this is simply not true for k-NN based estimators in Bulyan/Multikrum – ***choosing a slightly different value of k-NN often leads to significantly different performance*** which is not desirable in practice. So it is not that surprising that choosing too high can be detrimental to the aggregation. Now, conceptually speaking, our results with FA with high λ merely indicate that explicit pairwise distances are not needed for aggregation while all existing aggregation scheme need them explicitly!

- **In several cases, MultiKrum (Fig 5) and Bulyan (Fig 4) outperform Flag Aggregator, why is this the case? (R2-jupZ) Is the legend for Figure 4 incorrect? (R1-jQSZ) Figure 4: Seems like the Flag aggregator really suffers from small batch sizes. (R3-3Cod)**

  Answer: Yes, the legend is correct. In Figure 4, we are running the same experiment as Figure 3 but with a smaller batch size to highlight that in cases when larger batch size is a training requirement, FA outperforms existing state of the art aggregators. We believe that discrepancy in Figure 4 (Bulyan/Multikrum perform better than FA) may be related to tightness of SDP relaxation (as pointed by Reviewer 3Cod). We have plans to close this gap by studying the integrality gap and/or using careful initialization as suggested in recent works. For example, the analysis in Wang et al. [10] could be quite useful. Finally, it would be interesting to consider the relaxations solved by Bulyan and MultiKrum using techniques from [6] and [8] that have already presented few such relaxations that work in practical clustering instances. In essence, we believe that the empirical discrepancy in Figure 4/5 is just an artifact while using small batch sizes, which can be rectified in future work.

2 R1 - Reviewer jQSZ

Thank you for highlighting the mathematical benefits of using our FA aggregator – has an algebraic, and geometrical meaning that can be exploited for training purposes. Please find our answers to your technical and experiments below.

1. **Why is it reasonable to constrain $YY^T \in \mathbb{R}^{n\times n}$ to be rank one?**

   Answer: Great question! We do not constrain YY^T to be rank one, but its lifted representation $Z \in \mathbb{R}^{mn\times mn}$ to be rank one (line 199). In essence, if the optimal solution Z^* to the SDP relaxation is a rank one matrix, then by rank factorization theorem, Z^* can be written as $Z^* = \mathbf{vec}(Y^*)\mathbf{vec}(Y^*)^T$ where $\mathbf{vec}(Y^*) \in \mathbb{R}^{mn\times 1}$. So, after reshaping, we can obtain our optimal subspace estimate $Y^* \times \mathbb{R}^{m\times n}$ for aggregation purposes. We will clarify this.

2. **The optimization program for equation (2) should be presented in the preliminaries.**

   Answer: Thank you for your feedback. We will rephrase the preliminaries of section 2 with the optimization program for Flag Median to better reflect our inspiration.

3. **"FA converges to a significantly higher accuracy than MultiKrum," (line 366) is a bit misleading per Figure 5.**

   Answer: Thank you for pointing this out. Based on the current writing, we mention this with respect to the tolerance to communication loss experiment which is shown in Figure 6. In Figure 5, we fixed the number of byzantine workers, $f = 2$, and study the marginal utility of additional workers by increasing the total number of workers, $n_w = 9$ to $n_w = 15$. We are sorry for the confusion since we unintentionally referred to Figure 6 earlier in the text. We will make a correction and will also mention $f = 5$ in the caption of the current Figure 6 to clarify.

4. **Why is it fair to assume that Y should be orthogonal?**

   Answer: Yes, this assumption is without loss of generality. To see this, first note that in general, a (nondegenerate) subspace $\mathcal{S}$ of a vector space $\mathcal{V}$ is defined as a subset of $\mathcal{V}$ that is closed under linear combinations. Fortunately, in finite dimensions, we can represent $\mathcal{S}$ as a rectangular matrix M by Fundamental theorem of linear algebra. So, we simply use Y to represent the basis of this matrix M that represents the subspace $\mathcal{S}$ in our FA aggregator formulation.

5. **"Byzantine workers send random gradients." Random according to which distributions?**

   Answer: According to the uniform distribution. There are multiple sources of error in the physical setting where the distributed training is being performed. For example, DRAM memory corruptions not detected by hardware checks that mostly flip bits from 1 to 0, GPU failures such as double bit errors that are not corrected by Error Correction Code (ECC) leading to silent data corruption or program crashes, and network link or device failures that cause the parameter server to not receive part of the gradients which are all in turn modeled with replacing the affected gradients with random values drawn from a uniform distribution. [1, 7, 9]. This corresponds to the experiments for Figures 2-6.

6. **(Line 327), Which Multi-Krum experiment is using f=5 workers?**

   Answer: In the tolerance to communication loss experiment in Figure 6, we are using $f = 5$ byzantine workers, and $n_w = 15$ total number of workers. In this setting, $m = n_w - f - 2 = 8$.

3 R2 - Reviewer jupZ

1. **Why is the added regularization term in Flag Aggregator necessary?**

   Answer: Thank you for the question. Using pairwise distances explicitly in the aggregation scheme is the de-facto standard in our distributed training community, but not necessary in our optimization based Flag Aggregator formulation. We agree that our experimental results indicate that for analyzing publicly available datasets or standard benchmarks, it may not be necessary to set λ to be very high – which is consistent with our central hypothesis that the true gradient indeed lies on a subspace. In our formulation, we merely indicate how to use such pairwise distances in the Flag aggregation setup to help researchers appreciate the novelty of our formulation. From the technical standpoint, we believe that including pairwise distances in the formulation can be beneficial as it precisely specifies how they can be utilized for computational purposes: by lagrangian duality, we may be able to reformulate them as constraints, and use existing nearest neighbors based algorithm to initialize Y. We leave these implementations for future work.

2. **Where is Gr(k_i, n) defined?**

   Answer: Thank you for your suggestions on highlighting our conceptual advantage over Flag Median and defining Gr(k_i, n) at the beginning of the paper. By Grassmannian Gr(k_i, n) we mean the set of k_i-dimensional subspaces in an n-dimensional vector space. We will clarify, rephrase, and move it to Section 2.

3. **What is meant by the individual workers' gradients not being subspaces?**

   Answer: We mean that Flag Aggregator allows each worker to have gradients with different dimensions than other workers. This is possible due to some GPUs being faster than others.

4. **What are practical settings, e.g., in the case of the considered supervised image classification, where data augmentations introduce noise that requires Byzantine fault tolerance?**

   Answer: In adversarial training [3], it is known that if we add some unlabeled data to semi-supervised training, we can improve the robustness of the classifier, therefore data augmentation is good for adversarial robustness. However, forming these adversarial examples can be quite complex, and need to be carefully designed, especially in large scale instantions [11]. Our experiments show that Flag Aggregator would be a sensible approach even when some of the adversarial unlabeled data are noisy.

5. **Why not include an empirical comparison to Flag Median, which is most closely connected to the proposed method?**

   Answer: We have included them. When λ is close to zero, the estimators are equivalent.

4 R3 - Reviewer 3Cod

1. **What do you mean by chordal distances, Stiefel manifold, and Gr(k_i, n)?**

   Answer: Thank you for your feedback. We will clarify and cite references such as [4, 5] for these terms accordingly.

2. **When can the relaxation reliably find optimal solutions to the Flag Median problem? Is it robust only for small noise levels, $f = 1$, and not for larger noise levels $f = 3$? Essentially, is the relaxation tight?**

   Answer: Great question! First, regarding tightness of our formulation – to the best of our knowledge, tightness of the proposed relaxation terms of number of variables and/or dimensions, and especially the feasible set containment, is still an open problem. Recent results indicate that kronecker product based formulations may be the most robust to noise (or noise optimal) for some classes of optimization problems, for example, see Figures in [2]. We plan to investigate the tightness of our formulation for gradient aggregation purposes in the future.

3. **What is the number of good workers in Figure 3?**

   Answer: There are 15 total workers in this experiment, so Figure 3(a), (b), (c) have 14, 13, and 12 non-byzantine or good workers respectively.

4. **Why is Bulyan not reported in Figure 9? Why does decreasing λ improve performance?**

   Answer: This is a great suggestion. After comparing FA to other SOTA robust aggregators from different aspects, in Figure 9 we are specifically evaluating the scalability of FA to larger setups with more workers in real-world scenarios. We will add a comparison to Bulyan from this aspect as well. Please our response to the question **Why does decreasing λ towards zero in Figures 8 and 9 gives the best performance?** in the top.

5. **Why is $\|g_i - g_j\|_2^2$ denoted as D_{ij}? Does it mean that the denominator would be $\|g_i - g_j\|_2^4$?**

   Answer: Thank you for pointing this out. There is a typo and we should write $D_{ij}^2 = \|g_i - g_j\|_2^2$.

6. **Could you clarify whether offloading the Flag aggregator to switches is potentially better than, say, Multi-Krum and how?**

   Answer: Offloading FA to switches has great potential in improving its computational complexity because the switch would perform as a high-throughput streaming parameter server that accelerates aggregation as the gradients are being synchronized over the network. This means that FA would achieve the same performance as its current implementation but faster, i.e. better time to accuracy. Considering that FA's accuracy currently outperforms Multi-Krum's in several experiments, an offloaded FA can reach that accuracy even faster or it could reach a higher accuracy than Multi-Krum in the same amount of time.

7. **What do the authors expect to see with these data augmentations when they say "Byzantine behavior" (lines 912, 915)? Do they expect the average aggregator to collapse more than Multi-Krum, for example?**

   Answer: Yes, the goal of the experiments with data augmentation was to see how different patterns created by nonlinear augmentation routines such as Lotka Volterra and Arnold's Cat Map would introduce correlated noise in the gradients, and compare FA's performance to SOTA aggregators like Multi-Krum in removing this type of noise. Similar to what was shown for random noise, since the average is merely a linear aggregator of gradients, it should not recover the true gradient from the clean ones in some cases for the correlated noise. Multi-Krum relies on pairwise terms and is not a linear aggregator so it may not suffer as much in those cases. FA, however, as shown under various percentages of samples affected by nonlinear routines, significantly outperforms other aggregators.

5 R4 - Reviewer bzY7

1. **Could you please explain what you mean when you say, "linear combination of gradients is not robust to byzantine failures"?**

   Answer: Great question! The aggregation method in distributed SGD – the de-facto nonrobust distributed gradient aggregation schmes – computes the average of the gradients proposed by the workers, $\mathcal{A}(g_1,\ldots,g_p)=\frac{1}{p}\sum_{i=1}^{p} g_i$, which is a linear operation and has some desirable properties such as computational efficiency. However, in this case, a single byzantine worker p can suggest $pg_p-\sum_{i=1}^{p-1} g_i$ to make $\mathcal{A}(g_1,\ldots,g_p)=g_p$, with g_p being any arbitrary gradient, which is not desirable in distributed training. More generally, we mean that a **sparse** linear combination is more applicable in these settings. In our formulation, we estimate this sparsity using the subspace Y, **not** by explicitly using pairwise distances, as is done in existing implementation of aggregators. We are happy to clarify this.

2. **In the Abstract and the Introduction, what is the relationship between the data augmentation and the byzantine failure?**

   Answer: Data augmentation methods such as additive noise, random projections, and nonlinear routines similar to those introduced in section 3, introduce stochasticity in gradients. This could hamper the convergence of SGD to other attack models studied in the byzantine fault tolerance literature. Please see our answer to the "motivation" question in the beginning of our response for more details.

   Answer:

3. **How does FA solve byzantine faults?**

   Answer: We first simplify the question. Due to byzantine faults in distributed training, the gradients can have noise in them. For example, in a simplified scenario if workers $w_i, i=1,2,3$ all compute a gradient of $10=1010_2$ for a parameter, if due to a memory or network fault, one of the most significant bits flip from 1 to 0, the gradient would change to $2=0010_2$. With this, the average gradient would be $\frac{10+10+2}{3}\approx 7$ while the median would still remain 10, as desired more robust to the fault. This simple example shows that under noisy settings, there maybe better estimators than average. Our work provides a generalized notion of median to high dimensions that can be applied for gradient aggregation purposes under noise. Our proposed FA is a high-dimensional variation of the median problem and we use standard techniques from convex optimization and linear algebra to provide theoretical guarantees for it. For example, robustness is an important consequence of the mathematical fact that optimal solutions to convex optimization problems are continuous – so no sudden changes in estimated Y^* with respect to the data g_i.

4. **Did you experiment with more than one dataset/model?**

Answer: We used CIFAR10 and ResNet-18 for experiments regarding Figures 2-6. We augmented CIFAR10 based on routines described in section 3.2 and used the new dataset in byzantine workers regarding Figures 7 and 8. For the scalability experiment in Figure 9, we used augmented MNIST trained on a simple CNN with two fully-connected layers. We repeated some experiments with the Tiny ImageNet dataset trained on ResNet-18 and reported the results in appendix section C.2. We will add more clarity to the text by specifically describing the settings for each experiment.

5. **What is the motivation for your work? Why does dependent noise exist in the practical scenario? Is there any stronger evidence to support the motivation? (R4-bzY7)**

   Answer: The main motivation for our work is that data augmentation in some form such as hand chosen augmentations have become common to train large scale models in order to improve its generalization capabilities. In fact, Adversarial training seeks to do so by finding so-called "Adversarial" samples – that are close to training samples but are classified to be in a different class – and backpropagate with them as if they are true samples. However, often times this process of finding adversarial samples itself is known to be hard – see On Robustness to Adversarial Examples and Polynomial Optimization by Awasthi et al, NeurIPS 2020. So in this training paradigm of augmenting adversarial samples using outer optimization (with known failure probabilities to achieve optimality), noise in gradients is a natural phenomenon that can cause "Byzantine" Failures in the distributed context. Please see https://adversarial-ml-tutorial.org/adversarial_training/ and references within for practical non-distributed implementations for more details. So, it is safe to say that there is ample evidence of the presence of dependent noise in common training paradigms used in practice.

6. **How is FA aggregator different from competition (Bulyan and MultiKrum)?**

   Answer: Bulyan and Multikrum use k-NN based aggregators. Since these estimators are "discrete" in the sense that k is discrete, instance level robustness are not guaranteed. Our estimator comes with a well defined optimality criterion (say KKT conditions) that can be checked at an instance level. At the population level, we may follow the proof in Lemma 1 in AggregaThor to bound the "squared distance" with Y^* and obtain finer guarantees in terms of eigenvalues of the gradient matrix G, but this is not the focus of our paper and we leave this for future work. In essence, even if our performance is similar to competition (it is not in many cases from the empirical standpoint), our guarantee is still a big value addition from a practical perspective in our applications. Moreover, as R2-jupZ and R3-3Cod note, our FA method has a clear motivation and can be used even when individual workers return different number of gradients, a capability not currently available in off-the-shelf implementations of robust aggregation schemes.

7. **The constrained optimization problem of Eq (2) seems to be borrowd from [4]. So the technical contribution seems to be minor.**

   Answer: We disagree with the sentiment expressed here regarding novelty of our formulation. In numerical optimization research, it is common practice to utilize existing formulations and modify them to suit different use cases in order to communicate new findings – for example, most Integer Programming (or Combinatorial) solvers use Linear Programming relaxations to explain further technical developments. In our FA aggregator, the reviewer will notice that by Lagrangian duality our formulation further constrains the feasible set for large-scale gradient-based training purposes using pairwise terms, which is not present in the recently introduced

Flag Median estimator. This is the difference between median and aggregation – aggregation has to be suited for accelerating training purposes, and not a one time computation as in median.

References

[1] L. Bautista-Gomez, F. Zyulkyarov, O. Unsal, and S. McIntosh-Smith. Unprotected computing: A large-scale study of dram raw error rate on a supercomputer. In *SC '16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 645–655, 2016. doi: 10.1109/SC.2016.54.

[2] S. Burer and K. Park. A strengthened sdp relaxation for quadratic optimization over the stiefel manifold. *Journal of Optimization Theory and Applications*, pages 1–20, 2023.

[3] Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, and P. S. Liang. Unlabeled data improves adversarial robustness. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[4] J. H. Conway, R. H. Hardin, and N. J. Sloane. Packing lines, planes, etc.: Packings in grassmannian spaces. *Experimental mathematics*, 5(2):139–159, 1996.

[5] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. 20(2):303–353, apr 1999.

[6] P. Felzenszwalb, C. Klivans, and A. Paul. Clustering with semidefinite programming and fixed point iteration. *Journal of Machine Learning Research*, 23(190):1–23, 2022. URL `http://jmlr.org/papers/v23/21-0402.html`.

[7] P. Gill, N. Jain, and N. Nagappan. Understanding network failures in data centers: Measurement, analysis, and implications. *SIGCOMM Comput. Commun. Rev.*, 41(4):350–361, aug 2011. ISSN 0146-4833. doi: 10.1145/2043164.2018477. URL `https://doi.org/10.1145/2043164.2018477`.

[8] D. G. Mixon, S. Villar, and R. Ward. Clustering subgaussian mixtures by semidefinite programming. *Information and Inference: A Journal of the IMA*, 6(4):389–415, 03 2017. ISSN 2049-8764. doi: 10.1093/imaiai/iax001. URL `https://doi.org/10.1093/imaiai/iax001`.

[9] D. Tiwari, S. Gupta, J. Rogers, D. Maxwell, P. Rech, S. Vazhkudai, D. Oliveira, D. Londo, N. DeBardeleben, P. Navaux, L. Carro, and A. Bland. Understanding gpu errors on large-scale hpc systems and the implications for system design and operation. In *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*, pages 331–342, 2015. doi: 10.1109/HPCA.2015.7056044.

[10] P. Wang, H. Liu, A. M.-C. So, and L. Balzano. Convergence and recovery guarantees of the k-subspaces method for subspace clustering. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 22884–22918. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/wang22r.html`.

[11] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10687–10698, 2020.