We would like to thank the reviewers for their positive and helpful feedback. We adress the main questions below.

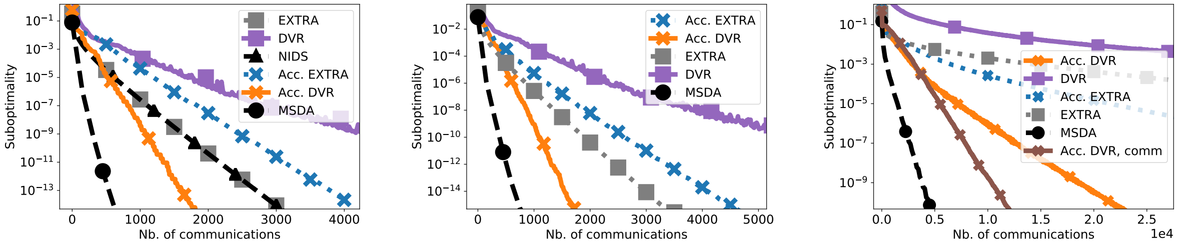
Typo: The result of Theorem 1 is on the expected norm of θ indeed, thank you for pointing this out.

Technical Challenge for the dual-free analysis: Our algorithm is inspired from [33] in the sense that we choose specific Bregman divergences (with respect to the f_{ij}^*) in order to obtain closed-form solutions of dual Bregman gradient steps as gradients of the primal functions ∇f_{ij}. Yet, [33] applies a randomized primal-dual algorithm with fixed Bregman divergences choice to a specific *primal-dual* formulation. Instead, we apply a generic randomized Bregman coordinate descent algorithm to a specific *dual* formulation (which has an augmented graph interpretation as explained in [11]). The main technical challenges that we faced were:

- Computing the relative strong convexity / smoothness constants, which now depend on the topology of the graph since our decentralized dual formulation is more complex than the formulation from [33].
- Proving convergence of Bregman Coordinate Descent in the relatively smooth setting. Although a similar algorithm is analyzed in Hanzely and Richtárik [2018], we give sharper results in the case of arbitrary sampling of blocks, and tightly adapt to the separability structure. This is crucial to our analysis since the probabilities to sample a local gradient and to communicate can be vastly different. Derivations can be found in Appendix A.
- Proving Catalyst acceleration: Theorem 4 (Appendix B) controls dual variables but we apply Catalyst to the primal variables, which thus requires both primal and dual warm-start errors, which is done in Appendix C.

Although dual-free SDCA is a dual approach in the spirit, the analysis in [33] is primal, which in particular explains why individual convexity is not needed. Yet, and although it leads to a related algorithm, our approach is different.

When is ∇f^* expensive to compute? For $x \in \mathbb{R}^d$, $\nabla f^*(x) = \arg\max_y x^\top y - f^*(y)$ so in general, computing the gradient of the conjugate is as hard as minimizing the function itself. Solving this subproblem requires inverting a $d \times d$ matrix for ridge regression, and it has no closed form solution for logistic regression. Although the dimensionality of the subproblems can be reduced in the case of stochastic algorithms for linear models, this leads to more complex implementations and increases numerical errors. In the MSDA implementation below, we obtained the dual gradients by solving each local subproblem up to precision 10^{-11} using accelerated gradient descent. Solving the subproblems with lower precision caused MSDA to plateau and not converge to the true optimum.



(a) Erdős-Rényi, $\sigma = m \cdot 10^{-5}$ (b) Grid, $\sigma = m \cdot 10^{-5}$ (c) Grid, $\sigma = m \cdot 10^{-7}$

Figure 1: Experimental results for the RCV1 dataset with different graphs of size $n = 81$, with $m = 2430$ samples per node, and with different regularization parameters.

Comparison with MSDA: Figure 1 presents the comparison between DVR and MSDA in terms of communication complexity. In Figure 1(c), *Acc. DVR, comm* (the brown line) refers to Accelerated DVR with Catalyst parameter chosen to favor communication complexity (as explained after Theorem 2). MSDA is the fastest algorithm as expected, but accelerated DVR is not too far behind, especially given the fact that it relies on generic Catalyst acceleration, which adds some complexity overhead. Therefore, the comparison with MSDA corroborates the fact that accelerated DVR is competitive with optimal methods in terms of communication while enjoying a drastically lower computational cost.

We will make sure to insist on the points discussed in this rebuttal in a revised version of the paper.

References

F. Hanzely and P. Richtárik. Fastest rates for stochastic mirror descent methods. *arXiv:1803.07374*, 2018.