

---

# Dual-Free Stochastic Decentralized Optimization with Variance Reduction

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

1 We consider the problem of training machine learning models on distributed data  
2 in a decentralized way. For finite-sum problems, fast single-machine algorithms  
3 for large datasets rely on stochastic updates combined with variance reduction. Yet,  
4 existing decentralized stochastic algorithms either do not obtain the full speedup  
5 allowed by stochastic updates, or require oracles that are more expensive than  
6 regular gradients. In this work, we introduce a Decentralized stochastic algorithm  
7 with Variance Reduction called DVR. DVR only requires computing stochastic  
8 gradients of the local functions, and is computationally as fast as a standard  
9 stochastic variance-reduced algorithms run on a  $1/n$  fraction of the dataset, where  
10  $n$  is the number of nodes. To derive DVR, we use Bregman coordinate descent  
11 on a well-chosen dual problem, and obtain a dual-free algorithm using a specific  
12 Bregman divergence. We give an accelerated version of DVR based on the Catalyst  
13 framework, and illustrate its effectiveness with simulations on real data.

## 14 1 Introduction

15 We consider the regularized empirical risk minimization problem distributed on a network of  $n$  nodes.  
16 Each node has a local dataset of size  $m$ , and the problem thus writes:

$$\min_{x \in \mathbb{R}^d} F(x) \triangleq \sum_{i=1}^n f_i(x), \text{ with } f_i(x) \triangleq \frac{\sigma_i}{2} \|x\|^2 + \sum_{j=1}^m f_{ij}(x), \quad (1)$$

17 where  $f_{ij}$  typically corresponds to the loss function for training example  $j$  of machine  $i$ , and  $\sigma_i$   
18 is the local regularization parameter for node  $i$ . We assume that each function  $f_{ij}$  is convex and  
19  $L_{ij}$ -smooth (see, e.g., [29]), and that each function  $f_i$  is  $M_i$ -smooth. Following [39], we denote  
20  $\kappa_i = (1 + \sum_{j=1}^m L_{ij})/\sigma_i$  the stochastic condition number of  $f_i$ , and  $\kappa_s = \max_i \kappa_i$ . Similarly, the  
21 batch condition number is  $\kappa_b = \max_i M_i/\sigma_i$ . It always holds that  $\kappa_b \leq \kappa_s \leq m\kappa_b$ , but generally  
22  $\kappa_s \ll m\kappa_b$ , which explains the success of stochastic methods. Indeed,  $\kappa_s \approx m\kappa_b$  when all Hessians  
23 are orthogonal to one another which is rarely the case in practice, especially for a large dataset.

24 Regarding the distributed aspect, we follow the standard *gossip* framework [5, 28, 8, 31] and assume  
25 that nodes are linked by a communication network which we represent as an undirected graph  $G$ . We  
26 denote  $\mathcal{N}(i)$  the set of neighbors of node  $i$  and  $\mathbf{1} \in \mathbb{R}^d$  the vector with all coordinates equal to 1.  
27 Communication is abstracted by multiplication by a positive semi-definite matrix  $W \in \mathbb{R}^{n \times n}$ , which  
28 is such that  $W_{k\ell} = 0$  if  $k \notin \mathcal{N}(\ell)$ , and  $\text{Ker}(W) = \text{Span}(\mathbf{1})$ . The matrix  $W$  is called the *gossip*  
29 *matrix*, and we denote its spectral gap by  $\gamma = \lambda_{\min}^+(W)/\lambda_{\max}(W)$ , the ratio between the smallest  
30 non-zero and the highest eigenvalue of  $W$ , which is a key quantity in decentralized optimization.  
31 We finally assume that nodes can compute a local stochastic gradient  $\nabla f_{ij}$  in time 1, and that  
32 communication (i.e., multiplication by  $W$ ) takes time  $\tau$ .

33 **Single-machine stochastic methods.** Problem (1) is generally solved using first-order methods.  
 34 When  $m$  is large, computing  $\nabla F$  becomes very expensive, and batch methods require  $O(\kappa_b \log(\varepsilon^{-1}))$   
 35 iterations, which takes time  $O(m\kappa_b \log(\varepsilon^{-1}))$ , to minimize  $F$  up to precision  $\varepsilon$ . In this case, updates  
 36 using the stochastic gradients  $\nabla f_{ij}$ , where  $(i, j)$  is selected randomly, can be much more effective [4].  
 37 Yet, these updates are noisy and plain stochastic gradient descent (SGD) does not converge to the  
 38 exact solution unless the step-size goes to zero, which slows down the algorithm. One way to fix this  
 39 problem is to use variance-reduced methods such as SAG [32], SDCA [34], SVRG [15] or SAGA [7].  
 40 These methods require  $O((nm + \kappa_s) \log(\varepsilon^{-1}))$  stochastic gradient evaluations, which can be much  
 41 smaller than  $O(m\kappa_b \log(\varepsilon^{-1}))$ .

42 **Decentralized methods.** Decentralized adaptations of gradient descent in the smooth and strongly  
 43 convex setting include EXTRA [36], DIGing [27] or NIDS [20]. These algorithms have sparked a  
 44 lot of interest, and the latest convergence results [13, 41, 19] show that EXTRA and NIDS require  
 45 time  $O((\kappa_b + \gamma^{-1})(m + \tau) \log(\varepsilon^{-1}))$  to reach precision  $\varepsilon$ . A generic acceleration of EXTRA  
 46 using Catalyst [19] obtains the (batch) optimal  $O(\sqrt{\kappa_b}(1 + \tau/\sqrt{\gamma}) \log(\varepsilon^{-1}))$  rate up to log factors.  
 47 Another line of work on decentralized algorithms is based on the *penalty method* [18, 9]. This  
 48 consists in performing traditional optimization algorithms to problems augmented with a Laplacian  
 49 penalty, and in particular enables the use of accelerated methods. Yet, these algorithms are sensitive  
 50 to the value of the penalty parameter (when it is fixed), since it directly influences the solution they  
 51 converge to. Another natural way to construct decentralized optimization algorithms is through dual  
 52 approaches [31, 37]. Although the dual approach leads to algorithms that are optimal both in terms of  
 53 number of communications and computations [30, 12], they generally assume access to the proximal  
 54 operator or the gradient of the Fenchel conjugate of the local functions, which is not very practical in  
 55 general since it requires solving a subproblem at each step.

56 **Decentralized stochastic optimization.** Although both stochastic and decentralized methods have  
 57 a rich literature, there exist few decentralized stochastic methods with linear convergence rate.  
 58 Although DSA [26], or GT-SAGA [40] propose such algorithms, they respectively take time  
 59  $O((m\kappa_s + \kappa_s^4\gamma^{-1}(1 + \tau) \log(\varepsilon^{-1})))$  and  $O((m + \kappa_s^2\gamma^{-2})(1 + \tau) \log(\varepsilon^{-1}))$  to reach precision  $\varepsilon$ .  
 60 Therefore, they have significantly worse rates than decentralized batch methods when  $m = 1$ ,  
 61 and than single-machine stochastic methods when  $n = 1$ . Other methods have better rates of  
 62 convergence [35, 11] but they require evaluation of proximal operators, which may be expensive.

63 **Our contributions.** This work develops a dual approach similar to that of [11], which leads to a  
 64 decentralized stochastic algorithm with rate  $O(m + \kappa_s + \tau\kappa_b/\sqrt{\gamma})$ , where the  $\sqrt{\gamma}$  factor comes from  
 65 Chebyshev acceleration, such as used in [31]. Yet, our algorithm, called DVR, can be formulated in  
 66 the primal only, thus avoiding the need for computing expensive dual gradients or proximal operators.  
 67 Besides, DVR is derived by applying Bregman coordinate descent to the dual of a specific augmented  
 68 problem. Thus, its convergence follows from the convergence of block coordinate descent with  
 69 Bregman gradients, which we prove as a side contribution. When executed on a single-machine, DVR  
 70 is similar to dual-free SDCA [33], and obtains similar rates. We believe that the same methodology  
 71 could be applied to tackle non-convex problems, but we leave these extensions for future work.

72 We present in Section 2 the derivations leading to DVR, namely the dual approach and the dual-free  
 73 trick. Then, Section 3 presents the actual algorithm along with a convergence theorem based on  
 74 block Bregman coordinate descent (presented in Appendix A). Section 4 shows how to accelerate  
 75 DVR, both in terms of network dependence (Chebyshev acceleration) and global iteration complexity  
 76 (Catalyst acceleration [22]). Finally, experiments on real-world data are presented in Section 5, that  
 77 demonstrate the effectiveness of DVR.

## 78 2 Algorithm Design

79 This section presents the key steps leading to DVR. We start by introducing a relevant dual formulation  
 80 from [11], then introduce the dual-free trick based on [16], and finally show how this leads to DVR, an  
 81 actual implementable decentralized stochastic algorithm, as a special case of the previous derivations.

### 82 2.1 Dual formulation

83 The standard dual formulation of Problem (1) is obtained by associating a parameter vector to each  
 84 node, and imposing that two neighboring nodes have the same parameters [6, 14, 31]. This leads to

the following constrained problem, in which we write  $\theta^{(i)} \in \mathbb{R}^d$  the local vector of node  $i$ :

$$\min_{\theta \in \mathbb{R}^{nd}} \sum_{i=1}^n f_i(\theta^{(i)}) \text{ such that } \forall k, \ell \in \mathcal{N}(k), \theta^{(k)} = \theta^{(\ell)}. \quad (2)$$

Following the approach of [11, 12], we further split the  $f_i(\theta^{(i)})$  term into  $\sigma_i \|\theta^{(i)}\|^2/2 + \sum_{j=1}^n f_{ij}(\theta^{(ij)})$ , with the constraint that  $\theta^{(i)} = \theta^{(ij)}$  for all  $j$ . This is equivalent to the previous approach performed on an augmented graph [11, 12] in which each node is split into a star network with the regularization in the center and a local summand at each tip of the star. Thus, the equivalent augmented constrained problem that we consider writes:

$$\min_{\theta \in \mathbb{R}^{n(m+1)d}} \sum_{i=1}^n \left[ \frac{\sigma_i}{2} \|\theta^{(i)}\|^2 + \sum_{j=1}^m f_{ij}(\theta^{(ij)}) \right] \text{ s.t. } \forall k, \ell \in \mathcal{N}(k), \theta^{(k)} = \theta^{(\ell)} \text{ and } \forall i, j, \theta^{(i)} = \theta^{(ij)}. \quad (3)$$

We now use Lagrangian duality, and introduce two kinds of multipliers. The variable  $x$  corresponds to multipliers associated with the constraints given by edges of the communication graph (i.e.,  $\theta^{(k)} = \theta^{(\ell)}$  if  $k \in \mathcal{N}(\ell)$ ), that we will call *communication edges*. Similarly,  $y$  corresponds to the constraints associated with the edges that are specific to the augmented graph (i.e.,  $\theta^{(i)} = \theta^{(ij)} \forall i, j$ ) that we call *computation* or *virtual edges*, since they are not present in the original graph and were constructed for the augmented problem. Therefore, there are  $E$  communication edges (number of edges in the initial graph), and  $nm$  virtual edges. The dual formulation of Problem (3) thus writes:

$$\min_{x \in \mathbb{R}^{Ed}, y \in \mathbb{R}^{nmd}} \frac{1}{2} q_A(x, y) + \sum_{i=1}^n \sum_{j=1}^m f_{ij}^*((A(x, y))^{(ij)}), \text{ with } q_A(x, y) \triangleq (x, y)^\top A^\top \Sigma A (x, y), \quad (4)$$

and where  $(x, y) \in \mathbb{R}^{(E+nm)d}$  is the concatenation of vectors  $x \in \mathbb{R}^{Ed}$ , which is associated with the communication edges, and  $y \in \mathbb{R}^{nmd}$ , which is the vector associated with computation edges. We denote  $\Sigma = \text{Diag}(\sigma_1^{-1}, \dots, \sigma_n^{-1}, 0, \dots, 0) \otimes I_d \in \mathbb{R}^{n(m+1)d \times n(m+1)d}$  and  $A$  is such that for all  $z \in \mathbb{R}^d$ ,  $A(e_{k,\ell} \otimes z) = \mu_{k\ell}(u_k - u_\ell) \otimes P_{k\ell}z$  for edge  $(k, \ell)$ , where  $P_{k\ell} = I_d$  if  $(k, \ell)$  is a communication edge,  $P_{ij}$  is the projector on  $\text{Ker}(f_{ij})^\perp \triangleq (\cap_{x \in \mathbb{R}^d} \text{Ker}(\nabla^2 f_{ij}(x)))^\perp$  if  $(i, j)$  is a virtual edge,  $z_1 \otimes z_2$  is the Kronecker product of vectors  $z_1$  and  $z_2$ , and  $e_{k,\ell} \in \mathbb{R}^{E+nm}$  and  $u_k \in \mathbb{R}^{n(m+1)}$  are the unit vectors associated with edge  $(k, \ell)$  and node  $k$  respectively.

Note that the upper left  $nd \times nd$  block of  $AA^\top$  (corresponding to the communication edges) is equal to  $W \otimes I_d$  where  $W$  is a gossip matrix (see, e.g., [31]) that depends on the  $\mu_{k\ell}$ . In particular,  $W$  is equal to the Laplacian of the communication graph if  $\mu_{k\ell}^2 = 1/2$  for all  $(k, \ell)$ . For computation edges, the projectors  $P_{ij}$  account for the fact that the parameters  $\theta^{(i)}$  and  $\theta^{(ij)}$  only need to be equal on the subspaces on which  $f_{ij}$  is not constant, and we choose  $\mu_{ij}$  such that  $\mu_{ij}^2 = \alpha L_{ij}$  for some  $\alpha > 0$ . Although this introduces heavier notations, explicitly writing  $A$  as an  $n(1+m)d \times (E+nm)d$  matrix instead of an  $n(1+m) \times (E+nm)$  matrix allows to introduce the projectors  $P_{ij}$ , which then yields a better communication complexity than choosing  $P_{ij} = I_d$ . See [11, 12] for more details on this dual formulation, and in particular on the construction on the augmented graph. Now that we have obtained a suitable dual problem, we would like to solve it without computing gradients or proximal operators of  $f_{ij}^*$ , which can be very expensive.

## 2.2 Dual-free trick

Dual methods are based on variants of Problem (4), and apply different algorithms to it. In particular, [31, 37] use accelerated gradient descent [29], and [10, 11] use accelerated (proximal) coordinate descent [23]. Let  $p_{\text{comm}}$  denote the probability of performing a communication step and  $p_{ij}$  be the probability that node  $i$  samples a gradient of  $f_{ij}$ , which are such that for all  $i$ ,  $\sum_{j=1}^m p_{ij} = 1 - p_{\text{comm}}$ . Applying a coordinate update with step-size  $\eta/p_{\text{comm}}$  to Problem (4) in the direction  $x$  (associated with communication edges) writes:

$$x_{t+1} = x_t - \eta p_{\text{comm}}^{-1} \nabla_x q_A(x_t, y_t), \quad (5)$$

where we denote  $\nabla_x$  the gradient in coordinates that correspond to  $x$  (communication edges), and  $\nabla_{y,ij}$  the gradient for coordinate  $(ij)$  (computation edge). Similarly, the standard coordinate update

of a local computation edge  $(i, j)$  can be written as:

$$y_{t+1}^{(ij)} = \arg \min_{y \in \mathbb{R}^d} \left\{ \left( \nabla_{y,ij} q_A(x_t, y_t) + \mu_{ij} \nabla f_{ij}^*(\mu_{ij} y_t^{(ij)}) \right)^\top y + \frac{p_{ij}}{2\eta} \|y - y_t^{(ij)}\|^2 \right\}, \quad (6)$$

where the minimization problem actually has a closed form solution. Yet, as mentioned before, solving Equation (6) requires computing the derivative of  $f_{ij}^*$ . In order to avoid this, a trick introduced by [16] and later used in [38] is to replace the Euclidean distance term by a well-chosen Bregman divergence. More specifically, the Bregman divergence of a convex function  $\phi$  is defined as:

$$D_\phi(x, y) = \phi(x) - \phi(y) - \nabla \phi(y)^\top (x - y). \quad (7)$$

Bregman gradient algorithms typically enjoy the same kind of guarantees as standard gradient algorithms, but with slightly different notions of *relative* smoothness and strong convexity [1, 24]. Note that the Bregman divergence of the squared Euclidean norm is the squared Euclidean distance, and the standard gradient descent algorithm is recovered in that case. We now replace the Euclidean distance by the Bregman divergence induced by function  $\phi : y \mapsto (L_{ij}/\mu_{ij}^2) f_{ij}^*(\mu_{ij} y^{(ij)})$ , which is normalized to be 1-strongly convex since  $f_{ij}^*$  is  $L_{ij}^{-1}$ -strongly convex. We introduce the constant  $\alpha > 0$  such that  $\mu_{ij}^2 = \alpha L_{ij}$  for all computation edges  $(i, j)$ . Using the definition of the Bregman divergence with respect to  $\phi$ , we write:

$$\begin{aligned} y_{t+1}^{(ij)} &= \arg \min_{y \in \mathbb{R}^d} \left( \nabla_{y,ij} q_A(x_t, y_t) + \mu_{ij} \nabla f_{ij}^*(\mu_{ij} y_t^{(ij)}) \right)^\top y + \frac{p_{ij}}{\eta} D_\phi(y, y_t^{(ij)}) \\ &= \arg \min_{y \in \mathbb{R}} \left( \frac{\alpha\eta}{p_{ij}} \nabla_{y,ij} q_A(x_t, y_t) - \left( 1 - \frac{\alpha\eta}{p_{ij}} \right) \mu_{ij} \nabla f_{ij}^*(\mu_{ij} y_t^{(ij)}) \right)^\top y + f_{ij}^*(\mu_{ij} y) \\ &= \frac{1}{\mu_{ij}} \nabla f_{ij} \left( \left( 1 - \frac{\alpha\eta}{p_{ij}} \right) \nabla f_{ij}^*(\mu_{ij} y_t^{(ij)}) - \frac{\alpha\eta}{\mu_{ij} p_{ij}} \nabla_{y,ij} q_A(x_t, y_t) \right). \end{aligned}$$

In particular, if we know  $\nabla f_{ij}^*(\mu_{ij} y_t^{(ij)})$  then it is possible to compute  $y_{t+1}^{(ij)}$ . Besides,

$$\nabla f_{ij}^*(\mu_{ij} y_{t+1}^{(ij)}) = (1 - \alpha\eta) \nabla f_{ij}^*(\mu_{ij} y_t^{(ij)}) - \frac{\alpha\eta}{\mu_{ij}} \nabla_{y,ij} q_A(x_t, y_t), \quad (8)$$

so we can also compute  $\nabla f_{ij}^*(\mu_{ij} y_{t+1}^{(ij)})$ , and we can use it for the next step. Therefore, instead of computing a dual gradient at each step, we can simply choose  $y_0^{(i)} = \mu_{ij}^{-1} \nabla f_{ij}(z_0^{(ij)})$  for any  $z_0^{(ij)}$ , and iterate from this. Therefore, the Bregman coordinate update applied to Problem (4) in the block of direction  $(i, j)$  with  $y_0^{(ij)} = \mu_{ij}^{-1} \nabla f_i(z_0^{(ij)})$  yields:

$$z_{t+1}^{(ij)} = \left( 1 - \frac{\alpha\eta}{p_{ij}} \right) z_t^{(ij)} - \frac{\alpha\eta}{p_{ij} \mu_{ij}} \nabla_{y,ij} q_A(x_t, y_t), \quad y_{t+1}^{(ij)} = \mu_{ij}^{-1} \nabla f_i(z_{t+1}^{(ij)}). \quad (9)$$

The iterations of (9) are called a *dual-free* algorithm because they are a transformation of the iterations from (6) that do not require computing  $\nabla f_{ij}^*$  anymore. This is obtained by replacing the Euclidean distance in (6) by the Bregman divergence of a function proportional to  $f_{ij}^*$ .

### 2.3 Distributed implementation

Iterations from (9) do not involve functions  $f_{ij}^*$  anymore, which was our first goal. Yet, they consist in updating dual variables associated with edges of the augmented graph, and have no clear distributed meaning yet. In this section, we rewrite the updates of (9) in order to have an easy to implement distributed algorithm. The key steps are (i) multiplication of the updates by  $A$ , (ii) expliciting the gossip matrix and (iii) remarking that  $\theta_t^{(i)} = (\Sigma A(x_t, y_t))^{(i)}$  converges to the primal solution for all  $i$ . For a vector  $z \in \mathbb{R}^{(n+nm)d}$ , we denote  $[z]_{\text{comm}} \in \mathbb{R}^{nd}$  its restriction to the communication nodes, and  $[M]_{\text{comm}} \in \mathbb{R}^{nd \times nd}$  similarly refers to the restriction on communication edges of a matrix  $M \in \mathbb{R}^{(n+nm)d \times (n+nm)d}$ . By abuse of notations, we call  $A_{\text{comm}} \in \mathbb{R}^{nd \times Ed}$  the restriction of  $A$  to communication nodes and edges. We denote  $P_{\text{comm}}$  the projector on communication edges, and  $P_{\text{comp}}$  the projector on  $y$ . We multiply the  $x$  (communication) update in (9) by  $A$  on the left (which is standard [31, 11]) and obtain:

$$A_{\text{comm}} x_{t+1} = A_{\text{comm}} x_t - \eta p_{\text{comm}}^{-1} [A P_{\text{comm}} A^\top]_{\text{comm}} [\Sigma A(x_t, y_t)]_{\text{comm}}. \quad (10)$$

158 Note that  $[P_{\text{comm}}A^\top \Sigma A(x_t, y_t)]_{\text{comm}} = [P_{\text{comm}}A^\top]_{\text{comm}}[\Sigma A(x_t, y_t)]_{\text{comm}}$  because  $P_{\text{comm}}$  and  $\Sigma$   
 159 are non-zero only for communication edges and nodes. Similarly, and as previously stated, one  
 160 can verify that  $A_{\text{comm}}[P_{\text{comm}}A^\top]_{\text{comm}} = [AP_{\text{comm}}A^\top]_{\text{comm}} = W \otimes I_d \in \mathbb{R}^{nd \times nd}$  where  $W$  is a  
 161 gossip matrix. We finally introduce  $\tilde{x}_t \in \mathbb{R}^{nd}$  which is a variable associated with nodes, and which is  
 162 such that  $\tilde{x}_t = A_{\text{comm}}x_t$ . With this rewriting, the communication update becomes:

$$\tilde{x}_{t+1} = \tilde{x}_t - \eta p_{\text{comm}}^{-1} (W \otimes I_d) \Sigma_{\text{comm}} [A(x_t, y_t)]_{\text{comm}}.$$

163 To show that  $[A(x_t, y_t)]_{\text{comm}}$  is locally accessible to each node, we write:

$$[A(x_t, y_t)]_{\text{comm}}^{(i)} = (A_{\text{comm}}x_t)^{(i)} - \left( \sum_{k=1}^n \sum_{j=1}^m (A(e_{kj} \otimes y_t^{(kj)}))^{(i)} \right) = (\tilde{x}_t)^{(i)} - \sum_{j=1}^m \mu_{ij} y_t^{(ij)}.$$

164 We note this rescaled local vector  $\theta_t = \Sigma_{\text{comm}}([A(x_t, y_t)]_{\text{comm}})$ , and obtain for variables  $\tilde{x}_t$  the  
 165 gossip update of (12). Note that we directly write  $y_t^{(ij)}$  instead of  $P_{ij}y_t^{(ij)}$  even though there has  
 166 been a multiplication by the matrix  $A$ . This is allowed because Equation (13) implies that (i)  
 167  $y_t^{(ij)} \in \text{Ker}(f_{ij})^\perp$  for all  $t$ , and (ii) the value of  $(I_d - P_{ij})z_t^{(ij)}$  does not matter since  $z_t^{(ij)}$  is only  
 168 used to compute  $\nabla f_{ij}$ . We now consider computation edges, and remark that:

$$\nabla_{y,ij} q_A(x_t, y_t) = -\mu_{ij} (\Sigma_{\text{comm}})_{ii} ([A(x_t, y_t)]_{\text{comm}})^{(i)} = -\mu_{ij} \theta_t. \quad (11)$$

169 Plugging Equation (11) into the updates of (9), we obtain the following updates:

$$\tilde{x}_{t+1} = \tilde{x}_t - \frac{\eta}{p_{\text{comm}}} (W \otimes I_d) \theta_t, \quad (12)$$

170 for communication edges, and for the local update of the  $j$ -th component of node  $i$ :

$$z_{t+1}^{(ij)} = \left(1 - \frac{\alpha\eta}{p_{ij}}\right) z_t^{(ij)} + \frac{\alpha\eta}{p_{ij}} \theta_t^{(i)}, \quad \theta_{t+1}^{(i)} = \frac{1}{\sigma_i} \left( \tilde{x}_{t+1}^{(i)} - \sum_{j=1}^m \nabla f_{ij}(z_{t+1}^{(ij)}) \right). \quad (13)$$

171 Finally, Algorithm 1 is obtained by expressing everything in terms of  $\theta_t$  and removing variable  $\tilde{x}_t$ .  
 172 To simplify notations, we further consider  $\theta$  as a matrix in  $\mathbb{R}^{n \times d}$  (instead of a vector in  $\mathbb{R}^{nd}$ ), and so  
 173 the communication update of Equation (12) is a standard gossip update with matrix  $W$ , which we  
 174 recall is such that  $W \otimes I_d = [AP_{\text{comm}}A^\top]_{\text{comm}}$ . We now discuss the local updates of Equation (13)  
 175 more in details, which are closely related to dual-free SDCA updates [33].

---

#### Algorithm 1 DVR( $z_0$ )

---

```

1:  $\alpha = 2\lambda_{\min}^+(A_{\text{comm}}^\top D_M^{-1} A_{\text{comm}})$ ,  $\eta = \min \left( \frac{p_{\text{comm}}}{\lambda_{\max}(A_{\text{comm}}^\top \Sigma_{\text{comm}} A_{\text{comm}})}, \frac{p_{ij}}{\alpha(1+\sigma_i^{-1}L_{ij})} \right)$  // Init.
2:  $\theta_0^{(i)} = -(\sum_{j=1}^m \nabla f_{ij}(z_0^{(ij)}))/\sigma_i$ . //  $z_0$  is arbitrary but not  $\theta_0$ .
3: for  $t = 0$  to  $K - 1$  do // Run for  $K$  iterations
4:   Sample  $u_t$  uniformly in  $[0, 1]$ . // Randomly decide the kind of update
5:   if  $u_t \leq p_{\text{comm}}$  then
6:      $\theta_{t+1} = \theta_t - \frac{\eta}{p_{\text{comm}}} \Sigma W \theta_t$  // Communication using  $W$ 
7:   else
8:     for  $i = 1$  to  $n$  do
9:       Sample  $j \in \{1, \dots, m\}$  with probability  $p_{ij}$ .
10:       $z_{t+1}^{(ij')} = z_t^{(ij')}$  for  $j \neq j'$  // Only one virtual node is updated
11:       $z_{t+1}^{(ij)} = \left(1 - \frac{\alpha\eta}{p_{ij}}\right) z_t^{(ij)} + \frac{\alpha\eta}{p_{ij}} \theta_t^{(i)}$  // Virtual node update
12:       $\theta_{t+1}^{(i)} = \theta_t^{(i)} - \frac{1}{\sigma_i} \left( \nabla f_{ij}(z_{t+1}^{(ij)}) - \nabla f_{ij}(z_t^{(ij)}) \right)$  // Local update using  $f_{ij}$ 
13: return  $\theta_K$ 

```

---

### 176 3 Convergence Rate

177 The goal of this section is to set parameters  $\eta$  and  $\alpha$  in order to get the best convergence guarantees.  
 178 We introduce  $\kappa_{\text{comm}} = \gamma \lambda_{\max}(A_{\text{comm}}^\top \Sigma_{\text{comm}} A_{\text{comm}}) / \lambda_{\min}^+(A_{\text{comm}}^\top D_M^{-1} A_{\text{comm}})$ , where  $\lambda_{\min}^+$  and

179  $\lambda_{\max}$  respectively refer to the smallest non-zero and the highest eigenvalue of the corresponding  
 180 matrices. We denote  $D_M$  the diagonal matrix such that  $(D_M)_{ii} = \sigma_i + \lambda_{\max}(\sum_{j=1}^m L_{ij}P_{ij})$ , where  
 181  $\nabla^2 f_{ij}(x) \preceq L_{ij}P_{ij}$  for all  $x \in \mathbb{R}^d$ . Note that we use notation  $\kappa_{\text{comm}}$  since it corresponds to a  
 182 condition number. In particular,  $\kappa_{\text{comm}} \leq \kappa_s$  when  $\sigma_i = \sigma_j$  for all  $i, j$ , and  $\kappa_{\text{comm}}$  more finely  
 183 captures the interplay between regularity of local functions (through  $D_M$  and  $\Sigma_{\text{comm}}$ ) and the  
 184 topology of the network (through  $A$ ) otherwise.

185 **Theorem 1.** We choose  $p_{\text{comm}} = (1 + \gamma \frac{m + \kappa_s}{\kappa_{\text{comm}}})^{-1}$ ,  $p_{ij} \propto (1 - p_{\text{comm}})(1 + L_{ij}/\sigma_i)$  and  $\alpha$  and  $\eta$   
 186 as in Algorithm 1. Then, there exists  $C_0 > 0$  that only depends on  $\theta_0$  (initial conditions) such that for  
 187 all  $t > 0$ , the error and the expected time  $T_\varepsilon$  required to reach precision  $\varepsilon$  are such that:

$$\sum_{i=1}^n \frac{1}{2} \|\theta_t^{(i)} - \theta^*\|^2 \leq C_0 \left(1 - \frac{\alpha\eta}{2}\right)^t, \text{ and so } T_\varepsilon = O\left(\left[2(m + \kappa_s) + \tau \frac{\kappa_{\text{comm}}}{\gamma}\right] \log \varepsilon^{-1}\right). \quad (14)$$

188 *Proof sketch.* We have seen in Section 2 that DVR is obtained by applying Bregman coordinate  
 189 descent on a well-chosen dual problem. Therefore, one of our key results consists in proving  
 190 convergence rates for Bregman coordinate descent. In order to ease the reading of the paper, we  
 191 present these results for a general setting in Appendix A, which is self-contained and which we  
 192 believe to be of independent interest (beyond its application to decentralized optimization).

193 Then, Appendix B focuses on the application to decentralized optimization. In particular, we recall  
 194 the Equivalence between DVR and Bregman coordinate descent applied to the dual problem of  
 195 Equation (4), and show that its structure is suited to the application of coordinate descent. Indeed,  
 196 no two virtual edges adjacent to the same node are updated at the same time with our sampling.  
 197 Then, we evaluate the relative smoothness and strong convexity constants of the augmented problem,  
 198 which allows to derive adequate values for parameters  $\alpha$  and  $\eta$ . Finally, we choose  $p_{\text{comm}}$  in order to  
 199 minimize the execution time of DVR.  $\square$

200 We would like to highlight the fact that the convergence theory of DVR decomposes nicely into  
 201 several building blocks, and thus simple rates are obtained. This is not so usual for decentralized  
 202 algorithms, for instance many follow-up papers were needed to obtain a tight convergence theory for  
 203 EXTRA [36, 13, 41, 19]. We now discuss the convergence rate of DVR more in details.

204 **Computation complexity.** The computation complexity of DVR is the same computation complexity  
 205 as locally running a stochastic algorithm with variance reduction at each node. This is not surprising  
 206 since, as we argue later, DVR can be understood as a decentralized version of an algorithm that is  
 207 closely related to dual-free SDCA [33]. Therefore, this improves the computation complexity of  
 208 EXTRA from  $O(m(\kappa_b + \gamma^{-1}))$  individual gradients to  $O(m + \kappa_s)$ , which is the expected improvement  
 209 for stochastic variance-reduced algorithm. In comparison, GT-SAGA [40], a recent decentralized  
 210 stochastic algorithm, has a computation complexity of order  $O(m + \kappa_s^2/\gamma^2)$ , which is significantly  
 211 worse than that of DVR, and generally worse than that of EXTRA as well.

212 **Communication complexity.** The communication complexity of DVR (*i.e.*, the number of commu-  
 213 nications, so the communication time is retrieved by multiplying by  $\tau$ ) is of order  $O(\kappa_{\text{comm}}/\gamma)$ , and  
 214 can be improved to  $O(\kappa_{\text{comm}}/\sqrt{\gamma})$  using Chebyshev acceleration (see Section 4). Yet, this is in  
 215 general worse than the  $O(\kappa_b + \gamma^{-1})$  communication complexity of EXTRA or NIDS, which can  
 216 be interpreted as a partly accelerated communication complexity since the optimal dependence is  
 217  $O(\sqrt{\kappa_b/\gamma})$  [30], and  $2\sqrt{\kappa_b/\gamma} = \kappa_b + \gamma^{-1}$  in the worst case ( $\kappa_b = \gamma^{-1}$ ). Yet, stochastic updates  
 218 are mainly intended to deal with cases in which the computation time dominates, and we show in the  
 219 experimental section that DVR outperforms EXTRA and NIDS for a wide range of communication  
 220 times  $\tau$  (the computation complexity dominates roughly as long as  $\tau < \sqrt{\gamma(m + \kappa_s)}/\kappa_{\text{comm}}$ ). Fi-  
 221 nally, the communication complexity of DVR is significantly lower than that of DSA and GT-SAGA,  
 222 the primal decentralized stochastic alternatives presented in Section 1.

223 **Homogeneous parameter choice.** In the homogeneous case ( $\sigma_i = \sigma_j$  for all  $i, j$ ), choosing the  
 224 optimal  $p_{\text{comp}}$  and  $p_{\text{comm}}$  described above leads to  $\eta\lambda_{\max}(W) = \sigma p_{\text{comm}}$ . Therefore, the commu-  
 225 nication update becomes  $\theta_{t+1} = (I - W/\lambda_{\max}(W))\theta_t$ , which is a gossip update with a standard  
 226 step-size (independent of the optimization parameters). Similarly,  $\alpha\eta(m + \kappa_s) = p_{\text{comp}}$ , and so the  
 227 step-size for the computation updates is independent of the network.

**Links with SDCA.** The single-machine version of Algorithm 1 ( $n = 1, p_{\text{comm}} = 0$ ) is closely related to dual-free SDCA [33]. The difference is in the stochastic gradient used: DVR uses  $\nabla f_{ij}(z_t^{(ij)})$ , where  $z_t^{(ij)}$  is a convex combination of  $\theta_k^{(i)}$  for  $k < t$ , whereas dual-free SDCA uses  $g_t^{(ij)}$ , which is a convex combination of  $\nabla f_{ij}(\theta_k^{(i)})$  for  $k < t$ . Both algorithms obtain the same rates.

**Local synchrony.** Instead of using the synchronous communications of Algorithm 1, it is possible to update edges one at a time, as in [11]. This can be very efficient in heterogeneous settings (both in terms of computation and communication times) and similar convergence results can be obtained using the same framework, and we leave the details for future work.

## 4 Acceleration

We show in this section how to modify DVR to improve the convergence rate of Theorem 1.

**Network acceleration.** Algorithm 1 depends on  $\gamma^{-1}$ , also called the *mixing time* of the graph, which can be as high as  $O(n^2)$  for a chain of length  $n$  [25]. However, it is possible to improve this dependency to  $\gamma^{-1/2}$  by using Chebyshev acceleration, as in [31]. To do so, the first step is to choose a polynomial  $P$  of degree  $k$  and communicate with  $P(W)$  instead of  $W$ . In terms of implementation, this comes down to performing  $k$  communication rounds instead of one, but this makes the algorithm depend on the spectral gap of  $P(W)$ . Then, the important fact is that there is a polynomial  $P_\gamma$  of degree  $\lceil \gamma^{-1/2} \rceil$  such that the spectral gap of  $P_\gamma(W)$  is of order 1. Each communication step with  $P_\gamma(W)$  only takes time  $\tau \deg(P_\gamma) = \tau \lceil \gamma^{-1/2} \rceil$ , and so the communication term in Theorem 1 can be replaced by  $\tau \kappa_{\text{comm}} \gamma^{-1/2}$ , thus leading to *network acceleration*. The polynomial  $P_\gamma$  can for example be chosen as a Chebyshev polynomial, and we refer the interested reader to [31] for more details. Finally, other polynomials yield even faster convergence when the graph topology is known [2].

**Catalyst acceleration.** Catalyst [21] is a generic framework that achieves acceleration by solving a sequence of subproblems. Because of space limitations, we only present the accelerated convergence rate without specifying the algorithm in the main text. Yet, only mild modifications to Algorithm 1 are required to obtain these rates, and the detailed derivations and proofs are presented in Appendix C.

**Theorem 2.** *DVR can be accelerated using catalyst, so that the time  $T_\varepsilon$  required to reach precision  $\varepsilon$  is equal (up to log factors) to  $T_\varepsilon = \tilde{O}([m + \sqrt{m\kappa_s} + \tau\sqrt{\kappa_{\text{comm}}/\gamma}\sqrt{m\kappa_{\text{comm}}/\kappa_s}] \log \varepsilon^{-1})$ .*

This rate recovers the computation complexity of optimal finite sum algorithms such as ADFS [11, 12]. Although the communication time is slightly increased (by a factor  $\sqrt{m\kappa_{\text{comm}}/\kappa_s}$ ), ADFS uses a stronger oracle than DVR (proximal operator instead of gradient), which is why we develop DVR in the first place. Although both ADFS and DVR are derived using the same dual formulation, both the approach and the resulting algorithms are rather different: ADFS uses accelerated coordinate descent, and thus has strong convergence guarantees at the cost of requiring dual oracles. DVR uses coordinate descent with the Bregman divergence of  $\phi_{ij} \propto f_{ij}^*$  in order to work with primal oracles, but thus loses direct acceleration, which is recovered through the Catalyst framework. Note that the parameters of accelerated DVR can also be set such that  $T_\varepsilon = \tilde{O}(\sqrt{\kappa_{\text{comm}}} [m + \tau/\sqrt{\gamma}] \log \varepsilon^{-1})$ , which recovers the convergence rate of optimal batch algorithms, but loses the finite-sum speedup.

## 5 Experiments

We investigate in this section the practical performances of DVR. We solve a regularized logistic regression problem on the RCV1 dataset [17] ( $d = 47236$ ) with  $n = 81$  (leading to  $m = 2430$ ) and two different graph topologies: an Erdős-Rényi random graph (see, e.g., [3]) and a grid. We choose  $\mu_{k\ell}^2 = 1/2$  for all communication edges, so the gossip matrix  $W$  is the Laplacian of the graph.

Figure 1 compares the performance of DVR with that of state-of-the-art primal algorithms such as EXTRA [36], NIDS [20], GT-SAGA [40], and Catalyst accelerated versions of EXTRA [19] and DVR. Suboptimality refers to  $F(\theta_t^{(0)}) - F(\theta^*)$ , where node 0 is chosen arbitrarily and  $F(\theta^*)$  is approximated by the minimal error over all iterations. Each subplot of Figure 1(a) shows the same run with different x axes. The left plot measures the complexity in terms of individual gradients ( $\nabla f_{ij}$ ) computed by each node whereas the center plot measures it in terms of communications (multiplications by  $W$ ). All other plots are taken with respect to (simulated) time (i.e., computing

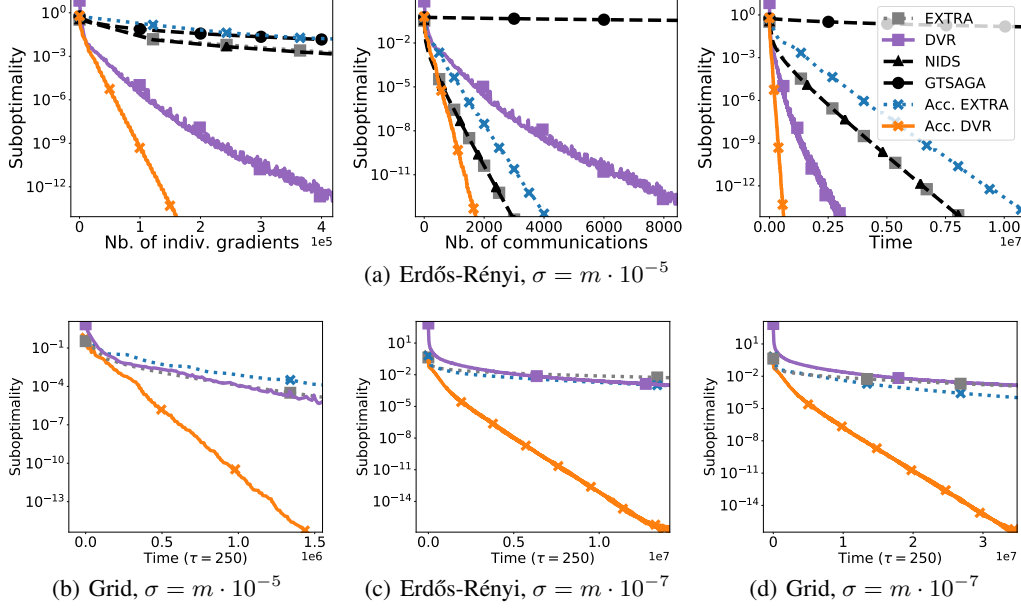


Figure 1: Experimental results for the RCV1 dataset with different graphs of size  $n = 81$ , with  $m = 2430$  samples per node, and with different regularization parameters.

$\nabla f_{ij}$  takes time 1 and multiplying by  $W$  takes time  $\tau$ ) with  $\tau = 250$  in order to report results that are independent of the computing cluster hardware and status. All parameters are chosen according to theory, except for the smoothness of the  $f_i$ , which requires finding the smallest eigenvalue of a  $d \times d$  matrix. For this, we start with  $L_b = \sigma_i + \sum_{j=1}^m L_{ij}$  (which is a known upper bound), and decrease it while convergence is ensured, leading to  $\kappa_b = 0.01\kappa_s$ . The parameters for accelerated EXTRA are chosen as in [19] since tuning the number of inner iterations does not significantly improve the results (at the cost of a high tuning effort). For accelerated DVR, we set the number of inner iterations to  $N/p_{\text{comp}}$  (one pass over the local dataset). We use Chebyshev acceleration for (accelerated) DVR but not for (accelerated) EXTRA since it is actually slower, as predicted by the theory.

As expected from their theoretical iteration complexities, NIDS and EXTRA perform very similarly [19], and GT-SAGA is the slowest method. Therefore, we only plot NIDS and GT-SAGA in Figure 1(a). We then see that though it requires more communications, DVR has a much lower computation complexity than EXTRA, which illustrates the benefits of stochastic methods. We see that DVR is faster overall if we choose  $\tau = 250$ , and both methods perform similarly for  $\tau \approx 1000$ , at which point communicating takes roughly as much time as computing a full local gradient. We then see that accelerated EXTRA has quite a lot of overhead and, despite our tuning efforts, is slower than EXTRA when the regularization is rather high. On the other hand, accelerated DVR consistently outperforms DVR by a relatively large margin. The communication complexity is in particular greatly improved, allowing accelerated DVR to be the fastest method regardless of the setting. Further experimental results are given in Appendix D, and the code is available in supplementary material.

## 6 Conclusion

This paper introduces DVR, a Decentralized stochastic algorithm with Variance Reduction obtained using Bregman block coordinate descent on a well-chosen dual formulation. Thanks to this approach, DVR inherits from the fast rates and simple theory of dual approaches without the computational burden of relying on dual oracles. Therefore, DVR has a drastically lower computational cost than standard primal decentralized algorithms, although sometimes at the cost of a slight increase in communication complexity. The framework used to derive DVR is rather general and could in particular be extended to analyze asynchronous algorithms. Finally, although deriving a direct acceleration of DVR is a challenging open problem, Catalyst and Chebyshev accelerations allow to significantly reduce DVR's communication overhead both in theory and in practice.



## Broader impact statement

This work does not present any foreseeable societal consequence.

## References

- [1] Heinz H. Bauschke, Jérôme Bolte, and Marc Teboulle. A descent lemma beyond Lipschitz gradient continuity: first-order methods revisited and applications. *Mathematics of Operations Research*, 42(2):330–348, 2017.
- [2] Raphaël Berthier, Francis Bach, and Pierre Gaillard. Accelerated gossip in networks of given dimension using jacobi polynomial iterations. *SIAM Journal on Mathematics of Data Science*, 2(1):24–47, 2020.
- [3] Béla Bollobás. *Random graphs*. Number 73 in Cambridge studies in advanced mathematics. Cambridge University Press, 2001.
- [4] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT*, pages 177–186. Springer, 2010.
- [5] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, 2006.
- [6] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.
- [7] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.
- [8] John C. Duchi, Alekh Agarwal, and Martin J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic Control*, 57(3):592–606, 2012.
- [9] Darina Dvinskikh and Alexander Gasnikov. Decentralized and parallelized primal and dual accelerated methods for stochastic convex programming problems. *arXiv preprint arXiv:1904.09015*, 2019.
- [10] Hadrien Hendrikx, Francis Bach, and Laurent Massoulié. Accelerated decentralized optimization with local updates for smooth and strongly convex objectives. In *Artificial Intelligence and Statistics*, 2019.
- [11] Hadrien Hendrikx, Francis Bach, and Laurent Massoulié. An accelerated decentralized stochastic proximal algorithm for finite sums. In *Advances in Neural Information Processing Systems*, 2019.
- [12] Hadrien Hendrikx, Francis Bach, and Laurent Massoulié. An optimal algorithm for decentralized finite sum optimization. *arXiv preprint arXiv:2005.10675*, 2020.
- [13] Dušan Jakovetić. A unification and generalization of exact distributed first-order methods. *IEEE Transactions on Signal and Information Processing over Networks*, 5(1):31–46, 2018.
- [14] Dušan Jakovetić, José M. F. Moura, and Joao Xavier. Linear convergence rate of a class of distributed augmented Lagrangian algorithms. *IEEE Transactions on Automatic Control*, 60(4):922–936, 2014.
- [15] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.
- [16] Guanghui Lan and Yi Zhou. An optimal randomized incremental gradient method. *Mathematical Programming*, pages 1–49, 2017.

- [17] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.
- [18] Huan Li, Cong Fang, Wotao Yin, and Zhouchen Lin. A sharp convergence rate analysis for distributed accelerated gradient methods. *arXiv preprint arXiv:1810.01053*, 2018.
- [19] Huan Li and Zhouchen Lin. Revisiting EXTRA for smooth distributed optimization. *arXiv preprint arXiv:2002.10110*, 2020.
- [20] Zhi Li, Wei Shi, and Ming Yan. A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *IEEE Transactions on Signal Processing*, 67(17):4494–4506, 2019.
- [21] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, pages 3384–3392, 2015.
- [22] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. Catalyst acceleration for first-order convex optimization: from theory to practice. *Journal of Machine Learning Research*, 18(1):7854–7907, 2017.
- [23] Qihang Lin, Zhaosong Lu, and Lin Xiao. An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization. *SIAM Journal on Optimization*, 25(4):2244–2273, 2015.
- [24] Haihao Lu, Robert M. Freund, and Yurii Nesterov. Relatively smooth convex optimization by first-order methods, and applications. *SIAM Journal on Optimization*, 28(1):333–354, 2018.
- [25] Bojan Mohar. Some applications of laplace eigenvalues of graphs. In *Graph Symmetry*, pages 225–275. Springer, 1997.
- [26] Aryan Mokhtari and Alejandro Ribeiro. DSA: Decentralized double stochastic averaging gradient algorithm. *Journal of Machine Learning Research*, 17(1):2165–2199, 2016.
- [27] Angelia Nedic, Alex Olshevsky, and Wei Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633, 2017.
- [28] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.
- [29] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87. Springer Science & Business Media, 2013.
- [30] Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Lee, and Laurent Massoulié. Optimal convergence rates for convex distributed optimization in networks. *Journal of Machine Learning Research*, 20:1–31, 2019.
- [31] Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *International Conference on Machine Learning*, pages 3027–3036, 2017.
- [32] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.
- [33] Shai Shalev-Shwartz. SDCA without duality, regularization, and individual convexity. In *International Conference on Machine Learning*, pages 747–754, 2016.
- [34] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(Feb):567–599, 2013.
- [35] Zebang Shen, Aryan Mokhtari, Tengfei Zhou, Peilin Zhao, and Hui Qian. Towards more efficient stochastic decentralized learning: Faster convergence and sparse communication. In *International Conference on Machine Learning*, pages 4631–4640, 2018.
- [36] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

- 397 [37] César A. Uribe, Soomin Lee, Alexander Gasnikov, and Angelia Nedić. A dual approach  
398 for optimal algorithms in distributed optimization over networks. *Optimization Methods and*  
399 *Software*, pages 1–40, 2020.
- 400 [38] Jialei Wang and Lin Xiao. Exploiting strong convexity from data with primal-dual first-order  
401 algorithms. In *International Conference on Machine Learning*, pages 3694–3702, 2017.
- 402 [39] Lin Xiao, Adams Wei Yu, Qihang Lin, and Weizhu Chen. DSCOVER: Randomized primal-dual  
403 block coordinate algorithms for asynchronous distributed optimization. *Journal of Machine*  
404 *Learning Research*, 20(43):1–58, 2019.
- 405 [40] Ran Xin, Soumya Kar, and Usman A Khan. Decentralized stochastic optimization and  
406 machine learning: A unified variance-reduction framework for robust performance and fast  
407 convergence. *IEEE Signal Processing Magazine*, 37(3):102–113, 2020.
- 408 [41] Jinming Xu, Ye Tian, Ying Sun, and Gesualdo Scutari. Distributed algorithms for composite  
409 optimization: Unified and tight convergence analysis. *arXiv preprint arXiv:2002.11534*, 2020.

This appendix contains the details of the derivations and proofs from the main text. More specifically, Appendix A is a self-contained appendix that specifies the Bregman coordinate descent algorithm and proves its convergence rate. Appendix B focuses on the application of Bregman coordinate descent to the dual problem (relative smoothness and strong convexity constants, sparsity structure), and how to retrieve guarantees on the primal parameters. Appendix C is devoted to presenting the Catalyst acceleration of DVR and proving its convergence speed, and Appendix D details the experimental setting, along with more experiments.

## A Block Coordinate descent

We focus in this section on the general problem minimizing  $f + g$  using coordinate Bregman gradient, where  $g$  is separable, i.e.,  $g(x) = \sum_{i=1}^d g_i(x^{(i)})$ . This is a self-contained section, and notations may differ from the rest of the paper. In particular, function  $f$  is for now arbitrary and not related to  $F$  or  $f_i$  from Problem (1), and the dimension  $d$  is arbitrary as well.

We first precise the blocks sampling rule. More specifically, we define a block  $b \subset \{1, \dots, d\}$  as a collection of coordinates, and  $\mathcal{B}$  is the set of all blocks that can be chosen for the updates. Then, the algorithm updates each block  $b \in \mathcal{B}$  with probability  $p(b)$ , so that the probability of updating a given coordinate is given by  $p_i = \sum_{i \in b} p(b)$ . Similarly to individual coordinates, we write  $x^{(b)}$  the restriction of  $x$  to coordinates in  $b$ . The Bregman coordinate gradient update for a block of coordinates  $b$  writes:

$$x_{t+1} = \arg \min_{x \in \mathbb{R}^d} \left\{ V_t^b(x) \triangleq \sum_{i \in b} \frac{\eta_t}{p_i} \left[ \nabla_i f(x_t)^\top x + g_i(x^{(i)}) \right] + D_\phi(x, x_t) \right\}, \quad (15)$$

where  $\nabla_i f$  denotes the gradient of  $f$  in direction  $i$ . Note that this update is more general than the one used to derive DVR, for which  $g = 0$ . In order to derive strong guarantees for this block coordinate descent algorithm, we need to ensure that there is some separability in functions  $f$  and  $\phi$ , and that the block structure is suited to this separability. All the assumptions about the separability structure of  $f$ ,  $g$  and  $\phi$  are contained in the following assumption.

**Assumption 1** (Separability). *The function  $g$  is separable and the function  $\phi$  is block-separable for  $b$ , meaning that for all  $b \in \mathcal{B}$ , there exist two convex functions  $\phi_b$  and  $\phi_b^\perp$  such that for all  $x$ ,*

$$\phi(x) = \phi_b(x^{(b)}) + \phi_b^\perp(x - x^{(b)}). \quad (16)$$

Besides, for all  $b \in \mathcal{B}$ , either of the following two hold:

1.  $\phi$  and  $f$  are separable for  $b$ , i.e.,  $\phi_b(x^{(b)}) = \sum_{i \in b} \phi_i(x^{(i)})$ , and

$$\sum_{i \in b} [f(x_t + \delta_i e_i) - f(x_t)] = f\left(x_t + \sum_{i \in b} \delta_i e_i\right) - f(x_t).$$

2.  $p_i = p_j$  for all  $i, j \in b$ .

If  $\phi$  is not block-separable, the support of the Bregman update in direction  $b$  may not be restricted to  $b$ . This causes some of the derivations below to fail, which is why we prevent it by assuming that Equation (16) holds.

Then, the first option ensures that within a block, the updates do not affect each other. The function  $f$  is not separable, but some directions can be updated independently from others. To have these independent updates, we also need to assume further separability of  $\phi$  within the blocks. The second option states that if only block-separability of  $\phi$  is assumed then within each block for which  $\phi$  and  $f$  are not separable, coordinates must be picked with the same probability.

Assumption 1 is a bit technical but we actually require all statements in order to derive DVR. In particular, the first option is verified when updating within the same block virtual edges that are adjacent to different nodes in the dual problem. The second option is verified when picking all communication edges at once within the same block.

Now that we have made assumptions on the structure of  $f$ ,  $g$  and  $\phi$ , we will make assumptions on their regularity. We start by a directional relative smoothness assumption between  $f$  and  $\phi$ , i.e., we

452 assume that for all  $i$ , there exists  $L_{\text{rel}}^i$  such that for all  $\delta > 0$  and  $e_i$  the unit vector of direction  $i$ ,

$$D_f(x + \delta e_i, x) \leq L_{\text{rel}}^i D_\phi(x + \delta e_i, x). \quad (17)$$

453 Similarly, for  $\sigma_{\text{rel}} > 0$ ,  $f$  is said to be  $\sigma_{\text{rel}}$ -strongly convex relatively to  $\phi$  if for all  $x, y$ :

$$D_f(x, y) \geq \sigma_{\text{rel}} D_\phi(x, y). \quad (18)$$

454 We finally assume that  $f$  and  $\phi$  are convex (but not necessarily smooth). We can now state the central  
455 theorem of this section:

456 **Theorem 3.** *Let  $f$  and  $\phi$  be such that  $f$  is  $L_{\text{rel}}^i$ -smooth in direction  $i$  and  $\sigma_{\text{rel}}$ -strongly convex  
457 relatively to  $\phi$ . Denote  $p_{\min} = \min_i p_i$ , and*

$$L_t = D_\phi(x, x_t) + \frac{\eta_t}{p_{\min}} (F(x_t) - F(x)).$$

458 *Then, if the blocks  $\mathcal{B}$  respect Assumption 1 (separability) and  $\eta_t L_{\text{rel}}^i < p_i$  for all  $i$ , the Bregman  
459 coordinate descent algorithm guarantees for all  $x$ :*

$$\mathbb{E}[L_{t+1}] \leq (1 - \eta_t \sigma_{\text{rel}}) L_t.$$

460 *The same result holds with  $L'_t = D_\phi(x, x_t) + \frac{1}{L_{\text{rel}}^{\max}} (F(x_t) - F(x))$ , where  $L_{\text{rel}}^{\max} = \max_i L_{\text{rel}}^i$ .*

461 To prove this theorem, we start by proving the monotonicity of such iterations.

462 **Lemma 1** (Monotonicity). *We note  $\delta_i = e_i^\top (x_{t+1} - x_t) e_i$ . If  $x_{t+1} = \arg \min_x V_t^b(x)$  then:*

463 1. *If  $\phi$  and  $f$  are separable for  $b$  then for all  $i \in b$ , if  $\eta_t L_{\text{rel}}^i \leq p_i$  then  $F(x_t) \geq F(x_t + \delta_i)$ .*

464 2. *If  $p_i = p_j$  for all  $i, j \in b$  and  $\eta_t L_{\text{rel}}^b \leq p_b$  then  $F(x_t) \geq F(x_{t+1})$ .*

465 *Proof.* We start by the first point. If  $\phi$  is separable for  $b$  then this means that each coordinate is  
466 updated independently. By definition of  $x_{t+1}^{(i)}$ , we have  $V_t^b(x_{t+1}^{(i)}) \leq V_t^b(x_t)$ . This writes, splitting  
467 over each  $i$  and using the fact that  $D_{\phi_i}(x_t, x_t) = 0$ :

$$\begin{aligned} g_i(x_t^{(i)}) - g_i(x_{t+1}^{(i)}) &\geq \nabla_i f(x_t)^\top (x_{t+1}^{(i)} - x_t^{(i)}) + \frac{p_i}{\eta} D_{\phi_i}(x_{t+1}^{(i)}, x_t^{(i)}) \\ &= \nabla_i f(x_t)^\top (x_t + \delta_i - x_t) + \frac{p_i}{\eta} D_{\phi_i}(x_{t+1}^{(i)}, x_t^{(i)}) \\ &= f(x_t + \delta_i) - f(x_t) - D_f(x_{t+1}^{(i)}, x_t^{(i)}) + \frac{p_i}{\eta} D_{\phi_i}(x_{t+1}^{(i)}, x_t^{(i)}) \\ &\geq f(x_t + \delta_i) - f(x_t) + \left( \frac{p_i}{\eta} - L_{\text{rel}}^i \right) D_{\phi_i}(x_{t+1}^{(i)}, x_t^{(i)}) \\ &\geq f(x_t + \delta_i) - f(x_t). \end{aligned}$$

468 The result follows from summing over all  $i \in b$ , and using Assumption 1. For the second point, it is  
469 not possible to split the update per coordinate since  $\phi$  is not separable. Yet, we can still write (using  
470 separability of  $g$ ):

$$\sum_{i \in b} \frac{\eta_t}{p_i} \left[ g(x_t^{(i)}) - g(x_{t+1}^{(i)}) - \nabla_i f(x_t)^\top (x_{t+1}^{(i)} - x_t^{(i)}) \right] \geq D_\phi(x_{t+1}, x_t). \quad (19)$$

471 Since  $g$  is separable and  $p_i = p_b$  for all  $i \in b$ , Equation (19) writes:

$$g(x_t) - g(x_{t+1}) \geq \nabla f(x_t)^\top (x_{t+1} - x_t) + \frac{p_b}{\eta} D_\phi(x_{t+1}, x_t). \quad (20)$$

472 Note that this crucially relies on  $x_{t+1} - x_t$  having support on  $b$ , which is enforced by the block-  
473 separability of  $\phi$ . Then, the proof is similar to that of the first point, using that  $\eta_t L_{\text{rel}}^b \leq p_b$ .  $\square$

474 Using this monotonicity result allows us to prove Theorem 3.

*Proof of Theorem 3.* First note that by convexity of all  $g_i$ ,

$$\nabla^2 V_t^b(x) = \sum_{i \in b} \frac{\eta_t}{p_i} \nabla^2 g_i(x^{(i)}) + \nabla^2 \phi(x) \succcurlyeq \nabla^2 \phi(x).$$

Therefore, we have  $D_{V_t^b}(x, y) \geq D_\phi(x, y)$  for all  $x, y \in \mathbb{R}^d$ . Applying this with  $y = x_{t+1}$  yields:

$$V_t^b(x) - V_t^b(x_{t+1}) - \nabla V_t^b(x_{t+1})^\top (x - x_{t+1}) \geq D_\phi(x, x_{t+1}). \quad (21)$$

Then,  $\nabla V_t^b(x_{t+1}) = 0$  by definition of  $x_{t+1}$ , so Equation (21) writes:

$$\begin{aligned} D_\phi(x, x_{t+1}) + \sum_{i \in b} \frac{\eta_t}{p_i} \left( g_i(x_{t+1}^{(i)}) - g_i(x^{(i)}) \right) &\leq \sum_{i \in b} \frac{\eta_t}{p_i} \nabla_i f(x_t)^\top (x - x_{t+1}) \\ &\quad + D_\phi(x, x_t) - D_\phi(x_{t+1}, x_t). \end{aligned}$$

We first consider that the first option of Assumption 1 holds, i.e., that  $f$  and  $\phi$  are separable in  $b$ . We note  $\delta_i = e_i^\top (x_{t+1} - x_t) e_i$ , so that:

$$\begin{aligned} -\nabla_i f(x_t)^\top (x_{t+1} - x_t) &= \nabla f(x_t)^\top (x_t + \delta_i - x_t) \\ &= f(x_t) - f(x_t + \delta_i) + D_f(x_t + \delta_i, x_t) \\ &\leq f(x_t) - f(x_t + \delta_i) + L_{\text{rel}}^i D_{\phi_i}(x_{t+1}^{(i)}, x_t^{(i)}). \end{aligned}$$

Therefore, if  $\eta_t L_{\text{rel}}^i \leq p_i$  for all  $i \in b$ ,

$$\begin{aligned} &-\sum_{i \in b} \frac{\eta_t}{p_i} \nabla_i f(x_t)^\top (x_{t+1} - x_t) - D_\phi(x_{t+1}, x_t) \\ &\leq \sum_{i \in b} \frac{\eta_t}{p_i} [f(x_t) - f(x_t + \delta_i)] + \sum_{i \in b} \left( \frac{\eta_t L_{\text{rel}}^i}{p_i} - 1 \right) D_{\phi_i}(x_{t+1}^{(i)}, x_t^{(i)}) \\ &\leq \sum_{i \in b} \frac{\eta_t}{p_i} [f(x_t) - f(x_t + \delta_i)] \end{aligned}$$

The  $g_i(x_{t+1}^{(i)}) - g_i(x^{(i)})$  term can be replaced by  $g(x_t + \delta_i) - g(x_t) + g_i(x_{t+1}^{(i)}) - g_i(x^{(i)})$  since  $g_j(x_{t+1}) = g_j(x_t)$  for  $j \neq i$ . Therefore, we obtain:

$$\begin{aligned} D_\phi(x, x_{t+1}) + \sum_{i \in b} \frac{\eta_t}{p_i} [F(x_t + \delta_i) - F(x_t)] + \sum_{i \in b} \frac{\eta_t}{p_i} \left( g_i(x_{t+1}^{(i)}) - g_i(x^{(i)}) \right) \\ \leq \sum_{i \in b} \frac{\eta_t}{p_i} \nabla_i f(x_t)^\top (x - x_t) + D_\phi(x, x_t). \end{aligned} \quad (22)$$

The separability of  $F$  in  $b$  and its monotonicity lead to, using the fact that  $x_{t+1} = x_t + \sum_{i \in b} \delta_i$ :

$$\sum_{i \in b} \frac{\eta_t}{p_i} [F(x_t + \delta_i) - F(x_t)] \geq \frac{\eta_t}{p_{\min}} \sum_{i \in b} [F(x_t + \delta_i) - F(x_t)] = \frac{\eta_t}{p_{\min}} [F(x_{t+1}) - F(x_t)].$$

Therefore, if the first option of Assumption 1 holds, we obtain:

$$\begin{aligned} D_\phi(x, x_{t+1}) + \frac{\eta_t}{p_{\min}} [F(x_{t+1}) - F(x_t)] + \sum_{i \in b} \frac{\eta_t}{p_i} \left( g_i(x_{t+1}^{(i)}) - g_i(x^{(i)}) \right) \\ \leq \sum_{i \in b} \frac{\eta_t}{p_i} \nabla_i f(x_t)^\top (x - x_t) + D_\phi(x, x_t). \end{aligned} \quad (23)$$

If the second option holds, i.e.,  $p_i = p$  for all  $i \in b$ , then

$$\sum_{i \in b} \frac{\eta_t}{p_i} \nabla_i f(x_t)^\top (x_{t+1} - x_t) = \frac{\eta_t}{p} \nabla f(x_t)^\top (x_{t+1} - x_t),$$

and Equation (23) can be obtained through similar derivations (at the block-level). Using the separability of  $g$ , we obtain that

$$\mathbb{E} \left[ \sum_{i \in b} \frac{1}{p_i} \left( g_i(x_{t+1}^{(i)}) - g_i(x^{(i)}) \right) \right] = g(x_t) - g(x).$$

Then, since  $\mathbb{E} \left[ \sum_{i \in b} \frac{1}{p_i} \nabla_i f(x_t) \right] = \sum_i p_i^{-1} \sum_{b: i \in b} p(b) \nabla_i f(x_t) = \nabla f(x_t)$ , and the relative strong convexity assumption yields:

$$\mathbb{E} \left[ \sum_{i \in b} \frac{1}{p_i} \nabla_i f(x_t)^\top (x - x_t) \right] = \nabla f(x_t)^\top (x - x_t) \leq f(x) - f(x_t) - \sigma_{\text{rel}} D_\phi(x, x_t).$$

Therefore, taking the expectation of Equation (22) yields:

$$\mathbb{E} \left[ D_\phi(x, x_{t+1}) + \frac{\eta_t}{p_{\min}} (F(x_{t+1}) - F(x_t)) \right] \leq \eta_t (F(x) - F(x_t)) + (1 - \eta_t \sigma_{\text{rel}}) D_\phi(x, x_t).$$

We obtain after some rewriting:

$$\begin{aligned} & \mathbb{E} \left[ D_\phi(x, x_{t+1}) + \frac{\eta_t}{p_{\min}} (F(x_{t+1}) - F(x)) \right] \\ & \leq (1 - p_{\min}) \frac{\eta_t}{p_{\min}} (F(x_t) - F(x)) + (1 - \eta_t \sigma_{\text{rel}}) D_\phi(x, x_t). \end{aligned}$$

Finally,  $\sigma_{\text{rel}} \leq L_{\text{rel}}^i$  so  $\eta_t \sigma_{\text{rel}} \leq \eta_t L_{\text{rel}}^i \leq p_i$  for all  $i$ , and in particular  $1 - p_{\min} \leq 1 - \eta_t \sigma_{\text{rel}}$ , which yields the desired result.

The result on  $L_t'$  is obtained by bounding  $\eta/p_{\min}$  by  $L_{\text{rel}}^{\max} = \max_i L_{\text{rel}}^i$  and remarking that  $1 - \eta_t L_{\text{rel}}^{\max} \leq 1 - \eta_t \sigma_{\text{rel}}$  since  $L_{\text{rel}}^{\max} \geq \sigma_{\text{rel}}$ .  $\square$

## B Convergence results for DVR

We now give a series of small results, that justify our approach. We start by showing the applicability of Theorem 3 to Problem (4), and the associated constants. Finally, we show how to obtain rates for the primal iterates  $\theta_t$ .

### B.1 Application to the dual of the augmented problem

In this section, we note  $f_{\text{sum}}^* = \sum_{i=1}^n \sum_{j=1}^m f_{ij}^*$ , so that Problem 4 writes:

$$\min_{x,y} q_A(x, y) + f_{\text{sum}}^*(y) \quad (24)$$

**Lemma 2.** *The iterations of Algorithm 1 are equivalent to the iteration of Equations (15) applied to Problem (4) with  $g = 0$  and  $\phi(x, y) = \phi_{\text{comm}}(x) + \sum_{i=1}^n \sum_{j=1}^m \phi_{ij}(y^{(ij)})$ , with  $\phi_{\text{comm}}(x) = \frac{1}{2} \|x\|_{A^\top A}^2$  for coordinates associated with communication edges, and  $\phi_{ij}(y^{(ij)}) = \frac{L_{ij}}{\mu_{ij}^2} f_{ij}^*(\mu_{ij} y_{ij})$  for coordinates associated with computation edges.*

*Proof.* This result follows from the dual-free and implementation-friendly derivations presented in the previous section.  $\square$

**Lemma 3.** *Let  $\alpha = 2\lambda_{\min}(A_{\text{comm}}^\top D_M^{-1} A_{\text{comm}})$ , and  $\phi$  as in Lemma 2, then:*

1.  $q_A + f_{\text{sum}}^*$  is  $(\alpha/2)$ -strongly convex relatively to  $\phi$ .
2.  $q_A + f_{\text{sum}}^*$  is  $(L_{\text{rel}}^{\text{comm}})$ -smooth relatively to  $\phi$  in the direction of communication edges, with

$$L_{\text{rel}}^{\text{comm}} = \lambda_{\max}(A_{\text{comm}}^\top \Sigma_{\text{comm}} A_{\text{comm}}).$$

3.  $q_A + f_{\text{sum}}^*$  is  $(L_{\text{rel}}^{ij})$ -smooth relatively to  $\phi$  in the direction of virtual edge  $(i, j)$ , with

$$L_{\text{rel}}^{ij} = \alpha \left( 1 + \frac{L_{ij}}{\sigma_i} \right).$$

*Proof.* First note that  $\nabla^2 f_{\text{sum}}^*$  is a block-diagonal matrix, and its  $ij$ -th block is equal to

$$(\nabla^2 f_{\text{sum}}^*(y))_{ij} = A^\top (u_{ij} u_{ij}^\top \otimes \nabla^2 f_{ij}^*(\mu_{ij} y^{(ij)})) A \succcurlyeq \frac{1}{L_{ij}} A^\top (u_{ij} u_{ij}^\top \otimes P_{ij}) A, \quad (25)$$

512 where  $u_{ij} \in \mathbb{R}^{n(1+m)}$  denotes the unit vector corresponding to virtual *node*  $(i, j)$ . We denote  
 513  $\tilde{\Sigma} = \Sigma + \sum_{i=1}^n \sum_{j=1}^m \frac{1}{L_{ij}} (u_{ij} u_{ij}^\top) \otimes I_d$ . Then,

$$\nabla^2 q_A(x, y) + \nabla^2 f_{\text{sum}}^*(y) = A^\top \tilde{\Sigma} A + \nabla^2 f_{\text{sum}}^*(y) - A^\top \left[ \sum_{i=1}^n \sum_{j=1}^m \frac{1}{L_{ij}} (u_{ij} u_{ij}^\top) \otimes P_{ij} \right] A. \quad (26)$$

514 **Relative strong convexity.** Then, [12, Lemma 6.5] leads to  $A^\top \tilde{\Sigma} A \succcurlyeq \sigma_F A^\top A$ . Note that the  
 515 notations are slightly different, and the matrix  $\tilde{\Sigma}$  in this paper is the same as the matrix  $\Sigma^\dagger$  in [12].  
 516 Then, remark that  $(A^\top A)_{ij} = P_{ij} = \frac{1}{\mu_{ij}^2} (A^\top [(u_{ij} u_{ij}^\top) \otimes P_{ij}] A)_{ij}$ , and  $\phi_{ij} = \alpha^{-1} f_{ij}^*$ , so that:

$$\begin{aligned} \nabla^2 q_A(x, y) + \nabla^2 f_{\text{sum}}^*(y) &\succcurlyeq \sigma_F \nabla^2 \phi(x, y) + \\ &\quad (1 - \alpha^{-1} \sigma_F) \left[ \nabla^2 f_{\text{sum}}^*(y) - A^\top \left[ \sum_{i=1}^n \sum_{j=1}^m \frac{1}{L_{ij}} (u_{ij} u_{ij}^\top) \otimes P_{ij} \right] A \right]. \end{aligned}$$

517 Finally, using that Equation (25) along with the fact that  $\sigma_F \leq \alpha$  implies that  $q_A + f_{\text{sum}}^*$  is  $\sigma_F$ -  
 518 relatively strongly convex with respect to  $\phi$ .

519 **Relative smoothness.** We first prove the relative smoothness property for communicate edges. For  
 520 any  $\tilde{x} \in R^{Ed}$ , Equation (26) leads to:

$$(\tilde{x}, 0)^\top [\nabla^2 q_A(x, y) + \nabla^2 f_{\text{sum}}^*(y)] (\tilde{x}, 0) = (\tilde{x}, 0)^\top A^\top \Sigma A (\tilde{x}, 0) \preccurlyeq L_{\text{rel}}^{\text{comm}} (\tilde{x}, 0)^\top \nabla^2 \phi(x, y) (\tilde{x}, 0).$$

521 Similarly, for any  $\theta \in \mathbb{R}^d$ , we consider  $\tilde{y} = e_{ij} \otimes \theta$  and write:

$$\begin{aligned} \tilde{y}^\top [\nabla^2 q_A(x, y) + \nabla^2 f_{\text{sum}}^*(y)] \tilde{y} &= \tilde{y}^\top A^\top \tilde{\Sigma} A \tilde{y} + \mu_{ij}^2 \theta^\top \left[ \nabla^2 f_{ij}^*(\mu_{ij} y^{(ij)}) - \frac{1}{L_{ij}} P_{ij} \right] \theta \\ &\preccurlyeq L_{\text{rel}}^i \tilde{y}^\top \nabla^2 \phi(x, y) \tilde{y} + (1 - \alpha^{-1} L_{\text{rel}}^i) \theta^\top \left[ \nabla^2 f_{ij}^*(\mu_{ij} y^{(ij)}) - \frac{1}{L_{ij}} P_{ij} \right] \theta, \end{aligned}$$

522 with

$$L_{\text{rel}}^i = \max_{\theta} \mu_{ij}^2 u_{ij}^\top \tilde{\Sigma} u_{ij} \frac{\theta^\top P_{ij} \theta}{\|\theta\|^2} \leq \alpha \left( 1 + \frac{L_{ij}}{\sigma_i} \right).$$

523 Finally,  $\nabla^2 f_{ij}^*(\mu_{ij} y^{(ij)}) \succcurlyeq P_{ij} / L_{ij}$ , and  $\alpha \leq L_{\text{rel}}^i$ , which ends the proof of the directional relative  
 524 smoothness result.  $\square$

525 **Lemma 4.** Assumption 1 holds with  $f = q_A + f_{\text{sum}}^*$ ,  $g = 0$ , and  $\phi$  as in Lemma 2, and when the  
 526 sampling is such that either:

- 527 • All communication edges are sampled at once, or
- 528 • Each node samples exactly one virtual edge.

529 *Proof.* First of all,  $g = 0$  is separable, and  $\phi$  is separable with respect to the communication and  
 530 computation blocks by construction.

531 We note  $b_{\text{comm}}$  the block of all communication edges, which is sampled with probability  $p_{\text{comm}}$ .  
 532 All communication edges are sampled at the same time, so  $p_i = p_{\text{comm}}$  for all  $i \in b_{\text{comm}}$  and so  $\phi$   
 533 respects option 2 for the communication block.

534 Let us now consider a computation block  $b$ . First of all,  $\phi$  is separable for the virtual edges. Then,  
 535 virtual blocks contain exactly one virtual edge per node, and so  $b = \{(1, j_1), \dots, (n, j_n)\}$ . Let  $k \neq \ell$ ,  
 536 then

$$e_{k,j_k}^\top A^\top \Sigma A e_{\ell,j_\ell} = \mu_{k,j_k} \mu_{\ell,j_\ell} (e_k - e_{k,j_k})^\top \Sigma (e_\ell - e_{\ell,j_\ell}) = 0.$$



Therefore,

$$\begin{aligned}
q_A \left( x_t + \sum_{(i,j) \in b} \delta_{ij} \right) - q_A(x_t) &= \frac{1}{2} \left( \sum_{(i,j) \in b} \delta_{ij} \right) A^\top \Sigma A \left( \sum_{(i,j) \in b} \delta_{ij} \right) + \left( \sum_{(i,j) \in b} A \delta_{ij} \right)^\top \Sigma A x_t \\
&= \sum_{(i,j) \in b} (q_A(\delta_{ij}) + \delta_{ij}^\top A^\top \Sigma A x_t) \\
&= \sum_{(i,j) \in b} (q_A(x_t + \delta_{ij}) - q_A(x_t)).
\end{aligned}$$

Finally,  $f_{\text{sum}}^*$  is separable, and so  $q_A + f_{\text{sum}}^*$  respects option 2.  $\square$

We can now prove the main theorem on the convergence rate of DVR.

**Theorem 4.** We choose  $p_{\text{comm}} = (1 + \gamma \frac{m + \kappa_s}{\kappa_{\text{comm}}})^{-1}$  and  $p_{ij} \propto (1 - p_{\text{comm}})(1 + L_{ij}/\sigma_i)$ . Then, for all  $\theta_0 \in \mathbb{R}^{n \times d}$  and all  $t > 0$ , the error is such that:

$$\frac{\eta_t}{p_{\min}} D_\phi(\lambda_\star, \lambda_t) + D(\lambda_t) - D(\lambda_\star) \leq \left(1 - \frac{\alpha \eta_t}{2}\right)^t \left[ \frac{\eta_t}{p_{\min}} D_\phi(\lambda_\star, \lambda_0) + D(\lambda_0) - D(\lambda_\star) \right], \quad (27)$$

with  $p_{\min} = \min(p_{\text{comm}}, \min_{ij} p_{ij})$ ,  $\lambda_t = (x_t, y_t)$  and  $D = -(q_A + f_{\text{sum}}^*)$ . Therefore, the expected time  $T_\varepsilon$  required to reach precision  $\varepsilon$  is equal to:

$$T_\varepsilon = O \left( \left[ 2(m + \kappa_s) + \tau \frac{\kappa_{\text{comm}}}{\gamma} \right] \log \varepsilon^{-1} \right).$$

*Proof.* Using Lemmas 4 and 3, we apply Theorem 3 (convergence of Bregman coordinate gradient descent), and obtain that the convergence rate is  $\eta_t \alpha / 2$ , with  $\eta_t \leq \min_{ij} p_{ij} / L_{\text{rel}}^i$  and  $\eta_t \leq p_{\text{comm}} / L_{\text{rel}}^{\text{comm}}$ . Therefore, for communication edges, we have that

$$\eta_t \leq \frac{p_{\text{comm}}}{L_{\text{rel}}^{\text{comm}}} = \frac{p_{\text{comm}}}{\lambda_{\max}(A_{\text{comm}}^\top \Sigma_{\text{comm}}^{-1} A_{\text{comm}})}.$$

For computation edges, we know that  $p_{ij} = p_{\text{comm}}(1 + L_{ij}/\sigma_i) / (\sum_{j=1}^m (1 + L_{ij}/\sigma_i))$ , and so

$$\eta_t \leq \frac{p_{ij}}{L_{\text{rel}}^{ij}} = \frac{p_{\text{comp}}}{\alpha \sum_{j=1}^m (1 + \sigma_i^{-1} L_{ij})} \leq \frac{p_{\text{comp}}}{\alpha(m + \kappa_s)},$$

with  $\kappa_s \geq \sigma_i^{-1} \sum_{j=1}^m L_{ij}$  for all  $i$ .

In the end, we would like these two bounds to be equal, so we choose  $p_{\text{comp}}$  and  $p_{\text{comm}}$  such that

$$p_{\text{comp}} = p_{\text{comm}} (m + \kappa_s) \frac{\lambda_{\min}^+(A_{\text{comm}}^\top D_M^{-1} A_{\text{comm}})}{\lambda_{\max}(A_{\text{comm}}^\top \Sigma_{\text{comm}}^{-1} A_{\text{comm}})}.$$

Yet, we also know that  $p_{\text{comm}} = 1 - p_{\text{comp}}$ , so

$$p_{\text{comp}} = \left( 1 + \frac{1}{m + \kappa_s} \frac{\lambda_{\max}(A_{\text{comm}}^\top \Sigma_{\text{comm}}^{-1} A_{\text{comm}})}{\lambda_{\min}^+(A_{\text{comm}}^\top D_M^{-1} A_{\text{comm}})} \right)^{-1}.$$

Equivalently, this corresponds to taking

$$p_{\text{comm}} = \left( 1 + \gamma \frac{m + \kappa_s}{\kappa_{\text{comm}}} \right)^{-1}.$$

With this choice, one can verify that  $\eta_t$  verifies both  $\eta_t \alpha \leq 2p_{\text{comm}}$  and  $\eta_t \alpha \leq 2 \min_{ij} p_{ij}$ , so the rate is:

$$1 - \frac{\eta_t \alpha}{2} = 1 - \frac{p_{\text{comp}}}{2(m + \kappa_s)}.$$

The expected execution time to reach precision  $\varepsilon$ , denoted  $T_\varepsilon$ , is equal to  $T_\varepsilon = \rho^{-1}(p_{\text{comp}} + \tau p_{\text{comm}}) K_\varepsilon$  with  $K_\varepsilon$  such that  $C(1 - \eta_t \alpha / 2)^{K_\varepsilon} < \varepsilon$  for some constant  $C$ , and so:

$$T_\varepsilon = O \left( 2(m + \kappa_s) + \tau \frac{\kappa_{\text{comm}}}{\gamma} \right).$$

$\square$

## 556 B.2 Primal guarantees

557 The goal of this section is to recover primal guarantees from dual guarantees. Although the initial  
 558 setting is inspired from [23], the proof is different, and in particular does not require smoothness of  
 559 the  $f_{ij}^*$  or an extra proximal step. We define for  $\beta \geq 0$  the Lagrangian function:

$$\mathcal{L}(\lambda, \theta) = \sum_{i=1}^n \sum_{j=1}^m f_{ij}(\theta^{(ij)}) + \frac{\sigma_i}{2} \|\theta^{(i)}\|^2 + \frac{\beta}{2} \|\theta^{(i)} - \omega^{(i)}\|^2 - \lambda^\top A^\top \theta. \quad (28)$$

560 The dual problem  $D(\lambda)$  is defined as

$$D(\lambda) = \min_{\theta} \mathcal{L}(\lambda, \theta).$$

561 Given an approximate dual solution  $\lambda_k$ , we can get an approximate primal solution  $\theta_k =$   
 562  $\arg \min_{\theta} \mathcal{L}(\lambda_k, \theta)$ , which is obtained as:

$$\theta_t^{(ij)} = \arg \min_v \left( f_{ij}(v) - \mu_{ij} \lambda_t^{(ij)} v \right) \in \partial f_{ij}^*(\mu_{ij} \lambda_t^{(ij)}), \quad (29)$$

$$\theta_t^{(i)} = \frac{1}{\sigma_i + \beta} \left( (A\lambda_t)^{(i)} + \beta \omega^{(i)} \right). \quad (30)$$

563 Note that  $\theta_t^{(ij)}$  corresponds to the  $z_t^{(ij)}$  from Algorithm 1. We chose to use a different notation in  
 564 the main text to emphasize on the fact that these are the parameters for the virtual nodes, but  $z_t^{(ij)}$   
 565 actually converge to the solution as well. Similarly,  $\lambda_t$  corresponds to  $(x_t, y_t)$  the concatenation of  
 566 the parameters for communication and virtual edges from Section 2. The last difference is that the  
 567 Lagrangian defined in Equation (28) actually corresponds to a Lagrangian associated to a perturbed  
 568 version of Problem (1) in which  $\tilde{f}_i(\theta) = f_i(\theta) + \frac{\beta}{2} \|\theta - \omega^{(i)}\|^2$ . The solution to the initial problem  
 569 can be retrieved by taking  $\beta = 0$ , but this more general formulation enables us to derive results that  
 570 also holds for the inner problems solved by the Catalyst accelerated version of DVR.

571 **Lemma 5.** Denote  $C_0 = \frac{(\beta + \sigma_{\max} + L_{\max})}{2(\sigma_{\min} + \beta)^2} \left( \frac{p_{\min}}{\eta_t} D_{\phi}(\lambda^*, \lambda_0) + (D(\lambda^*) - D(\lambda_0)) \right)$ , then

$$\sum_{i=1}^n \|\theta_t^{(i)} - \theta^*\|^2 \leq C_0 (1 - \rho)^t. \quad (31)$$

572 *Proof.* Using the fact that  $\theta_t^{(i)} = \frac{1}{\sigma_i + \beta} ((A\lambda_t)^{(i)} + \omega_t^{(i)})$  (and similarly for  $\theta^*$ ), where  $\Sigma_{\beta}$  is the block  
 573 diagonal matrix such that  $(\Sigma_{\beta})_{ii} = (\sigma_i + \beta)^{-1} I_d$ , we obtain:

$$\begin{aligned} \sum_{i=1}^n \|\theta_t^{(i)} - \theta^*\|^2 &= \sum_{i=1}^n \frac{1}{(\sigma_i + \beta)^2} \|(A\lambda_t)^{(i)} - (A\lambda^*)^{(i)}\|^2 \\ &\leq \frac{1}{(\sigma_{\min} + \beta)^2} \|A\lambda_t - A\lambda^*\|^2. \end{aligned}$$

574 Using the  $\min((\sigma_{\max} + \beta)^{-1}, L_{ij}^{-1})$ -strong convexity of  $\theta \mapsto \frac{1}{2} x^\top \Sigma_{\beta} x + \sum_{i,j} f_{ij}^*(x^{(ij)})$ , we obtain:

$$\sum_{i=1}^n \|\theta_t^{(i)} - \theta^*\|^2 \leq 2 \frac{(\beta + \sigma_{\max} + L_{\max})}{(\sigma_{\min} + \beta)^2} (D(\lambda^*) - D(\lambda_t)). \quad (32)$$

576 Then, we add  $p_{\min} \eta_t^{-1} D_{\phi}(\lambda^*, \lambda_t) \geq 0$  and apply Theorem 4, which yields

$$\sum_{i=1}^n \|\theta_t^{(i)} - \theta^*\|^2 \leq \frac{2(\beta + \sigma_{\max} + L_{\max})}{(\sigma_{\min} + \beta)^2} (1 - \rho)^t \left( \frac{p_{\min}}{\eta_t} D_{\phi}(\lambda^*, \lambda_0) + (D(\lambda^*) - D(\lambda_0)) \right).$$

577 □

578 Then, Theorem 1 is a direct consequence of Theorem 4 and Lemma 5.

## C Catalyst acceleration

We show in this Section how to apply Catalyst acceleration to DVR, and prove the convergence speed in this case.

### C.1 Derivation and rates

In the main text, we derived DVR to solve regularized finite sum problems. Although not so different, the subproblem obtained with Catalyst is not in the form of Problem (1), and some adjustments need to be made. More specifically, we would like to solve problems of the form:

$$\min_{\theta} \left\{ F_t(\theta) \triangleq \sum_{i=1}^n \left[ \frac{\sigma_i}{2} \|\theta\|^2 + \frac{\beta}{2} \|\theta - \omega_t^{(i)}\|^2 + \sum_{j=1}^m f_{ij}(\theta) \right] \right\}. \quad (33)$$

An easy way to adapt the algorithm is to consider the extra  $(\beta/2)\|\theta - \omega_t^{(i)}\|^2$  as just another component of the sum. Yet, the point of this extra term is to make the problem easier to solve by adding strong convexity. This would not be the case if this term were is treated as just another term in the sum. Therefore, we want to include it with the quadratic term. We define:

$$h(x) = \frac{\sigma_i}{2} \|\theta\|^2 + \frac{\beta}{2} \|\theta - \omega_t^{(i)}\|^2,$$

then  $h^*(x) = \frac{1}{2(\beta+\sigma)} \|x + \beta\omega_t^{(i)}\|^2 - \frac{\beta}{2} \|\omega_t^{(i)}\|^2$ . Therefore, Problem (4) becomes:

$$\min_{\lambda \in \mathbb{R}^{(E+mn)d}} \frac{1}{2} \lambda^\top A^\top \Sigma_\beta A \lambda + \beta \omega_t^\top \Sigma_\beta A \lambda + \sum_{i=1}^n \sum_{j=1}^m f_{ij}^*((A\lambda)_{ij}), \quad (34)$$

with  $(\Sigma_\beta)_{ii} = (\sigma_i + \beta)^{-1}$  for  $i \in \{1, \dots, n\}$ . The linear term does not affect the Hessians, and thus the convergence rate is the same as before, with  $\sigma$  replaced by  $\sigma + \beta$ . In terms of algorithms, we just need to modify the gradient term, and obtain Algorithm 2. The only term that changes is  $\nabla q_A(x, y)$ , to which an extra  $\beta \Sigma_\beta \omega_t$  term is added. Therefore, the updates to  $\theta_t$  and  $z_t$  remain unchanged, and only the initial expression of  $\theta_t$  requires some adjustments since we now have that (as written in Equation 30):

$$\theta_{t,k}^{(i)} = \frac{1}{\sigma_i + \beta} \left( (A\lambda_{t,k})^{(i)} + \beta \omega_t^{(i)} \right).$$

If we only consider 1 inner loop then the only thing that changes is the initial condition. If we consider several outer loops, then the we must choose the new parameter as  $\theta_0^{t+1} = \theta_T^t + \Sigma_\beta(\omega_{t+1} - \omega_t)$  in order to maintain the invariant, but a remarkable fact is that the inner iterations remain the same, with the only exception that  $\Sigma$  is replaced by  $\Sigma_\beta$ . Note that it is possible to warm-start the  $z_{t+1,0}$  as well, but this requires updating  $\theta_{t,0}$  accordingly with  $\nabla f_{ij}(z_{t,0}^{(ij)})$ , which requires a full pass over the local dataset. We therefore choose not to do it.

However, it is not obvious that Algorithm 2 corresponds to a genuine Catalyst acceleration yet. Indeed, Catalyst acceleration requires having a feasible  $\varepsilon_t$ -approximations for the primal problem, *i.e.*, points  $\theta_t \in \mathbb{R}^d$  such that  $F_t(\theta_t) - \min_{\theta} F(\theta) \leq \varepsilon_t$ . In our case, we only have dual guarantees and approximate feasibility. We know that the parameters converge to consensus, but they do not reach it at any time. This is a problem because it is then not possible to adequately define  $F_{t+1}$  based on the local approximations of the solutions of  $F_t$ . Yet, following the approach of [19], we note that

$$\sum_{i=1}^n \|\theta - \omega_t^{(i)}\|^2 = n \|\theta - \bar{\omega}_t\|^2 + \sum_{i=1}^n \|\omega_t^{(i)}\|^2 - n \|\bar{\omega}_t\|^2,$$

where  $\bar{\omega}_t = \frac{1}{n} \sum_{i=1}^n \omega_t^{(i)}$ . This means that although  $F_t$  is only defined with the local variables  $\omega_t^{(i)}$ , solving  $F_t$  is equivalent to solving a problem involving  $\bar{\omega}_t$  only. Besides, the Catalyst iterations are linear, meaning that performing the extrapolation step on  $\bar{\theta}_t$  is equivalent to performing it on each  $\theta_t^{(i)}$  individually. Therefore, although Catalyst is implemented in a fully decentralized manner (each node

---

**Algorithm 2** Accelerated DVR( $z_0$ )
 

---

```

1:  $\alpha = 2\lambda_{\min}^+(A_{\text{comm}}^\top D_M^{-1} A_{\text{comm}})$ ,  $\eta = \min \left( \frac{p_{\text{comm}}}{\lambda_{\max}(A_{\text{comm}}^\top \Sigma_{\beta, \text{comm}} A_{\text{comm}})}, \frac{p_{ij}}{\alpha(1+\sigma_i^{-1}L_{ij})} \right)$ 
2:  $q = \frac{\sigma_{\min}}{\sigma_{\min} + \beta}$  // Initialization
3:  $\omega_0^{(i)} = -\frac{1}{\sigma_i + \beta} \sum_{j=1}^m \nabla f_{ij}(z_0^{(ij)})$ ,  $\theta_0^{(i)} = \left(1 + \frac{\beta}{\sigma_i + \beta}\right) \omega_0^{(i)}$ . //  $z_0$  is arbitrary but not  $\theta_0$ .
4: for  $t = 0$  to  $T - 1$  do //  $T$  outer loops
5:   for  $k = 0$  to  $K - 1$  do // Inner loop runs for  $K$  iterations
6:      $z_{t,k+1} = z_{t,k}$ .
7:     Sample  $u_t$  uniformly in  $[0, 1]$ . // Randomly decide the kind of update
8:     if  $u_t \leq p_{\text{comm}}$  then
9:        $\theta_{t,k+1} = \theta_{t,k} - \frac{\eta_t}{p_{\text{comm}}} \Sigma_\beta W \theta_{t,k}$  // Communication using  $W$ 
10:    else
11:      for  $i = 1$  to  $n$  do
12:        Sample  $j \in \{1, \dots, m\}$  with probability  $p_{ij}$ .
13:         $z_{t,k+1}^{(ij)} = \left(1 - \frac{\alpha\eta}{p_{\text{comp}}}\right) z_{t,k}^{(ij)} + \frac{\alpha\eta}{p_{\text{comp}}} \theta_{t,k}^{(i)}$  // Computing new virtual node parameter
14:         $\theta_{t,k+1}^{(i)} = \theta_{t,k}^{(i)} - \frac{1}{\sigma_i + \beta} \left( \nabla f_{ij}(z_{t,k+1}^{(ij)}) - \nabla f_{ij}(z_{t,k}^{(ij)}) \right)$  // Local update using  $f_{ij}$ 
15:       $\omega_{t+1} = \theta_{t,K} + \frac{1-\sqrt{q}}{1+\sqrt{q}} (\theta_{t,K} - \theta_{t-1,K})$ 
16:       $\theta_{t+1,0} = \theta_{t,K} + \frac{\beta}{\beta + \sigma_i} (\omega_{t+1} - \omega_t)$ 
17:       $z_{t+1,0} = z_{t,K}$ 
18: return  $\theta_T$ 

```

---

613 knowing only its own parameter), it is conceptually applied to a mean parameter  $\bar{\theta}_t$  (that is never  
 614 explicitly computed). In the following, we thus analyze the performances of the following algorithm:

$$\begin{aligned} \bar{\theta}_{t+1} &\approx \arg \min_{\theta} F(\theta) + \frac{n\beta}{2} \|\theta - \bar{\omega}_t\|^2 \\ \bar{\omega}_t &= \bar{\theta}_{t+1} + \frac{1 - \sqrt{q}}{1 + \sqrt{q}} (\bar{\theta}_{t+1} - \bar{\theta}_t), \end{aligned} \quad (35)$$

615 where we recall that  $q = \sigma_{\min}/(\sigma_{\min} + \beta)$ . Recall that the inner problem is approximated using DVR  
 616 and the means do not need to be computed explicitly. Let  $\kappa_s^\beta = \max_i 1 + (\sum_{j=1}^m L_{ij})/(\beta + \sigma_i)$ ,  
 617 and  $\kappa_{\text{comm}}^\beta$  be obtained similarly to  $\kappa_{\text{comm}}$  but replacing  $\Sigma$  by  $\Sigma_\beta$ . We consider in this section that  
 618  $\sigma_i = \sigma$  for all  $i \in \{1, \dots, n\}$  in order to simplify exposition, but the results hold more generally.  
 619 Note that  $\alpha$  and  $\eta$  have slightly different expressions than in the main text since  $\beta$  is now involved in  
 620 their definitions. We define the sequence  $\varepsilon_t$  which is such that:

$$\varepsilon_t = \frac{2}{9} (F(\theta_0) - F(\theta^*)) (1 - \rho^{\text{out}})^t \text{ with } \rho^{\text{out}} < \sqrt{q}, \text{ and } q = \frac{\sigma}{\sigma + \beta}. \quad (36)$$

621 We then prove the following theorem:

622 **Theorem 5.** Consider Algorithm 2 with  $p_{\text{comm}} = (1 + \gamma \frac{m + \kappa_s^\beta}{\kappa_{\text{comm}}^\beta})^{-1}$ ,  $p_{ij} \propto (1 - p_{\text{comm}})(1 + L_{ij}/(\sigma_i +$   
 623  $\beta))$ . If  $K = \tilde{O}(1/(\eta_t \alpha))$  then for all  $t \leq T$ ,  $F_t(\bar{\theta}_t) - F_t(\theta_t^*) \leq \varepsilon_t$  and

$$F(\bar{\theta}_t) - F(\theta^*) \leq \frac{8}{(\sqrt{q} - \rho^{\text{out}})^2} (1 - \rho^{\text{out}})^{t+1} (F(\bar{\theta}_0) - F(\theta^*)). \quad (37)$$

624 Note that the error is on the mean parameter, and we also want  $\theta_t^{(i)}$  to be close to  $\bar{\theta}_t$  for all  $i$ . This is  
 625 ensured by Lemma 5. Before we start the proof of Theorem 5, we show that Theorem 2 is a corollary  
 626 of Theorem 5.

627 *Proof of Theorem 2.* Using the same argument as in Theorem 1, we obtain that each inner loop takes  
 628 time

$$T_{\text{inner}} = O \left( m + \frac{L_s + \sigma}{\beta + \sigma} + \tau \frac{L_{\text{comm}} + \beta}{\gamma(\beta + \sigma)} \right)$$

in expectation, so the total number of inner iterations is of order:

$$T_\varepsilon = \tilde{O} \left( \sum_{k=0}^{\lceil 1/\rho^{\text{out}} \rceil} T_{\text{inner}} \right) = \tilde{O} \left( \sqrt{1 + \frac{\beta}{\sigma}} \left( m + \frac{L_s + \sigma}{\beta + \sigma} + \tau \frac{L_{\text{comm}} + \beta}{\gamma(\beta + \sigma)} \right) \log \frac{1}{\varepsilon} \right). \quad (38)$$

Therefore, we see that if we choose  $\beta + \sigma = L_{\text{comm}}$  then, taking into account the fact that  $\kappa_s \leq m\kappa_{\text{comm}}$ , the algorithm takes time:

$$T_\varepsilon = \tilde{O} \left( \sqrt{\kappa_{\text{comm}}} \left( m + \frac{\tau_c}{\gamma} \right) \right).$$

Therefore, using Chebyshev acceleration allows to recover the rate of optimal batch algorithms (up to log factors). On the other hand, if we choose  $\beta = L_s/m - \sigma$  then if  $\beta \geq 0$  (i.e.,  $\kappa_s \geq m$ ), the time to convergence is equal to:

$$T_\varepsilon = \tilde{O} \left( \sqrt{\frac{\kappa_s}{m}} \left( m + \tau \frac{m\kappa_{\text{comm}} + \kappa_s}{\gamma\kappa_s} \right) \right).$$

This can be rewritten as:

$$T_\varepsilon = \tilde{O} \left( \sqrt{m\kappa_s} + \tau \frac{\sqrt{\kappa_{\text{comm}}}}{\gamma} \sqrt{\frac{m\kappa_{\text{comm}}}{\kappa_s}} \right).$$

Therefore, we obtain the optimal  $\sqrt{m\kappa_s}$  computation complexity in this case, with a slightly suboptimal communication complexity due to the  $\sqrt{m\kappa_{\text{comm}}/\kappa_s}$  term. When this term is equal to 1 then  $\sqrt{m\kappa_s} = m\sqrt{\kappa_b}$  and so nothing is gained from using a stochastic algorithm. Otherwise, this allows to trade-off communications for computations.  $\square$

The proof of Theorem 5 is obtained in several steps, that we emphasize below:

1. Equivalent decentralized implementation of Catalyst.
2. Bounding the primal suboptimality as  $F_t(\bar{\theta}_t) - \min_{\theta} F_t(\theta) \leq (1 - (\eta\alpha)/2)^k D_0^t$ , with  $k$  the number of inner iterations and  $D_0^t$  a dual error. This quantifies how precisely the inner problem is solved.
3. Evaluating the initial dual suboptimality  $D_0^t$ , which depends on  $\theta_{t-1}$  (and its associated dual parameter  $\lambda_{t-1}$ ). This quantifies how good  $\bar{\theta}_{t-1}$  already is as a solution to  $F_t$ .

In the end, this allows us to use the catalyst general results with primal criterion, and with simple warm-start scheme (warm-start on the last iterate of the last outer iteration). The first point is presented at the beginning of this section and the second one is addressed by Lemma 5. The following section deals the last point.

## C.2 Proof of Theorem 5

We now show a bound on the initial error of an inner loop when warm-starting on the last iterate of the previous inner loop. Indeed, the convergence results for DVR depend on the initial dual error and so results from [22] cannot be used directly. Yet, it can be adapted, as we show in this section. We note  $D_t(\lambda)$  the dual function at outer step  $t$  (which should not be mistaken with the Bregman divergence  $D_\phi$ ), and  $\lambda_\star^t$  its minimizer. Similarly, we note  $\theta_\star^t = \arg \min_{\theta} F_t(\theta)$ , whereas  $\theta^\star$  is the global minimizer of  $F$ . The following theorem ensures convergence of  $\bar{\theta}_t$  to the true optimum, given that the subproblems are solved precisely enough.

**Theorem 6.** [22, Proposition 5]. *If  $F_k(\bar{\theta}_k) - F_k(\theta_\star^k) \leq \varepsilon_k$  for all  $k \leq t$  then*

$$F(\bar{\theta}_t) - F(\theta^\star) \leq \frac{8}{(\sqrt{q} - \rho^{\text{out}})^2} (1 - \rho^{\text{out}})^{t+1} (F(\bar{\theta}_0) - F(\theta^\star)). \quad (39)$$

Therefore, our goal is to prove that  $F_t(\bar{\theta}_{t+1}) - F_t(\theta_\star^t) \leq \varepsilon_t$  for all  $t$ . The smoothness of  $F_t$  ensures that this is achieved if

$$\sum_{i=1}^n \|\theta_{t+1}^{(i)} - \theta_\star^t\|^2 \leq \frac{n}{L} \varepsilon_t. \quad (40)$$

Yet, using Lemma 5, we know that, since  $\theta_{t+1}^{(i)}$  is obtained by applying  $K$  steps of DVR to  $F_t$  starting from  $\lambda_0^t$ .

$$\sum_{i=1}^n \|\theta_{t+1}^{(i)} - \theta_\star^t\|^2 \leq \frac{(\beta + \sigma_{\max} + L_{\max})}{(\sigma_{\min} + \beta)^2} (1 - \rho)^K \left( \frac{p_{\min}}{\eta_t} D_\phi(\lambda_\star^t, \lambda_0^t) + D_t(\lambda_\star^t) - D_t(\lambda_0^t) \right).$$

Unfortunately, we have no control over the dual error at this point. In the remainder of this section, we prove by recursion that Equation (40) holds for all  $t$ . More specifically, we start by assuming that:

$$\frac{1}{2} \sum_{i=1}^n \|\theta_{t+1}^{(i)} - \theta_\star^t\|^2 \leq \frac{n}{L} \varepsilon_t, \quad (41)$$

$$\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^m \|\theta_{t+1}^{(ij)} - \theta_\star^t\|^2 \leq C_1 \varepsilon_t, \quad (42)$$

$$D_t(\lambda_\star^t) - D_t(\lambda_{t+1}) \leq C_2 \varepsilon_t, \quad (43)$$

where  $C_1$  and  $C_2$  are such that the conditions are verified for  $t = -1$ , with  $D_{-1} = D_0$ ,  $\theta_\star^{-1} = \theta_\star^0$ , and  $\lambda_\star^{-1} = \lambda_\star^0$ . Equation (41) may not hold for  $t = -1$ , but making it hold at time  $t = 0$  would only require a slightly longer first inner iteration, meaning at most an extra log factor. Therefore we assume without loss of generality that it is the case, since the final complexities are given up to logarithmic factors. The rest of this section is devoted to showing that if  $K$  is chosen as in Theorem 5 then Equations (41), (42) and (43) hold regardless of  $t$ . The first part focuses on assessing the initial error of outer iteration  $t + 1$  when the conditions hold at the end of outer iteration  $t$ , and the second part on showing how these errors shrink during outer iteration  $t + 1$ .

### C.2.1 Warm-start error

We know that DVR converges linearly, and so the error for each subproblem decreases exponentially fast. Yet, we need to know how big the error is when solving a new problem in order to make sure that the progress from solving previous subproblems is not lost. The point of this is to avoid an extra  $\log(\varepsilon^{-1})$  factor in the rate, which would come from having to solve each subproblem from a  $O(1)$  precision to an  $\varepsilon$  precision using DVR. We show in this section that the initial error is actually much lower than  $O(1)$  and decreases with the outer iterations. We first start by bounding the variations of  $\omega_t$  across iterations, which we will need for the next proofs.

**Lemma 6** (Distance between subproblems). *It holds that*

$$\|\omega_t - \omega_{t-1}\|^2 \leq C_\omega \varepsilon_{t-1}, \text{ with } C_\omega = \frac{1080n}{1 - \rho^{\text{out}}} \left( \frac{8(1 - \rho^{\text{out}})}{\sigma_{\min}(\sqrt{q} - \rho^{\text{out}})^2} + \frac{4}{9L} \right).$$

*Proof.* The form of the updates yields that (see [22, Proposition 12] or [19, Proof of Lemma 10])

$$\|\omega_t^{(i)} - \omega_{t-1}^{(i)}\| \leq 40 \max\{\|\theta_t^{(i)} - \theta_\star\|, \|\theta_{t-1}^{(i)} - \theta_\star\|, \|\theta_{t-2}^{(i)} - \theta_\star\|\}.$$

Note that here,  $\theta_\star$  is the actual solution of the primal problem without the catalyst perturbation. Then, the error can be decomposed as:

$$\begin{aligned} \sum_{i=1}^n \|\theta_t^{(i)} - \theta_\star\|^2 &\leq 3 \sum_{i=1}^n \left( \|\theta_t^{(i)} - \theta_\star^t\|^2 + \|\theta_\star^t - \bar{\theta}_t\|^2 + \|\bar{\theta}_t - \theta_\star\|^2 \right) \\ &\leq 3n \|\bar{\theta}_t - \theta_\star\|^2 + 6 \sum_{i=1}^n \|\theta_t^{(i)} - \theta_\star^t\|^2. \end{aligned}$$

Finally, the strong convexity of  $F$  leads to

$$\frac{\sigma_{\min}}{2} \|\bar{\theta}_t - \theta_\star\|^2 \leq F(\bar{\theta}_t) - F(\theta_\star) \leq \frac{8}{(\sqrt{q} - \rho^{\text{out}})^2} (1 - \rho^{\text{out}})^{t+1} (F(\theta_0) - F(\theta_\star)) \quad (44)$$

where in the last inequality we use [22, Proposition 5], which holds because  $F_k(\bar{\theta}_k) - F_k(\theta_\star^k) \leq \varepsilon_k$  for all  $k < t$ . Indeed,  $K$  is such that for all  $k \leq t$ ,  $\frac{1}{2} \sum_{i=1}^n \|\theta_k^{(i)} - \theta_\star^k\|^2 \leq \frac{n}{L} \varepsilon_k$ , which yields:

$$F_k(\bar{\theta}_k) - F_k(\theta_\star^k) \leq \frac{L}{2} \|\bar{\theta}_k - \theta_\star^k\|^2 \leq \frac{L}{2n} \sum_{i=1}^n \|\theta_k^{(i)} - \theta_\star^k\|^2 \leq \varepsilon_k.$$

Therefore,

$$\sum_{i=1}^n \|\theta_t^{(i)} - \theta^*\|^2 \leq 6n(1 - \rho^{\text{out}})^t (F(\theta_0) - F(\theta^*)) \left( \frac{8(1 - \rho^{\text{out}})}{\sigma_{\min}(\sqrt{q} - \rho^{\text{out}})^2} + \frac{4}{9L} \right),$$

and a similar bound can be used for  $\theta_{t-1}^{(i)}$  and  $\theta_{t-2}^{(i)}$ . Then, we finish proof by plugging in the expression of  $\varepsilon_{t-1}$ .  $\square$

We then use Lemma 6 to bound the initial dual error. We denote  $\theta_k^t$  (and  $\lambda_k^t$ ) the parameters at inner iteration  $k$  of outer iteration  $t$ .

**Lemma 7** (Dual error warm-start). *The warm-started dual error verifies:*

$$D_t(\lambda_\star^t) - D_t(\lambda_t) \leq C_D \varepsilon_{t-1}, \text{ with } C_D = \left( C_2 + C_\omega + 4 \frac{\beta n}{L} \right). \quad (45)$$

Note that we simply warm-start the dual coordinates for an outer iteration using the last iterate from the previous one. Yet, this leads to  $\theta_0^t = \theta_K^{t-1} + \beta \Sigma_\beta^{-1}(\omega_{t+1} - \omega_t)$ , as in Algorithm 2.

*Proof.* Equation (34) implies that  $D_t(\lambda)$  can be written as:

$$D_t(\lambda) = - \sum_{i=1}^n \frac{1}{\beta + \sigma_i} \left[ \frac{1}{2} (A\lambda)^{(i)} + \beta \omega_t^{(i)} \right]^\top (A\lambda)^{(i)} + R_{\text{comp}}(\lambda), \quad (46)$$

with  $R_{\text{comp}}(\lambda)$  that only depends on  $\lambda^{(ij)}$  and not on  $\omega_t^{(i)}$  for  $i \in \{1, \dots, n\}$ . Therefore,

$$\begin{aligned} D_t(\lambda_\star^t) - D_t(\lambda_K^{t-1}) \\ = D_{t-1}(\lambda_\star^t) - D_{t-1}(\lambda_K^{t-1}) - \beta \sum_{i=1}^n \left[ (A\lambda_\star^t)^{(i)} - (A\lambda_K^{t-1})^{(i)} \right]^\top \Sigma_\beta \left[ \omega_t^{(i)} - \omega_{t-1}^{(i)} \right]. \end{aligned}$$

Equation (30) writes  $(A\lambda_\star^t)^{(i)} = (\beta + \sigma_i)\theta_\star^t - \beta\omega_t^{(i)}$ , and so:

$$A\lambda_\star^t - A\lambda_K^{t-1} = A\lambda_\star^t - A\lambda_\star^{t-1} + A\lambda_\star^{t-1} - A\lambda_K^{t-1} = \Sigma_\beta^{-1}(\theta_\star^t - \theta_\star^{t-1}) + A\lambda_\star^{t-1} - A\lambda_K^{t-1} - \beta(\omega_t - \omega_{t-1}).$$

Then, we know from the equivalent reformulation of Equation (35) that  $\theta_\star^t = \arg \min F(\theta) + \frac{\beta}{2} \|\theta - \bar{\omega}_t\|^2$ , so using the 1-Lipschitzness of the proximal operator yields

$$\|\theta_\star^t - \theta_\star^{t-1}\|^2 \leq \|\bar{\omega}_t - \bar{\omega}_{t-1}\|^2 \leq \frac{1}{n} \sum_{k=1}^n \|\omega_t^{(k)} - \omega_{t-1}^{(k)}\|^2 = \frac{1}{n} \|\omega_t - \omega_{t-1}\|^2. \quad (47)$$

Similarly,  $\Sigma_\beta(A\lambda_\star^{t-1} - A\lambda_K^{t-1}) = \theta_\star^{t-1} - (\theta_K^{t-1})^{(i)}$ , and so:

$$\begin{aligned} \sum_{i=1}^n \left[ (A\lambda_\star^t)^{(i)} - (A\lambda_K^{t-1})^{(i)} \right]^\top \Sigma_\beta \left[ \omega_t^{(i)} - \omega_{t-1}^{(i)} \right] &\leq \sum_{i=1}^n \left\| \frac{(A\lambda_\star^t)^{(i)} - (A\lambda_K^{t-1})^{(i)}}{\beta + \sigma_i} \right\| \left\| \omega_t^{(i)} - \omega_{t-1}^{(i)} \right\| \\ &\leq \sum_{i=1}^n 2\|\theta_\star^t - \theta_\star^{t-1}\|^2 + 2\|\theta_\star^{t-1} - (\theta_K^{t-1})^{(i)}\|^2 + \left( \frac{\beta}{\beta + \sigma_i} + 4 \right) \|\omega_t^{(i)} - \omega_{t-1}^{(i)}\|^2. \end{aligned}$$

Plugging in Equation (47) yields:

$$D_t(\lambda_\star^t) - D_t(\lambda_K^{t-1}) \leq D_{t-1}(\lambda_\star^t) - D_{t-1}(\lambda_K^{t-1}) + 2\beta \sum_{i=1}^n \|(\theta_K^{t-1})^{(i)} - \theta_\star^{t-1}\|^2 + 7\beta \|\omega_t - \omega_{t-1}\|^2$$

Finally note that  $D_{t-1}(\lambda_\star^t) \leq D_{t-1}(\lambda_\star^{t-1})$  since  $\lambda_\star^{t-1}$  is the maximizer of  $D_{t-1}$ , and  $(\theta_K^{t-1})^{(i)} = \theta_t^{(i)}$  since it is the output of DVR after inner iteration  $t$ . The final expression is obtained using 6 and the recursion assumptions given by Equations (41) and (43).  $\square$

Finally, the warm-start error on the nodes parameters is given by the two following lemmas.

708 **Lemma 8** (Virtual parameters warm-starts). Denote  $\|\theta_1 - \theta_2\|_{\text{comp}}^2 = \sum_{i=1}^n \sum_{j=1}^m \|\theta_1^{(ij)} - \theta_2^{(ij)}\|^2$ .  
 709 Then,

$$\|\theta_0^t - \theta_\star^t\|_{\text{comp}}^2 \leq 2(C_\omega + 2mC_1)\varepsilon_{t-1}. \quad (48)$$

710 *Proof.* We use the fact that  $(\theta_t)^{(ij)} = (\theta_0^t)^{(ij)} = (\theta_K^{t-1})^{(ij)}$  to write:

$$\|\theta_0^t - \theta_\star^t\|_{\text{comp}}^2 = \|\theta_K^{t-1} - \theta_\star^{t-1} + \theta_\star^{t-1} - \theta_\star^t\|_{\text{comp}}^2 \leq 2\|\theta_t - \theta_\star^{t-1}\|_{\text{comp}}^2 + 2nm\|\theta_\star^{t-1} - \theta_\star^t\|^2.$$

711 Then, as before, the 1-Lipchitzness of the prox operator yields  $\|\theta_\star^{t-1} - \theta_\star^t\| \leq \frac{1}{n}\|\omega_t - \omega_{t-1}\|$ .  $\square$

712 **Lemma 9** (Parameters warm-start). Denote  $\|\theta_1 - \theta_2\|_{\text{comp}}^2 = \sum_{i=1}^n \sum_{j=1}^m \|\theta_1^{(ij)} - \theta_2^{(ij)}\|^2$ . Then,

$$\sum_{i=1}^n \|(\theta_0^t)^{(i)} - \theta_\star^t\|^2 \leq 6\left(C_\omega + \frac{n}{L}\right)\varepsilon_{t-1}. \quad (49)$$

713 *Proof.* We use the fact that since  $\lambda_0^t = \lambda_K^{t-1}$  then  $(\theta_0^t)^{(i)} = (\theta_0^t)^{(i)} + \frac{\beta}{\beta + \sigma_i}(\omega_t^{(i)} - \omega_{t-1}^{(i)})$  to write:

$$\begin{aligned} \sum_{i=1}^n \|(\theta_0^t)^{(i)} - \theta_\star^t\|^2 &\leq \sum_{i=1}^n \|(\theta_K^{t-1})^{(i)} - \theta_\star^{t-1} + \theta_\star^{t-1} - \theta_\star^t + \frac{\beta}{\sigma_i + \beta}(\omega_t^{(i)} - \omega_{t-1}^{(i)})\|^2 \\ &\leq 3\|\omega_t - \omega_{t-1}\|^2 + 3n\|\theta_\star^{t-1} - \theta_\star^t\|^2 + 3\sum_{i=1}^n \|(\theta_t)^{(i)} - \theta_\star^{t-1}\|^2. \end{aligned}$$

714  $\square$

715 We finish this part on warm starts by proving the following lemma, that links the initial dual parameters  
 716 error (computed with the Bregman divergence of  $\phi$ ), to the other parameters which we already know  
 717 how to control.

**Lemma 10** (Dual parameters warm-start, as measured by the Bregman divergence).

$$D_\phi(\lambda_\star^t, \lambda_0^t) \leq C_\phi \varepsilon_{t-1}, \quad (50)$$

718 with  $C_\phi = \frac{6(C_\omega + n/L) + L_{\max}^2(2C_\omega + 2mC_1)}{\lambda_{\min}^+(A^\top \Sigma_\beta^2 A)} + \frac{2L_{\max}(C_\omega + 2mC_1)}{\alpha}$ .

719 *Proof.* We first decompose the Bregman divergence as:

$$D_\phi(\lambda_0^t, \lambda_\star^t) \leq \frac{1}{2}\|(\lambda_0^t)^{\text{comm}} - (\lambda_\star^t)^{\text{comm}}\|_{A_{\text{comm}}^\dagger A_{\text{comm}}} + \sum_{i=1}^n \sum_{j=1}^m D_{\phi_{ij}}((\lambda_\star^t)^{(ij)}, (\lambda_0^t)^{(ij)}). \quad (51)$$

720 Then, we bound the communication term as:

$$\begin{aligned} \|(\lambda_0^t)^{\text{comm}} - (\lambda_\star^t)^{\text{comm}}\|_{A_{\text{comm}}^\dagger A_{\text{comm}}} &\leq \|\lambda_0^t - \lambda_\star^t\|_{A^\dagger A} \leq \frac{1}{\lambda_{\min}^+(A^\top \Sigma_\beta^2 A)} \|\Sigma_\beta A (\lambda_0^t - \lambda_\star^t)\|^2 \\ &= \frac{1}{\lambda_{\min}^+(A^\top \Sigma_\beta^2 A)} \left( \sum_{i=1}^n \|(\theta_0^t)^{(i)} - \theta_\star^t\|^2 + \sum_{i=1}^n \sum_{j=1}^m \mu_{ij}^2 \|(\lambda_0^t)^{(ij)} - (\lambda_\star^t)^{(ij)}\|^2 \right) \\ &= \frac{1}{\lambda_{\min}^+(A^\top \Sigma_\beta^2 A)} \left( \sum_{i=1}^n \|(\theta_0^t)^{(i)} - \theta_\star^t\|^2 + \sum_{i=1}^n \sum_{j=1}^m \|\nabla f_{ij}((\theta_0^t)^{(ij)}) - \nabla f_{ij}((\theta_\star^t)^{(ij)})\|^2 \right) \\ &= \frac{1}{\lambda_{\min}^+(A^\top \Sigma_\beta^2 A)} \left( \sum_{i=1}^n \|(\theta_0^t)^{(i)} - \theta_\star^t\|^2 + \sum_{i=1}^n \sum_{j=1}^m L_{ij}^2 \|(\theta_0^t)^{(ij)} - (\theta_\star^t)^{(ij)}\|^2 \right). \end{aligned}$$

721 Using Lemmas 8 and 9, we obtain:

$$\frac{1}{2}\|(\lambda_0^t)^{\text{comm}} - (\lambda_\star^t)^{\text{comm}}\|_{A_{\text{comm}}^\dagger A_{\text{comm}}} \leq \frac{(6(C_\omega + \frac{n}{L}) + L_{\max}^2(2C_\omega + 2mC_1))}{\lambda_{\min}^+(A^\top \Sigma_\beta^2 A)} \varepsilon_{t-1}. \quad (52)$$



For the computation part, we use the duality property of the Bregman divergence, which yields

$$\begin{aligned}
D_{\phi_{ij}}((\lambda_{\star}^t)^{(ij)}, (\lambda_0^t)^{(ij)}) &= \frac{L_{ij}}{\mu_{ij}^2} D_{f_{ij}^*}(\mu_{ij}(\lambda_{\star}^t)^{(ij)}, \mu_{ij}(\lambda_0^t)^{(ij)}) \\
&= \frac{L_{ij}}{\mu_{ij}^2} D_{f_{ij}}(\nabla f_{ij}(\mu_{ij}(\lambda_0^t)^{(ij)}), \nabla f_{ij}(\mu_{ij}(\lambda_{\star}^t)^{(ij)})) \\
&= \frac{L_{ij}}{\mu_{ij}^2} D_{f_{ij}}((\theta_0^t)^{(ij)}, (\theta_{\star}^t)^{(ij)}) \leq \frac{L_{ij}^2}{\mu_{ij}^2} \|(\theta_0^t)^{(ij)} - (\theta_{\star}^t)^{(ij)}\|^2
\end{aligned}$$

Therefore,

$$\sum_{i=1}^n \sum_{j=1}^m D_{\phi_{ij}}((\lambda_0^t)^{(ij)}, (\lambda_{\star}^t)^{(ij)}) \leq \frac{2L_{\max}(C_{\omega} + 2mC_1)}{\alpha} \varepsilon_{t-1}. \quad (53)$$

Substituting Equations (52) and (53) into Equation (51) finishes the proof.  $\square$

### C.2.2 Inner iteration error decrease

Now that we have bounded the error at the beginning of each outer iteration, we bound error at the end of each outer iteration by using the convergence results for DVR. We first prove the following Lemma, which controls the distance between the virtual parameters and the actual one:

**Lemma 11** (Virtual error decrease). *For all  $(i, j)$ ,*

$$\mathbb{E} \left[ \sum_{i,j} \|(\theta_{t+1}^t)^{(ij)} - \theta_{\star}^t\|^2 \right] \leq (1 - \rho)^K \left[ \|\theta_0^t - \theta_{\star}^t\|_{\text{comp}}^2 + \frac{\rho_{\text{sum}} K}{1 - \rho} C_0(t) \right]. \quad (54)$$

*Proof.* We cannot retrieve direct control over the  $\theta_{t+1}^{(ij)}$  from control over the dual variables or the dual error, since this would require the  $f_{ij}^*$  functions to be smooth, which they may not be. Yet, we leverage the fact that  $\theta_{t+1}^{(ij)}$  is obtained by a convex combination between  $\theta_t^{(ij)}$  and  $\theta_t^{(i)}$  to obtain convergence of to  $\theta_{\star}^t$ . We note  $j_{t,k}(i)$  the virtual node that is updated at time  $(t, k)$  for node  $i$ . We note  $\mathbb{E}_k$  the expectation relative to the value of  $j_{t,k}(i)$ . We start by remarking that:

$$\begin{aligned}
&\mathbb{E}_{k+1} \left[ \|(\theta_{k+1}^t)^{(ij)} - \theta_{\star}^t\|^2 \right] \\
&= (1 - p_{ij}) \|(\theta_k^t)^{(ij)} - \theta_{\star}^t\|^2 + p_{ij} \|(1 - \rho_{ij})(\theta_k^t)^{(ij)} + \rho_{ij}(\theta_k^t)^{(i)} - \theta_{\star}^t\|^2 \\
&\leq (1 - p_{ij}\rho_{ij}) \|(\theta_k^t)^{(ij)} - \theta_{\star}^t\|^2 + p_{ij}\rho_{ij} \|(\theta_k^t)^{(i)} - \theta_{\star}^t\|^2,
\end{aligned}$$

where in the last inequality we used the convexity of the squared norm. We use that  $p_{ij}\rho_{ij} \geq \rho$  (equal for the smallest one), and write that:

$$\mathbb{E} \left[ \|(\theta_K^t)^{(ij)} - \theta_{\star}^t\|^2 \right] \leq (1 - \rho)^K \|(\theta_0^t)^{(ij)} - \theta_{\star}^t\|^2 + p_{ij}\rho_{ij} \sum_{k=1}^K (1 - \rho)^{k-1} \|(\theta_{K-k}^t)^{(i)} - \theta_{\star}^t\|^2. \quad (55)$$

Noting  $\rho_{\text{sum}} = \max_i \sum_{j=1}^m \rho_{ij} p_{ij}$  and  $\|\theta_k^t - \theta_{\star}^t\|_{\text{comp}, i}^2 = \sum_{j=1}^m \|(\theta_k^t)^{(ij)} - \theta_{\star}^t\|^2$ , we obtain

$$\mathbb{E} \left[ \|\theta_K^t - \theta_{\star}^t\|_{\text{comp}, i}^2 \right] \leq (1 - \rho)^K \|\theta_0^t - \theta_{\star}^t\|_{\text{comp}, i}^2 + \rho_{\text{sum}} \sum_{k=1}^K (1 - \rho)^{k-1} \|(\theta_{K-k}^t)^{(i)} - \theta_{\star}^t\|^2. \quad (56)$$

Using Lemma 5, we know that  $\sum_{i=1}^n \|(\theta_k^t)^{(i)} - \theta_{\star}^t\|^2 \leq C_0(t)(1 - \rho)^k$ , with  $C_0(t)$  a constant that depends on the initial conditions of outer iteration  $t$ . Therefore,

$$\sum_{i=1}^n \sum_{k=1}^K (1 - \rho)^{k-1} \|(\theta_{K-k}^t)^{(i)} - \theta_{\star}^t\|^2 \leq K(1 - \rho)^{K-1} C_0(t). \quad (57)$$

In the end,

$$\mathbb{E} \left[ \|\theta_K^t - \theta_{\star}^t\|_{\text{comp}}^2 \right] \leq (1 - \rho)^K \left[ \|\theta_0^t - \theta_{\star}^t\|_{\text{comp}}^2 + \frac{\rho_{\text{sum}} K}{1 - \rho} C_0(t) \right]. \quad (58)$$

$\square$

742 This lemma has the following corollary:

743 **Corollary 1** (Warm-started virtual error decrease). *For all  $(i, j)$ ,*

$$\mathbb{E} \left[ \sum_{i,j} \|(\theta_{t+1})^{(ij)} - \theta_\star^t\|^2 \right] \leq (1 - \rho)^K \left[ 6 \left( C_\omega + \frac{n}{L} \right) + K \frac{\rho_{\text{sum}} C_{\text{comp}}}{1 - \rho} \right] \varepsilon_{t-1}, \quad (59)$$

with

$$C_{\text{comp}} = \frac{(\beta + \sigma_{\max} + L_{\max})}{(\sigma_{\min} + \beta)^2} \left( \frac{p_{\min}}{\eta_t} C_\phi + C_2 + C_\omega + 4 \frac{\beta n}{L} \right)$$

744 *Proof.* Using Lemmas 5, 10 and 7, we write:

$$C_0(t) = \frac{(\beta + \sigma_{\max} + L_{\max})}{(\sigma_{\min} + \beta)^2} \left( \frac{p_{\min}}{\eta_t} D_\phi(\lambda_\star^t, \lambda_0^t) + (D(\lambda_\star^t) - D(\lambda_0^t)) \right) \leq C_{\text{comp}} \varepsilon_{t-1}$$

745 We use Lemma 8 for the first term. □

**Lemma 12** (Condition on  $K$ ). *If Equations (41), (42) and (43) hold at time  $t$ , and  $K$  is such that:*

$$(1 - \rho)^K \leq \min \left( \frac{C_1(1 - \rho^{\text{out}})}{12(C_\omega + n/L)}, \frac{C_1(1 - \rho^{\text{out}})(1 - \rho)}{K \rho_{\text{sum}} C_{\text{comp}}}, \frac{C_2}{C_L}, \frac{n(\sigma_{\min} + \beta)^2}{2LC_L(\beta + \sigma_{\max} + L_{\max})} \right),$$

746 *then they also hold at time  $t + 1$ .*

*Proof.* Using Corollary 1, we obtain that if  $K$  is set such that

$$(1 - \rho)^K \left[ 6 \left( C_\omega + \frac{n}{L} \right) + K \frac{\rho_{\text{sum}} C_{\text{comp}}}{1 - \rho} \right] \leq C_1(1 - \rho^{\text{out}}),$$

747 then the recursion condition is respected for the virtual parameters. This yields the first and second  
748 conditions on  $K$ . Now, we write  $C_L = \left( \frac{p_{\min}}{\eta_t} C_\phi + C_D \right)$ , then using Lemmas 10 and 7 (where  $C_\phi$   
749 and  $C_D$  are defined), we obtain using Theorem 4 that

$$D_t(\lambda_\star^t) - D_t(\lambda_{t+1}) \leq C_L(1 - \rho)^K \varepsilon_{t-1},$$

750 since  $\lambda_{t+1}$  is obtained by performing  $K$  iterations of DVR to minimize  $F_t$  starting from  $\lambda_t$ . This  
751 yields the third condition on  $K$ . Finally, the last condition on  $K$  is obtained by leveraging Lemma 5.

752 □

## 753 D Experiments

754 For the experiments, the following logistic regression problem is solved:

$$\min_{\theta \in \mathbb{R}^d} \sum_{i=1}^n \left[ \frac{\sigma}{2} \|\theta\|^2 + \sum_{j=1}^m \frac{1}{m} \log(1 + \exp(-y_{ij} X_{ij}^\top \theta)) \right], \quad (60)$$

755 where the pairs  $(X_{ij}, y_{ij}) \in \mathbb{R}^d \times \{-1, 1\}$  are taken from the RCV1 dataset, which we downloaded  
756 from <https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html>.

757 Figure 2 is the full version of Figure 1, in which we report the number of individual gradients  
758 and number of communications for each configuration. We see that accelerated EXTRA actually  
759 outperforms EXTRA when the regularization is small, as already mentioned in the main text. We also  
760 see that Accelerated EXTRA and Accelerated DVR have comparable communication complexity on  
761 the grid graph, when  $\gamma$  is smaller. Yet, the computation complexity of (accelerated) DVR is much  
762 smaller, so accelerated DVR is much faster overall as long as  $\tau$  is not too big.

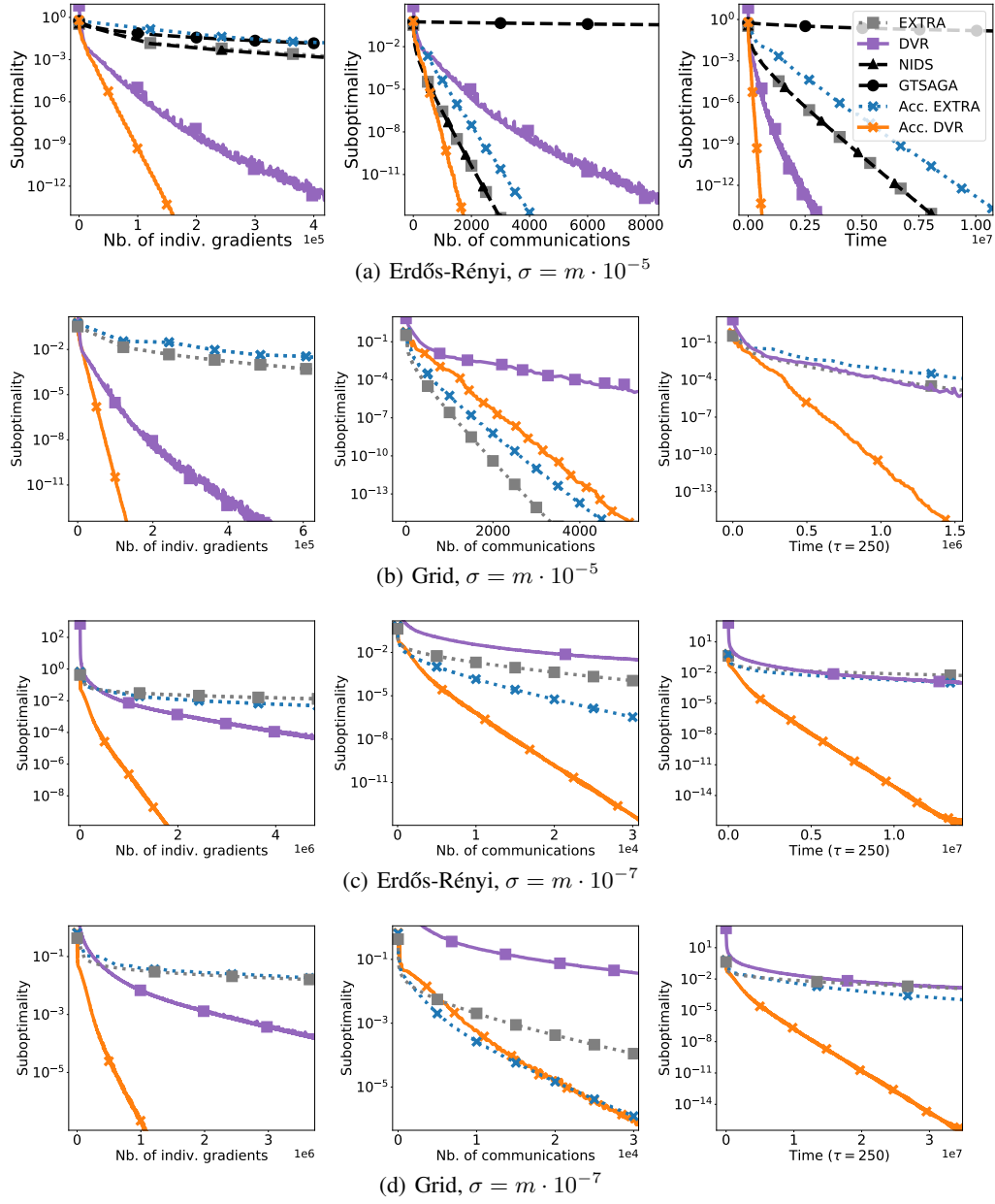


Figure 2: Experimental results for the RCV1 dataset with different graphs of size  $n = 81$ , with  $m = 2430$  samples per node, and with different regularization parameters.