Stochastic Mirror Descent for Convex Optimization with Consensus Constraints

A. Borovykh, N. Kantas, P. Parpas, G. A. Pavliotis

Abstract. The mirror descent algorithm is known to be effective in situations where it is beneficial to adapt the mirror map to the underlying geometry of the optimization model. However, the effect of mirror maps on the geometry of distributed optimization problems has not been previously addressed. In this paper we study an exact distributed mirror descent algorithm in continuous-time under additive noise. We establish a linear convergence rate of the proposed dynamics for the setting of convex optimization. Our analysis draws motivation from the Augmented Lagrangian and its relation to gradient tracking. To further explore the benefits of mirror maps in a distributed setting we present a preconditioned variant of our algorithm with an additional mirror map over the Lagrangian dual variables. This allows our method to adapt to both the geometry of the primal variables, as well as to the geometry of the consensus constraint. We also propose a Gauss-Seidel type discretization scheme for the proposed method and establish its linear convergence rate. For certain classes of problems we identify mirror maps that mitigate the effect of the graph's spectral properties on the convergence rate of the algorithm. Using numerical experiments we demonstrate the efficiency of the methodology on convex models, both with and without constraints. Our findings show that the proposed method outperforms other methods, especially in scenarios where the model's geometry is not captured by the standard Euclidean norm.

Key words. Distributed optimization, mirror descent, pre-conditioning, interacting particles, stochastic optimization.

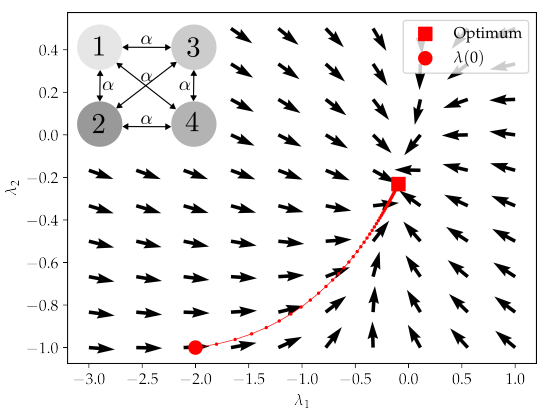
AMS subject classifications. 65K10, 68Q87, 60G07, 68W15

1. Introduction.

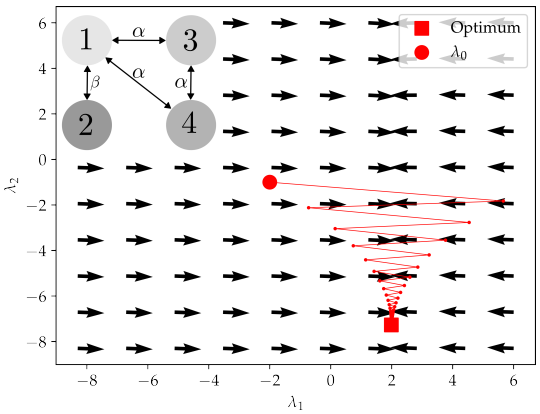
The choice of mirror map has a significant impact on both the theoretical and numerical performance of the Mirror Descent (MD) algorithm [3, 7, 29]. With an appropriate choice of the mirror map, MD captures the geometry of the optimization model more faithfully than other first-order methods, and for certain classes of problems it is known to outperform other methods both in theory and in practice [3, 7]. There is an extensive literature on the mirror descent algorithm. However, the effect of the choice of the mirror map for *distributed* optimization problems has received much less attention (see Section 1.1 for related work). Distributed optimization problems, even when otherwise unconstrained, have to satisfy a consensus constraint. Existing algorithms do not capture the geometry of the consensus constraint. Motivated by the attractive theoretical and real-world performance of the mirror descent algorithm, in this paper we attempt to answer the following question: *Does there exist a distributed variant of mirror descent that can accurately capture the geometry of distributed optimization models?* To answer this question, we propose a distributed mirror descent algorithm for the following optimization model,

$$f^\star = \min_{x^i \in \mathcal{X}} f(\mathbf{x}) = \sum_{i=1}^{N} f_i(x^i), \quad \text{s.t. } \mathcal{L}\mathbf{x} = 0. \tag{1.1}$$

Where f_i denotes the objective function of node or particle i, $x^i \in \mathcal{X} \subset \mathbb{R}^d$, $i = 1, \dots, N$. When it is not necessary to distinguish between the particles we write $\mathbf{x}^\top = [x^1 \ \dots \ x^N]$ with $\mathbf{x} \in \mathcal{X}^N$,



(a) Dual ascent with a complete interaction graph $\alpha = 1$.



(b) Dual ascent with an incomplete interaction graph $\alpha = 1$, $\beta = 0.1$.

Figure 1: Vector fields of the gradient of the dual function in (1.2) for (a) a complete graph and (b) an incomplete graph. The two graphs are shown inset in the left-hand side corners of (a) and (b) respectively (for details see Example 1.1). In (a) the Laplacian of the graph is well-conditioned, in (b) the Laplacian is ill-conditioned. An example solution trajectory, obtained using an explicit discretization scheme, is shown in red for both cases. The explicit discretization scheme leads to oscillatory behavior and to eliminate it a small step-size needs to be used resulting in a slow convergence rate.

where $\mathcal{X}^N = \mathcal{X} \times \ldots \times \mathcal{X}$. We assume that the particles communicate through a strongly connected, weighted, undirected graph $\mathcal{G} := (V, E, A)$. We use V to represent the nodes of the graph, E its edges, A is the weighted adjacency matrix and $\mathcal{L}$ is the graph's Laplacian (see Subsection 3.1 for exact definitions). Note that $\mathcal{L}\mathbf{x} = 0 \iff x_i = x_j, \forall (i,j) \in E$. We assume that each particle has access to its own objective function $f_i : \mathbb{R}^d \to \mathbb{R}$, and constraint set $\mathcal{X} \subset \mathbb{R}^d$.

In mirror descent algorithms the mirror map is selected according to the geometry of $\mathcal{X}$, and is independent of the consensus constraint. Unlike the $\mathcal{X}$ constraint, the consensus constraint couples all the particles together, and can therefore have a significant impact on the convergence properties of distributed optimization methods for (1.1). The impact of the consensus constraint becomes apparent by examining the Lagrangian dual function of (1.1) given by,

$$q(\boldsymbol{\lambda}) = \inf_{\mathbf{x} \in \mathcal{X}^N} \left\{ f(\mathbf{x}) + \boldsymbol{\lambda}^\top \mathcal{L}\mathbf{x} \right\}. \tag{1.2}$$

Under the usual convexity assumptions strong duality holds i.e. $f^\star = \sup_{\boldsymbol{\lambda} \in \mathbb{R}^{Nd}} \{q(\boldsymbol{\lambda})\}$ and $q(\boldsymbol{\lambda})$ is concave and differentiable (see e.g. [2, Section 6.3]). Therefore, a natural starting point for a mirror descent algorithm is to generalize the dual-ascent method,

$$\dot{\boldsymbol{\lambda}}_t = \nabla_{\boldsymbol{\lambda}} q(\boldsymbol{\lambda}_t), \tag{1.3}$$

by introducing a mirror map for the Lagrange multipliers $\boldsymbol{\lambda}$. The example below illustrates the challenges associated with a naive application of the dual ascent dynamics in (1.3) to distributed optimization problems.

Example 1.1. *Consider a simple distributed optimization problem in the form of* (1.1) *where the objective function is given by the quadratic* $\sum_{i=1}^{4}(x^i - b^i)^2$*, with* $x^i \in \mathbb{R}$*, and* $b^i \in \mathbb{R}$ *chosen randomly. To illustrate the importance of the spectral properties of* $\mathcal{L}$ *we consider two different strongly connected graphs with* $N = 4$ *nodes. In Figure* 1a *we visualize a* $2D$ *projection of the vector field of* $\nabla q(\lambda)$ *for the fully connected graph shown inset. We obtained the* $2D$ *projection on* λ_1 *and* λ_2 *by fixing the other two Lagrange multipliers to their optimal value. The red line shows the path to a particular numerical solution of* (1.3) *when the explicit Euler method is used to solve* (1.3)*. Now consider the same problem but with a different strongly connected but incomplete graph. Such a graph is shown in the inset of Figure* 1b*. In this case the vector field of* (1.3) *has completely different qualitative features. In Figure* 1b *we observe that the optimal solution is along* $\lambda_2 = 2$*, and all paths reach this point after a* rapid transient phase*. Again, the red line in Figure* 1b *shows the trajectory of the explicit Euler discretization scheme applied to* (1.3)*. The oscillations in the solution trajectory are a typical feature of stiff dynamical systems (see [45, IV.1]). It is well known that unless a small step-size is used it is common to observe oscillations in the numerical solution of stiff differential equations. Stiff dynamics present a significant challenge to the development of computationally efficient algorithms because optimization algorithms typically rely on explicit discretization schemes. The problem is even worse in the stochastic setting where constant step-sizes are used. We will revisit this example in Section* 4 *and show that even when using an explicit discretization scheme the proposed algorithm converges in a single step independently of the graph's Laplacian.*

Guided by the observations above, it is tempting to consider the following natural generalization of the Mirror Descent algorithm,

$$\partial_t \Psi(\boldsymbol{\lambda}_t) = \nabla_{\boldsymbol{\lambda}} q(\boldsymbol{\lambda}_t). \tag{1.4}$$

One could then select the mirror map for the dual variables, Ψ, in such a way as to better reflect the geometry of the interaction graph $\mathcal{G}$. An additional advantage of such an approach is that we could apply existing results from the literature to establish both the convergence and convergence rate of (1.4). However, there are two significant challenges associated with the introduction of the mirror map Ψ to the dual-ascent dynamics of (1.3). Firstly, computing the gradient of the dual requires the solution of the maximization problem in (1.2). Requiring the solution of an optimization problem in every iteration is only possible for problems with specific structure. Therefore using (1.4) may be computationally infeasible for many problems. Secondly, even if the objective function is strongly convex the dual function in (1.2) is concave and not strongly concave. The dual function would be strongly concave if the Laplacian matrix had full rank. It is well known that the Laplacian matrix associated with a graph $\mathcal{G}$ does not have full rank (see [8, Theorem 6.6]). Therefore applying existing convergence results to (1.4) would only guarantee a sub-linear rate such as $O(1/T)$. The objective of this paper is to address these challenges and to do so for the case where only noisy estimates of the gradients are available. Our results are based on a continuous-time analysis of distributed

stochastic mirror descent dynamics from a dynamical systems perspective. Our contributions are summarized below.

- We propose a variant of the distributed mirror descent algorithm called Exact Preconditioned Interacting Stochastic Mirror Descent (EPISMD), that is able to converge exponentially fast to a neighborhood of the solution. The neighborhood can be made arbitrary small in the case where there is no noise.
- In order to avoid the computational challenges of computing the gradient of the dual we use an Augmented Lagrangian formulation. For distributed gradient descent methods the links between the Augmented Lagrangian, gradient tracking and distributed optimization are well known. The novel aspect here is the analysis in the context of mirror descent methods where the link between the Augmented Lagrangian and the dynamics of mirror descent are less developed.
- We avoid the technical difficulty associated with the fact that the Laplacian matrix is positive semi-definite (and not positive definite) by showing that if the algorithm is initialized correctly then it will never visit a solution that is in the null space of the Laplacian matrix (see Lemma 4.4).
- The analysis is performed under general assumptions regarding the choice of mirror maps. We propose a Gauss-Seidel type discretization scheme and show that it too converges exponentially fast (Proposition (4.6)). We also identify suitable mirror maps that enable an improved convergence rate when the interaction graph is ill-conditioned (see Section 4). We note that the discretization analysis is based on the Lyapunov function identified from the continuous time analysis.
- We use the insights obtained from the properties of the dual function in order to propose a suitable mirror map for the Lagrange multipliers. See Subsection 3.3 for motivation, Section 4 for the analysis in discrete time and Section 5 for a numerical implementation.
- In Section 5 we illustrate the performance of the proposed algorithm in constrained and unconstrained convex optimization problems. The performance of the proposed method is in agreement with the theoretical results and also outperforms other state-of-the-art algorithms. When the graph is ill-conditioned then the proposed algorithm can be several orders of magnitude faster than other state-of-the-art methods.

1.1. Previous work. Distributed optimization has a variety of applications. A classic reference for distributed optimization is [4], and more recent applications in statistical learning are described in [6]. The authors in [8] also describe several interesting applications. The literature on distributed optimization algorithms is vast. Since this paper focuses on exact distributed first-order algorithms for convex optimization models, we will focus on this class of algorithms. Note that in this context an exact algorithm is one that, in the absence of noise, converges to a solution of (1.1). Two algorithmic techniques can be used to develop exact distributed optimization algorithms. The first technique uses diminishing step-sizes, and the second one relies on gradient tracking. Gradient tracking is closely related to Augmented Lagrangian methods, however we remark that in the case of mirror maps the relationship between the two is less straightforward due to the mirrored variables. Algorithms with diminishing step-sizes tend to be very slow. So recent literature focuses on using constant step-sizes in combination

Reference	Mirror (Primal,Dual)	Linear Rate	Noise	Constant step
Liang et al. [19]	$(\times, \times)$	$\times$	$\times$	$\times$
[20, 15, 21, 47]	$(\times, \times)$	$\times$	$\times$	$\times$
[11, 46]	$(\times, \times)$	$\times$	✓	✓
Ram et. al. [34]	$(\times, \times)$	$\times$	✓	✓
Shi et al. [37]	$(\times, \times)$	✓	$\times$	✓
Qu & Li [32]	$(\times, \times)$	✓	$\times$	✓
Pu & Nedic [31], Sun et al.[40]	$(\times, \times)$	✓	✓	✓
Duchi et al. [13]	$(\checkmark, \times)$	$\times$	✓	✓
Nedic et al. [28]	$(\checkmark, \times)$	$\times$	✓	$\times$
Shahrampour et al. [35]	$(\checkmark, \times)$	$\times$	✓	$\times$
Sun et. al. [41, 42]	$(\checkmark, \times)$	$\times$	$\times$	✓
Sun et. al. [39]	$(\checkmark, \times)$	✓	$\times$	✓
This work	$(\checkmark, \checkmark)$	✓	✓	✓

Table 1: Overview of related work. Note that linear convergence rate refers to a global linear rate.

with gradient tracking. In Table 1, we summarize selected related works that show how this paper fits within the existing literature. The top half of the table shows that there are many distributed algorithms that are based on the gradient descent method. Many of these works achieve a global linear convergence rate while also addressing the case of noisy gradients. As noted in the introduction there are many settings (e.g. optimization over the simplex) where mirror descent algorithms outperform projected gradient methods. For this reason there have been many attempts to generalize mirror descent to the distributed setting. The bottom half of the table summarizes various recent works that study the mirror descent algorithm in the distributed setting. Among these only [39] is an exact algorithm with linear convergence but does not deal with noisy gradient evaluations or mirroring of the Lagrangian dual variables. Moreover, the proof in [39] is based on the solution of a Semi-Definite Programing (SDP) optimization model, and it is unclear if the proof technique can be extended to deal with noisy gradient evaluations. Therefore compared to existing works, the algorithm we propose in this paper is developed in both continuous and discrete time, allows for additive Brownian noise, and achieves a global linear convergence rate, while being exact in the no-noise setting. Moreover, the mirror maps in the existing literature are used only to model the geometry of the separable constraints and/or the local objective functions and not the consensus constraint that is the main focus and distinctive feature of this paper. We achieve the latter by allowing for an additional mirror map for the Lagrange multipliers of the consensus constraint. For our analysis we use a Lyapunov function that suitably combines both the preconditioning of the primal and Lagrangian dual variables to attain non-asymptotic convergence results.

1.2. Notation. We use $\otimes$ to denote the Kronecker product, I_d the d-dimensional identity matrix and $\mathbf{1}_d$ denotes the d-dimensional vector of ones. Diag(a) with $a \in \mathbb{R}^d$ denotes a matrix with diagonal elements $[a_1, ..., a_d]$. We use A to denote the $N \times N$ weighted adjacency matrix associated with a graph $\mathcal{G} = (V, E)$. The graph Laplacian is given by $L := \text{Diag}(A\mathbf{1}_N) - A$ and we use the following notation $\mathcal{L} := L \otimes I_d$ with $\mathcal{L} \in \mathbb{R}^{Nd \times Nd}$ to denote the vectorized version of the graph Laplacian. We use $\langle x, y\rangle = x^\top y$ for the standard dot product, and $\langle x, y\rangle_Q = \langle x, Qy\rangle = x^\top Qy$ for the Q-inner product, for some positive definite matrix Q. We use $A \succeq B$ to denote that $A - B$ is positive semi-definite. We assume that $\mathcal{X} \subseteq \mathbb{R}^d$ is a convex set. We use $\mathcal{D}$ to denote an open set such that $\mathcal{X} \subset \text{cl}(\mathcal{D})$. The set $\mathcal{D}$ will be used to denote the domain of the mirror maps of the mirror descent algorithm. We use $\mathcal{X}^*$ to denote the dual space of $\mathcal{X}$. The normal cone of $\mathcal{X}$ is defined as $N_{\mathcal{X}}(x) = \{z \in \mathcal{X}^* \mid \langle z, y - x\rangle \leq 0 \ \ \forall y \in \mathcal{X}\}$.

Given an arbitrary norm $||\cdot||$ on $\mathbb{R}^d$, we will define $B_{\|\cdot\|} := \{v \in \mathbb{R}^d : \|v\| \leq 1\}$. The dual norm $\|\cdot\|_*$ is defined as $\|z\|_* := \sup\{\langle z, v\rangle : v \in B_{\|\cdot\|}\}$. If A is a matrix then $\|A\|_2$ denotes its spectral norm and we assume that the dual norm is compatible with the spectral norm, i.e. $\|Az\|_* \leq \|A\|_2 \, \|z\|_*$. We will make use of the following generalized Cauchy inequality,

$$|\langle v, w\rangle| \leq \|v\|_* \|w\| \quad \forall w \in \mathcal{X}, \ v \in \mathcal{X}^*.$$

Since $0 \leq (\|v\|_* - \|w\|)^2 = \|v\|_*^2 + \|w\|^2 - 2\|v\|_*\|w\|$, we also have,

$$|\langle v, w\rangle| \leq \frac{1}{2}\|v\|_*^2 + \frac{1}{2}\|w\|^2. \tag{1.5}$$

A function g is said to be L-Lipschitz continuous with respect to a norm $\|\cdot\|$ if $\|g(x) - g(y)\| \leq L\|x - y\|$, $\forall x, y \in \mathcal{X}$. The Bregman divergence associated with a convex, differentiable function $g : \mathcal{X} \to \mathbb{R}$ is defined as follows,

$$D_g(x, y) = g(x) - g(y) - \langle \nabla g(y), x - y\rangle.$$

If the second-order derivative of g exists it furthermore holds,

$$\nabla_x D_g(x, y) = \nabla g(x) - \nabla g(y), \ \ \nabla_y D_g(x, y) = \nabla^2 g(y)(y - x).$$

The aggregate cost function will be written as $f(\mathbf{x}) = \sum_{i=1}^N f_i(x^i)$, where $\mathbf{x} = [x^1, \ldots, x^N]^T$ denotes the stacked vector of particles and each $x^i \in \mathcal{X}$. For a function $f : \mathcal{X}^N \mapsto \mathbb{R}$, unless specified otherwise gradient vectors ∇f are taken with respect to the joint particle vector $\mathbf{x}$ following the convention that $\nabla f(\mathbf{x})^\top = [\nabla f_1(x_1) \ \nabla f_2(x_2) \ \ldots \ \nabla f_N(x_N)]$ and for Hessian matrices we write $\nabla^2 f(\mathbf{x}) = \text{diag}(\nabla^2 f_1(x_1), \nabla^2 f_2(x_2), \ \ldots, \nabla^2 f_N(x_N))$. We will use $(X^\star, \Lambda^\star)$ to denote the set of primal-dual variables that satisfy the first order optimality conditions for (1.1).

Remark 1.2. *The space of the Lagrange multipliers for the consensus constraint, $\boldsymbol{\lambda} \in \Lambda \subset \mathbb{R}^{Nd}$, will play an important role in the definition of the algorithms below. We note that the norm associated with $\boldsymbol{\lambda} \in \Lambda$ will not necessarily be the same as the one used for the primal variables $\mathbf{x} \in \mathcal{X}^N$. We will however use the same notation: $\|\cdot\|$, and its dual $\|\cdot\|_*$ for both spaces, and it will be clear from context which norm is being used. For $\mathbf{w} = [\mathbf{x}^\top, \ \boldsymbol{\lambda}^\top]^\top$ we*

will use the following mixed norm convention $\|\mathbf{w}\| = \|\mathbf{x}\| + \|\boldsymbol{\lambda}\|$*, with the understanding that the two norms could be different. For example, the norm in* $\mathcal{X}^N$ *could be the* ℓ_1*, and in* Λ *the* Q*-norm (for some positive definite matrix* Q*) so that,* $\|\mathbf{w}\| = \|\mathbf{x}\|_1 + \|\boldsymbol{\lambda}\|_Q$.

2. Distributed Stochastic Mirror Descent: Exact and Preconditioned Dynamics. In this section we introduce the proposed distributed MD algorithm. We adopt a dynamical systems point of view for our analysis. The discretization of the proposed method is discussed in Section 4 and we report on numerical experiments in Section 5. For an introduction to the original mirror descent algorithm we refer the interested reader to [3, Ch. 9]. We also remark that Langevin dynamics with mirror maps have been used in the sampling literature [48, 10, 18] but the the primary emphasis in this paper is on optimization.

The most widely studied algorithm for distributed Mirror Descent is the Interacting Stochastic Mirror Descent (ISMD) algorithm (also known as distributed dual averaging). In the continuous time setting the dynamics of ISMD are as follows,

$$dz_t^i = -\eta \nabla f_i(x_t^i)dt + \epsilon \sum_{j=1}^{N} A_{ij}(z_t^j - z_t^i)dt + \sigma dB_t^i, \quad x_t^i = \nabla \Phi^*(z_t^i), \tag{2.1}$$

for particles $i = 1, ..., N$, and where B_t^i are independent Brownian motions. The matrix $A = \{A_{ij}\}_{i,j=1}^N$ is an $N \times N$ doubly-stochastic matrix representing the interaction weights and η, ϵ are tuning constants representing the learning rate and interaction strength, respectively. For simplicity in most of the subsequent analysis we set $\eta = \epsilon = 1$, but it is straightforward to extend the results to arbitrary values of η and ϵ. In the context of modern large scale applications, we note that understanding convergence under the presence of noise is often motivated from computational considerations such as when sub-sampling the gradient of f or sub-sampling the interaction graph.

Using the graph Laplacian, we can rewrite the evolution in vector form as,

$$d\mathbf{z}_t = (-\nabla f(\mathbf{x}_t) - \mathcal{L}\mathbf{z}_t)\, dt + \sigma d\mathbf{B}_t, \;\; \mathbf{x}_t = \nabla \Phi^*(\mathbf{z}_t), \tag{2.2}$$

where $\mathbf{B}_t := ((B_t^1)^T, ..., (B_t^N)^T)^T$. If $\Phi = I$ then the algorithm reduces to the standard Distributed Gradient Descent (DGD) in continuous time. This algorithm was proposed in [33] but its convergence was only established for strongly convex quadratic functions. The case where all the functions are strongly convex and identical (i.e. $f_1 = \ldots = f_N$), but not necessarily quadratic, was analyzed in [5]. The discrete time version of the algorithm for the general convex case was discussed in [13], but exact convergence was only established under a diminishing step-size strategy. Since we could not find a convergence analysis of (2.2) we provide a convergence proof in Appendix A that shows that the dynamics of (2.2) will converge to an approximate solution of (1.1).

It is known that DGD will not converge to the exact solution of (1.1). When $\sigma = 0$ and $\Phi = I$ then this observation follows from the fact that the fixed points of (2.2) do not coincide with the first-order necessary and sufficient conditions for optimality for the optimization problem in (1.1); see [37, 36] for more details in this case. In Appendix A.1 we adapt arguments of [37, 36] to show that the dynamics in (2.2) also fail to converge to the exact solution of

(1.1) and that the convergence of (2.2) to the exact solution of (1.1) can only occur under additional and restrictive assumptions on f. In simple terms, our result indicates that one cannot make the dynamics of DGD exact just by an appropriate choice of mirror map.

2.1. Exact Preconditioned Interacting Stochastic Mirror Descent (EPISMD). The ISMD algorithm described above has two limitations: (1) even if $\sigma = 0$ the stable points of (2.2) do not coincide with the ones of (1.1) and (2) the mirror map Φ is only chosen to reflect the geometry of $\mathcal{X}$ and not the consensus constraint. To address the first limitation we propose to exploit the links between the constrained optimization problem in (1.1) and its Augmented Lagrangian,

$$L(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \langle \boldsymbol{\lambda}, \mathcal{L}\mathbf{x} \rangle + \frac{1}{2}\|\mathcal{L}^{\frac{1}{2}}\mathbf{x}\|_2^2. \tag{2.3}$$

Since we are in a convex setting we have by strong duality that $f^\star = \max_{\boldsymbol{\lambda}} \min_{\mathbf{x}} L(\mathbf{x}, \boldsymbol{\lambda})$. To address the first shortcoming it is entirely natural to perform a descent step on (2.3) w.r.t $\mathbf{x}$ and an ascent step w.r.t the Lagrange multipliers $\boldsymbol{\lambda}$. Note that the Augmented Lagrangian in the form of (2.3) is preferred to the classical Lagrangian because of the regularizing effect of the quadratic term during the ascent/descent steps (see [2, Chapter 9]). To address the second shortcoming we use two mirror maps. The first mirror map Φ, is used to reflect the geometry of the primal space $\mathcal{X}$. Then, a second mirror map, Ψ, is used to adapt to the geometry associated with the consensus constraint in (1.1). We use $\boldsymbol{\mu}$ to denote the dual of the Lagrange multiplier, i.e. $\boldsymbol{\mu} = \nabla\Psi(\boldsymbol{\lambda})$. Figure 2 explains the main steps in mirror descent with the two maps. At time-step t the primal-dual pair $(\mathbf{x}_t, \boldsymbol{\lambda}_t)$ is mapped to $(\mathbf{z}_t, \boldsymbol{\mu}_t) = (\nabla\Phi(\mathbf{x}_t), \nabla\Psi(\boldsymbol{\lambda}_t))$. The inverse $(\nabla\Phi^{-1}, \nabla\Psi^{-1})$ maps the duals back to the primal space $\mathcal{X} \times \Lambda$. The proposed algorithm, is given below,

$$\begin{aligned} &d\mathbf{z}_t = -\nabla f(\mathbf{x}_t)dt - \mathcal{L}\mathbf{x}_t dt - \mathcal{L}\boldsymbol{\lambda}_t dt + \sigma d\mathbf{B}_t, \quad \mathbf{x}_t = \nabla\Phi^*(\mathbf{z}_t) \\ &d\boldsymbol{\mu}_t = \mathcal{L}\mathbf{x}_t dt, \quad \boldsymbol{\lambda}_t = \nabla\Psi^*(\boldsymbol{\mu}_t). \end{aligned} \tag{2.4}$$

The motivation for the dynamics above stems from the Bregman Augmented Lagrangian. One can view the dynamics in (2.4) as performing a descent step with respect to primal variables $\mathbf{x}$ and an ascent step with respect to the Lagrangian dual variables. To see the idea behind this intuition, we note that in the noise-free case ($\sigma = 0$) we can characterize (2.4) as a limiting case of the classical proximal algorithm of [9]

$$\begin{aligned} \mathbf{z}_{t+\delta} &= \inf_{\mathbf{y}} \left\{ L(\mathbf{y}, \lambda_t) + \frac{1}{\delta} D_\Phi(\mathbf{y}, \mathbf{x}_t) \right\}, \\ \boldsymbol{\mu}_{t+\delta} &= \sup_{\nu} \left\{ L(\mathbf{x}_t, \nu) - \frac{1}{\delta} D_\Psi(\nu, \boldsymbol{\lambda}_t) \right\}. \end{aligned}$$

Applying the first order optimality conditions to the two optimization problems above and taking formally the limit $\delta \downarrow 0$ we can derive (2.4) with $\sigma = 0$; see [1] for a more rigorous exposition.

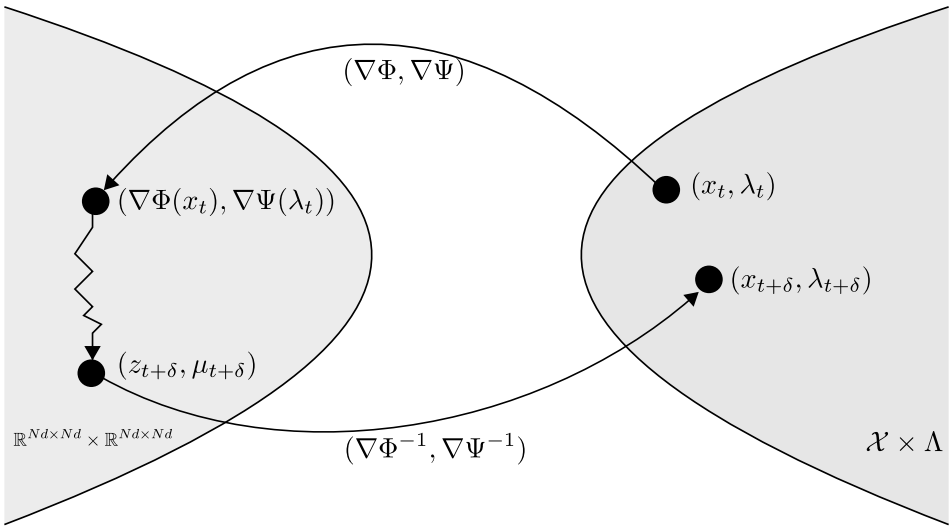


Figure 2: Stochastic Mirror Descent with two mirror maps. Φ maps the primal variables to the dual space, and Ψ maps the Lagrangian dual variables associated with the consensus constraint to the dual of the Lagrange multipliers.

3. Convergence Analysis of Distributed Stochastic Mirror Descent: Exact and Preconditioned Dynamics.

In this section we show that the dynamics of EPISMD in (2.4) converge exponentially fast to an area of the optimum. This area depends on σ. If $\sigma = 0$ the algorithm converges to the exact solution of (1.1). We start this section by collecting all our assumptions and some preliminary results that will be useful later on.

3.1. Preliminaries and Assumptions.

We split our assumptions into three broad categories. We start in Subsection 3.1.1 with our assumptions regarding the optimization model. In Subsection 3.1.2 we state our assumptions regarding the network, and finally in Subsection 3.1.3 we discuss our assumptions for the mirror maps. These assumptions will hold throughout the paper. We make some additional assumptions in Section 4 when we introduce our discretization scheme. We also state some technical lemmas that will be useful for the analysis of the method and whose proofs are presented in Appendix A.

3.1.1. Optimality Conditions and Model Assumptions.

The consensus constraint in (1.1) is satisfied if and only if $\mathcal{L}\mathbf{x} = 0$, where $\mathcal{L}$ denotes the vectorized graph Laplacian. Therefore the optimality conditions for (1.1) are as follows,

$$-\nabla f(\mathbf{x}^\star) - \mathcal{L}\boldsymbol{\lambda}^\star \in N_{\mathcal{X}^N}(\mathbf{x}^\star),$$
$$\mathcal{L}\mathbf{x}^\star = 0.$$

We make the following assumptions regarding the objective function and solution set.

Assumption 1. *Each f_i in (1.1) is proper, convex and twice differentiable. The elements of $X^\star$ are in the interior of $\mathcal{X}$.*

If the solution of (1.1) is in the interior of $\mathcal{X}$, and if f is convex, then we must have that $N_{\mathcal{X}}(\mathbf{x}^\star) = \{\mathbf{0}\}$ for any $\mathbf{x}^\star \in X^\star$. Because the focus of this paper is on the effect of the consensus constraint and its impact on the dynamics of the algorithm, we will assume that the optimal solution of (1.1) is in the interior of $\mathcal{X}$. For certain applications, especially in machine learning, the assumption that the solution lies in the interior of the feasible set holds (e.g. [25, 38]). The extension to the general case requires modifications to our convergence analysis similar to [25] and will be an interesting extension to our work. The derivatives of f are required so that we can apply Itô's formula.

Definition 3.1. *We say that $f : \mathcal{X}^N \to \mathbb{R}$ is μ-strongly convex w.r.t. some norm $||\cdot||$ provided that $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_* \geq \mu\|\mathbf{x}-\mathbf{y}\|$. Similarly, a function f is L-smooth w.r.t. some norm $\|\cdot\|$ when $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_* \leq L\|\mathbf{x}-\mathbf{y}\|$.*

Some of our results will use the notion of relative strong convexity and smoothness. We refer the reader to [23]. Below we present some definitions and properties that will be useful later on.

Definition 3.2 (Relative strong convexity). *A function $g : \mathcal{X}^N \to \mathbb{R}$ is μ-strongly convex with respect to some convex function h if for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}^N$ the following holds,*

$$g(\mathbf{x}) \geq g(\mathbf{y}) + \nabla g(\mathbf{y})^T(\mathbf{x}-\mathbf{y}) + \mu D_h(\mathbf{x}, \mathbf{y}).$$

Or equivalently, $\langle \mathbf{x}-\mathbf{y}, \nabla g(\mathbf{x}) - \nabla g(\mathbf{y})\rangle \geq \mu\langle \mathbf{x}-\mathbf{y}, \nabla h(\mathbf{x}) - \nabla h(\mathbf{y})\rangle$.

Definition 3.3 (Relative smoothness). *A function $g : \mathcal{X}^N \to \mathbb{R}$ is α-smooth with respect to some function h if for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}^N$ the following holds,*

$$g(\mathbf{x}) \leq g(\mathbf{y}) + \nabla g(\mathbf{y})^T(\mathbf{x}-\mathbf{y}) + \alpha D_h(\mathbf{x}, \mathbf{y}).$$

Or equivalently, $\langle \mathbf{x}-\mathbf{y}, \nabla g(\mathbf{x}) - \nabla g(\mathbf{y})\rangle \leq \alpha\langle \mathbf{x}-\mathbf{y}, \nabla h(\mathbf{x}) - \nabla h(\mathbf{y})\rangle$.

If we assume that g is μ-strongly convex and α-smooth with respect to h it holds,

$$\mu D_h(\mathbf{x}, \mathbf{y}) \leq D_g(\mathbf{x}, \mathbf{y}) \leq \alpha D_h(\mathbf{x}, \mathbf{y}).$$

We adopt the following definition for the convex conjugate of a relatively strong convex function.

Definition 3.4 (Convex conjugate). *Let $g : \mathcal{X}^N \to \mathbb{R}$ be a μ-strongly convex function with respect to some h. Then $g^*(\mathbf{z}) := \max_{\mathbf{x}\in\mathcal{X}^N}\{\langle \mathbf{z}^T, \mathbf{x}\rangle - g(\mathbf{x})\}$ is its Legendre-Fenchel convex conjugate. When g is differentiable, we also have $\nabla g^*(\mathbf{z}) := \arg\max_{\mathbf{x}\in\mathcal{X}^N}\{\langle \mathbf{z}^T, \mathbf{x}\rangle - g(\mathbf{x})\}$, and $\nabla g \circ \nabla g^*(\mathbf{z}) = \mathbf{z}$.*

3.1.2. Network Assumptions.

We first state our assumptions on the network topology.

Assumption 2. *The graph $\mathcal{G}$ is connected, undirected and the adjacency matrix A is doubly stochastic.*

These assumptions imply that the graph Laplacian $\mathcal{L}$ is a real symmetric matrix with nonnegative eigenvalues. We will denote the pseudo-inverse of $\mathcal{L}$ by $\mathcal{L}^+$ such that,

$$\mathcal{L}\mathcal{L}^+\mathcal{L} = \mathcal{L}. \tag{3.1}$$

We will use the following definition of the β-regularized Laplacian [8],

$$\mathcal{L}_\beta = \mathcal{L} + \frac{\beta}{N}\mathbf{1}_N\mathbf{1}_N^\top \otimes I_d. \tag{3.2}$$

In the remainder we will assume $\beta > 0$ as this results in $\mathcal{L}_\beta$ being positive definite. We define the Rayleigh quotient associated with $\mathcal{L}_\beta$ as follows,

$$\kappa_{\beta,N} = \max_{\mathbf{d}_x \in \mathbb{R}^{Nd}\setminus\{0\}} \frac{\|\mathcal{L}_\beta^{\frac{1}{2}}\mathbf{d}_x\|_2^2}{\|\mathbf{d}_x\|_2^2}. \tag{3.3}$$

It holds that [8, p.116],

$$\mathcal{L}_\beta^{-1} = \mathcal{L}^+ + \frac{1}{\beta N}\mathbf{1}_N\mathbf{1}_N^\top \otimes I_d \succeq \mathcal{L}^+, \tag{3.4}$$

where the latter inequality follows from the fact that $\mathbf{1}_d\mathbf{1}_d^\top \otimes I_N$ is positive semidefinite. In Subsection 3.2 we will use the following technical Lemma.

Lemma 3.5. *Let Assumption 2 hold and suppose that $\kappa_{\beta,N}$ is as defined in (3.3) then,*

$$\langle \mathbf{x}, \mathcal{L}\mathbf{x}\rangle \geq \frac{1}{\kappa_{\beta,N}}\|\mathcal{L}\mathbf{x}\|_2^2.$$

3.1.3. Mirror Maps. In this section we state our assumption for the two mirror maps Φ and Ψ. These assumptions are fairly standard in the literature of the mirror descent algorithm, see for example [3, Chapter 9]. We also state some identities and relationships that will be useful later on.

Assumption 3 (Mirror map). *$\Phi : \mathcal{D} \to \mathbb{R}$ is proper, twice differentiable, μ_Φ-strongly convex and L_Φ-smooth w.r.t. some norm $||\cdot||$. The same holds for $\Psi : \Lambda \to \mathbb{R}$ with constants μ_Ψ and L_Ψ. respectively. We furthermore make the additional assumption $\nabla\Phi^*(\mathbb{R}^d) = \mathcal{X}$, $\nabla\Psi^*(\mathbb{R}^d) = \Lambda$, Φ^* is L_{Φ^*}-smooth and assume uniform boundedness of the Laplacians of Φ^*, Ψ^* such that $||\Delta\Phi^*||_\infty, ||\Delta\Psi^*||_\infty < \infty$.*

The assumption that $\nabla\Phi^*$ maps directly to $\mathcal{X}$ (and similarly for Ψ) avoids the need for projections. Extending our results without this assumption is possible by following a route similar to [25]. A common mirror map that satisfies our assumptions for the primal variables is the negative entropy $\Phi(x) = \sum_{i=1}^d x_i \ln(x_i)$. For the Lagrange multipliers a Q-norm may be more appropriate i.e. $\Psi(\lambda) = \frac{1}{2}\lambda^\top Q\lambda$. The assumptions that $||\Delta\Phi^*||_\infty$, and $||\Delta\Psi^*||_\infty$ are bounded from above are needed in order to apply Itô's lemma to the Lyapunov function we will analyze in Subsection 3.2.

A useful property of the Bregman divergence induced by mirror maps that satisfy our assumptions is the following,

$$D_{\Phi^*}(z, z') = D_\Phi(x', x), \tag{3.5}$$

where $z = \nabla\Phi(x)$ and $z' = \nabla\Phi(x')$. For $x, y, z \in \mathbb{R}^d$ we have the triangle property for Bregman divergences (see [3, Lemma 9.11])

$$\langle x - y, \nabla\Phi(z) - \nabla\Phi(y)\rangle = D_\Phi(x, y) + D_\Phi(y, z) - D_\Phi(x, z). \tag{3.6}$$

We also make use of the following property,

$$D_f(\mathbf{x}, \mathbf{x}') \leq \alpha(\Phi) D_\Phi(\mathbf{x}', \mathbf{x}), \tag{3.7}$$

where $\alpha(\Phi) = \frac{L_f L_\Phi}{\mu_\Phi}$. This property follows from the relative smoothness assumption combined with the strong convexity and Lipschitz assumption on Φ,

$$D_f(\mathbf{x}, \mathbf{x}') \leq L_f D_\Phi(\mathbf{x}, \mathbf{x}') \leq \frac{L_f L_\Phi}{2} \|\mathbf{x}' - \mathbf{x}\|^2 \leq \frac{2 L_f L_\Phi}{2\mu_\Phi} D_\Phi(\mathbf{x}', \mathbf{x}),$$

where we used the notation $\Phi(\mathbf{x}) = \sum_{i=1}^N \Phi(x^i)$. We will use the following Rayleigh quotient,

$$\kappa_N = \max_{\mathbf{d}_x, \mathbf{d}_\lambda \in \mathbb{R}^{Nd} \setminus \{0\}} \max \left\{ \frac{\|\mathcal{L}^{\frac{1}{2}} \mathbf{d}_x\|_2^2}{\|\mathbf{d}_x\|^2}, \frac{\|\mathcal{L}^{\frac{1}{2}} \mathbf{d}_\lambda\|_2^2}{\|\mathbf{d}_\lambda\|^2} \right\}. \tag{3.8}$$

Note that the norms for $\mathbf{d}_x$ and $\mathbf{d}_\lambda$ in the definition above may be different (see Remark 1.2). We will also make use of following generalized Rayleigh quotient,

$$\kappa_g = \inf_{\mathbf{x} \in \mathcal{X}^N, \mathbf{d}_x, \mathbf{d}_\lambda \in \mathbb{R}^{Nd} \setminus \{0\}} \frac{\|A(\mathbf{x})[\mathbf{d}_x^T \ \mathbf{d}_\lambda^T]^\top\|^2_{\nabla^2 \Phi^*(\mathbf{z})}}{\|\mathbf{d}_x\|^2 + \|\mathbf{d}_\lambda\|^2}, \tag{3.9}$$

where $A(\mathbf{x}) = [\nabla^2 f(\mathbf{x}) + \mathcal{L}, \mathcal{L}] \in \mathbb{R}^{Nd \times 2Nd}$ and the notation $[X, Y]$ is used to denote a concatenation of matrices.

Lemma 3.6. *Suppose Assumptions 1 to 3 hold and that f is relatively strongly convex with respect to Φ, then κ_g defined in (3.9) is positive.*

This result is not obvious since $A(\mathbf{x})$ is not a square matrix, and the norm used in the definition of (3.9) is not standard; a short proof is provided in Appendix A.

Lastly, we will need the following relationship between the gradient of the Augmented Lagrangian and the two mirror maps.

Lemma 3.7. *Suppose Assumptions 1 to 3 hold. Then for an arbitrary optimal primal dual pair $(\mathbf{x}^\star, \boldsymbol{\lambda}^*)$ we have*

$$\begin{aligned} \|\nabla f(\mathbf{x}) + \mathcal{L}\boldsymbol{\lambda} + \mathcal{L}\mathbf{x}\|_{\nabla \Phi^*(\mathbf{z})} &\geq \frac{2\kappa_g}{\hat{\mu}} \left(\sum_{i=1}^N D_\Phi(x^\star, x^i) + D_\Psi(\lambda^\star, \lambda^i) \right) \\ &= \frac{2\kappa_g}{\hat{\mu}} \left(\sum_{i=1}^N D_{\Phi^*}(z^i, z^\star) + D_{\Psi^*}(\mu^i, \mu^\star) \right), \end{aligned}$$

where $\hat{\mu} = \min\{\mu_\Phi, \mu_\Psi\}$

The lemma above suggests that making progress towards reducing the norm of the gradient of the Augmented Lagrangian also means that we will be getting closer to the optimal solution. This is not surprising and it is a generalization of a similar inequality for the gradient descent method, see e.g. [30, Chapter 2]

3.2. Convergence Analysis. Ler $(\mathbf{x}^\star, \boldsymbol{\lambda}^\star) = (\mathbf{1}_N \otimes x^\star, \mathbf{1}_N \otimes \lambda^\star)$ be a point that satisfies the first order optimality conditions for (1.1). Note that when $\sigma = 0$, $(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)$ is also a fixed point of (2.4). In the case that f is only convex and thus multiple minimizers might exist, then we define the optimal primal dual pair, $(x^\star, \lambda^\star)$ as follows,

$$(x^\star, \lambda^\star) = \arg \min_{x,\lambda \in (X^\star, \Lambda^\star)} \{D_\Phi(x, x_0) + D_\Psi(\lambda, \lambda_0)\}, \tag{3.10}$$

where x_0, λ_0 are the initial point of the algorithm.

The connection of the dynamics of (2.4) and the Augmented Lagrangian is key to the convergence analysis below. The analysis of the algorithm in (2.4) is based on the following Lyapunov function,

$$V(\mathbf{x}, \boldsymbol{\lambda}) = c(V_1(\mathbf{x}) + V_2(\boldsymbol{\lambda})) + V_3(\mathbf{x}, \boldsymbol{\lambda}), \tag{3.11}$$

where,

$$\begin{aligned} V_1(\mathbf{x}) = \sum_{i=1}^N D_\Phi(x^\star, x^i), \quad V_2(\boldsymbol{\lambda}) = \sum_{i=1}^N D_\Psi(\lambda^\star, \lambda^i), \\ V_3(\mathbf{x}, \boldsymbol{\lambda}) = D_f(\mathbf{x}, \mathbf{x}^\star) + \langle \mathbf{x} - \mathbf{x}^\star, \mathcal{L}(\boldsymbol{\lambda} - \boldsymbol{\lambda}^\star) \rangle + \frac{1}{2} \|\mathcal{L}^{\frac{1}{2}} \mathbf{x}\|_2^2, \end{aligned} \tag{3.12}$$

and $c > 0$ will be specified below. The role of V_1 and V_2 are clear, they measure the distance between the point $(\mathbf{x}, \boldsymbol{\lambda})$ and the optimal Karush–Kuhn–Tucker (KKT) primal-dual pair $(\mathbf{x}^\star, \boldsymbol{\lambda}^\star)$. The role of V_3 becomes more obvious once re-arranged as follows,

$$V_3(\mathbf{x}, \boldsymbol{\lambda}) = L(\mathbf{x}, \boldsymbol{\lambda}) - L(\mathbf{x}^\star, \boldsymbol{\lambda}^\star).$$

Therefore V_3 is also a measure of optimality since $V_3(\mathbf{x}^\star, \boldsymbol{\lambda}^\star) = 0$. Unfortunately, neither $V_1 + V_2$ or V_3 on their own are Lyapunov functions for the stochastic dynamics in (2.4). Instead the proof below balances the two terms using the constant c. Note that given our assumptions on f, Φ and Ψ it follows that V is a proper function (has compact sub-level sets).

First we establish useful upper and lower bounds on (3.11). Note that the Lemma below is enough to establish the convergence of the algorithm but not the actual rate that V dissipates. This is one of the reasons that the Lyapunov function in (3.11) is different than other works (e.g. [41]) that establish the convergence of distributed mirror descent but not the convergence rate for V.

Lemma 3.8. *Let Assumptions 1-3 hold. Then V_t in (3.11) satisfies the following,*
(I) $V(\mathbf{x}^\star, \boldsymbol{\lambda}^\star) = 0$.
(I.1) *If* $c \geq \max\{\kappa_N/\mu_\Phi, \kappa_N/\mu_\Psi\}$,

$$V(\mathbf{x}, \boldsymbol{\lambda}) \geq \frac{1}{2}(\mu_\Phi c - \kappa_N)||\mathbf{x} - \mathbf{x}^\star||^2 + \frac{1}{2}(\mu_\Psi c - \kappa_N)||\boldsymbol{\lambda} - \boldsymbol{\lambda}^\star||^2 \geq 0.$$

If in addition f is μ_f-strongly convex relative to Φ then:

(II.2) *For any* $c \geq \max\{((\kappa_N - \mu_f L_\Phi)/\mu_\Phi, \kappa_N/\mu_\Psi\}$,

$$V(\mathbf{x}, \boldsymbol{\lambda}) \geq \frac{1}{2}(\mu_\Phi c + \mu_f L_\Phi - \kappa_N)||\mathbf{x} - \mathbf{x}^\star||^2 + \frac{1}{2}(\mu_\Psi c - \kappa_N)||\boldsymbol{\lambda} - \boldsymbol{\lambda}^\star||^2 \geq 0.$$

(III) *Let* $\hat{\mu} = \min\{\mu_\Phi, \mu_\Psi\}$. *Then,*

$$V(\mathbf{x}, \boldsymbol{\lambda}) \leq \left(c + \frac{3\kappa_N + (1+\hat{\mu})\alpha(\Phi)}{\hat{\mu}}\right)\left(\sum_{i=1}^N D_\Phi(x^\star, x^i) + \sum_{i=1}^N D_\Psi(\lambda^\star, \lambda^i)\right). \tag{3.13}$$

Proof. Property (I) is obvious. For (I.1) we bound V_1, V_2 using the strong convexity of Φ and Ψ respectively:

$$\begin{aligned} V_1(\mathbf{x}) &= \sum_{i=1}^N D_\Phi(x^\star, x^i) \geq \frac{\mu_\Phi}{2}\|\mathbf{x}^\star - \mathbf{x}\|^2, \\ V_2(\boldsymbol{\lambda}) &= \sum_{i=1}^N D_\Psi(\lambda^\star, \lambda^i) \geq \frac{\mu_\Psi}{2}\|\boldsymbol{\lambda}^\star - \boldsymbol{\lambda}\|^2. \end{aligned} \tag{3.14}$$

We note that the convexity of f implies that $D_f(\mathbf{x}, \mathbf{x}^\star) \geq 0$, and therefore we can bound V_3 as follows,

$$\begin{aligned} V_3(\mathbf{x}, \boldsymbol{\lambda}) &\geq \langle \mathbf{x} - \mathbf{x}^\star, \mathcal{L}(\boldsymbol{\lambda} - \boldsymbol{\lambda}^\star)\rangle \geq -\frac{1}{2}(\|\mathcal{L}^{\frac{1}{2}}(\mathbf{x} - \mathbf{x}^\star)\|_2^2 + \|\mathcal{L}^{\frac{1}{2}}(\boldsymbol{\lambda} - \boldsymbol{\lambda}^\star)\|_2^2) \\ &\geq -\frac{\kappa_N}{2}(\|\mathbf{x} - \mathbf{x}^\star\|^2 + \|\boldsymbol{\lambda} - \boldsymbol{\lambda}^\star\|^2), \end{aligned}$$

where in the second inequality we used (1.5) and in the third one (3.8). Using the bound for V_3 above and the two bounds in (3.14) we obtain (I.1).

If, in addition, f is strongly convex relative to Φ then,

$$\sum_{i=1}^N D_f(x^i, x^\star) \geq \mu_f \sum_{i=1}^N D_\Phi(x^i, x^\star) \geq \frac{\mu_f \mu_\Phi}{2}\|\mathbf{x}^\star - \mathbf{x}\|^2.$$

Using the preceding inequality to bound V_3 along with (3.14) we obtain the (II.2).

For the upper bound in (III) we bound the the first term in V_3 using the symmetry bound in (3.7),

$$\sum_{i=1}^N D_f(x_i, x^\star) \leq \alpha(\Phi) \sum_{i=1}^N D_\Phi(x^\star, x_i).$$

For the second term in V_3 we use (1.5) again and for any $\gamma > 0$,

$$\begin{aligned}\langle \mathbf{x} - \mathbf{x}^\star, \mathcal{L}(\boldsymbol{\lambda} - \boldsymbol{\lambda}^\star)\rangle &\leq \frac{\kappa_N}{2}(\gamma\|\mathbf{x} - \mathbf{x}^\star\|^2 + \frac{1}{\gamma}\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^\star\|^2) \\ &\leq \frac{\kappa_N\gamma}{\mu_\Phi}\sum_{i=1}^N D_\Phi(x^\star, x^i) + \frac{\kappa_N}{\gamma\mu_\Psi}\sum_{i=1}^N D_\Psi(\lambda^\star, \lambda^i) \\ &\leq \frac{\kappa_N}{\hat{\mu}}\left(\gamma\sum_{i=1}^N D_\Phi(x^\star, x^i) + \frac{1}{\gamma}\sum_{i=1}^N D_\Psi(\lambda^\star, \lambda^i)\right) \\ &\leq \frac{\kappa_N}{\hat{\mu}}\sum_{i=1}^N D_\Phi(x^\star, x^i) + \frac{\kappa_N + \alpha(\Phi)}{\hat{\mu}}\sum_{i=1}^N D_\Psi(\lambda^\star, \lambda^i),\end{aligned}$$

where in the second inequality we used the relative strong convexity of Φ and for the last inequality we set $\gamma = \frac{\kappa_N}{\kappa_N + \alpha(\Phi)} \leq 1$. Finally, for the last term in V_3 we use the bound from (3.8), the strong convexity of Φ and the definition of $\hat{\mu}$,

$$\|\mathcal{L}^{\frac{1}{2}}\mathbf{x}\|_2^2 = \|\mathcal{L}^{\frac{1}{2}}(\mathbf{x} - \mathbf{x}^\star)\|_2^2 \leq \kappa_N\|\mathbf{x} - \mathbf{x}^\star\|^2 \leq \frac{2\kappa_N}{\hat{\mu}}\sum_{i=1}^N D_\Phi(x^\star, x^i)$$

Using the upper bounds for the three terms in V_3 we obtain the bound in (3.13). ∎

Below we state and prove the main result of this section.

Proposition 3.9 (Convergence of the preconditioned dynamics in (2.4)). *Let Assumptions 1-3 hold and suppose that $\kappa_g > 0$ (see (3.9). Consider the dynamics in (2.4). Let the Lyapunov function V_t be defined as in (3.11), then*

$$\mathbb{E}[V_T] \leq e^{-rT}\mathbb{E}[V_0] + \frac{\sigma^2}{2}\mathbb{E}\left[\int_0^T e^{-r(T-s)}M_s ds\right]$$

where $V_T := V(\mathbf{x}_T, \boldsymbol{\lambda}_T)$,

$$r = \frac{2\kappa_g}{c\hat{\mu} + (1+\hat{\mu})\alpha(\Phi) + 3\kappa_N} > 0, \tag{3.15}$$

$$M_s = c\mathrm{tr}(\nabla^2\Phi^*(\mathbf{z}_s)) + \left(\Delta\cdot\nabla\Phi^*(\mathbf{z}_s) + \mathrm{tr}(\nabla^2\Phi^*(\mathbf{z}_s)\nabla^2_{xx}L(\mathbf{x}_s, \boldsymbol{\lambda}_s)\nabla^2\Phi^*(\mathbf{z}_s))\right)$$

where $\hat{\mu} = \min\{\mu_\Phi, \mu_\Psi\}$ and $c \geq \max\{2\kappa_{\beta,N}\mu_\Psi, \kappa_N/\mu_\Phi, \kappa_N/\mu_\Psi\}$, and the operator $\Delta\cdot\nabla\Phi^*(\mathbf{z})$ is given by $[\Delta\cdot\nabla\Phi^*(z^i)]_j = \sum_{k=1}^d \partial^2_{kk}\partial_j\Phi^*(z^i)$, $i = 1, \ldots, N$, $j = 1, \ldots, d$.

Proof. Since $x^i = \nabla\Phi^*(z^i)$ it follows from Itô's Lemma that,

$$dx_t^i = \nabla^2\Phi^*(z^i)dz_t^i + \frac{1}{2}\sigma^2\Delta\cdot\nabla\Phi^*(z^i)dt,$$

where the j^{th} element of the Itô correction term is $[\Delta \cdot \nabla\Phi^*(z^i)]_j = \sum_{k=1}^{d} \partial^2_{kk}\partial_j\Phi^*(z^i)$. For ease of exposition we make the following definition,

$$M_1(t) = \frac{\sigma^2}{2}\int_0^t \mathrm{tr}(\nabla^2\Phi^*(z_s))ds - \sigma\int_0^t \langle \nabla\Phi^*(\mathbf{z}_s) - \nabla\Phi^\star(\mathbf{z}^\star), dB_s\rangle.$$

Next note that V_1 defined in (3.12) can be written as follows $V_1(\mathbf{x}) = \sum_{i=1}^N D^*_\Phi(x^i, x^\star)$ and therefore,

(3.16)
$$\begin{aligned} dV_1(t) &= -\langle \nabla\Phi(\mathbf{z}_t) - \nabla\Phi^\star(\mathbf{z}^\star), \nabla f(x_t) + \mathcal{L}\mathbf{x}_t + \mathcal{L}\boldsymbol{\lambda}_t\rangle dt + dM_1(t) \\ &= -(\langle \mathbf{x}_t - \mathbf{x}^\star, \nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}^\star)\rangle + \langle \mathbf{x}_t - \mathbf{x}^\star, \nabla f(\mathbf{x}^\star) + \mathcal{L}\mathbf{x}_t + \mathcal{L}\boldsymbol{\lambda}_t\rangle)dt + dM_1(t) \\ &\leq -\langle \mathbf{x}_t - \mathbf{x}^\star, \mathcal{L}\mathbf{x}_t + \mathcal{L}(\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star)\rangle dt + dM_1(t), \end{aligned}$$

where for the last inequality we used the convexity of f and the optimality condition $\nabla f(\mathbf{x}^\star) = -\mathcal{L}\boldsymbol{\lambda}^\star$. We also re-write the second equation in (2.4) in terms of λ_t,

$$d\boldsymbol{\lambda}_t = \nabla^2\Psi(\boldsymbol{\lambda}_t)^{-1}\mathcal{L}\mathbf{x}_t$$

Using the equation above we obtain the following expression for V_2 in (3.12),

$$\begin{aligned} dV_2(t) &= \langle -\nabla^2\Psi(\boldsymbol{\lambda}_t)(\boldsymbol{\lambda}^\star - \boldsymbol{\lambda}_t), \nabla^2\Psi(\boldsymbol{\lambda}_t)^{-1}\mathcal{L}\mathbf{x}_t\rangle dt \\ &= (\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^\star)^T\mathcal{L}(\mathbf{x}_t - \mathbf{x}^\star)dt. \end{aligned}$$

Adding the bounds for V_1 and V_2 we obtain,

(3.17)
$$\begin{aligned} d(V_1(t) + V_2(t)) &\leq -\langle \mathbf{x}_t - \mathbf{x}^\star, \mathcal{L}(\mathbf{x}_t - \mathbf{x}^\star)\rangle dt + dM_1(t) \\ &\leq -\frac{1}{\kappa_{\beta,N}}\|\mathcal{L}(\mathbf{x}_t - \mathbf{x}^\star)\|_2^2 dt + dM_1(t) \end{aligned}$$

where we used Lemma 3.5 to bound the first term. Using the definition of the Augmented Lagrangian in (2.3) we can re-write V_3 from (3.12) as follows,

$$V_3(\mathbf{x}, \boldsymbol{\lambda}) = L(\mathbf{x}, \boldsymbol{\lambda}) - f(\mathbf{x}^\star).$$

Therefore,

$$\begin{aligned} dV_3(t) &= -\|\nabla_x L(\mathbf{x}, \boldsymbol{\lambda})\|_{\nabla^2\Phi^*(\mathbf{z})}dt + \|\nabla_\lambda L(\mathbf{x}, \boldsymbol{\lambda})\|_{\nabla^2\Psi^*(\boldsymbol{\mu})}dt + dM_2(t), \\ &\leq -\frac{2\kappa_g}{\hat{\mu}}\left(D_\Phi(\mathbf{x}^\star, \mathbf{x}_t) + D_\Psi(\boldsymbol{\lambda}^\star, \boldsymbol{\lambda}_t)\right)dt + \frac{2}{\mu_\Psi}\|\mathcal{L}(\mathbf{x} - \mathbf{x}^\star)\|^2 dt + dM_2(t) \end{aligned}$$

where the last inequality follows from Lemma 3.7 and $M_2(t)$ is defined as follows,

$$\begin{aligned} M_2(t) = \frac{\sigma^2}{2}\int_0^t \left(\Delta \cdot \nabla\Phi^*(\mathbf{z}_s) + \mathrm{tr}(\nabla^2\Phi^*(\mathbf{z}_s)\nabla^2_{xx}L(\mathbf{x}_s, \boldsymbol{\lambda}_s)\nabla^2\Phi^*(\mathbf{z}_s))\right)ds \\ + \sigma\int_0^t \langle \nabla L(\mathbf{x}_s, \boldsymbol{\lambda}_s), \nabla^2\Phi^*(\mathbf{z}_s)d\mathbf{B}_s\rangle \end{aligned}$$

Using the bounds in (3.16) and (3.17) we obtain,

$$\begin{aligned} dV_t &\leq -\frac{2\kappa_g}{\hat{\mu}}\left(D_\Phi(\mathbf{x}^\star,\mathbf{x}_t)+D_\Psi(\boldsymbol{\lambda}^\star,\boldsymbol{\lambda}_t)\right)dt+\left(\frac{2}{\mu_\Psi}-\frac{c}{\kappa_{\beta,N}}\right)\|\mathcal{L}(\mathbf{x}-\mathbf{x}^\star)\|^2dt+cdM_1(t)+dM_2(t)\\ &\leq -\frac{2\kappa_g}{\hat{\mu}}\left(c+\frac{3\kappa_N+(1+\hat{\mu})\alpha(\Phi)}{\hat{\mu}}\right)^{-1}V(\mathbf{x}_t,\boldsymbol{\lambda}_t)dt+cdM_1(t)+dM_2(t)\\ &= -rV_tdt+cdM_1(t)+dM_2(t) \end{aligned}$$

where in the second inequality we used (3.13) in Lemma 3.11 and the condition $c \geq 2\kappa_{\beta,N}/\mu_\Psi$, and r is defined in (3.15). Taking expectations and applying Gronwall's Lemma we obtain the final result. ∎

Remark 3.10. *Convergence to an invariant distribution: Proposition 3.9 can be extended to establish a standard Foster-Lyapunov condition, $\mathcal{A}V_t \leq rV_t + \bar{d}$, where $\mathcal{A}$ is the infinitesimal generator of (2.4). By imposing some additional (and not particularly restrictive) assumptions on Φ, f such that $c\mathrm{tr}(\nabla^2\Phi^*(z_s)) + \left(\Delta\cdot\nabla\Phi^*(\mathbf{z}_s) + \mathrm{tr}(\nabla^2\Phi^*(\mathbf{z}_s)\nabla^2_{xx}L(\mathbf{x}_s,\boldsymbol{\lambda}_s)\nabla^2\Phi^*(\mathbf{z}_s))\right) < \bar{d}$ holds, one then establishes the existence of an invariant distribution, [27, Theorem 5.3]. One could go further to establish exponential convergence to equilibrium via [27, Theorem 6.1], but this would require establishing a minorization condition for irreducibility or equivalently showing controllability of the dynamics in (2.4) when the noise is replaced with a deterministic control input. This requires further assumptions on Φ and for the sake of brevity details are omitted.*

The result in Proposition 3.9 is quite general since it allows *any* mirror map (within assumptions) in the primal and/or dual. In the standard decentralized case we can only establish the superiority of the mirror descent method over gradient methods for specific cases. For example the case where Φ is the negative entropy and $\mathcal{X}$ is the simplex, then mirror descent is known to be more efficient both in theory and in practice (see [3, Example 9.10]). Therefore in order to generate similar insights into the distributed case it is necessary to investigate some specific choices of mirror maps. In the next section we motivate a particular preconditioner that relies on convex duality theory.

3.3. Choosing the Network Preconditioner. For many interesting applications good mirror maps are known, and the advantages of mirror descent over gradient descent are well understood. Unfortunately, it is not clear how to select a good mirror map for the space of Lagrange multipliers. However, if we restrict the mirror map to be quadratic, then it is possible to identify one possible mirror map that works well in practice (see also the numerical results in Section 5). In Section 4 (see Example 4.5) we show that if we choose the mirror map to be the Hessian of the dual of (1.1) and if the objective function is quadratic, then the algorithm would converge in a single iteration (when $\sigma = 0$). This observation is reminiscent of the Newton method for unconstrained problems and so we propose that using the Hessian of the Lagrangian is a good choice for Ψ. Of course, computing the Hessian of the Lagrangian could be expensive but we argue that the extra computation associated with approximating the Hessian of the dual function could be justified in the scenario where the Laplacian matrix is ill-conditioned. We can back this claim exactly using the example in Example 4.5 and also

using more realistic problems in Section 5. We also note that our discretization scheme does not rely on the exact computation of the Hessian of the dual.

Below we briefly outline the derivation of the Hessian for the dual function in the deterministic setting and when f is strongly convex. For simplicity we only consider the Lagrangian $f(\mathbf{x})+\langle\boldsymbol{\lambda}, \mathcal{L}\mathbf{x}\rangle$ (and not the Augmented Lagrangian). To extend for the Augmented Lagrangian the extra term $\frac{1}{2}\|\mathcal{L}^{\frac{1}{2}}\mathbf{x}\|_2^2$ can easily be added in the expressions below but for simplicity is omitted. In particular, the (negative) dual function $q(\boldsymbol{\lambda})$ is defined as follows,

$$q(\boldsymbol{\lambda}) = \max_{\mathbf{x}\in\mathcal{X}^N} \left\{-(f(\mathbf{x}) + \langle\boldsymbol{\lambda}, \mathcal{L}\mathbf{x}\rangle)\right\}, \tag{3.18}$$

Let $\mathbf{x}(\boldsymbol{\lambda})$ be the maximizer of (3.18). We know that for the strongly convex case, $\mathbf{x}(\boldsymbol{\lambda})$ is unique and the gradient of the dual function is given by ([2, Theorem 6.3.3]),

$$\nabla_{\boldsymbol{\lambda}} q(\boldsymbol{\lambda}) = -\mathcal{L}\mathbf{x}(\boldsymbol{\lambda}).$$

Applying the KKT conditions to (3.18) and differentiating with respect to λ we also have that,

$$\nabla^2 f(\mathbf{x}(\boldsymbol{\lambda}))\frac{d\mathbf{x}(\boldsymbol{\lambda})}{d\boldsymbol{\lambda}} + \mathcal{L} = 0.$$

Using the preceding equation we obtain the following expression for the Hessian of the dual function,

$$\nabla^2 q(\boldsymbol{\lambda}) = -\mathcal{L}\frac{d\mathbf{x}(\boldsymbol{\lambda})}{d\boldsymbol{\lambda}} = \mathcal{L}(\nabla^2 f(\mathbf{x}(\boldsymbol{\lambda})))^{-1}\mathcal{L}.$$

Unfortunately, even if f is strongly convex, the dual function in (3.18) is only convex (and not strongly convex). So we cannot directly use the Hessian of the dual function above, and instead we proposed to use,

$$\nabla^2 \Psi(\boldsymbol{\lambda}) = \mathcal{L}_\beta(\nabla^2 f(\mathbf{x}(\boldsymbol{\lambda})))^{-1}\mathcal{L}_\beta,$$

where $\mathcal{L}_\beta$ is the β-regularized Laplacian defined in (3.2). Note that we can avoid inverting the Hessian of f at every iteration and instead work with the convex conjugate of Ψ which occurs a one-off cost of diagonalizing the regularized Laplacian,

$$\nabla^2 \Psi^*(\boldsymbol{\lambda}) = \mathcal{L}_\beta^{-1}(\nabla^2 f(\mathbf{x}(\boldsymbol{\lambda})))\mathcal{L}_\beta^{-1}.$$

A major computational challenge associated with the derivation above is the computation of $\mathbf{x}(\boldsymbol{\lambda})$. Unless f is a particularly simple function, it is not possible to solve (3.18) in closed form. In the next section we will address this issue using a discretization approach that is similar to a Gauss-Seidel scheme.

4. Discretization Analysis.

In the convergence analysis of Section 3.2, any mirror map that is compatible with Assumption 3 can be used. However, this flexibility means that the dynamics in (2.4) have the potential to lead to good or bad algorithms depending upon the specific mirror map selection. As a result the convergence rates we obtained in the previous

section are not necessarily the most tight. To address this issue we examine the algorithm once the dynamics in (2.4) are discretized and we specifically focus on quadratic mirror maps. For the remainder we will restrict to

$$\Phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top Q\mathbf{x}, \quad \Psi(\boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{\lambda}^\top R\boldsymbol{\lambda},$$

where Q, R are positive-definite matrices. Nevertheless, the insights of the continuous time analysis obtained previously is crucial since the results of this section rely on the same Lyapunov function from Section 3.2. By restricting the analysis to the case of quadratic mirror maps we will be able to identify mirror maps for both the primal and Lagrangian dual variables that lead to favorable condition numbers. Like all preconditioning methods, a balance needs to be struck between the computational cost of computing the preconditioner and the algorithm's convergence rate. One of the aims of the analysis in this section is to provide a detailed description of this inherent trade-off (see Figure 3 and the discussion in the last paragraph of this section).

We propose to apply an explicit Gauss-Seidel discretization scheme to (2.4),

$$\begin{aligned} \mathbf{z}_k &= \mathbf{z}_{k-1} - \delta(\nabla f(\mathbf{x}_{k-1}) + \mathcal{L}\mathbf{x}_{k-1} + \mathcal{L}\boldsymbol{\lambda}_{k-1}) + \sqrt{\delta}\sigma\ \boldsymbol{\xi}_k, \\ \boldsymbol{\mu}_k &= \boldsymbol{\mu}_{k-1} + \delta\mathcal{L}\mathbf{x}_k, \end{aligned} \tag{4.1}$$

where $\boldsymbol{\xi}_k \sim N(0, I_{Nd})$. Note that we use the notation $\mathbf{x}_k$ to denote the approximation to $\mathbf{x}(k\delta)$, with similar notation for the rest of the variables. In this work, our emphasis primarily lies on employing this scheme for optimization rather than for sampling purposes. Consequently, the convergence of $\mathbf{x}_k$ to $\mathbf{x}(k\delta)$ or the density of $\mathbf{x}_k$ are outside the scope of this paper. Note that if we were to follow the motivation discussed in Subsection 3.3 then we would have $\mathbf{x}_k$ to be the solution of the dual problem. Instead, in our scheme below we just use a simple Gauss-Seidel scheme. The two schemes (i.e. Gauss-Seidel and exact solution of the dual) would coincide if the solution of the dual problem could be obtained via a single mirror-descent iteration.

Let $(\mathbf{z}^\star, \boldsymbol{\mu}^\star)$ denote the duals of the optimal solution defined in (3.10). We introduce the following notation,

$$\begin{aligned} \hat{\mathbf{z}}_k &= \mathbf{z}_k - \mathbf{z}^\star, \\ \hat{\boldsymbol{\mu}}_k &= \boldsymbol{\mu}_k - \boldsymbol{\mu}^\star, \\ \nabla\hat{f}_k &= \nabla f(\mathbf{x}_k) + \mathcal{L}\mathbf{x}_k - (\nabla f(\mathbf{x}^\star) + \mathcal{L}\mathbf{x}^\star). \end{aligned}$$

Subtracting $\mathbf{z}^\star$ and $\boldsymbol{\mu}^\star$) from both sides of the first (second) equation in (4.1), using $\hat{\mathbf{x}}_k = Q^{-1}\hat{\mathbf{z}}_k$, $\hat{\boldsymbol{\lambda}}_k = R^{-1}\hat{\boldsymbol{\mu}}_k$ and the KKT optimality conditions we obtain,

$$\begin{aligned} \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k-1} - \delta Q^{-1}(\nabla\hat{f}(\mathbf{x}_{k-1}) + \mathcal{L}\hat{\boldsymbol{\lambda}}_{k-1}) + \sqrt{\delta}\sigma Q^{-1}\boldsymbol{\xi}_k, \\ \hat{\boldsymbol{\lambda}}_k &= \hat{\boldsymbol{\lambda}}_{k-1} + \delta R^{-1}\mathcal{L}\hat{\mathbf{x}}_k. \end{aligned} \tag{4.2}$$

Remark 4.1. *The assumption that the mirror maps are quadratic is not as restrictive as it might first appear. Firstly, the consensus constraint is an equality, therefore the Lagrangian dual variables are vectors in $\mathbb{R}^{Nd}$. Therefore using a Q-norm for some positive definite matrix*

Q is the natural choice. For the Lagrangian duals any other commonly used mirror map such as the negative entropy would not lead to convergent algorithms. The domain of the negative entropy is the positive orthant and it would therefore violate our assumptions. Secondly, regarding the mirror map of the primal variables, we can extend our setting in the discretization scheme using local linearization and a first order Taylor expansion and set $Q_k = \nabla^2\Phi(\mathbf{x}_{k-1})$, i.e. discretize the right-hand side of (2.4) using

$$\nabla\Phi(\mathbf{x}_k) - \nabla\Phi(\mathbf{x}_{k-1}) = \nabla^2\Phi(\mathbf{x}_{k-1})(\mathbf{x}_k - \mathbf{x}_{k-1}) + O(\|\mathbf{x}_k - \mathbf{x}_{k-1}\|^2),$$

and similarly for R_k and $\boldsymbol{\lambda}_k$. Our analysis below is for the case where Q and R are the same at each iteration but similar arguments can be used to extend it to the non homogeneous case.

We begin by specializing our assumptions on relatively strong convexity and relative smoothness to the case of quadratic maps.

Lemma 4.2. *Suppose that $\Phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top Q\mathbf{x}$, where Q is a positive definite matrix, then:*
(i) *If f is L_f-smooth relative to Φ then,*

$$\|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x}) + \mathcal{L}(\mathbf{y} - \mathbf{x})\|^2_{Q^{-1}} \leq L'_f\langle\nabla f(\mathbf{x}) - \nabla f(\mathbf{y}) + \mathcal{L}(\mathbf{x} - \mathbf{y}), \mathbf{x} - \mathbf{y}\rangle \tag{4.3}$$

where $L'_f = L_f + \kappa_N/\mu_\Phi$.
(ii) *If f is μ_f-strongly convex relative to Φ we have,*

$$\langle\nabla f(\mathbf{x}) - \nabla f(\mathbf{y}) + \mathcal{L}(\mathbf{x} - \mathbf{y}), \mathbf{x} - \mathbf{y}\rangle \geq \mu_f\|\mathbf{x} - \mathbf{y}\|^2_Q \tag{4.4}$$

Proof. (i) Consider the function $g(\mathbf{y}) = f(\mathbf{y}) + \|\mathcal{L}^{\frac{1}{2}}\mathbf{y}\|^2 - \langle\nabla f(\mathbf{x}) + \mathcal{L}\mathbf{x}, \mathbf{y}\rangle$ and note that it is minimized by $\mathbf{x}$. Moreover, since f is relatively smooth w.r.t Φ implies that for any $\mathbf{y}'$,

$$\nabla^2 f(\mathbf{y}') + \mathcal{L} \preceq L_f Q + \kappa_N I \preceq (L_f + \frac{\kappa_N}{\mu_\Phi})Q,$$

where we used Assumption 3 and κ_N was defined in (3.8). Therefore g is also relatively smooth w.r.t Φ:

$$D_g(\mathbf{y}', \mathbf{y}) \leq L'_f D_\Phi(\mathbf{y}', \mathbf{y}) = \frac{L'_f}{2}\|\mathbf{y}' - \mathbf{y}\|^2_Q$$

Using the equation above at with $\mathbf{y}' = \mathbf{y} - \frac{Q^{-1}}{L'_f}(\nabla f(\mathbf{y}) + \mathcal{L}\mathbf{y})$

$$g(\mathbf{x}) \leq g(\mathbf{y} - \frac{1}{L'_f}Q^{-1}(\nabla f(\mathbf{y}) + \mathcal{L}\mathbf{y})) \leq g(\mathbf{y}) - \frac{1}{2L'_f}\|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x}) + \mathcal{L}(\mathbf{x} - \mathbf{y})\|^2_{Q^{-1}}$$

Where the first inequality follows from the fact that $\mathbf{x}$ minimizes g. Using the definition of g on the preceding equation we obtain,

$$\begin{aligned}\langle\nabla f(\mathbf{x}) + \mathcal{L}\mathbf{x}, \mathbf{x} - \mathbf{y}\rangle - (f(\mathbf{x}) + \|\mathcal{L}^{\frac{1}{2}}\mathbf{x}\|^2) \geq &- (f(\mathbf{y}) + \|\mathcal{L}^{\frac{1}{2}}\mathbf{y}\|^2) \\ &+ \frac{1}{2L'_f}\|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x}) + \mathcal{L}(\mathbf{x} - \mathbf{y})\|^2_{Q^{-1}}.\end{aligned}$$

Finally, using the inequality above with the role of $\mathbf{x}$ and $\mathbf{y}$ reversed and adding the two inequalities we obtain (4.3).

(ii) For the second part we use the mean-value theorem to obtain,

$$\nabla f(\mathbf{x}) - \nabla f(\mathbf{y}) + \mathcal{L}(\mathbf{x}-\mathbf{y}) = (\nabla^2 f(\mathbf{x}') + \mathcal{L})(\mathbf{x}-\mathbf{y})$$

for some $\mathbf{x}' = \tau\mathbf{x} + (1-\tau)\mathbf{y}$ with $0 < \tau < 1$. Multiplying both sides by $\mathbf{x}-\mathbf{y}$ and using the assumption that f is μ_f-strongly convex relative to Φ we obtain,

$$\begin{aligned}\langle \nabla f(\mathbf{x}) - \nabla f(\mathbf{y}) + \mathcal{L}(\mathbf{x}-\mathbf{y}), \mathbf{x}-\mathbf{y}\rangle &= \langle (\mathbf{x}-\mathbf{y})(\nabla^2 f(\mathbf{x}') + \mathcal{L}), \mathbf{x}-\mathbf{y}\rangle \\ &\geq \mu_f \langle \mathbf{x}-\mathbf{y}, Q(\mathbf{x}-\mathbf{y})\rangle = \mu_f \|\mathbf{x}-\mathbf{y}\|_Q^2,\end{aligned}$$

where for the inequality we used the fact that $\mathcal{L}$ is positive semi-definite. ∎

Next we introduce two generalized Rayleigh quotients that will play a key role in our analysis and are inspired by the definitions in (3.8) and (3.9). However, because the mirror maps in this section are assumed to be quadratic (and therefore symmetric) we will be able to obtain more refined estimates with the definitions below:

$$\overline{\sigma}(\mathcal{L})^2 = \sup_{\mathbf{x}\in\mathcal{X}^N} \frac{\langle \mathcal{L}\mathbf{x}, R^{-1}\mathcal{L}\mathbf{x}\rangle}{\|\mathbf{x}\|_Q^2}, \tag{4.5}$$

$$\underline{\sigma}(\mathcal{L})^2 = \inf_{\boldsymbol{\lambda}\in \mathrm{range}(\mathcal{L})} \frac{\langle \mathcal{L}\boldsymbol{\lambda}, Q^{-1}\mathcal{L}\boldsymbol{\lambda}\rangle}{\|\boldsymbol{\lambda}\|_R^2}. \tag{4.6}$$

In the standard setting where $Q = R = I$ (i.e. when our algorithm reduces to an exact distributed gradient descent algorithm) then $\overline{\sigma}(\mathcal{L})$ is the largest eigenvalue value of $\mathcal{L}$ and $\underline{\sigma}(\mathcal{L}) > 0$ the second smallest. However, in the framework we analyze in this paper we have the opportunity to design the algorithm by choosing the Φ and Ψ. To see the utility of the definitions above suppose that $\mathcal{L}$ was invertible; (of course $\mathcal{L}$ is not invertible, but for the sake of argument suppose that it was). Then if $R = \mathcal{L}Q^{-1}\mathcal{L}$ and then $\underline{\sigma}(\mathcal{L}) = \overline{\sigma}(\mathcal{L}) = 1$. Therefore, the condition number of the graph Laplacian will play no role in the convergence rate of the algorithm. While not possible to invert $\mathcal{L}$, we can achieve a similar effect by using the $\beta-$regularized Laplacian (see Remark 4.5 below). The definition of the two Rayleigh quotients above will be useful in the analysis, but the Lemma below is useful for computing these constants using standard methods.

Lemma 4.3. **(i)** *The second smallest eigenvalue value of* $R^{-\frac{1}{2}}\mathcal{L}Q^{-1}\mathcal{L}R^{-\frac{1}{2}}$ *is equal to* $\underline{\sigma}(\mathcal{L})$. **(ii)** *The largest eigenvalue value of* $Q^{-\frac{1}{2}}\mathcal{L}R^{-1}\mathcal{L}Q^{-\frac{1}{2}}$ *is equal to* $\overline{\sigma}(\mathcal{L})$.

Proof. (i) follows from the change of variables $\kappa = R^{-\frac{1}{2}}\boldsymbol{\lambda}$ and noting that the range of $\boldsymbol{\lambda} \in \mathcal{L}$ excludes the possibility of $\mathcal{L}\boldsymbol{\lambda} = 0$ which would otherwise correspond to smallest eigenvalue. (ii) follows using $\mathbf{x} = Q^{-\frac{1}{2}}\mathbf{y}$. ∎

In order to be able to use the lower bound in (4.6) we need to ensure that the $\boldsymbol{\lambda}_k$ generated by the algorithm satisfies the condition $\boldsymbol{\lambda}_k \in \mathrm{range}(\mathcal{L})$. The Lemma below shows that this is easy to ensure as long as the algorithm is initialized correctly.

Lemma 4.4. *Suppose that (4.2) is initialized at $\boldsymbol{\lambda}_0 = 0$, $\mathbf{x}_0 \in \mathcal{X}^N$ and R has full rank. Then $\boldsymbol{\lambda}_k \in \text{range}(\mathcal{L})$, $\forall k > 0$.*

Proof. We proceed by induction. For $\boldsymbol{\lambda}_0 = 0$ then $\boldsymbol{\lambda}_1 = R^{-1}\mathcal{L}\mathbf{x}_1$ therefore $\boldsymbol{\lambda}_1 \in \text{range}(R^{-1}\mathcal{L})$. Since R^{-1} has full rank it follows from [26, Corollary 4.2.12] that $\text{range}(R^{-1}\mathcal{L}) = \text{range}(\mathcal{L})$. Now suppose that $\boldsymbol{\lambda}_{m-1} \in \text{range}(\mathcal{L})$, then

$$\boldsymbol{\lambda}_m = \boldsymbol{\lambda}_{m-1} + \delta R^{-1}\mathcal{L}\mathbf{x}_m = \delta R^{-1}\mathcal{L}\left(\sum_{i=1}^{m-1}\mathbf{x}_i + (\mathbf{x}_m - \mathbf{x}^\star)\right),$$

and therefore $\boldsymbol{\lambda}_m \in \text{range}(\mathcal{L})$. ∎

Example 4.5. *It is instructive to examine the case where the conditioning of the optimization model is solely due to the condition number of the graph. Suppose $\mathcal{X} = \mathbb{R}$ and $f(\mathbf{x}) = \frac{\alpha}{2}\|\mathbf{x}-\mathbf{b}\|^2$, where $\alpha > 0$, and $\mathbf{x}$, and $\mathbf{b}$ are in $\mathbb{R}^N$. In this case minimizing $f(\mathbf{x})$ subject to the constraint $\mathcal{L}\mathbf{x} = 0$ has the unique solution $\frac{1}{N}\sum_{i=1}^N b_i$. Let $C = \{\mathbf{x} \in \mathbb{R}^N \mid \mathcal{L}\mathbf{x} = 0\}$ and let $C^\perp$ be its orthogonal complement. We also denote the unit vector as $e = \mathbf{1}_N$ and $\Pi_{C^\perp}$ as the orthogonal projection on C so that*

$$\Pi_C = \frac{1}{N}ee^\top$$
$$\Pi_{C^\perp} = I - \frac{1}{N}ee^\top$$

It follows from the above that $\mathcal{L}\Pi_C = 0$ and $\mathcal{L}\Pi_{C^\perp} = \mathcal{L}$. We will use these properties of the orthogonal projections on C below. Next, suppose that $Q = \alpha I$ and $R = \mathcal{L}_\beta Q^{-1}\mathcal{L}_\beta$ then using Lemma 4.3 we find $\overline{\sigma}(\mathcal{L}) = \underline{\sigma}(\mathcal{L}) = 1$. If $\sigma = 0$, $\boldsymbol{\lambda}_0 = 0$ and $\delta = 1$ in (4.2) then the algorithm converges in one iteration. To see this note that,

$$\mathbf{x}_1 = \mathbf{x}_0 - (\alpha I + \mathcal{L})^{-1}((\alpha I + \mathcal{L})\mathbf{x}_0 - \alpha\mathbf{b})) = (\alpha\mathbf{I} + \mathcal{L})^{-\mathbf{1}}(\alpha\mathbf{b}),$$
$$\boldsymbol{\lambda}_1 = \mathcal{L}_\beta^{-1}(\alpha I + \mathcal{L})\mathcal{L}_\beta^{-1}\mathcal{L}\mathbf{x}_1.$$

Starting at an arbitrary $\mathbf{x}_0$, after some straightforward algebra we obtain,

$$\begin{aligned}
\mathbf{x}_2 &= (\alpha I + \mathcal{L})^{-1}(\alpha\mathbf{b} - \mathcal{L}\boldsymbol{\lambda}_\mathbf{1}) \\
&= (\alpha I + \mathcal{L})^{-1}(\alpha\mathbf{b} - \mathcal{L}\mathcal{L}_\beta^{-\mathbf{1}}(\alpha\mathbf{I} + \mathcal{L})\mathcal{L}_\beta^{-\mathbf{1}}\mathcal{L}(\alpha\mathbf{I} + \mathcal{L})^{-\mathbf{1}}(\alpha\mathbf{b})) \\
&= (\alpha I + \mathcal{L})^{-1}(\alpha\mathbf{b} - \Pi_{C^\perp}(\alpha I + \mathcal{L})\Pi_{C^\perp}(\alpha I + \mathcal{L})^{-1}(\alpha\mathbf{b})) \\
&= (\alpha I + \mathcal{L})^{-1}(I - \Pi_{C^\perp})(\alpha\mathbf{b}) = (\alpha I + \mathcal{L})^{-1}\Pi_C(\alpha\mathbf{b}),
\end{aligned}$$

where we used the facts that orthogonal projections are idempotent and the fact that $\mathcal{L}\Pi_\perp = \mathcal{L}$. Finally we use the fact that,

$$\begin{aligned}
(\alpha I + \mathcal{L})^{-1} = \alpha^{-1}(I + \alpha^{-1}\mathcal{L})^{-1} &= \alpha^{-1}(I + \alpha^{-1}\mathcal{L})^{-1}(I + \alpha^{-1}\mathcal{L} - \alpha^{-1}\mathcal{L}) \\
&= \alpha^{-1}I - \alpha^{-2}(I + \alpha^{-1}\mathcal{L})^{-1}\mathcal{L}.
\end{aligned}$$

Therefore,

$$\mathbf{x}_2 = \alpha^{-1}I - \alpha^{-2}(I + \alpha^{-1}\mathcal{L})^{-1}\mathcal{L}\Pi_C(\alpha\mathbf{b}),$$

and since $\mathcal{L}\Pi_C(\alpha\mathbf{b}) = 0$ *it follows that,*

$$\mathbf{x}_2 = \Pi_C(\mathbf{b}) = \frac{1}{N}\sum_{i=1}^{N} b_i.$$

Note that if we chose $Q = R = I$ *then the convergence (and step-size* δ*) of the algorithm would depend on the condition number of the graph.*

We are now ready to state the main result of this section.

Proposition 4.6. *Suppose that,*

$$\delta < \min\left\{\frac{1}{2L'_f}, \frac{\mu_f}{\underline{\sigma}(\mathcal{L})^2}\right\}. \tag{4.7}$$

Then,

$$\mathbb{E}[V(\mathbf{x}_k, \boldsymbol{\lambda}_k)] \leq \rho^k \mathbb{E}[V(\mathbf{x}_0, \boldsymbol{\lambda}_0)] + \frac{\delta\sigma^2 \mathrm{tr}(Q^{-1})}{2(1-\rho)}. \tag{4.8}$$

where,

$$V(\mathbf{x}_k, \boldsymbol{\lambda}_k) = (1 - \delta^2\overline{\sigma}(\mathcal{L})^2) D_\Phi(\mathbf{x}_k, \mathbf{x}^\star) + D_\Psi(\boldsymbol{\lambda}_k, \boldsymbol{\lambda}^\star), \tag{4.9}$$

and $0 < \rho = \max\{\rho_x, \rho_\lambda\} < 1$*, with* $\rho_x = 1 + \delta^2\mu_f L'_f - \delta\mu_f$*,* $\rho_\lambda = 1 - \delta^2\underline{\sigma}(\mathcal{L})^2$*.*

Proof. To simplify notation we will use $I(\xi_k)$ to denote a random variable with zero expectation and we will omit the exact terms when they will not play a role in the estimates. We expand the first term in (4.9) as follows,

$$\begin{aligned}
D_\Phi(\mathbf{x}_k, \mathbf{x}^\star) &= \frac{1}{2}\|\hat{\mathbf{x}}_k\|_Q^2 \\
&= \frac{1}{2}\|Q^{\frac{1}{2}}\hat{\mathbf{x}}_{k-1} - \delta Q^{-\frac{1}{2}}(\nabla \hat{f}_{k-1} + \mathcal{L}\hat{\boldsymbol{\lambda}}_{k-1}) + \sqrt{\delta}\sigma Q^{-\frac{1}{2}}\boldsymbol{\xi}_k\|^2 \\
&= \frac{1}{2}\|Q^{\frac{1}{2}}\hat{\mathbf{x}}_{k-1} - \delta Q^{-\frac{1}{2}}\nabla \hat{f}_{k-1}\|^2 + \frac{\delta^2}{2}\|Q^{-\frac{1}{2}}\mathcal{L}\hat{\boldsymbol{\lambda}}_{k-1}\|^2 \\
&\quad - \delta\langle \mathcal{L}\hat{\boldsymbol{\lambda}}_{k-1}, \hat{\mathbf{x}}_{k-1} - \delta Q^{-1}\nabla \hat{f}_{k-1}\rangle + \frac{\delta\sigma^2}{2}\|Q^{-\frac{1}{2}}\xi_k\|^2 + I(\xi_k).
\end{aligned}$$

Similarly for the second term in (4.9) we have,

$$\begin{aligned}
&D_\Psi(\boldsymbol{\lambda}_k, \boldsymbol{\lambda}^\star) = \frac{1}{2}\|\hat{\boldsymbol{\lambda}}_k\|_R^2 \\
&= \frac{1}{2}\|R^{\frac{1}{2}}\hat{\boldsymbol{\lambda}}_{k-1} + \delta R^{-\frac{1}{2}}\mathcal{L}\hat{\mathbf{x}}_k\|^2 \\
&= \frac{1}{2}\|R^{\frac{1}{2}}\hat{\boldsymbol{\lambda}}_{k-1}\|^2 + \frac{\delta^2}{2}\|R^{-\frac{1}{2}}\mathcal{L}\hat{\mathbf{x}}_k\| + \delta\langle \mathcal{L}\hat{\boldsymbol{\lambda}}_{k-1}, \hat{\mathbf{x}}_k\rangle \\
&= \frac{1}{2}\|R^{\frac{1}{2}}\hat{\boldsymbol{\lambda}}_{k-1}\|^2 + \frac{\delta^2}{2}\|R^{-\frac{1}{2}}\mathcal{L}\hat{\mathbf{x}}_k\| + \delta\langle \mathcal{L}\hat{\boldsymbol{\lambda}}_{k-1}, \hat{\mathbf{x}}_{k-1} - \delta Q^{-1}(\nabla \hat{f}_{k-1} + \mathcal{L}\hat{\boldsymbol{\lambda}}_{k-1})\rangle + I(\xi_k).
\end{aligned}$$

Therefore,

$$D_\Phi(\mathbf{x}_k,\mathbf{x}^\star)+D_\Psi(\boldsymbol{\lambda}_k,\boldsymbol{\lambda}^\star)=\frac{1}{2}\|Q^{\frac{1}{2}}\hat{\mathbf{x}}_{k-1}-\delta Q^{-\frac{1}{2}}\nabla\hat{f}_{k-1}\|^2-\frac{\delta^2}{2}\|Q^{-\frac{1}{2}}\mathcal{L}\hat{\boldsymbol{\lambda}}_{k-1}\|^2 + \frac{1}{2}\|R^{\frac{1}{2}}\hat{\boldsymbol{\lambda}}_{k-1}\|^2+\frac{\delta^2}{2}\|R^{-\frac{1}{2}}\mathcal{L}\hat{\mathbf{x}}_k\|^2+\frac{\delta\sigma^2}{2}\|Q^{-\frac{1}{2}}\xi_k\|^2+I(\xi_k). \tag{4.10}$$

It follows from (4.5),

$$\overline{\sigma}(\mathcal{L})^2 D_\Phi(\mathbf{x}_k,\mathbf{x}^\star)=\frac{1}{2}\overline{\sigma}(\mathcal{L})^2\|Q^{\frac{1}{2}}\hat{\mathbf{x}}_k\|^2\geq\frac{1}{2}\|R^{-\frac{1}{2}}\mathcal{L}\hat{\mathbf{x}}_k\|^2.$$

It follows from Lemma 4.4 $\boldsymbol{\lambda}_{k-1}\in\text{range}(\mathcal{L})$ and therefore using (4.6) we obtain,

$$\underline{\sigma}(\mathcal{L})^2 D_\Psi(\boldsymbol{\lambda}_{k-1},\boldsymbol{\lambda}^\star)=\frac{1}{2}\underline{\sigma}(\mathcal{L})^2\|R^{\frac{1}{2}}\hat{\boldsymbol{\lambda}}_{k-1}\|^2\leq\frac{1}{2}\|Q^{-\frac{1}{2}}\mathcal{L}\hat{\boldsymbol{\lambda}}_{k-1}\|^2.$$

We use the last two inequalities to bound (4.10) as follows,

$$\begin{aligned}\left(1-\delta^2\overline{\sigma}(\mathcal{L})^2\right)D_\Phi(\mathbf{x}_k,\mathbf{x}^\star)+D_\Psi(\boldsymbol{\lambda}_k,\boldsymbol{\lambda}^\star)\leq{}&\frac{1}{2}\|Q^{\frac{1}{2}}\hat{\mathbf{x}}_{k-1}-\delta Q^{-\frac{1}{2}}\nabla\hat{f}_{k-1}\|^2\\&+(1-\delta^2\underline{\sigma}(\mathcal{L})^2)D_\Psi(\boldsymbol{\lambda}_{k-1},\boldsymbol{\lambda}^\star)\\&+\frac{\delta\sigma^2}{2}\|Q^{-\frac{1}{2}}\xi_k\|^2+I(\xi_k).\end{aligned} \tag{4.11}$$

We can bound the first term in the inequality above using (4.3) from Lemma 4.2,

$$\begin{aligned}\frac{1}{2}\|Q^{\frac{1}{2}}\hat{\mathbf{x}}_{k-1}-\delta Q^{-\frac{1}{2}}\nabla\hat{f}_{k-1}\|^2&\leq D_\Phi(\mathbf{x}_{k-1},\mathbf{x}^\star)+\left(\frac{1}{2}\delta^2L_f'-\delta\right)\langle\nabla\hat{f}_{k-1},\hat{\mathbf{x}}_{k-1}\rangle\\&\leq(1+\delta^2\mu_fL_f'-2\delta\mu_f)D_\Phi(\mathbf{x}_{k-1},\mathbf{x}^\star)\\&=\rho_x(1-\delta^2\overline{\sigma}(\mathcal{L})^2)D_\Phi(\mathbf{x}_{k-1},\mathbf{x}^\star)+(\rho_x\delta^2\overline{\sigma}(\mathcal{L})^2-\delta\mu_f)D_\Phi(\mathbf{x}_{k-1},\mathbf{x}^\star)\\&\leq\rho_x(1-\delta^2\overline{\sigma}(\mathcal{L})^2)D_\Phi(\mathbf{x}_{k-1},\mathbf{x}^\star),\end{aligned}$$

where in the second inequality we used (4.4) and the step-size condition (4.7) that implies that $(\frac{1}{2}\delta^2L_f'-\delta)<0$. In the last inequality we used the fact that the step-size condition (4.7) means that $0<\rho_x<1$ and that the last term is negative. Using the bound above in (4.11) and the definition of in (4.9),

$$\begin{aligned}V(\mathbf{x}_k,\boldsymbol{\lambda}_k)&\leq\rho_x(1-\delta^2\overline{\sigma}(\mathcal{L})^2)D_\Phi(\mathbf{x}_{k-1},\mathbf{x}^\star)+\rho_\lambda D_\Psi(\boldsymbol{\lambda}_{k-1},\boldsymbol{\lambda}^\star)+\frac{\delta\sigma^2}{2}\|Q^{-\frac{1}{2}}\xi_k\|^2+I(\xi_k)\\&\leq\rho V(\mathbf{x}_{k-1},\boldsymbol{\lambda}_{k-1})+\frac{\delta\sigma^2}{2}\|Q^{-\frac{1}{2}}\xi_k\|^2+I(\xi_k)\end{aligned}$$

Taking expectation on both sides,

$$\mathbb{E}[V(\mathbf{x}_k,\boldsymbol{\lambda}_k)]\leq\rho\mathbb{E}[V(\mathbf{x}_{k-1},\boldsymbol{\lambda}_{k-1})]+\frac{\delta\sigma^2\text{tr}(Q^{-1})}{2}.$$

Iterating the inequality above,

$$\mathbb{E}[V(\mathbf{x}_k, \boldsymbol{\lambda}_k)] \leq \rho^k \mathbb{E}[V(\mathbf{x}_0, \boldsymbol{\lambda}_0)] + \frac{\delta\sigma^2 \text{tr}(Q^{-1})}{2} \sum_{i=0}^{k-1} \rho^k \leq \rho^k \mathbb{E}[V(\mathbf{x}_0, \boldsymbol{\lambda}_0)] + \frac{\delta\sigma^2 \text{tr}(Q^{-1})}{2(1-\rho)}$$

where in the last inequality we used the fact that $\sum_{i=0}^{k-1} \rho^k \leq \sum_{i=0}^{\infty} \rho^k = 1/(1-\rho)$. Finally note that the step-size condition (4.7) means that $0 < \rho < 1$ ∎

The convergence rate in (4.8) makes explicit the effect of the condition number of the objective function (L_f/μ_f) and the condition number of the graph ($\overline{\sigma}(\mathcal{L})/\underline{\sigma}(\mathcal{L})$). We illustrate this point in Figure 3. In Figure 3 we plot the convergence rate ρ as a function of the objective function's condition number. We set $Q = R = I$ and plot two different cases: (a) a fully connected graph ($\overline{\sigma}(\mathcal{L})/\underline{\sigma}(\mathcal{L}) = 1$) and (b) a graph with a less favorable condition number ($\overline{\sigma}(\mathcal{L})/\underline{\sigma}(\mathcal{L}) = 3$). When the graph is well conditioned (solid blue line) we see that ρ depends only on the condition number of the objective function. However, when the graph is not fully connected (dotted red line) we see that the convergence rate is independent of the objective function properties and is dominated by the condition number of the graph. After a certain threshold (indicated by the dotted vertical line, $L_f/\mu_f \approx 4.5$), the convergence rate is again dominated by the properties of the objective function. This example illustrates the effect of the graph's spectral properties on the convergence rate. As discussed previously setting $Q = I$, and $R = \mathcal{L}_\beta Q^{-1} \mathcal{L}_\beta$ eliminates the dependence of the convergence rate to the graph's spectral properties. Other choices for Φ, e.g. $\Phi = \nabla^2 f$ means that Ψ has a dual job of preconditioning both the primal and Lagrangian dual variables. We note that a graph condition number of $\overline{\sigma}(\mathcal{L})/\underline{\sigma}(\mathcal{L}) = 3$ is not particularly high. For example an undirected Erdös-Rènyi graph with $N = 100$ nodes and a probability of connection 0.25 has an average condition number of approximately 3.

5. Numerical Results. In this section we illustrate the performance of the proposed algorithm in different settings. Our intention is to illustrate that the proposed method can be used to reduce and sometimes eliminate the impact of the graph's spectral properties. All numerical experiments are run on a modern laptop with 64Gb memory and an 8-core CPU. Because we implemented several algorithms with different iteration cost we report CPU-times as opposed to the number of iterations required to find a solution.

The test problem is given by the following quadratic optimization problem,

$$\min_{x_i \in \mathcal{X}} \sum_{i=1}^{N} \|A_i x_i - b_i\|_2^2 \quad \text{s.t.} \mathcal{L}\mathbf{x} = 0. \tag{5.1}$$

Where $\mathcal{X} \subset \mathbb{R}^d$, $A \in \mathbb{R}^{m \times d}$, and $b_i \in \mathbb{R}^m$. In the simulations below we take $d = 50$, $M = 75$ and $N = 60$. The matrices A_i are generated randomly with a condition number of $\rho = 2$. We also generate an ill-conditioned variant of the problem in (5.1) by taking the absolute value of the entries in A_i and b_i. For the graph associated with the problem above we consider three different cases: (a) a fully connected graph, (b) a ring-of-cliques graph (with 5 cliques of 12 nodes) and (c) an Erdős-Rènyi graph. For $N = 20$ the three cases are illustrated in Figure 5d. For the experiments below we use EPISMD$-(\nabla^2\Phi, \nabla^2\Psi)$ to denote how (4.1) is implemented. For example EPISMD$-(\nabla^2 f, I)$ denotes that we use $\nabla^2\Phi = \nabla^2 f$ and that we

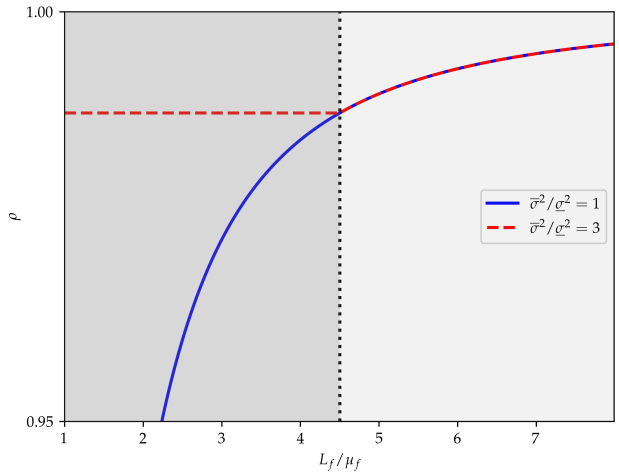
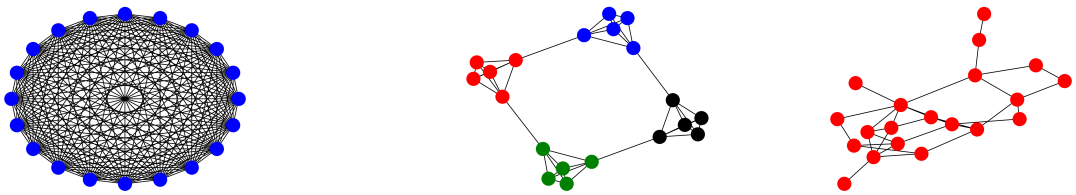


Figure 3: The algorithms convergence rate without preconditioning $Q = R = I$. When the graph is well-conditioned ($\overline{\sigma}^2/\underline{\sigma}^2 = 1$) the convergence rate of the algorithm (solid blue line) is dominated by the problem's condition number L_f/μ_f. When the graph is ill-conditioned ($\overline{\sigma}^2/\underline{\sigma}^2 = 3$) the convergence rate (dotted red line) is dominated by the spectral properties of the graph up to a certain condition number (in this case $L_f/\mu_f \approx 4.5$), after this point the convergence rate is again dominated by the model's condition number.



(a) A fully connected graph with $N = 20$ nodes

(b) A ring-of-cliques graph with $N = 20$ nodes and 4 cliques

(c) An Erdős-Rènyi graph with $N = 20$ nodes and probability of connection $p = 0.2$

Figure 4: Different types of graphs considered in the numerical experiments.

do not precondition the Lagrangian dual variables. Whenever we precondition the Lagrangian dual variables we use $R = \mathcal{L}_\beta(\nabla^2 f)^{-1}\mathcal{L}_\beta$ with $\beta = 0.0001$. For reasons of computational efficiency we use $R^{-1} = \mathcal{L}_\beta^{-1}(\nabla^2 f)\mathcal{L}_\beta^{-1}$ in our implementation. This is because we can compute $\mathcal{L}_\beta^{-1}$ using a one-off cost of an eigenvalue decomposition of an N dimensional matrix. The Hessian of f can be computed at a cost of $O(Nd^2)$. Assessing the efficiency or relative benefit

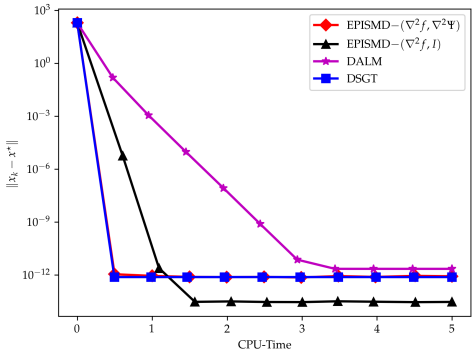
of computing the additional preconditioner is problem dependent, but we note this is only computed once at initialization. As will be illustrated in the numerical examples below, when the graph is ill-conditioned then it can improve the performance by orders of magnitude. We also note that there are a lot of ways to improve the way we compute the preconditioner. For example, we could use quasi-Newton style methods, statistical preconditioning (see [16]), multilevel methods (e.g. [43]) and specialized techniques for solving equations involving the Laplacian (see [44]).

Below we compare several algorithms with constant step-sizes. In order to be fair we tuned the step-size of all algorithms separately for best performance. Only the Distributed Projected Gradient Descent (DPGD) algorithm uses a diminishing step-size strategy, and we used the schedule described in [22]. The exact solution was computed using CVXPY [12].

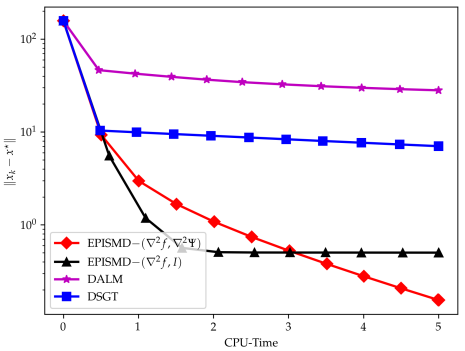
5.1. The Unconstrained Case. In the first set of numerical experiments we consider the unconstrained case, i.e. we take $\mathcal{X} = \mathbb{R}^d$ in (5.1). Since the problem is unconstrained we compare against the state-of-the-art for unconstrained problems. In our case we compare against the Distributed Stochastic Gradient Tracking (DSGT) method from [31]. DSGT is exact, has a linear convergence rate and can handle noisy gradient evaluations. In [31] it was compared against other algorithms and was found to outperform many of them. We also implemented the Distributed Augmented Lagrangian Method (DALM) analyzed in [24]. DALM is a good baseline to compare against since in the unconstrained case our algorithm reduces to the DALM if we take $Q = R = I$. Therefore any improvement against DALM can be attributed to our choice of mirror maps.

In our first experiment we consider a fully connected network, $\sigma = 0$, and a well conditioned problem. As can be seen from Figure 5a all algorithms converge to the same solution, any small variations are due to rounding errors. In our second experiment we change the condition number of the problem as described above. In Figure 5b we observe that EPISMD$-(\nabla^2 f, I)$ and EPISMD$-(\nabla^2 f, \nabla^2 \Psi)$ outperform the other methods. This observation is not surprising since the problem is ill-conditioned and EPISMD$-(\nabla^2 f, I)$ and EPISMD$-(\nabla^2 f, \nabla^2 \Psi)$ pre-condition the primal variables. We also see a small advantage of EPISMD$-(\nabla^2 f, \nabla^2 \Psi)$. This is because $\nabla^2 \Psi$ also contains information from the Hessian of f. In our next set of experiments we analyze the impact of using a ring-of-cliques graph (we used twelve cliques with five nodes in each clique). In Figure 5c we plot the results for the case where the problem is well-conditioned and in Figure 5d we plot the case where the problem is also ill-conditioned. It is clear from these two figures that the impact of the graph's spectral properties is significant for all methods except for EPISMD$-(\nabla^2 f, \nabla^2 \Psi)$. This is of course to be expected given the results of the previous section. When the problem is unconstrained EPISMD$-(\nabla^2 f, \nabla^2 \Psi)$ pre-conditions both the primal and dual variables and can therefore mitigate ill-conditioning that originates from either the model, the graph or both. Note that in the experimental setup of this section EPISMD$-(\nabla^2 f, \nabla^2 \Psi)$ is a preconditioned version of DALM. Therefore it would be interesting to develop a variant of DSGT that also adapts to the geometry of the graph's Laplacian.

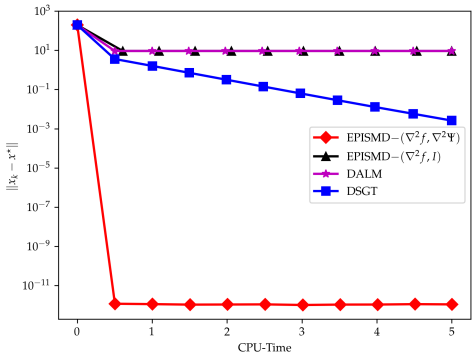
5.2. The Constrained Case. In this section we repeat the experiments from Subsection 5.1 but when $\mathcal{X} = \Delta$, where Δ denotes the d-dimensional simplex. In this case we take Φ to be the negative entropy $\Phi(x) = \sum_{i=1}^{d} \ln(x_i)$. Again EPISMD-$(\nabla^2 \Phi, \nabla^2 \Psi)$ is implemented



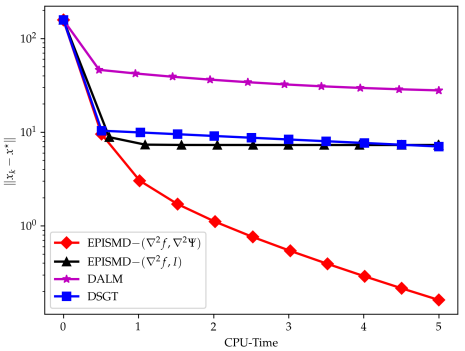
(a) Well conditioned model with a fully connected graph.



(b) Ill conditioned model with a fully connected graph.



(c) Well conditioned model on a ring-of-cliques graph.

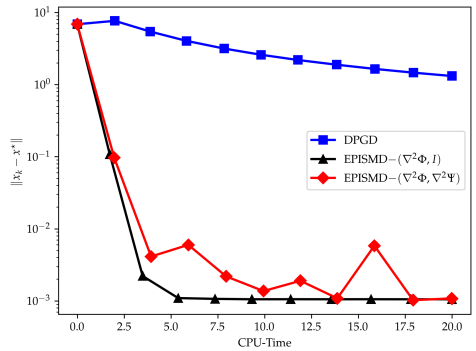


(d) Ill conditioned model on a ring-of-cliques graph.

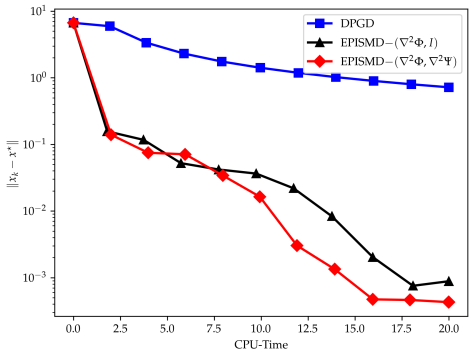
Figure 5: Numerical comparisons of distributed optimization algorithms for unconstrained problems. (a) When both the problem and graph are well-conditioned the proposed method performs similarly to other methods. (b-d) However when the model and/or graph are ill-conditioned then the proposed algorithm provides a significant improvement over other methods.

.

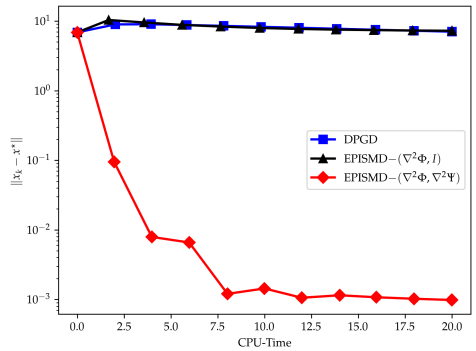
with $\nabla^2 \Psi = \mathcal{L}_\beta \nabla^2 f \mathcal{L}_\beta$. As a benchmark we use the Distributed Projected Gradient Method (DPGD) method described in [22]. Note that the DSGT and DPALM from the previous section cannot handle constraints. We repeat the same set of four experiments from Subsection 5.1 and plot the results in Figure 6. The results are similar as in the unconstrained case, but the impact of using the proposed preconditioner in conjunction with the conventional mirror maps for the primal variables is even more significant. The results demonstrate that using



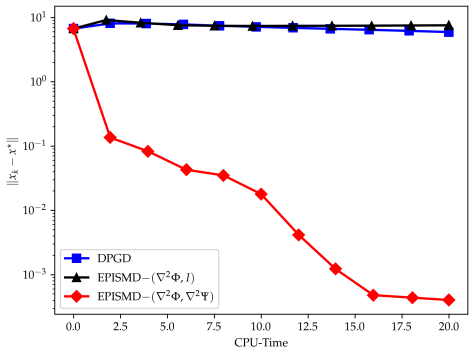
(a) Well conditioned model with a fully connected graph.



(b) Ill conditioned model with a fully connected graph.



(c) Well conditioned model with a “ring of cliques" graph.

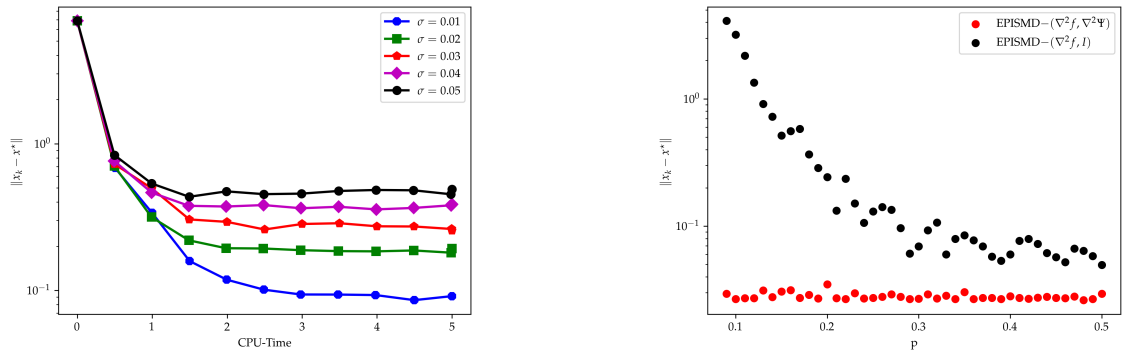


(d) Ill conditioned model with a “ring of cliques" graph.

Figure 6: Numerical comparisons of distributed optimization algorithms for simplex-constrained problems. Like the unconstrained case in Figure 5 the proposed method is particularly efficient when the graph is ill-conditioned.

the proposed mirror map on the Lagrangian dual variables has a significant impact when the graph is ill conditioned.

5.3. The Effect of Noise. As shown in Proposition 3.9 (continuous time) and Proposition 4.6 (discrete time) the effect of noise is an extra error term that depends on σ. To show that the conclusions of the numerical experiments above are still valid for the stochastic case we re-run all the experiments using different levels of noise. To save space we only show one particular instance of the model in (5.1). The instance we consider is when the graph is a ring of cliques, the constraint is the $d-$dimensional simplex and the problem is ill-conditioned.



(a) (b)

Figure 7: (a) The impact of different noise levels in a ring of cliques graph with a d-dimensional simplex constraint. (b) Experiment with different Erdős-Rènyi graphs as we vary the probability p that two nodes are connected (x-axis). In this case we see that the convergence rate of the preconditioned algorithm EPISMD$-(\nabla^2 f, \nabla^2 \Psi)$ is independent of p. This is not the case if we only precondition the primal variables using EPISMD$-(\nabla^2 f, I)$.

In other words, we used the same experimental setup used to produce Figure 6d but with different levels of noise. The results are shown in Figure 7a. As can be seen from these results the conclusions from the deterministic experiments are still valid in the stochastic case.

5.4. The Effect of the Graph's Connectivity. In our last set of experiments we show how the efficiency of the preconditioned dynamics in (4.1) are essentially unaffected by the graph's connectivity. For this experiment we take an Erdős-Rènyi graph with $N = 100$ nodes and run EPISMD$-(\nabla^2 \Phi, I)$ (i.e. no graph preconditioning) and EPISMD$-(\nabla^2 \Phi, \nabla^2 \Psi)$ (i.e. full preconditioning) on (5.1) over the $d-$dimensional simplex and when the problem is well-conditioned. We vary the probability p of two nodes being connected (the parameter in Erdős-Rènyi graphs) and run both algorithms for a fixed amount of CPU-time (10 seconds). For each probability p we generated 20 graphs and average the results. The results are shown in Figure 7b. As can be seen from the figure, the efficiency of EPISMD$-(\nabla^2 \Phi, \nabla^2 \Psi)$ is unaffected by the graph's connectivity; the same level of accuracy is reached independently of the probability p. On the other hand, the efficiency of EPISMD$-(\nabla^2 \Phi, I)$ depends on the probability of two nodes being connected. After a certain level $p \approx 0.5$ in this experiment we see that EPISMD$-(\nabla^2 \Phi, \nabla^2 \Psi)$ does not have a significant advantage over EPISMD$-(\nabla^2 \Phi, I)$, and only performs marginally better, but still the extra cost of preconditioning seems justified even if the graph is well connected.

6. Discussion. For future work several interesting extensions can be considered. One interesting direction is to study the proposed algorithm for sampling. For example, the discretization we discussed in this paper will lead to a biased method and the amount of bias will depend on the discretization interval. In this sense exactness is lost, but could be recovered

again using different methods from the sampling literature [14]. Addressing this in detail and comparing the differences between sampling and optimization in the context of distributed methods is an interesting direction for future work.

The analysis in this paper relies heavily on convexity. In preliminary numerical results not presented here EPISMD was used on a non-convex setting to train a small neural network with a negative entropy mirror map with promising performance. Future work can include a theoretical analysis of the non-convex case as well as exploring further the benefits of different mirror maps in the distributed setting. This is a challenging direction since the links with the dual are not exact. Nevertheless, the proposed preconditioner could be adapted to the non-convex setting using Levenberg–Marquardt regularization [2]. Finally, the proposed algorithm relies on second-order information. While this may seem prohibitive in practice, several recent works have shown that second order methods can be applied to problems where $d > 10^6$ using sub-sampling and multilevel optimization techniques [43]. Jointly sampling second-order information from the Hessian and the graph is an interesting direction to consider for large scale models.

Acknowledgements. We extend our sincere gratitude to the associate editors and the three reviewers for their valuable time, insightful comments, and thorough review of our manuscript. Their constructive feedback significantly improved the quality and clarity of our work. This project was funded by JPMorgan Chase & Co under J.P. Morgan A.I. Research Awards in 2019 and 2021. G.A.P. was partially supported by the EPSRC through grant number EP/P031587/1. P.P was partially supported by the EPSRC through grant number EP/W003317/1.

REFERENCES

[1] F. Alvarez, J. Bolte, and O. Brahic, *Hessian riemannian gradient flows in convex programming*, SIAM journal on control and optimization, 43 (2004), pp. 477–501.
[2] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: theory and algorithms*, John Wiley & Sons, 2013.
[3] A. Beck, *First-order methods in optimization*, vol. 25, SIAM, 2017.
[4] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*, 2015.
[5] A. Borovykh, P. Parpas, N. Kantas, and G. Pavliotis, *On stochastic mirror descent with interacting particles: convergence properties and variance reduction*, Physica D.Nonlinear Phenomena 418, (2021).
[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn., 3 (2011), p. 1–122.
[7] S. Bubeck, *Convex optimization: Algorithms and complexity*, arXiv preprint arXiv:1405.4980, (2014).
[8] F. Bullo, *Lectures on Network Systems*, Kindle Direct Publishing, 1.5 ed., 2021.
[9] Y. Censor and S. A. Zenios, *Proximal minimization algorithm with d-functions*, Journal of Optimization Theory and Applications, 73 (1992), pp. 451–464.
[10] S. Chewi, T. L. Gouic, C. Lu, T. Maunu, P. Rigollet, and A. Stromme, *Exponential ergodicity of mirror-Langevin diffusions*, arXiv preprint arXiv:2005.09669, (2020).
[11] P. Di Lorenzo and G. Scutari, *Next: In-network nonconvex optimization*, IEEE Transactions on Signal and Information Processing over Networks, 2 (2016), pp. 120–136.
[12] S. Diamond and S. Boyd, *Cvxpy: A python-embedded modeling language for convex optimization*, The Journal of Machine Learning Research, 17 (2016), pp. 2909–2913.
[13] J. C. Duchi, A. Agarwal, and M. J. Wainwright, *Dual averaging for distributed optimization: Con-*

vergence analysis and network scaling, IEEE Transactions on Automatic control, 57 (2011), pp. 592–606.

[14] A. Durmus and E. Moulines, *Sampling from strongly log-concave distributions with the unadjusted Langevin algorithm*, arXiv preprint arXiv:1605.01559, 5 (2016).

[15] B. Gharesifard and J. Cortés, *Distributed continuous-time convex optimization on weight-balanced digraphs*, IEEE Transactions on Automatic Control, 59 (2013), pp. 781–786.

[16] H. Hendrikx, L. Xiao, S. Bubeck, F. Bach, and L. Massoulie, *Statistically preconditioned accelerated gradient method for distributed optimization*, in International Conference on Machine Learning, PMLR, 2020, pp. 4203–4227.

[17] R. A. Horn and C. R. Johnson, *Topics in matrix analysis*, Cambridge university press, 1994.

[18] R. Li, M. Tao, S. S. Vempala, and A. Wibisono, *The mirror Langevin algorithm converges with vanishing bias*, in International Conference on Algorithmic Learning Theory, PMLR, 2022, pp. 718–742.

[19] S. Liang, L. Wang, and G. Yin, *Exponential convergence of distributed primal–dual convex optimization algorithm without strong convexity*, Automatica, 105 (2019), pp. 298–306.

[20] P. Lin, W. Ren, and J. A. Farrell, *Distributed continuous-time optimization: nonuniform gradient gains, finite-time convergence, and convex constraint set*, IEEE Transactions on Automatic Control, 62 (2016), pp. 2239–2253.

[21] S. Liu, Z. Qiu, and L. Xie, *Continuous-time distributed convex optimization with set constraints*, IFAC Proceedings Volumes, 47 (2014), pp. 9762–9767.

[22] ———, *Convergence rate analysis of distributed optimization with projected subgradient algorithm*, Automatica J. IFAC, 83 (2017), pp. 162–169.

[23] H. Lu, R. M. Freund, and Y. Nesterov, *Relatively smooth convex optimization by first-order methods, and applications*, SIAM Journal on Optimization, 28 (2018), pp. 333–354.

[24] D. Mateos-Nunez and J. Cortes, *p th moment noise-to-state stability of stochastic differential equations with persistent noise*, SIAM Journal on Control and Optimization, 52 (2014), pp. 2399–2421.

[25] P. Mertikopoulos and M. Staudigl, *On the convergence of gradient-like flows with noisy gradient input*, SIAM Journal on Optimization, 28 (2018), pp. 163–197.

[26] C. D. Meyer, *Matrix analysis and applied linear algebra*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second ed., [2023] ©2023.

[27] S. P. Meyn and R. L. Tweedie, *Stability of Markovian processes iii: Foster–Lyapunov criteria for continuous-time processes*, Advances in Applied Probability, 25 (1993), pp. 518–548.

[28] A. Nedić, S. Lee, and M. Raginsky, *Decentralized online optimization with global objectives and local communication*, in 2015 American Control Conference (ACC), IEEE, 2015, pp. 4497–4503.

[29] A. S. Nemirovsky and D. B. Yudin, *Problem complexity and method efficiency in optimization.*, (1983).

[30] Y. Nesterov, *Lectures on convex optimization*, vol. 137, Springer, 2018.

[31] S. Pu and A. Nedić, *Distributed stochastic gradient tracking methods*, Mathematical Programming, 187 (2021), pp. 409–457.

[32] G. Qu and N. Li, *Harnessing smoothness to accelerate distributed optimization*, IEEE Transactions on Control of Network Systems, 5 (2017), pp. 1245–1260.

[33] M. Raginsky and J. Bouvrie, *Continuous-time stochastic mirror descent on a network: Variance reduction, consensus, convergence*, in 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), IEEE, 2012, pp. 6793–6800.

[34] S. S. Ram, A. Nedić, and V. V. Veeravalli, *Distributed stochastic subgradient projection algorithms for convex optimization*, Journal of optimization theory and applications, 147 (2010), pp. 516–545.

[35] S. Shahrampour and A. Jadbabaie, *Distributed online optimization in dynamic environments using mirror descent*, IEEE Transactions on Automatic Control, 63 (2017), pp. 714–725.

[36] G. Shi, A. Proutiere, and K. H. Johansson, *Network synchronization with convexity*, SIAM Journal on Control and Optimization, 53 (2015), pp. 3562–3583.

[37] W. Shi, Q. Ling, G. Wu, and W. Yin, *EXTRA: An exact first-order algorithm for decentralized consensus optimization*, SIAM Journal on Optimization, 25 (2015), pp. 944–966.

[38] S. Sra, S. Nowozin, and S. J. Wright, *Optimization for machine learning*, Mit Press, 2012.

[39] Y. Sun, M. Fazlyab, and S. Shahrampour, *On centralized and distributed mirror descent: Conver-*

gence analysis using quadratic constraints, IEEE Transactions on Automatic Control, (2022).
[40] Y. Sun, G. Scutari, and A. Daneshmand, *Distributed optimization based on gradient tracking revisited: enhancing convergence rate via surrogation*, SIAM J. Optim., 32 (2022), pp. 354–385.
[41] Y. Sun and S. Shahrampour, *Distributed mirror descent with integral feedback: asymptotic convergence analysis of continuous-time dynamics*, IEEE Control Syst. Lett., 5 (2021), pp. 1507–1512.
[42] ———, *Linear convergence of distributed mirror descent with integral feedback for strongly convex problems*, in 2021 60th IEEE Conference on Decision and Control (CDC), IEEE, 2021, pp. 1683–1688.
[43] N. Tsipinakis and P. Parpas, *A multilevel method for self-concordant minimization*, arXiv preprint arXiv:2106.13690, (2021).
[44] N. K. Vishnoi, *Lx = b Laplacian solvers and their algorithmic applications*, Found. Trends Theor. Comput. Sci., 8 (2012), pp. front matter, 1–141.
[45] G. Wanner and E. Hairer, *Solving ordinary differential equations II*, vol. 375, Springer Berlin Heidelberg New York, 1996.
[46] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, *Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes*, in 2015 54th IEEE Conference on Decision and Control (CDC), IEEE, 2015, pp. 2055–2060.
[47] X. Zeng, P. Yi, and Y. Hong, *Distributed continuous-time algorithm for constrained convex optimizations via nonsmooth analysis approach*, IEEE Transactions on Automatic Control, 62 (2016), pp. 5227–5233.
[48] K. S. Zhang, G. Peyré, J. Fadili, and M. Pereyra, *Wasserstein control of mirror Langevin Monte Carlo*, in Conference on Learning Theory, PMLR, 2020, pp. 3814–3841.

Appendix A. Auxiliary results.

Proof of Lemma 3.5. Using the definition of the pseudo-inverse in (3.1) and its relationship with the inverse of the regularized Laplacian in (3.4) we obtain,

$$\begin{aligned}\langle \mathbf{x}, \mathcal{L}\mathbf{x}\rangle &= \langle \mathbf{x}, \mathcal{L}\mathcal{L}^{+}\mathcal{L}\mathbf{x}\rangle = \langle \mathcal{L}\mathbf{x}, (\mathcal{L}_\beta^{-1} - \frac{1}{\beta N}\mathbf{1}_d\mathbf{1}_d^\top \otimes I_N)\mathcal{L}\mathbf{x}\rangle \\ &= \langle \mathcal{L}\mathbf{x}, \mathcal{L}_\beta^{-1}\mathcal{L}\mathbf{x}\rangle,\end{aligned}$$

where in the last equality we used the fact that $(\mathbf{1}_d\mathbf{1}_d^\top \otimes I_N)\mathcal{L} = 0$. Since $\mathcal{L}_\beta \preceq \kappa_{\beta,N} I$ then $\mathcal{L}_\beta^{-1} \succeq \kappa_{\beta,N}^{-1} I$ and the result follows. ∎

Proof of Lemma 3.6. We note that $A(\mathbf{x})$ can be obtained by removing the last Nd columns and rows of the following matrix,

$$B(\mathbf{x}) := \begin{bmatrix} \nabla^2 f(\mathbf{x}) + \mathcal{L} & \mathcal{L} \\ -\mathcal{L} & 0 \end{bmatrix}.$$

Let $\mathbf{d} = [\mathbf{d}_x^\top, \mathbf{d}_\lambda^\top]^\top$ and note that $\langle \mathbf{d}, B(\mathbf{x})\mathbf{d}\rangle = \mathbf{d}_x^\top \nabla^2 f(\mathbf{x})\mathbf{d}_x$. It follows from the relative strong convexity assumption that $B(\mathbf{x}) \succeq \mu_\Phi \nabla^2\Phi(\mathbf{x})$ and therefore $\|B(\mathbf{x})\|^2_{\nabla^2\Phi(\mathbf{z})^{-1}} > 0$. Since $A(\mathbf{x})$ can be obtained by removing the last Nd columns and rows of $B(\mathbf{x})$ the result follows from the interlacing theorem for singular values, see e.g. Theorem 3.1.3 in [17]. ∎

Proof of Lemma 3.7. Since f is twice differentiable there exists an $\mathbf{y}$ on the line segment joining $\mathbf{x}$ and $\mathbf{x}^\star$ such that $\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}^\star) = \langle \nabla^2 f(\mathbf{y}), \mathbf{x} - \mathbf{x}^\star \rangle$. We then have,

$$\begin{aligned}
\|\nabla f(\mathbf{x}) + \mathcal{L}\boldsymbol{\lambda} + \mathcal{L}\mathbf{x}\|^2_{\nabla^2\Phi^*(\mathbf{z})} &= \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}^\star) + \mathcal{L}(\boldsymbol{\lambda}_t - \boldsymbol{\lambda}^*) + \mathcal{L}(\mathbf{x} - \mathbf{x}^*)\|^2_{\nabla^2\Phi^*(\mathbf{z})} \\
&= \|A(\mathbf{y})[\mathbf{x}_t - \mathbf{x}^*, \boldsymbol{\lambda} - \boldsymbol{\lambda}^*]^T\|^2_{\nabla^2\Phi^*(\mathbf{z})} \\
&\geq \kappa_g \left(\|\mathbf{x}^* - \mathbf{x}_t\|^2 + \|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}\|^2\right) \\
&\geq \frac{2\kappa_g}{\hat{\mu}} \left(\sum_{i=1}^N D_\Phi(x^\star, x^i) + D_\Psi(\lambda^\star, \lambda^i) \right). \blacksquare
\end{aligned}$$

We use (3.5) to obtain the bound in terms of the dual variables.

A.1. When first-order optimization fails. Our first result shows that if there exists an $x^\star$ such that $\nabla f_i(x^\star) = 0$ for all $i = 1, ..., N$, exact consensus can be obtained for ISMD.

Lemma A.1. *Let Assumptions 1-3 hold. Consider the dynamics in (2.2) with $\sigma = 0$. If*

$$\bigcap_{i=1}^N \{\nabla f_i(x) = 0\} \neq \emptyset, \tag{A.1}$$

then $\lim_{t\to\infty} x_t^i = x^\star$, where $x^\star$ is an optimal point for (1.1) such that $x^\star = \min_{x\in\mathcal{X}^\star} D_\Phi(x, x_0)$.

Proof. Let x_0 be the initial point of the algorithm, and let $z^\star$ be an optimal (dual) point closest to $z_0 = \nabla\Phi(x_0)$ with respect to the divergence generated by Φ^*,

$$z^\star = \arg\min_{z\in Z^\star} D_{\Phi^*}(z, z_0)$$

where $Z^\star = \{z \mid z = \nabla\Phi(x), \exists x \in \mathcal{X} : \nabla f_i(x) = 0, i = 1, \ldots, N\}$. By assumption (A.1), $Z^\star$ is not empty. With a slight abuse of notation we let $x^\star = \nabla\Phi^*(z^\star)$ and note that $(x^\star, z^\star)$ is an equilibrium point for (2.2) (for a strongly convex function it is also the unique equilibrium point, but here we only assume convexity of f).

Define the Lyapunov candidate function $V_t = \sum_{i=1}^N D_{\Phi^*}(z_t^i, z^\star)$, and note that given our assumptions on f, and Φ it follows that V is a proper function (has compact sub-level sets). Then we obtain,

$$dV_t = \sum_{i=1}^N (x^\star - x_t^i)^T \nabla f_i(x_t^i) dt + \sum_{i=1}^N (x_t^i - x^\star)^T \sum_{j=1}^N A_{ij}(z_t^j - z_t^i) dt.$$

Under convexity of f and optimality at $x^\star$ we have

$$\sum_{i=1}^N (x^\star - x_t^i)^T \nabla f_i(x_t^i) \leq \sum_{i=1}^N (f_i(x^\star) - f_i(x_t^i)) \leq 0.$$

By the triangle equality of the Bregman divergence in (3.6),

$$\begin{aligned}
(x_t^i - x^\star)^T (z_t^j - z_t^i) &= -(x^\star - x_t^i)^T (z_t^j - z_t^i) \\
&= -(\nabla\Phi^*(z^\star) - \nabla\Phi^*(z_t^i))^T (z_t^j - z_t^i) \\
&= -D_{\Phi^*}(z_t^j, z_t^i) - D_{\Phi^*}(z_t^i, z^\star) + D_{\Phi^*}(z_t^j, z^\star).
\end{aligned}$$

Then,

$$\sum_{i=1}^{N}\sum_{j=1}^{N}A_{ij}(x_t^i-x^\star)^T(z_t^j-z_t^i) \tag{A.2}$$
$$=\sum_{i=1}^{N}\sum_{j=1}^{N}A_{ij}\left(-D_{\Phi^*}(z_t^j,z_t^i)-D_{\Phi^*}(z_t^i,z^\star)+D_{\Phi^*}(z_t^j,z^\star)\right)\leq 0,$$

where we used $A_{ij}\geq 0$, $\sum_{i=1}^N\sum_{j=1}^N D_{\Phi^*}(z_t^i,z^\star)=\sum_{i=1}^N\sum_{j=1}^N D_{\Phi^*}(z_t^j,z^\star)$, and $D_{\Phi^*}(z_t^j,z_t^i)\geq 0$. Since $V_t>0$ for $\mathbf{z}\neq \mathbf{1}_N\otimes z^\star$, $V_t=0$ when $\mathbf{z}=\mathbf{1}_N\otimes z^\star$ and $dV_t\leq 0$ with equality only at $\mathbf{z}=\mathbf{1}_N\otimes z^\star$ we conclude that V_t is a Lyapunov function for $\mathbf{z}_t$. Also note that since $D_{\Phi^*}(z_t^i,z^\star)=D_\Phi(x^\star,x_t^i)$ V_t is also Lyapunov function for $\mathbf{x}_t$. Since all the Lyapunov stability criteria are satisfied (see e.g. c[8, Theorem 15.4]) it follows that $x^\star$ is globally asymptotically stable for the dynamics in (2.2). ∎

The Lemma above can be extended to an if and only if statement based on the arguments of [36, Theorem 1], but precise details lie beyond the scope of this paper. If (A.1) is violated, even with the right choice of mirror map, achieving exact consensus is not possible. In general imposing $x^\star$ to satisfy (A.1) is quite restrictive as $\nabla f(\mathbf{x})=\sum_{i=1}^N\nabla f_i(x^\star)=0$ does not necessarily imply $\nabla f_i(x^\star)=0$ for all $i=1,...,N$. The crucial point to realize here is that if and only if (A.1) holds then $(\mathbf{x}^\star,\mathbf{z}^\star)$ will also be the minimizer of $f(\mathbf{x})+\frac{1}{2}\mathbf{z}^T\mathcal{L}\mathbf{z}$; see [36, Lemma 7] for details. As a result one can establish consensus at equilibrium and V_t will approach zero at large t.

If $\mathbf{x}^\star$ does not satisfy (A.1) and one has just $\nabla f(\mathbf{x}^\star)=0$, even in the deterministic case exact consensus and optimality at convergence can no longer be achieved. But the arguments above can be used to establish exponential convergence of (2.2) to a neighborhood of $(\mathbf{x}^\dagger,\mathbf{z}^\dagger)$ both minimizing $f(\mathbf{x})+\frac{1}{2}\mathbf{z}^T\mathcal{L}\mathbf{z}$. The size of this neighborhood naturally depends on the noise and f.

Proposition A.2 (Approximate convergence of (2.2)). *Let Assumptions 1-3 hold and assume that f is μ_f-strongly convex w.r.t. Φ. Let $\mathbf{x}^\dagger=\arg\min\{f(\mathbf{x})+\frac{1}{2}\nabla\Phi(\mathbf{x})^T\mathcal{L}\nabla\Phi(\mathbf{x})\}$ with $\mathbf{x}^\dagger=1_N\otimes x^\dagger$ and $V_t=\frac{1}{N}\sum_{i=1}^N D_{\Phi^*}(z_t^i,z^\dagger)$, where z_t^i obeys the dynamics of (2.1) (or (2.2)). Then we have:*

$$\mathbb{E}\left[V_t\right]\leq e^{-\eta\mu_f t}\frac{1}{N}\sum_{i=1}^N D_{\Phi^*}(z_0^i,z^\dagger)+\frac{\sigma^2}{2\eta\mu_f}||\Delta\Phi^*||_\infty+\frac{1}{\mu_f}\left(f(\mathbf{x}^\dagger)-f(\mathbf{x}^\circ)\right),$$

where $x^\circ=\arg\min_{\mathbf{x}\in\mathcal{X}}f(\mathbf{x})$.

Proof. Denote $\mathbf{x}^\dagger=\arg\inf\{f(\mathbf{x})+\frac{1}{2}\nabla\Phi(\mathbf{x})^T\mathcal{L}\nabla\Phi(\mathbf{x})\}$, $\mathbf{z}^\dagger=\nabla\Phi(\mathbf{x}^\dagger)$. Then

$$dV_t=-\frac{1}{N}\sum_{i=1}^N(x_t^i-x^\dagger)^T\eta\nabla f_i(x_t^i)dt+\epsilon\frac{1}{N}\sum_{i=1}^N(x_t^i-x^\dagger)^T\sum_{j=1}^N A_{ij}(z_t^j-z_t^i)dt$$
$$+\frac{1}{2}\sigma^2\frac{1}{N}\sum_{i=1}^N\mathrm{tr}(\Delta\Phi^*(z_t^i))dt+\frac{1}{N}\sum_{i=1}^N(x_t^i-x^\dagger)^T dB_t^i. \tag{A.3}$$

Using the μ_f-strong convexity of f we obtain,

$$\begin{aligned}\nabla f(\mathbf{x}_t)^T(\mathbf{x}^\dagger - \mathbf{x}_t) \leq & f(\mathbf{x}^\dagger) - f(\mathbf{x}_t) - \mu_f D_\phi(\mathbf{x}^\dagger, \mathbf{x}_t) \\ \leq & f(\mathbf{x}^\dagger) - f(\mathbf{x}^\circ) + f(\mathbf{x}^\circ) - f(\mathbf{x}_t) - \mu_f D_\phi(\mathbf{x}^\dagger, \mathbf{x}_t) \\ \leq & f(\mathbf{x}^\dagger) - f(\mathbf{x}^\circ) - \mu_f D_\phi(\mathbf{x}^\dagger, \mathbf{x}_t)\end{aligned}$$

Substituting in (A.3) and using (3.5), (A.2) and taking expectations gives

$$\frac{d\mathbb{E}[V_t]}{dt} \leq -\eta\mu_f \mathbb{E}[V_t] + \eta\left(f(\mathbf{x}^\dagger) - f(\mathbf{x}^\circ)\right) + \frac{1}{2}\sigma^2 ||\Delta\Phi^*||_\infty.$$

Standard Grönwall arguments gives the result. ∎

While the relative strong convexity of the objective function can speed up convergence, even in the no noise setting the above bound does not guarantee that the algorithm is able to converge exactly to the optimum. In this setup the additional preconditioning via the mirror map does not facilitate exact convergence nor consensus. When (A.1) holds, then $\mathbf{x}^\circ$, $\mathbf{x}^\star$ and $\mathbf{x}^\star$ coincide [36, Lemma 7], so $f(\mathbf{x}^\dagger) - f(\mathbf{x}^\circ) = 0$.