Data-Driven Distributed Optimization via Aggregative Tracking and Deep Learning

Riccardo Brumali, *Graduate Student Member, IEEE*, Guido Carnevale, *Member, IEEE*, and Giuseppe Notarstefano, *Member, IEEE*

***Abstract*—In this article, we propose a novel distributed data-driven optimization scheme. In particular, we focus on the so-called aggregative framework, namely, the scenario in which a set of agents aim to cooperatively minimize the sum of local costs, each depending on both local decision variables and an aggregation of all of them. We consider a data-driven setup in which each objective function is unknown and can be only sampled at a single point per iteration (thanks to, e.g., feedback from human users or physical sensors). We address this scenario through a distributed algorithm that combines three key components: first, a learning part that leverages neural networks to learn the local cost functions' descent direction, second, an optimization routine that steers the estimates according to the learned direction to minimize the global cost, and third, a tracking mechanism that locally reconstructs the unavailable global quantities. By using tools from system theory, i.e., timescale separation and averaging theory, we formally prove that, in strongly convex setups, the overall distributed strategy linearly converges in a neighborhood of the optimal solution whose radius depends on the given accuracy capabilities of the neural networks. Finally, we corroborate the theoretical results with numerical simulations.**

***Index Terms*—Data-driven distributed optimization, distributed algorithms/control, learning, networks of autonomous agents, optimization.**

I. Introduction

DISTRIBUTED optimization has recently gained significant attention across various domains. The comprehensive overview provided in the surveys [1], [2], [3] discusses the most popular setups and the algorithms used to address these challenges.

The mentioned surveys do not include the so-called aggregative optimization framework, which is particularly suited to model tasks arising in cooperative robotics; see the recent tutorial [4]. This novel distributed optimization scenario has been introduced by the pioneering work [5] and deals with networks of agents aiming at cooperatively minimizing the sum of local functions, each depending on both microscopic variables (e.g., the position of a single robot) and macroscopic ones (e.g., the spatial distribution of the entire team). Online and constrained versions of the aggregative problem are considered in [6]. Other works on distributed aggregative optimization include [7], in which a distributed method based on the Frank–Wolfe update is proposed, and [8] and [9], in which momentum-based algorithms are proposed. Authors in [10] propose a continuous-time algorithm with nonuniform gradient gains that only requires the sign of relative state information between agents' neighbors. While in [11], a compressed communication scheme is interlaced with the solution proposed in [5]. Aggregative optimization scenarios with uncertain environments have been explored in [12], where the distributed scheme proposed in [5] is enhanced with a Recursive Least Square (RLS) method estimating the unknown cost via feedback from the users. A similar framework is addressed in [13], where instead the learning part relies on a neural network.

Unknown environments as those mentioned above are particularly interesting in the context of the so-called *personalized* optimization frameworks. In this field, the goal is to minimize cost functions given by the sum of a known part, named engineering function and related to measurable quantities (e.g., time or energy), and an unknown part representing the user's (dis)satisfaction with the current solution. Since synthetic models based on human preferences often perform well only on average, failing to address the unique preferences of individual users, personalized optimization emphasizes data-driven strategies. These approaches leverage feedback from specific users about the current solution to adapt and better meet their individual needs. With growing human–robot collaboration, personalized optimization techniques can greatly enhance the control of cooperative robotic networks. In centralized optimization, an initial step toward incorporating user feedback to define human discomfort was undertaken by [14], where a trajectory design problem was addressed using a cost function influenced by human complaints. In [15], personalized optimization is explored by integrating a learning mechanism based on Gaussian processes (GP) with an optimization approach. Similarly, [16] applies a personalized framework within the context of game theory. As for personalized distributed frameworks, the authors in [17] and [18], respectively, use GP combined with a primal–dual method and RLS merged with the gradient tracking algorithm.

Zeroth-Order (ZO) optimization methods are a popular solution to address problems with unknown cost functions; see [19] for an overview on this topic. Within this class, the so-called 1-

Received 14 August 2025; revised 12 December 2025; accepted 17 December 2025. Date of publication 12 January 2026; date of current version 22 June 2026. This work was supported by the European Union - NextGenerationEU under the National Recovery and Resilience Plan (PNRR) - Mission 4 Education and research - Component 2 From research to business - Investment 1.1 Notice Prin 2022, from title ECODREAM Energy COmmunity management: DistRibutEd AlgorithMs and toolboxes for efficient and sustainable operations, under Grant 202228CTKY002 - CUP J53D23000560006. Recommended by Associate Editor J. A. Taylor. *(Corresponding author: Riccardo Brumali.)*

The authors are with the Department of Electrical, Electronic, and Information Engineering, Alma Mater Studiorum - Università di Bologna, 40136 Bologna, Italy (e-mail: riccardo.brumali@unibo.it; guido.carnevale@unibo.it; giuseppe.notarstefano@unibo.it).

Digital Object Identifier 10.1109/TCNS.2026.3650964

point optimization methods are particularly appealing since they only require a single function evaluation per iteration and, thus, are particularly suited to scenarios where cost function evaluations are expensive or multipoint evaluations are not possible, such as when the environment is nonstationary (see [20]). Examples arise in online optimization, control, and reinforcement learning settings (e.g. [21]). These methods have been widely explored in the centralized literature (see the early work [22] and [23] or the most recent ones [20], [24], [25], and [26]), while a recent extension to the distributed setting is proposed in [27]. Despite their efficiency, 1-point methods typically underperform compared to multievaluation ZO methods. As we will better detail later, we overcome this issue by embedding the use of neural networks in our 1-point architecture.

The use of neural networks in optimization and control is a recent interesting research trend. For example, in [28] and [29], they are integrated into feedback controllers to estimate the system state from perceptual information. In [30], neural networks are combined with a ZO scheme, while [31] introduced a general framework for the verification of neural network controllers. Authors in [32] employed neural networks to learn stabilizing policies for nonlinear systems. The work [33] proposes a framework to learn from data algorithms for optimizing nonconvex functions. Further neural networks have been widely adopted in model predictive control algorithms, where they act as approximators for control policies. In this context, [34] proposes a method to certify the reliability of the neural network-based controller, while in [35], the problem is addressed via HardTanh deep neural networks. Finally, in [36], a set point tracking scenario for a robot manipulator is considered.

In this article, we introduce DEep-Learning aggregative TrAcking (DELTA), a novel distributed data-driven optimization scheme for aggregative problems. The peculiarity of DELTA is to tackle the typical convergence issues of standard 1-point methods based on static gradient approximations by integrating a dynamic learning strategy based on neural networks that does not increase the number of cost evaluations per iteration. More in detail, DELTA operates on a single timescale and integrates three core components: 1) a learning-oriented module, 2) an optimization-oriented routine, and 3) a tracking-oriented mechanism. The learning-oriented component uses local neural networks to asymptotically estimate the correct costs (and their gradients) in a data-driven fashion. Specifically, each neural network employs only a single cost evaluation per iteration taken in the neighborhood of the current local estimates. The optimization-oriented component applies an approximated distributed gradient method to the aggregative problem. Its inexactness arises from two factors: 1) the fact that the cost functions are unknown and approximated by the networks, and 2) the need for global quantities that are locally unavailable. The tracking-oriented component addresses the second challenge by locally reconstructing the global quantities, i.e., the aggregative variable and the derivative of the global cost with respect to that variable. We analyze DELTA using tools from system theory based on timescale separation and averaging theory to formally prove that, in strongly convex settings, the algorithm linearly converges in a neighborhood of the optimal solution whose radius depends on the given accuracy capabilities of the neural networks. A preliminary version of this work focusing on the time-varying settings is available in [13]. However, in that version, the addressed scenario is significantly simpler. Indeed, at each iteration, agents can 1) access multiple samples of their cost functions and 2) completely train their neural networks until convergence between two consecutive optimization steps. In contrast, here, at each iteration, we can access the cost functions only at a single point, and the neural networks are trained only for a single step. From a technical point of view, this translates in suitably integrating also the learning dynamics analysis in the optimization and consensus one. In addition, in [13], the local cost functions are partially known, while in this article, they are completely unknown.

The rest of this article is organized as follows. In Section II, we present the problem setup. In Section III, we introduce the DELTA algorithm. The main theoretical result is provided in Section III-D, while its analysis is detailed in Section IV. In Section V, we validate our analysis via numerical simulations. Finally, Section VI concludes this article.

Notations: We denote the column stacking of vectors $x_1, \dots, x_N$ with $\mathrm{col}(x_1, \dots, x_N)$. The m-dimensional identity matrix is denoted by I_m. The symbols 1_N and 0_m denote the vectors of N ones and m zeros, respectively, while $\mathbf{1}_{N,d} := 1_N \otimes I_d$, where $\otimes$ denotes the Kronecker product. Dimensions are omitted when they are clear from the context. Given $f : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^{n_3} \to \mathbb{R}^n$, we define $\nabla_1 f(x, y, \theta) := \frac{\partial}{\partial s} f(s, y, \theta)|_{s=x}^{\top}$, $\nabla_2 f(x, y, \theta) := \frac{\partial}{\partial s} f(x, s, \theta)|_{s=y}^{\top}$, and $\nabla_3 f(x, y, \theta) := \frac{\partial}{\partial s} f(x, y, s)|_{s=\theta}^{\top}$. Given $v \in \mathbb{R}^n$ and $r > 0$, we define $\mathcal{B}_r(v) := \{x \in \mathbb{R}^n \mid \|x - v\| \leq r\}$. Given the matrices $M_1, \dots, M_N$, we use $\mathrm{blkdiag}(M_1, \dots, M_N)$ to denote the block-diagonal matrix having M_i on the i-th block.

II. Problem Formulation

We consider N agents that aim to cooperatively solve

$$\min_{(x_1,\dots,x_N)\in\mathbb{R}^n} \sum_{i=1}^{N} f_i\left(x_i, \sigma(x)\right), \tag{1}$$

in which $x := \mathrm{col}(x_1, \dots, x_N) \in \mathbb{R}^n$ is the global decision vector, with each $x_i \in \mathbb{R}^{n_i}$ and $n := \sum_{i=1}^{N} n_i$. For all $i \in \{1, \dots, N\}$, each function $f_i : \mathbb{R}^{n_i} \times \mathbb{R}^d \to \mathbb{R}$ represents the *unknown* local objective function of agent i depending on both the local decision variable x_i and the aggregative one given by

$$\sigma(x) := \frac{1}{N} \sum_{i=1}^{N} \phi_i(x_i), \tag{2}$$

where each $\phi_i : \mathbb{R}^{n_i} \to \mathbb{R}^d$ is the i-th contribution to σ. As already mentioned above, the functions f_i are unknown. More in detail, we focus on data-driven scenarios where each agent i can only receive a single feedback per iteration about the unknown local cost f_i. We formalize this aspect as follows.

Assumption 1: For all $i \in \{1, \dots, N\}$, f_i is unknown but agent i can receive a single feedback $f_i(u_{x,i}, u_{\sigma,i})$ per iteration, where $(u_{x,i}, u_{\sigma,i}) \in \mathbb{R}^{n_i} \times \mathbb{R}^d$ can be arbitrarily chosen. ■

The choice of $(u_{x,i}, u_{\sigma,i})$ will represent a design aspect of the overall distributed algorithm we aim to propose. As one may expect and as we will see later, agent i will choose $(u_{x,i}, u_{\sigma,i})$ in the neighborhood of the current local estimates about 1) the i-th block of a problem solution and 2) the aggregative variable. The arising data-driven scenario is sketched in Fig. 1.

Now, we formally characterize the functions appearing in problem (1). To this end, we first introduce $f_\sigma : \mathbb{R}^n \to \mathbb{R}$ to denote the overall cost function of problem (1), namely

$$f_\sigma(x) := \sum_{i=1}^{N} f_i(x_i, \sigma(x)).$$

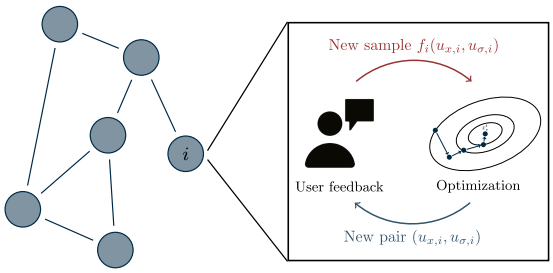

Fig. 1. Graphical representation of the problem framework.

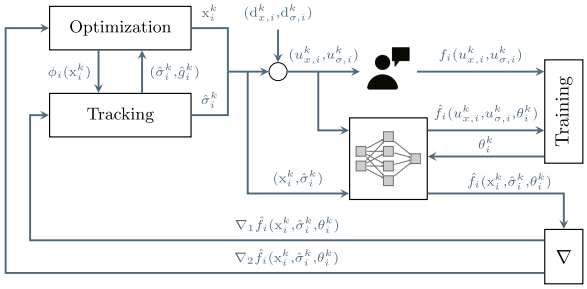

Fig. 2. Block diagram of the proposed distributed algorithm, where $\hat{\sigma}_i^k := \mathrm{w}_i^k + \phi_i(\mathrm{x}_i^k)$ and $\hat{g}_i^k := \mathrm{z}_i^k + \nabla_2 \hat{f}_i(\mathrm{x}_i^k, \hat{\sigma}_i^k, \theta_i)$.

Assumption 2: The function f_σ is μ_f-strongly convex and its gradient is L-Lipschitz continuous, for some $\mu_f, L > 0$. Further, for all $i \in \{1, \dots, N\}$, $\nabla_1 f_i$, $\nabla_2 f_i$, and ϕ_i are Lipschitz continuous with parameters $L_x, L_\sigma, L_\phi > 0$, respectively. ■

We remark that Assumption 2 implies the existence of a unique solution $x^\star \in \mathbb{R}^n$ to problem (1).

In this article, we want to develop an optimization algorithm to iteratively solve problem (1) in a distributed manner. Namely, we want to design an algorithm in which each agent only uses local information and exchanges data with its neighbors. Indeed, the considered N agents communicate according to a graph $\mathcal{G} = (\{1, \dots, N\}, \mathcal{E}, \mathcal{A})$, where $\{1, \dots, N\}$ is the set of agents, $\mathcal{E} \subseteq \{1, \dots, N\} \times \{1, \dots, N\}$ is the set of edges, and $\mathcal{A} \in \mathbb{R}^{N \times N}$ is the weighted adjacency matrix whose (i, j)-entry satisfies $a_{ij} > 0$ if $(j, i) \in \mathcal{E}$ and $a_{ij} = 0$, otherwise. The symbol $\mathcal{N}_i$ denotes the in-neighbor set of agent i, namely, $\mathcal{N}_i := \{j \in \{1, \dots, N\} \mid (j, i) \in \mathcal{E}\}$. The next assumption formalizes the class of graphs considered in this work.

Assumption 3: The graph $\mathcal{G}$ is strongly connected and the matrix $\mathcal{A}$ is doubly stochastic. ■

III. DELTA: Distributed Algorithm Design

In this section, we show the design of DELTA, i.e., a novel data-driven distributed method to iteratively address the problem formalized in Section II. DELTA exploits the concurrent action of a learning-oriented part tailored to estimating the unknown gradients of f_i, an optimization-oriented one aimed at solving (1), and a tracking one devoted to reconstructing the unavailable global quantities. The combination of these updates results in the DELTA algorithm, whose graphical description is provided in Fig. 2 and formal description is reported in Algorithm 1.

Although we analyze DELTA in a static and deterministic scenario, we point out that its design is particularly well suited for setups in which the cost functions vary over time and where the cost samples are possibly subjected to disturbances. Indeed,

Algorithm 1: DELTA (agent i).

Learning Update:

$$\theta_i^{k+1} = \theta_i^k - \gamma \nabla_3 \ell_i(\mathrm{x}_i^k + \mathrm{d}_{x,i}^k, \mathrm{w}_i^k + \phi_i(\mathrm{x}_i^k) + \mathrm{d}_{\sigma,i}^k, \theta_i^k)$$

Optimization Update:

$$\mathrm{x}_i^{k+1} = \mathrm{x}_i^k - \gamma \Big[\nabla_1 \hat{f}_i\left(\mathrm{x}_i^k, \mathrm{w}_i^k + \phi_i(\mathrm{x}_i^k), \theta_i^k\right) + \nabla \phi_i(\mathrm{x}_i^k) \left(\mathrm{z}_i^k + \nabla_2 \hat{f}_i\left(\mathrm{x}_i^k, \mathrm{w}_i^k + \phi_i(\mathrm{x}_i^k), \theta_i^k\right)\right)\Big]$$

Tracking Update:

$$\mathrm{w}_i^{k+1} = \sum_{j \in \mathcal{N}_i} a_{ij} \left(\mathrm{w}_j^k + \phi_j\left(\mathrm{x}_j^k\right)\right) - \phi_i\left(\mathrm{x}_i^k\right)$$

$$\mathrm{z}_i^{k+1} = \sum_{j \in \mathcal{N}_i} a_{ij} \left(\mathrm{z}_j^k + \nabla_2 \hat{f}_j\left(\mathrm{x}_j^k, \mathrm{w}_j^k + \phi_j\left(\mathrm{x}_j^k\right), \theta_j^k\right)\right) - \nabla_2 \hat{f}_i\left(\mathrm{x}_i^k, \mathrm{w}_i^k + \phi_i\left(\mathrm{x}_i^k\right), \theta_i^k\right)$$

the synergy between the learning, optimization, and tracking components of DELTA enables the algorithm to adapt to such changes without requiring a full restart of the neural network training, the optimization task, or the tracking process. In the following, we detail the three main components of DELTA.

A. Learning Update

We introduce in each agent i a local neural network to reconstruct f_i and the gradients $\nabla_1 f_i$ and $\nabla_2 f_i$ from data. Let $\hat{f}_i : \mathbb{R}^{n_i} \times \mathbb{R}^d \times \mathbb{R}^{m_i} \to \mathbb{R}$ be the estimate of f_i, where $m_i \in \mathbb{N}$ is the number of parameters of the neural network. In other words, for a generic parameter $\theta_i \in \mathbb{R}^{m_i}$, we consider

$$\hat{f}_i(x_i, s_i, \theta_i) \quad \text{proxy for} \quad f_i(x_i, s_i).$$

Once each network structure is fixed (i.e., the number of layers, the number of neurons, and the activation functions are chosen), the function $\hat{f}_i$ is analytically available. Thus, by considering differentiable functions $\hat{f}_i$, we also introduce their gradients

$$\nabla_1 \hat{f}_i(x_i, s_i, \theta_i) \quad \text{proxy for} \quad \nabla_1 f_i(x_i, s_i) \tag{3a}$$

$$\nabla_2 \hat{f}_i(x_i, s_i, \theta_i) \quad \text{proxy for} \quad \nabla_2 f_i(x_i, s_i). \tag{3b}$$

We remark that these derivatives are typically available in most of the deep learning packages via automatic differentiation. The next assumption characterizes the estimates $\nabla_1 \hat{f}_i$ and $\nabla_2 \hat{f}_i$.

Assumption 4: For all $i \in \{1, \dots, N\}$, the function $\hat{f}_i$ is differentiable and its gradients $\nabla_1 \hat{f}_i$ and $\nabla_2 \hat{f}_i$ are Lipschitz continuous with parameters $\hat{L}_x, \hat{L}_\sigma > 0$, respectively. ■

Remark 1: The requirement in Assumption 4 is satisfied by choosing differentiable activation functions with Lipschitz-continuous gradients, such as Tanh, Sigmoid, or Softplus. ■

To iteratively train each neural network using new sampled data, coherently with Assumption 1, we exploit feedback $f_i(\mathrm{x}_i^k + \mathrm{d}_{x,i}^k, \mathrm{w}_i^k + \phi(\mathrm{x}_i^k) + \mathrm{d}_{\sigma,i}^k)$, with $\mathrm{d}_{x,i}^k \in \mathbb{R}^{n_i}$ and $\mathrm{d}_{\sigma,i}^k \in \mathbb{R}^d$. As we will formalize in the next, $\mathrm{d}_{x,i}^k$ and $\mathrm{d}_{\sigma,i}^k$ are two dither signals that we add to force persistency-of-excitation-like properties in the learning process. Formally, given $k \in \mathbb{N}$ and a generic pair $(\mathrm{x}_i, \mathrm{w}_i) \in \mathbb{R}^{n_i} \times \mathbb{R}^d$, we consider the learning

problem

$$\min_{\theta_i\in\mathbb{R}^{m_i}} \frac{1}{k}\sum_{\tau=0}^{k}\ell_i(\mathrm{x}_i+\mathrm{d}_{x,i}^{\tau},\mathrm{w}_i+\mathrm{d}_{\sigma,i}^{\tau},\theta_i), \tag{4}$$

where each $\ell_i:\mathbb{R}^{n_i}\times\mathbb{R}^d\times\mathbb{R}^{m_i}\to\mathbb{R}$ is the loss function computed using a sample of the observed cost. A popular (but not unique) choice for ℓ_i is the squared loss, namely

$$\ell_i(u_{x,i},u_{\sigma,i},\theta_i):=\frac{1}{2}\left\|f_i(u_{x,i},u_{\sigma,i})-\hat{f}_i(u_{x,i},u_{\sigma,i},\theta_i)\right\|^2. \tag{5}$$

The next assumption characterizes the dithers' sequences $\{\mathrm{d}_{x,i}^{\tau},\mathrm{d}_{\sigma,i}^{\tau}\}_{\tau\in\mathbb{N}}$ and problem (4).

Assumption 5: There exist $\mathcal{K}\in\mathbb{N}\setminus\{0\}$, $\mu_\ell>0$, $\theta_i^\star\in\mathbb{R}^{m_i}$, and $\bar{r}_i>0$ such that, for all $i\in\{1,\dots,N\}$, the sequence $\{\mathrm{d}_{x,i}^{\tau},\mathrm{d}_{\sigma,i}^{\tau}\}_{\tau\in\mathbb{N}}$ is $\mathcal{K}$-periodic and, for all $(\mathrm{x}_i,\mathrm{w}_i)\in\mathbb{R}^{n_i}\times\mathbb{R}^d$, problem (4) with $k=\mathcal{K}$ is μ_ℓ-strongly convex in $\mathcal{B}_{\bar{r}_i}(\theta_i^\star)$ and $\theta_i^\star$ is its local minimum in $\mathcal{B}_{\bar{r}_i}(\theta_i^\star)$. Further, for all $i\in\{1,\dots,N\}$, $\nabla_3\ell_i(u_{x,i},u_{\sigma,i},\theta_i)$ is L_ℓ-Lipschitz continuous for all $(u_{x,i},u_{\sigma,i})\in\mathbb{R}^{n_i}\times\mathbb{R}^d$ and some $L_\ell>0$. ■

Remark 2: In the ideal case in which $(\mathrm{x}_i,\mathrm{w}_i)$ is fixed and the $\mathcal{K}$ samples taken in its neighborhood are all concurrently available, Assumption 5 ensures 1) local strong convexity of each learning problem (4) into $\mathcal{B}_{\bar{r}_i}(\theta_i^\star)$ and 2) that $\theta_i^\star$ is the unique minimizer of its problem (4) into $\mathcal{B}_{\bar{r}_i}(\theta_i^\star)$. ■

Once the local minimizers $\theta_i^\star$ have been introduced, we characterize as follows their reconstruction capabilities.

Assumption 6: For all $i\in\{1,\dots,N\}$, there exist Lipschitz-continuous functions $\mathrm{p}_{1,i}:\mathbb{R}^n\to\mathbb{R}^{n_i}$, $\mathrm{p}_{2,i}:\mathbb{R}^n\to\mathbb{R}^d$, and $\mathrm{p}_{\ell,i}:\mathbb{R}^n\times\mathbb{N}\to\mathbb{R}^{m_i}$ such that, for all $\mathrm{x}\in\mathbb{R}^n$ and $k\in\{1,\dots,\mathcal{K}\}$, it holds

$$\nabla_1\hat{f}_i(\mathrm{x}_i,\sigma(\mathrm{x}),\theta_i^\star)=\nabla_1 f_i(\mathrm{x}_i,\sigma(\mathrm{x}))+\mathrm{p}_{1,i}(\mathrm{x}) \tag{6a}$$

$$\nabla_2\hat{f}_i(\mathrm{x}_i,\sigma(\mathrm{x}),\theta_i^\star)=\nabla_2 f_i(\mathrm{x}_i,\sigma(\mathrm{x}))+\mathrm{p}_{2,i}(\mathrm{x}) \tag{6b}$$

$$\nabla_3\ell_i(\mathrm{x}_i+\mathrm{d}_{x,i}^k,\sigma(\mathrm{x})+\mathrm{d}_{\sigma,i}^k,\theta_i^\star)=\mathrm{p}_{\ell,i}(\mathrm{x},k). \tag{6c}$$

Moreover, there exists $\epsilon>0$ such that

$$\|\mathrm{p}_{1,i}(\mathrm{x})\|\le\epsilon,\quad \|\mathrm{p}_{2,i}(\mathrm{x})\|\le\epsilon,\quad \|\mathrm{p}_{\ell,i}(\mathrm{x},k)\|\le\epsilon, \tag{7}$$

for all $\mathrm{x}\in\mathbb{R}^n$, $k\in\{1,\dots,\mathcal{K}\}$, and $i\in\{1,\dots,N\}$. ■

Remark 3: Assumption 6 specifies each neural network's ability to approximate the functions f_i and their gradients. This capability is captured by the parameter ϵ, the maximum reconstruction error at convergence of the training process (i.e., when $\theta_i=\theta_i^\star$). Its value depends on the network design and the unknown f_i, but it is not required by the algorithm. ■

Once the capabilities of the neural network have been enforced, we can formalize the learning mechanism embedded in DELTA. An immediate but computationally expensive approach would be to apply the gradient descent method to problem (4), iteratively updating an estimate $\theta_i^k\in\mathbb{R}^{m_i}$ as

$$\theta_i^{k+1}=\theta_i^k-\gamma\sum_{\tau=0}^{k}\nabla_3\ell_i\left(\mathrm{x}_i^k+\mathrm{d}_{x,i}^{\tau},\mathrm{w}_i^k+\phi_i(\mathrm{x}_i^k)+\mathrm{d}_{\sigma,i}^{\tau},\theta_i^k\right). \tag{8}$$

However, the strategy in (8) would require storing the entire cost samples history and, more importantly, computing an increasing number k of gradients in the current estimate θ_i^k. To replace the computationally prohibitive method (8), we adopt a data-driven update based on a single sample per k, namely

$$\theta_i^{k+1}=\theta_i^k-\gamma\nabla_3\ell_i(\mathrm{x}_i^k+\mathrm{d}_{x,i}^k,\mathrm{w}_i^k+\phi_i(\mathrm{x}_i^k)+\mathrm{d}_{\sigma,i}^k,\theta_i^k). \tag{9}$$

Indeed, with an eye to the structure formalized in (5), we underline that (9) can be implemented by only relying on a single feedback $f_i(\mathrm{x}_i^k+\mathrm{d}_{x,i}^k,\mathrm{w}_i^k+\phi_i(\mathrm{x}_i^k)+\mathrm{d}_{\sigma,i}^k)$ in a neighborhood of $(\mathrm{x}_i^k,\mathrm{w}_i^k+\phi_i(\mathrm{x}_i^k))$, i.e., of the local estimate of agent i about the i-th solution block $x_i^\star$ and $\sigma(\mathrm{x}^k)$, respectively. Finally, we compactly represent the global cost addressed by the neural networks via $\ell:\mathbb{R}^n\times\mathbb{R}^{Nd}\times\mathbb{R}^m\to\mathbb{R}$ defined as

$$\ell(x,s,\theta):=\sum_{i=1}^{N}\ell_i(x_i,s_i,\theta_i), \tag{10}$$

where $s:=\mathrm{col}(s_1,\dots,s_N)$ and $\theta:=\mathrm{col}(\theta_1,\dots,\theta_N)$. With the notation of Assumption 6 at hand, we also define $m:=\sum_{i=1}^N m_i$, $\Theta:=\prod_{i=1}^N\mathcal{B}_{\bar{r}_i}(\theta_i^\star)\subseteq\mathbb{R}^m$, and $\theta^\star:=\mathrm{col}(\theta_1^\star,\dots,\theta_N^\star)\in\mathbb{R}^m$.

B. Optimization Update

In a full information setup, problem (1) could be solved with a parallel implementation of the gradient method. Namely, by denoting with $\mathrm{x}_i^k\in\mathbb{R}^{n_i}$ the estimate of agent i at iteration k about the i-th block $x_i^\star\in\mathbb{R}^{n_i}$ of $x^\star$, each agent would run

$$\mathrm{x}_i^{k+1}=\mathrm{x}_i^k-\gamma\nabla_1 f_i\left(\mathrm{x}_i^k,\sigma(\mathrm{x}^k)\right)-\gamma\frac{\nabla\phi_i(\mathrm{x}_i^k)}{N}\sum_{j=1}^{N}\nabla_2 f_j\left(\mathrm{x}_j^k,\sigma(\mathrm{x}^k)\right) \tag{11}$$

where $\gamma>0$ is the step size. Nevertheless, update (11) requires the gradients of the unknown f_i. We therefore modify it using the approximations (3), namely

$$\mathrm{x}_i^{k+1}=\mathrm{x}_i^k-\gamma\nabla_1\hat{f}_i\left(\mathrm{x}_i^k,\sigma(\mathrm{x}^k),\theta_i^k\right)-\gamma\frac{\nabla\phi_i(\mathrm{x}_i^k)}{N}\sum_{j=1}^{N}\nabla_2\hat{f}_j\left(\mathrm{x}_j^k,\sigma(\mathrm{x}^k),\theta_j^k\right). \tag{12}$$

In addition, the unavailable global quantities $\sigma(\mathrm{x}^k)$ and $\sum_{j=1}^N\nabla_2\hat{f}_j(\mathrm{x}_j^k,\sigma(\mathrm{x}^k),\theta_j^k)/N$ are needed in (12). We thus interlace (12) with a tracking-oriented mechanism.

C. Tracking Update

Being $\sigma(\mathrm{x}^k)$ and $\sum_{j=1}^N\nabla_2\hat{f}_j(\mathrm{x}_j^k,\sigma(\mathrm{x}^k),\theta_j^k)/N$ unavailable, we introduce two auxiliary variables $\mathrm{w}_i^k,\mathrm{z}_i^k\in\mathbb{R}^d$ for all $i\in\{1,\dots,N\}$ to estimate them via $\mathrm{w}_i^k+\phi_i(\mathrm{x}_i^k)$ and $\nabla_2\hat{f}_i(\mathrm{x}_i^k,\mathrm{w}_i^k+\phi_i(\mathrm{x}_i^k),\theta_i^k)+\mathrm{z}_i^k$, respectively. To this end, we update them according to the perturbed consensus dynamics

$$\mathrm{w}_i^{k+1}=\sum_{j\in\mathcal{N}_i}a_{ij}\left(\mathrm{w}_j^k+\phi_j(\mathrm{x}_j^k)\right)-\phi_i(\mathrm{x}_i^k) \tag{13a}$$

$$\mathrm{z}_i^{k+1}=\sum_{j\in\mathcal{N}_i}a_{ij}\left(\mathrm{z}_j^k+\nabla_2\hat{f}_j(\mathrm{x}_j^k,\mathrm{w}_j^k+\phi_j(\mathrm{x}_j^k),\theta_j^k)\right)-\nabla_2\hat{f}_i(\mathrm{x}_i^k,\mathrm{w}_i^k+\phi_i(\mathrm{x}_i^k),\theta_i^k), \tag{13b}$$

where we recall that the weights a_{ij} are the entries of the weighted adjacency matrix $\mathcal{A}$ matching the graph $\mathcal{G}$. We remark that the update (13) is fully distributed since it requires only local information and neighboring communication.

D. DELTA: Convergence Properties

To shorten the notation, consider $\theta^k := \mathrm{col}(\theta_1^k, \dots, \theta_N^k)$, $\mathrm{x}^k := \mathrm{col}(\mathrm{x}_1^k, \dots, \mathrm{x}_N^k)$, $\mathrm{w}^k := \mathrm{col}(\mathrm{w}_1^k, \dots, \mathrm{w}_N^k)$, $\mathrm{z}^k := \mathrm{col}(\mathrm{z}_1^k, \dots, \mathrm{z}_N^k)$, and $\phi(\mathrm{x}^k) := \mathrm{col}(\phi(\mathrm{x}_1^k), \dots, \phi(\mathrm{x}_N^k))$, and for all $c \in \{1, 2\}$, we define

$$\hat{G}_c(x, w, \theta) := \begin{bmatrix} \nabla_c \hat{f}_1(x_1, w_1, \theta_1) \\ \vdots \\ \nabla_c \hat{f}_N(x_N, w_N, \theta_N) \end{bmatrix}. \tag{14}$$

Let the function $\Xi : \mathbb{R}^n \times \mathbb{R}^{Nd} \times \mathbb{R}^{Nd} \times \mathbb{R}^m \to \mathbb{R}^{n+2Nd+m}$ be

$$\Xi(\mathrm{x}, \mathrm{w}, \mathrm{z}, \theta) := \begin{bmatrix} \mathrm{x} - x^\star \\ \mathrm{w} + \phi(\mathrm{x}) - \mathbf{1}\sigma(\mathrm{x}) \\ \mathrm{z} + (I - \frac{\mathbf{1}\mathbf{1}^\top}{N})\hat{G}_2(\mathrm{x}, \mathrm{w} + \phi(\mathrm{x}), \theta) \\ \theta - \theta^\star \end{bmatrix}.$$

Finally, for all $i \in \{1, \dots, N\}$, consider the set

$$\mathcal{S}_i := \mathcal{B}_{\bar{r}_i}(\theta_i^\star) \times \mathbb{R}^{n_i} \times \{0_d\} \times \{0_d\}.$$

The next theorem states the convergence properties of DELTA.

Theorem 1: Consider DELTA and let Assumptions 1–6 hold. Further, assume that $(\theta_i^0, \mathrm{x}_i^0, \mathrm{w}_i^0, \mathrm{z}_i^0) \in \mathcal{S}_i$ for all $i \in \{1, \dots, N\}$. Then, there exist $\bar{\gamma}, B, \kappa_1 > 0$ and $\kappa_2 \in (0, 1)$ such that, for all $\gamma \in (0, \bar{\gamma})$, it holds

$$\left\|\mathrm{x}^k - x^\star\right\| \le \kappa_1 \kappa_2^k \left\|\Xi(\mathrm{x}^0, \mathrm{w}^0, \mathrm{z}^0, \theta^0)\right\| + \epsilon B,$$

for all $k \in \mathbb{N}$. ■

The proof of Theorem 1 is provided in Section IV. It is based on system theory tools aimed at finding a uniform ultimate bound for the trajectories of the dynamical system describing DELTA. To this end, the proof relies on discrete-time averaging theory (see Appendix A) and timescale separation.

Besides the term B that depends on many problem parameters (e.g., the network connectivity, the Lipschitz continuity constants), we note that the bound in Theorem 1 directly depends on the accuracy capabilities of the neural networks (cf., Assumption 5) through the parameter ϵ. In particular, in the case of perfect estimation (i.e., if $\epsilon = 0$, see Assumption 5), Theorem 1 ensures exact and linear convergence of DELTA toward the optimal solution $x^\star$.

IV. DELTA: Stability Analysis

In this section, we carry out the stability analysis of DELTA to prove Theorem 1. Our analysis relies on the following five main steps.

1) In Section IV-A, we reformulate DELTA and consider a "nominal" version in which the "finite accuracy" of the neural networks (see Assumption 6) is compensated. We interpret this nominal scheme as a time-varying two-timescales system, i.e., as the interconnection of two subsystems identified, respectively, as *slow* and *fast* dynamics. This allows us to study the stability properties of the interconnected system by analyzing two auxiliary dynamics called boundary-layer and reduced systems related, respectively, to the fast and slow dynamics.
2) In Section IV-B, we show that the origin is an exponentially stable equilibrium of the boundary layer system.
3) In Section IV-C, we carry out the reduced system analysis in detail and show exponential stability of the origin via *averaging theory* (see Theorem 2 in Appendix A).
4) In Section IV-D, we exploit the results of steps 2) and 3) to conclude the analysis of the nominal system;
5) Finally, in Section IV-E, we return to the original system, interpreting it as a perturbed version of the nominal one. This interpretation allows us to use the stability properties of the nominal system (step 4) to establish those of the original one, thereby providing the proof of Theorem 1.

Assumptions 1–6 hold true throughout the entire section.

Remark 4: Although the stability analysis of DELTA is carried out in a static and deterministic environment, it paves the way for extensions to dynamic and uncertain environments subject to, e.g., variations of f_i over time and noisy cost samples. ■

A. DELTA as a Time-Varying Two-Time-Scale System

By collecting all the local updates in Algorithm 1, we can compactly rewrite DELTA as

$$\theta^{k+1} = \theta^k - \gamma \nabla_3 \ell(\mathrm{x}^k + \mathrm{d}_x^k, \mathrm{w}^k + \phi(\mathrm{x}^k) + \mathrm{d}_\sigma^k, \theta^k) \tag{15a}$$

$$\mathrm{x}^{k+1} = \mathrm{x}^k - \gamma \Big[\hat{G}_1(\mathrm{x}^k, \mathrm{w}^k + \phi(\mathrm{x}^k), \theta^k) + \nabla\phi\left(\mathrm{x}^k\right)\Big(\hat{G}_2(\mathrm{x}^k, \mathrm{w}^k + \phi(\mathrm{x}^k), \theta^k) + \mathrm{z}^k\Big)\Big] \tag{15b}$$

$$\mathrm{w}^{k+1} = A\mathrm{w}^k + (A - I_{Nd})\,\phi(\mathrm{x}^k) \tag{15c}$$

$$\mathrm{z}^{k+1} = A\mathrm{z}^k + (A - I_{Nd})\,\hat{G}_2(\mathrm{x}^k, \mathrm{w}^k + \phi(\mathrm{x}^k), \theta^k), \tag{15d}$$

where $\mathrm{d}_x^k := \mathrm{col}(\mathrm{d}_{x,1}^k, \dots, \mathrm{d}_{x,N}^k)$, $\mathrm{d}_\sigma^k := \mathrm{col}(\mathrm{d}_{\sigma,1}^k, \dots, \mathrm{d}_{\sigma,N}^k)$, and $A := \mathcal{A} \otimes I_d$. It is worth noting that the subspace

$$\mathcal{S} := \{(\theta, \mathrm{x}, \mathrm{w}, \mathrm{z}) \in \mathbb{R}^m \times \mathbb{R}^n \times \mathbb{R}^{Nd} \times \mathbb{R}^{Nd} \mid \mathbf{1}^\top \mathrm{w} = 0, \mathbf{1}^\top \mathrm{z} = 0\},$$

is invariant for system (15). Thus, we introduce a change of coordinate to isolate the invariant part of the state and define the error coordinates relative to $\theta^\star$ and $x^\star$. Formally, it is

$$\begin{bmatrix} \theta \\ \mathrm{x} \\ \mathrm{w} \\ \mathrm{z} \end{bmatrix} \mapsto \begin{bmatrix} \tilde{\theta} \\ \tilde{\mathrm{x}} \\ \bar{\mathrm{w}} \\ \mathrm{w}_\perp \\ \bar{\mathrm{z}} \\ \mathrm{z}_\perp \end{bmatrix} := \begin{bmatrix} \theta - \theta^\star \\ \mathrm{x} - x^\star \\ \frac{\mathbf{1}^\top}{N}\mathrm{w} \\ R^\top \mathrm{w} \\ \frac{\mathbf{1}^\top}{N}\mathrm{z} \\ R^\top \mathrm{z} \end{bmatrix}, \tag{16}$$

with $R \in \mathbb{R}^{Nd \times (N-1)d}$ such that $R^\top \mathbf{1} = 0$ and $R^\top R = I_{Nd}$. By using the definitions of $\bar{\mathrm{w}}^k$ and $\bar{\mathrm{z}}^k$ in (16) and since $\mathcal{A}$ is doubly stochastic (cf., Assumption 3), it holds

$$\bar{\mathrm{w}}^{k+1} = \bar{\mathrm{w}}^k, \quad \bar{\mathrm{z}}^{k+1} = \bar{\mathrm{z}}^k,$$

which implies $(\bar{\mathrm{w}}^k, \bar{\mathrm{z}}^k) = (\bar{\mathrm{w}}^0, \bar{\mathrm{z}}^0)$ for all $k \in \mathbb{N}$. Further, the initialization $\mathrm{w}_i^k = \mathrm{z}_i^k = 0$ leads to $(\bar{\mathrm{w}}^0, \bar{\mathrm{z}}^0) = (0, 0)$, and thus, it holds $(\bar{\mathrm{w}}^k, \bar{\mathrm{z}}^k) = (0, 0)$ for all $k \in \mathbb{N}$. Hence, we rewrite (15) in the new coordinates (16) and neglect the trivial dynamics of $\bar{\mathrm{w}}^k$ and $\bar{\mathrm{z}}^k$, thus obtaining an equivalent restricted system

$$\tilde{\theta}^{k+1} = \tilde{\theta}^k - \gamma \nabla_3 \tilde{\ell}\left(\tilde{\mathrm{x}}^k, \mathrm{w}_\perp^k, \tilde{\theta}^k, k\right) \tag{17a}$$

$$\tilde{\mathrm{x}}^{k+1} = \tilde{\mathrm{x}}^k - \gamma \Big[\hat{G}_1(\tilde{\mathrm{x}}^k + x^\star, R\mathrm{w}_\perp^k + \phi(\tilde{\mathrm{x}}^k + x^\star), \tilde{\theta}^k + \theta^\star) + \nabla\phi(\tilde{\mathrm{x}}^k + x^\star)\hat{G}_2(\tilde{\mathrm{x}}^k + x^\star, R\mathrm{w}_\perp^k + \phi(\tilde{\mathrm{x}}^k + x^\star), \tilde{\theta}^k + \theta^\star) + \nabla\phi(\tilde{\mathrm{x}}^k + x^\star)R\mathrm{z}_\perp^k\Big] \tag{17b}$$

$$\mathrm{w}_\perp^{k+1} = R^\top A R \mathrm{w}_\perp^k + M\phi(\tilde{\mathrm{x}}^k + x^\star) \tag{17c}$$

$$z_\perp^{k+1} = R^\top A R z_\perp^k + M\hat{G}_2(\tilde{x}^k + x^\star, w_\perp^k + \phi(\tilde{x}^k + x^\star), \tilde{\theta}^k + \theta^\star), \tag{17d}$$

where $M := R^\top(A - I_{Nd})$ and we introduce the compact notation $\tilde{\ell} : \mathbb{R}^n \times \mathbb{R}^{(N-1)d} \times \mathbb{R}^m \times \mathbb{N} \to \mathbb{R}$, defined as

$$\tilde{\ell}(\tilde{x}, w_\perp, \tilde{\theta}, k) := \ell\left(\tilde{x} + x^\star + d_x^k, Rw_\perp + \phi(x) + d_\sigma^k, \tilde{\theta} + \theta^\star\right).$$

Now, we introduce the "nominal" version of (17), i.e., an auxiliary system in which the finite accuracy of the neural networks (see Assumption 6) is suitably compensated by adding some fictitious terms. Such a nominal system reads as

$$\tilde{\theta}^{k+1} = \tilde{\theta}^k - \gamma\left[\nabla_3\tilde{\ell}\left(\tilde{x}^k, w_\perp^k, \tilde{\theta}^k, k\right) - p_\ell\left(\tilde{x}^k + x^\star, k\right)\right] \tag{18a}$$

$$\begin{aligned}\tilde{x}^{k+1} = \tilde{x}^k - \gamma\Big[&\hat{G}_1\left(\tilde{x}^k + x^\star, Rw_\perp^k + \phi\left(\tilde{x}^k + x^\star\right), \tilde{\theta}^k + \theta^\star\right)\\ &+ \nabla\phi\left(\tilde{x}^k + x^\star\right)\hat{G}_2\left(\tilde{x}^k + x^\star, Rw_\perp^k + \phi\left(\tilde{x}^k + x^\star\right), \tilde{\theta}^k + \theta^\star\right)\\ &+ \nabla\phi\left(\tilde{x}^k + x^\star\right)Rz_\perp^k\\ &- p_1\left(\tilde{x}^k + x^\star\right) - \nabla\phi\left(\tilde{x}^k + x^\star\right)\frac{\mathbf{1}\mathbf{1}^\top}{N}p_2\left(\tilde{x}^k + x^\star\right)\Big]\end{aligned} \tag{18b}$$

$$w_\perp^{k+1} = R^\top A R w_\perp^k + M\phi\left(\tilde{x}^k + x^\star\right) \tag{18c}$$

$$z_\perp^{k+1} = R^\top A R z_\perp^k + M\hat{G}_2\left(\tilde{x}^k + x^\star, w_\perp^k + \phi(\tilde{x}^k + x^\star), \tilde{\theta}^k + \theta^\star\right), \tag{18d}$$

where $p_c(x) := \mathrm{col}(p_{c,1}(x), \dots, p_{c,N}(x))$ for all $c \in \{1, 2\}$, while $p_\ell(x, k) := \mathrm{col}(p_{\ell,1}(x, k), \dots, p_{\ell,N}(x, k))$. Due to p_1, p_2, and p_ℓ, $(0, 0, -R^\top\phi(x^\star), -R^\top\hat{G}_2(x^\star, \mathbf{1}\sigma(x^\star), \theta^\star))$ is an equilibrium of (18) (cf., Assumption 6). In detail, the equilibrium points of the subsystems (18c) and (18d) are parameterized in the states (θ, x) of (18a) and (18b) through $h : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^{2(N-1)d}$ defined as

$$h(\theta, x) := \begin{bmatrix} -R^\top\phi(x) \\ -R^\top\hat{G}_2(x, \mathbf{1}\sigma(x), \theta)\end{bmatrix}. \tag{19}$$

Further, the variations of (18a) and (18b) can be arbitrarily reduced via γ. From these observations, we interpret system (18) as a time-varying two-time-scale system, where (18a) and (18b) are the so-called slow dynamics, while (18c) and (18d) are the so-called fast parts. According to this view, we now proceed by analyzing the boundary-layer and reduced system associated with (18).

B. Boundary-Layer System Analysis

In this section, we analyze the boundary-layer system associated with (18), obtained by fixing an arbitrary $(\tilde{\theta}, \tilde{x}) \in \mathbb{R}^m \times \mathbb{R}^n$ in (18c) and (18d), written in the error coordinates

$$\begin{bmatrix}\tilde{w}_\perp^k \\ \tilde{z}_\perp^k\end{bmatrix} := \begin{bmatrix} w_\perp^k \\ z_\perp^k\end{bmatrix} - h\left(\tilde{\theta} + \theta^\star, \tilde{x} + x^\star\right).$$

In light of the stochasticity property of $\mathcal{A}$ (cf., Assumption 3) and the definition of h [cf., (19)], we use the properties $R^\top\mathbf{1} = 0$ and $RR^\top = I_{Nd} - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$ to write this system as

$$\tilde{w}_\perp^{k+1} = R^\top A R\tilde{w}_\perp^k \tag{20a}$$

$$\tilde{z}_\perp^{k+1} = R^\top A R\tilde{z}_\perp^k + M\Delta\hat{G}_2\left(\tilde{x} + x^\star, \tilde{w}_\perp^k, \tilde{\theta} + \theta^\star\right), \tag{20b}$$

where $\Delta\hat{G}_2 : \mathbb{R}^n \times \mathbb{R}^{(N-1)d} \times \mathbb{R}^m \to \mathbb{R}$ is defined as

$$\begin{aligned}\Delta\hat{G}_2(\tilde{x} + x^\star, \tilde{w}_\perp, \tilde{\theta} + \theta^\star) := &\hat{G}_2(\tilde{x} + x^\star, R\tilde{w}_\perp + \mathbf{1}\sigma(\tilde{x} + x^\star), \tilde{\theta} + \theta^\star)\\ &- \hat{G}_2(\tilde{x} + x^\star, \mathbf{1}\sigma(\tilde{x} + x^\star), \tilde{\theta} + \theta^\star).\end{aligned} \tag{21}$$

Let us introduce $\zeta := \mathrm{col}(\zeta_1, \zeta_2) := \mathrm{col}(\tilde{w}_\perp, \tilde{z}_\perp)$, $\tilde{A} := \mathrm{blkdiag}(R^\top A R, R^\top A R)$, and $g_{\text{B}} : \mathbb{R}^n \times \mathbb{R}^{2(N-1)d} \times \mathbb{R}^m \to \mathbb{R}^{2(N-1)d}$ defined as

$$g_{\text{B}}\left(\tilde{x}, \zeta, \tilde{\theta}\right) := \begin{bmatrix} 0 \\ R^\top A\Delta\hat{G}_2(\tilde{x} + x^\star, \zeta_1, \tilde{\theta} + \theta^\star)\end{bmatrix}, \tag{22}$$

which allow us to compactly rewrite (20) as

$$\zeta^{k+1} = \tilde{A}\zeta^k + g_{\text{B}}(\tilde{x}, \zeta^k, \tilde{\theta}). \tag{23}$$

With this notation at hand, we are ready to formalize the stability properties of the boundary-layer system (23).

Lemma 1: There exist a Lyapunov function $U : \mathbb{R}^{2(N-1)d} \to \mathbb{R}$ and $b_1, b_2, b_3, b_4 > 0$ such that

$$b_1\|\zeta\|^2 \le U(\zeta) \le b_2\|\zeta\|^2 \tag{24a}$$

$$U\left(\tilde{A}\zeta + g_{\text{B}}\left(\tilde{x}, \zeta, \tilde{\theta}\right)\right) - U(\zeta) \le -b_3\|\zeta\|^2 \tag{24b}$$

$$\left|U(\zeta) - U\left(\bar{\zeta}\right)\right| \le b_4\left\|\zeta - \bar{\zeta}\right\|\left(\|\zeta\| + \left\|\bar{\zeta}\right\|\right), \tag{24c}$$

for any $\zeta, \bar{\zeta} \in \mathbb{R}^{2(N-1)d}$ and $(\tilde{\theta}, \tilde{x}) \in \mathbb{R}^m \times \mathbb{R}^n$. ■

The proof of Lemma 1 is reported in Appendix B.

C. Reduced System Analysis

In this section, we analyze the reduced system associated with (18), i.e., the system obtained by plugging $\mathrm{col}(w_\perp^k, z_\perp^k) = h(\tilde{\theta}^k + \theta^\star, \tilde{x}^k + x^\star)$ for all $k \in \mathbb{N}$ into (18a) and (18b), namely

$$\tilde{\theta}^{k+1} = \tilde{\theta}^k - \gamma(\nabla_3\tilde{\ell}(\tilde{x}^k, \mathbf{1}\sigma\left(\tilde{x}^k + x^\star\right), \tilde{\theta}^k, k) - p_\ell(k)) \tag{25a}$$

$$\tilde{x}^{k+1} = \tilde{x}^k - \gamma\hat{G}^d(\tilde{x}^k, \tilde{\theta}^k), \tag{25b}$$

in which we introduce $\hat{G}^d : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ defined as

$$\begin{aligned}\hat{G}^d(\tilde{x}, \tilde{\theta}) := &\hat{G}_1(\tilde{x} + x^\star, \mathbf{1}\sigma(\tilde{x} + x^\star), \tilde{\theta} + \theta^\star) - p_1(\tilde{x} + x^\star)\\ &+ \nabla\phi(\tilde{x} + x^\star)\frac{\mathbf{1}\mathbf{1}^\top}{N}\hat{G}_2(\tilde{x} + x^\star, \mathbf{1}\sigma(\tilde{x} + x^\star), \tilde{\theta} + \theta^\star)\\ &- \nabla\phi\left(\tilde{x} + x^\star\right)\frac{\mathbf{1}\mathbf{1}^\top}{N}p_2(\tilde{x} + x^\star).\end{aligned} \tag{26}$$

Using the accuracy characterization (6c) of $\theta^\star$ and the optimality of $x^\star$ for (1), we point out that the origin is an equilibrium of (25). We then study its stability via the corresponding averaged system, obtained by averaging the time-varying vector field of (25) over an infinite time horizon (see Appendix A). To this end, let $\tilde{\ell}_{\mathcal{K}} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ be

$$\tilde{\ell}_{\mathcal{K}}(\tilde{x}, \tilde{\theta}) := \frac{1}{\mathcal{K}}\sum_{k=1}^{\mathcal{K}}\ell(\tilde{x} + x^\star + d_x^k, \mathbf{1}\sigma(\tilde{x} + x^\star) + d_\sigma^k, \tilde{\theta} + \theta^\star).$$

The periodicity of d_x^k and d_σ^k (cf., Assumption 6) ensures the existence of the averaged system associated with (25a), namely

$$\tilde{\theta}_{\text{AV}}^{k+1} = \tilde{\theta}_{\text{AV}}^{k} - \gamma \nabla_2 \tilde{\ell}_{\mathcal{K}}(\tilde{\mathrm{x}}_{\text{AV}}, \tilde{\theta}_{\text{AV}}^{k}) + \frac{\gamma}{\mathcal{K}} \sum_{k=1}^{\mathcal{K}} \mathrm{p}_\ell(\tilde{\mathrm{x}}_{\text{AV}} + x^\star, k) \quad (27a)$$

$$\tilde{\mathrm{x}}_{\text{AV}}^{k+1} = \tilde{\mathrm{x}}_{\text{AV}}^{k} - \gamma \hat{G}^d(\tilde{\mathrm{x}}_{\text{AV}}^{k}, \tilde{\theta}_{\text{AV}}^{k}). \quad (27b)$$

From Assumption 5, $\nabla_2 \tilde{\ell}_{\mathcal{K}}(\cdot, \tilde{\theta}_{\text{AV}})$ is the gradient of a μ_ℓ-strongly convex function for all $(\tilde{\theta}_{\text{AV}} + \theta^\star) \in \Theta$. We thus obtain the following stability properties of the averaged system (27).

Lemma 2: There exists $\bar{\gamma}_2 > 0$ such that, for all $\gamma \in (0, \bar{\gamma}_2)$, the origin is an exponentially stable equilibrium of (27) with domain of attraction $\{(\tilde{\theta}_{\text{AV}}, \tilde{\mathrm{x}}_{\text{AV}}) \in \mathbb{R}^m \times \mathbb{R}^n \mid (\tilde{\theta}_{\text{AV}} + \theta^\star) \in \Theta\}$.

The proof of Lemma 2 is reported in Appendix C.

Let us introduce $\chi := \mathrm{col}(\chi_1, \chi_2) := \mathrm{col}(\tilde{\theta}, \tilde{\mathrm{x}})$ and $g_{\text{R}} : \mathbb{R}^{m+n} \times \mathbb{N} \to \mathbb{R}^{m+n}$ defined as

$$g_{\text{R}}(\chi, k) := \begin{bmatrix} g_{\text{R},1}(\chi_1, \chi_2, k) \\ g_{\text{R},2}(\chi_1, \chi_2) \end{bmatrix} := \begin{bmatrix} \nabla_3 \tilde{\ell}(\chi_2, \mathbf{1}\sigma(\chi_2 + x^\star), \chi_1, k) - \mathrm{p}_\ell(\xi_2 + x^\star, k) \\ \hat{G}^d(\chi_2, \chi_1) \end{bmatrix}.$$

Then, we can compactly rewrite system (25) as

$$\chi^{k+1} = \chi^k - \gamma g_{\text{R}}(\chi^k, k). \quad (28)$$

Let $\mathcal{X} := \{\chi \in \mathbb{R}^{m+n} \mid \chi_1 + \theta^\star \in \Theta\}$. The next lemma ensures exponential stability properties of the origin for (28).

Lemma 3: There exist a Lyapunov function $W : \mathbb{R}^{m+n} \times N \to \mathbb{R}$ and $\bar{\gamma}_1 > 0$ such that, for all $\gamma \in (0, \bar{\gamma}_1)$, it holds

$$c_1 \|\chi\|^2 \le W(\chi, k) \le c_2 \|\chi\|^2 \quad (29a)$$

$$W(\chi - \gamma g_{\text{R}}(\chi, k), k+1) - W(\chi, k) \le -\gamma c_3 \|\chi\|^2 \quad (29b)$$

$$|W(\chi, k) - W(\bar{\chi}, k)| \le c_4 \|\chi - \bar{\chi}\| \left(\|\chi\| + \|\bar{\chi}\|\right), \quad (29c)$$

for all $\chi, \bar{\chi} \in \mathcal{X}$, and some $c_1, c_2, c_3, c_4 > 0$. ■

The proof of Lemma 3 is reported in Appendix D.

D. Stability of Nominal System

In this section, we combine the stability properties of the boundary layer (cf., Lemma 1) and reduced system (cf., Lemma 3) to formalize those of the nominal system (18). To this end, let $n_\xi := n + m + 2(N-1)d$ and define $\xi \in \mathbb{R}^{n_\xi}$ as

$$\xi := \begin{bmatrix} \tilde{\theta} \\ \tilde{\mathrm{x}} \\ \mathrm{col}(\mathrm{w}_\perp, \mathrm{z}_\perp) - h(\tilde{\theta} + \theta^\star, \tilde{\mathrm{x}} + x^\star) \end{bmatrix}. \quad (30)$$

With this notation at hand, we rewrite (18) as

$$\xi^{k+1} = F_\gamma(\xi^k, k), \quad (31)$$

where $F_\gamma : \mathbb{R}^{n_\xi} \times \mathbb{N} \to \mathbb{R}^{n_\xi}$ compactly represents the update (18) in the new coordinates (30). The next lemma ensures that the origin is an exponentially stable equilibrium point of (31). We first introduce the set $D \subset \mathbb{R}^{n_\xi}$ defined as

$$D := \{\xi := \mathrm{col}(\xi_1, \xi_2, \xi_3, \xi_4) \in \mathbb{R}^{n_\xi} \mid \xi_1 + \theta^\star \in \Theta\}.$$

Lemma 4: There exist a function $V : \mathbb{R}^{n_\xi} \times N \to \mathbb{R}$ and $\bar{\gamma}, a_1, a_2, a_3, a_4 > 0$ such that, for all $\gamma \in (0, \bar{\gamma})$, it holds

$$a_1 \|\xi\|^2 \le V(\xi, k) \le a_2 \|\xi\|^2 \quad (32a)$$

$$V(F_\gamma(\xi, k), k+1) - V(\xi, k) \le -\gamma a_3 \|\xi\|^2 \quad (32b)$$

$$\left|V(\xi, k) - V(\bar{\xi}, k)\right| \le a_4 \left\|\xi - \bar{\xi}\right\| \left(\|\xi\| + \left\|\bar{\xi}\right\|\right), \quad (32c)$$

for all $\xi, \bar{\xi} \in D$ and $k \in \mathbb{N}$. ■

The proof of Lemma 4 is reported in Appendix E.

E. Proof of Theorem 1

The proof consists in finding a uniform ultimate bound to the trajectories of system (17) by using the Lyapunov function V studied in Lemma 4. To this end, by following the notation in (31), we compactly rewrite the original system (17) as

$$\xi^{k+1} := F_\gamma^{\text{or}}(\xi^k, k) = F_\gamma(\xi^k, k) - \gamma \mathrm{p}(\xi^k), \quad (33)$$

with $F_\gamma^{\text{or}} : \mathbb{R}^{n_\xi} \to \mathbb{R}^{n_\xi}$ and $\mathrm{p} : \mathbb{R}^{n_\xi} \to \mathbb{R}^{n_\xi}$, which reads as

$$\mathrm{p}(\xi) := \begin{bmatrix} \mathrm{p}_\ell(\xi_2 + x^\star, k) \\ \mathrm{p}_1(\xi_2 + x^\star) + \nabla\phi(\xi_2 + x^\star) \frac{\mathbf{1}\mathbf{1}^\top}{N} \mathrm{p}_2(\xi_2 + x^\star) \\ \frac{R^\top}{\gamma} \left(\phi\left(F_{\gamma,2}^{\text{or}}(\xi, k) + x^\star\right) - \phi\left(F_{\gamma,2}(\xi, k) + x^\star\right)\right) \\ \frac{R^\top}{\gamma} \left(\hat{G}_2^{\text{or}}\left(F_\gamma^{\text{or}}(\xi, k)\right) - \hat{G}_2^{\text{nom}}\left(F_\gamma(\xi, k)\right)\right) \end{bmatrix},$$

in which $F_{\gamma,i}(\xi^k, k)$ and $F_{\gamma,i}^{\text{or}}(\xi^k, k)$ denote the i-th components of $F_\gamma(\xi^k, k)$ and $F_\gamma^{\text{or}}(\xi^k, k)$, respectively, while $\hat{G}_2^{\text{or}}(F_\gamma^{\text{or}}(\xi, k))$ and $\hat{G}_2^{\text{nom}}(F_\gamma(\xi, k))$ represent the function $\hat{G}_2$ evaluated, respectively, at $(F_{\gamma,2}^{\text{or}}(\xi, k) + x^\star, \mathbf{1}\sigma(F_{\gamma,2}^{\text{or}}(\xi, k) + x^\star), F_{\gamma,1}^{\text{or}}(\xi, k) + \theta^\star)$ and $(F_{\gamma,2}(\xi, k) + x^\star, \mathbf{1}\sigma(F_{\gamma,2}(\xi, k) + x^\star), F_{\gamma,1}(\xi, k) + \theta^\star)$. We proceed bounding $\|F_{\gamma,2}^{\text{or}}(\xi^k, k) - F_{\gamma,2}(\xi^k, k)\|^2$ as

$$\begin{aligned} &\left\|F_{\gamma,2}^{\text{or}}(\xi^k, k) - F_{\gamma,2}(\xi^k, k)\right\|^2 \\ &= \left\|\mathrm{p}_1(\xi_2 + x^\star) + \nabla\phi(\xi_2 + x^\star) \frac{\mathbf{1}\mathbf{1}^\top}{N} \mathrm{p}_2(\xi_2 + x^\star)\right\|^2 \\ &\overset{(a)}{\le} \|\mathrm{p}_1(\xi_2 + x^\star)\|^2 + \left\|\nabla\phi(\xi_2 + x^\star) \frac{\mathbf{1}\mathbf{1}^\top}{N} \mathrm{p}_2(\xi_2 + x^\star)\right\|^2 \\ &\quad + 2\mathrm{p}_1(\xi_2 + x^\star)^\top \nabla\phi(\xi_2 + x^\star) \frac{\mathbf{1}\mathbf{1}^\top}{N} \mathrm{p}_2(\xi_2 + x^\star) \\ &\overset{(b)}{\le} \epsilon^2 N \left(1 + L_\phi^2 + 2L_\phi\right) = \epsilon^2 N (1 + L_\phi)^2, \end{aligned} \quad (34)$$

where in (a), we expanded the square norm, while in (b), we used the Cauchy–Schwarz inequality, the Lipschitz continuity of ϕ (see Assumption 2), and the accuracy bounds enforced by Assumption 6. Similarly, we compute the following bound:

$$\begin{aligned} &\left\|\mathbf{1}\sigma(F_{\gamma,2}^{\text{or}}(\xi^k, k) + x^\star) - \mathbf{1}\sigma(F_{\gamma,2}(\xi^k, k) + x^\star)\right\|^2 \\ &= \left\|\frac{\mathbf{1}\mathbf{1}^\top}{N} \left(\phi(F_{\gamma,2}^{\text{or}}(\xi^k, k)) - \phi(F_{\gamma,2}(\xi^k, k))\right)\right\|^2 \\ &\le L_\phi^2 \left\|F_{\gamma,2}^{\text{or}}(\xi^k, k) - F_{\gamma,2}(\xi^k, k)\right\|^2 \overset{(a)}{\le} \epsilon^2 N L_\phi^2 (1 + L_\phi)^2, \end{aligned} \quad (35)$$

where in (a), we used the bound (34). Lastly, we notice that

$$\left\|F_{\gamma,1}^{\text{or}}(\xi^k, k) - F_{\gamma,1}(\xi^k, k)\right\|^2 = \|\gamma \mathrm{p}_\ell(\xi_2 + x^\star, k)\|^2 \le \gamma^2 \epsilon^2 N. \quad (36)$$

At this point, $\|\mathrm{p}(\xi)\|$ can be bounded as

$$\begin{aligned} \|\mathrm{p}(\xi)\|^2 &= \|\mathrm{p}_\ell(\xi_2 + x^\star, k)\|^2 \\ &\quad + \left\|\mathrm{p}_1(\xi_2 + x^\star) + \nabla\phi(\xi_2 + x^\star) \frac{\mathbf{1}\mathbf{1}^\top}{N} \mathrm{p}_2(\xi_2 + x^\star)\right\|^2 \end{aligned}$$

$$
\begin{aligned}
&+\left\|\frac{R^\top}{\gamma}\left(\phi\left(F^{\text{or}}_{\gamma,2}(\xi,k)+x^\star\right)-\phi(F_{\gamma,2}(\xi,k)+x^\star)\right)\right\|^2\\
&+\left\|\frac{R^\top}{\gamma}\hat{G}^{\text{or}}_2\left(F^{\text{or}}_\gamma(\xi,k)\right)-\hat{G}^{\text{nom}}_2\left(F_\gamma(\xi,k)\right)\right\|^2\\
&\overset{(a)}{\leq}\|\mathrm{p}_\ell(\xi_2+x^\star,k)\|^2\\
&+\left\|\mathrm{p}_1(\xi_2+x^\star)+\nabla\phi(\xi_2+x^\star)\,\frac{\mathbf{1}\mathbf{1}^\top}{N}\mathrm{p}_2(\xi_2+x^\star)\right\|^2\\
&+\frac{L_\phi^2}{\gamma^2}\|R\|^2\left\|F^{\text{or}}_{\gamma,2}(\xi^k,k)-F_{\gamma,2}(\xi^k,k)\right\|^2\\
&+\frac{L_\sigma^2}{\gamma^2}\|R\|^2\Big(\left\|F^{\text{or}}_{\gamma,2}(\xi^k,k)-F_{\gamma,2}(\xi^k,k)\right\|^2\\
&+\left\|\mathbf{1}\sigma(F^{\text{or}}_{\gamma,2}(\xi^k,k)+x^\star)-\mathbf{1}\sigma(F_{\gamma,2}(\xi^k,k)+x^\star)\right\|^2\\
&+\left\|F^{\text{or}}_{\gamma,1}(\xi^k,k)-F_{\gamma,1}(\xi^k,k)\right\|^2\Big)\\
&\overset{(b)}{\leq}\epsilon^2 N\Bigg[1+(1+L_\phi)^2+\frac{L_\phi^2}{\gamma^2}\|R\|^2(1+L_\phi)^2\\
&+\frac{L_\sigma^2}{\gamma^2}\|R\|^2\Big((1+L_\phi)^2+L_\phi^2(1+L_\phi)^2+\gamma^2\Big)\Bigg]=:\epsilon^2 a_5^2,
\end{aligned}
$$

where in (a), we used the Cauchy–Schwarz inequality and the Lipschitz continuity of $\phi_i(\mathrm{x}_i)$ and $\nabla_2\hat{f}_i$ (cf., Assumptions 2 and 4), while in (b), we used (34)–(36). With this bound at hand, we now consider the Lyapunov function V introduced in Lemma 4 to study the stability properties of the original system (33). Then, by evaluating the increment of V along the trajectories of (33), for all $k\in\mathbb{N}$ and $\xi\in D$, we get

$$
\begin{aligned}
&\Delta V(\xi,k):=V(F_\gamma(\xi,k)-\gamma\mathrm{p}(\xi),k+1)-V(\xi,k)\\
&\overset{(a)}{=}V(F_\gamma(\xi,k),k+1)-V(\xi,k)\\
&\quad+V(F_\gamma(\xi,k)-\gamma\mathrm{p}(\xi),k+1)-V(F_\gamma(\xi,k),k+1)\\
&\overset{(b)}{\leq}-\gamma a_3\|\xi\|^2+a_4\gamma\|\mathrm{p}(\xi)\|\left(\|F_\gamma(\xi,k)-\gamma\mathrm{p}(\xi)\|+\|F_\gamma(\xi,k)\|\right)\\
&\overset{(c)}{\leq}-\gamma a_3\|\xi\|^2+\gamma 2a_4\|\mathrm{p}(\xi)\|\,\|F_\gamma(\xi,k)\|+\gamma^2 a_4\|\mathrm{p}(\xi)\|^2\\
&\overset{(d)}{\leq}-\gamma a_3\|\xi\|^2+\gamma 2a_4a_5\epsilon\|F_\gamma(\xi,k)\|+\gamma^2 a_4a_5^2\epsilon^2, \qquad (37)
\end{aligned}
$$

where in (a), we add $\pm V(F_\gamma(\xi,k),k+1)$, in (b), we use (32b) and (32c) (cf., Lemma 4), in (c), we use the triangle inequality, while in (d), we use (7). Now, since $F_\gamma(\xi,k)$ is the sum of Lipschitz continuous functions [see (18) and Assumptions 2 and 4–6], it is Lipschitz continuous too. Further, we recall that $F_\gamma(0,k)=0$ for all $k\in\mathbb{N}$. Hence, by denoting with $L_\ell>0$ its Lipschitz continuity constant, we bound (37) as

$$
\begin{aligned}
\Delta V(\xi,k)&\leq-\gamma a_3\|\xi\|^2+2\gamma a_4a_5L_\ell\epsilon\|\xi\|+\gamma^2a_4a_5^2\epsilon^2\\
&\overset{(a)}{\leq}-\gamma a_3\|\xi\|^2+\gamma a_4a_5L_\ell p\|\xi\|^2\\
&\quad+\gamma a_4a_5L_\ell\epsilon^2/p+\gamma^2a_4a_5^2\epsilon^2\\
&\overset{(b)}{\leq}-\gamma a_2\tilde{a}\|\xi\|^2+\gamma a_4a_5(L_\ell/p+\gamma a_5)\epsilon^2\\
&\overset{(c)}{\leq}-\gamma\tilde{a}V(\xi,k)+\gamma a_4a_5(L_\ell/p+\gamma a_5)\epsilon^2, \qquad (38)
\end{aligned}
$$

where in (a), we apply the Young's inequality with parameter $p\in(0,(a_3-a_2\tilde{a})/(a_4a_5L_\ell))$ considering an arbitrarily chosen $\tilde{a}\in(0,a_3/a_2)$, in (b), we rearrange the terms according to the choice of p, and in (c), we use (32a). Now, we arbitrarily fix $\gamma\in(0,\bar{\gamma})$, define $a_6:=\frac{a_4a_5}{\tilde{a}}(\frac{L_\ell}{p}+\gamma a_5)$, and bound (38) as

$$
\Delta V(\xi,k)\leq-\gamma\tilde{a}\left(V(\xi,k)-\epsilon^2a_6\right). \qquad (39)
$$

Because of the definition of ΔV, the inequality (39) leads to

$$
V(F_\gamma(\xi,k),k+1)\leq(1-\gamma\tilde{a})V(\xi,k)+\gamma\epsilon^2a_6\tilde{a}. \qquad (40)
$$

Thus, by iteratively applying (40) and since $1-\gamma\tilde{a}<1$, we get

$$
V\left(\xi^k,k\right)\leq(1-\gamma\tilde{a})^kV\left(\xi^0,0\right)+\epsilon^2a_6. \qquad (41)
$$

The proof follows by combining (41) with (32a) and $\|\mathrm{x}^k-x^\star\|\leq\|\xi^k\|$ [see the definition of ξ in (30)], and by setting $\kappa_1:=\sqrt{a_2/a_1}$, $\kappa_2:=\sqrt{1-\gamma\tilde{a}}$, and $B:=\sqrt{a_6/a_1}$.

V. Numerical Simulations

In this section, we test DELTA through numerical simulations. We consider an instance of problem (1) with $N=20$ agents. In detail, for all $i\in\{1,\dots,N\}$, we consider the following setup. We define the local cost functions f_i as

$$
\begin{aligned}
f_i(x_i,\sigma(x)):=&\frac{1}{2}\begin{bmatrix}x_i\\ \sigma(x)\end{bmatrix}^\top P_i\begin{bmatrix}x_i\\ \sigma(x)\end{bmatrix}+v_i^\top\begin{bmatrix}x_i\\ \sigma(x)\end{bmatrix}\\
&+a_ie^{-b_i^\top\begin{bmatrix}x_i\\ \sigma(x)\end{bmatrix}+c_i}+q_i,
\end{aligned}
$$

where $x_i\in\mathbb{R}$ and $\sigma(x):=\sum_{i=1}^N\pi_ix_i/N$. Further, the parameters π_i,a_i,b_i, and c_i and the entries of $P_i\in\mathbb{R}^{2\times2}$ are drawn uniformly from (0, 1) enforcing P_i to be positive definite. The entries of $v_i\in\mathbb{R}^2$ and $q_i\in\mathbb{R}$ are drawn uniformly from (0, 20). Then, we set $\mathrm{d}^k_{x,i}(k)=5\cos(2\pi k/4)$ and $\mathrm{d}^k_{\sigma,i}(k)=5\sin(2\pi k/4)$. We pick local neural networks composed of two layers with 300 neurons equipped with the softplus function. As for the learning problem (4), we consider the local cost functions

$$
\ell_i(u_{x,i},u_{\sigma,i},\theta_i):=\frac{1}{2}(f_i(u_{x,i},u_{\sigma,i})-\hat{f}_i(u_{x,i},u_{\sigma,i},\theta_i))^2+\|\theta_i\|^2.
$$

Further, we pick x_i^0 sampled from a Gaussian distribution, while θ_i^0 is chosen by considering the Xavier uniform initialization [37]. We empirically tune $\gamma=10^{-4}$ and consider a random Erdős–Rényi graph with connectivity $p=0.5$. Fig. 3(a) shows the evolution of the relative cost error $(f_\sigma(\mathrm{x}^k)-f_\sigma(x^\star))/|f_\sigma(x^\star)|$, where $x^\star$ is the optimizer of (1). We compare Algorithm 1 with an implementation of the ZO single-point method of [22] and with the distributed aggregative gradient tracking (DAGT) method of [5], which uses the exact gradients of f_i. All simulations use the same stepsize and initial conditions. All neural networks are implemented in TensorFlow, which provides built-in automatic differentiation [38].

As predicted by Theorem 1, Fig. 3(a) shows that the sequence $\{\mathrm{x}^k\}_{k\in\mathbb{N}}$ generated by Algorithm 1 converges to a neighborhood of $x^\star$. Then, Fig. 3(b) shows the evolution over

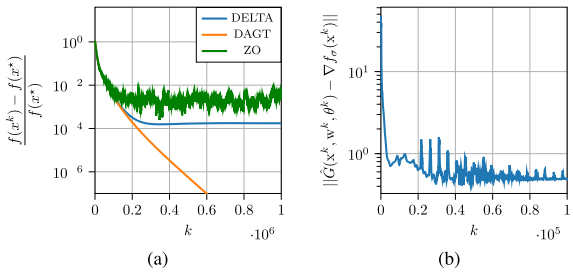

Fig. 3. (a) Comparison between DELTA, DAGT [5], and DAGT with ZO gradient approximation by [22]. (b) Evolution over time of $||\hat{G}(\mathrm{x}^k, \mathrm{w}^k, \theta^k) - \nabla f_\sigma(\mathrm{x}^k)||$.

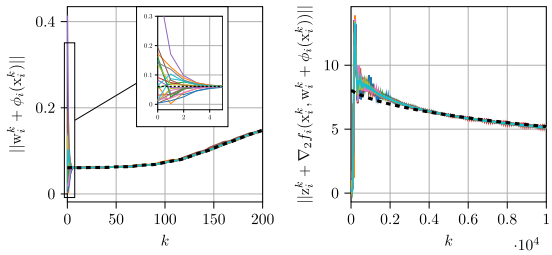

Fig. 4. Evolution of the tracking dynamics. The dashed lines are the global $\sigma(\mathrm{x}^k)$ (left) and $\sum_{j=1}^N \nabla_2 \hat{f}_j(\mathrm{x}_j^k, \sigma(\mathrm{x}^k), \theta_j^k)/N$ (right). The solid lines are their local agents estimates.

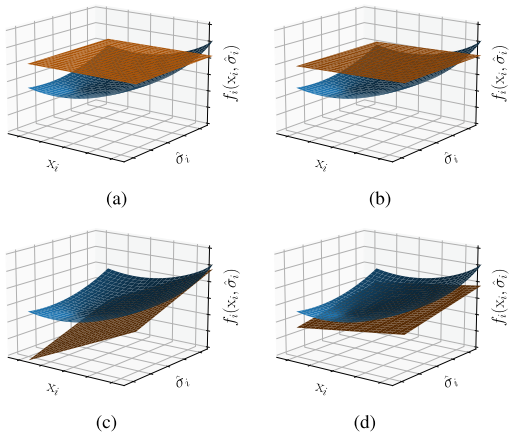

Fig. 5. Agent i perspective: evolution of the learned tangent space (in orange) over the iteration compared to f_i (in blue). (a) Iteration $k = 0$. (b) Iteration $k = 20$. (c) Iteration $k = 500$. (d) Iteration $k = 10\,000$.

time of the estimation error of the descent direction induced both by the neural networks and the tracking mechanism, i.e., $||\hat{G}(\mathrm{x}^k, \mathrm{w}^k, \theta^k) - \nabla f_\sigma(\mathrm{x}^k)||$, where $\hat{G}(\mathrm{x}^k, \mathrm{w}^k, \theta^k)$ is the estimated descent direction at iteration k, and thus, it reads as

$$\hat{G}(\mathrm{x}^k, \mathrm{w}^k, \theta^k) := \hat{G}_1(\mathrm{x}^k, \mathrm{w}^k + \phi(\mathrm{x}^k), \theta^k) + \nabla\phi(\mathrm{x}^k)(\hat{G}_2\left(\mathrm{x}^k, \mathrm{w}^k + \phi(\mathrm{x}^k), \theta^k\right) + \mathrm{z}^k).$$

Fig. 4 illustrates the timescale separation in Algorithm 1, showing that the consensus mechanism converges in fewer iterations than the optimization and learning components [cf., Fig. 3(a) and (b)]. In Fig. 5, we provide a graphical understanding of the learning capabilities of an agent. In detail, we show the evolution of the learned tangent space to the local cost f_i around the current pair $(\mathrm{x}_i^k, \mathrm{w}_i^k + \phi_i(\mathrm{x}_i^k))$. Finally, we tested the robustness of DELTA to changes in the local f_i. In particular, at iteration $k = 10^5$, the parameters of each f_i are perturbed by subtracting random values drawn from a uniform distribution in (0,0.1). Fig. 6 shows that after this sudden change, the algorithm

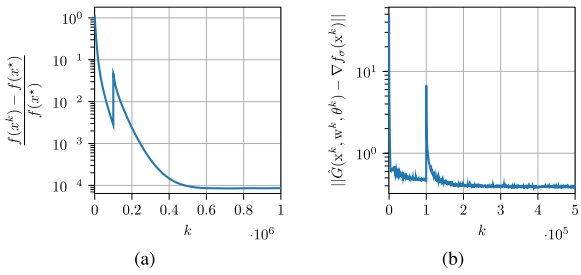

Fig. 6. Robustness of DELTA to changes in the local costs. (a) Evolution of the relative cost error. (b) Evolution of the norm of the descent estimation error.

relearns the new cost function gradients and converges to a neighborhood of the new optimal solution.

VI. Conclusion

In this article, we proposed DELTA, a data-driven distributed algorithm for aggregative optimization in unknown scenarios. DELTA combines an optimization-oriented part with a tracking mechanism and a learning one. The latter takes on local neural networks elaborating cost samples around the current solution estimates to approximate their gradients. We analyzed DELTA by using tools from system theory based on timescale separation and averaging theory. In detail, we theoretically proved linear convergence in a neighborhood of the optimal solution whose size depends on the given accuracy capabilities of the neural networks. Some numerical tests confirmed our findings.

Appendix

A. Preliminaries on Averaging Theory

We report [39, Thm. 2.2.2], which is a useful result in the context of averaging theory. Consider the time-varying system

$$\chi^{k+1} = \chi^k + \gamma f(\chi^k, \gamma, k), \tag{42}$$

where $\chi^k \in \mathbb{R}^{n_\chi}$ is the system state, $f : \mathbb{R}^{n_\chi} \times \mathbb{R} \times \mathbb{N} \to \mathbb{R}^{n_\chi}$ describes its dynamics, and $\gamma > 0$ is a tunable parameter. We now introduce a set of assumptions that allow for studying the stability properties of (42) by considering the so-called averaged system associated with (42). In particular, we impose these conditions within a sphere $\mathcal{B}_r(0_{n_\chi})$ of radius $r > 0$.

Assumption 7: The function f is piecewise continuous in k and the limit

$$f_{\text{AV}}(\chi) := \lim_{T\to\infty} \frac{1}{T} \sum_{\tau=\bar{k}+1}^{\bar{k}+T} f(\chi, 0, \tau), \tag{43}$$

exists uniformly in $\bar{k} \in \mathbb{N}$ and $\chi \in \mathcal{B}_r(0_{n_\chi})$. ■

Once f_{AV} is well posed, we can introduce the *averaged system* associated with (42), namely

$$\chi_{\text{AV}}^{k+1} = \chi_{\text{AV}}^k + \epsilon f_{\text{AV}}(\chi_{\text{AV}}^k), \tag{44}$$

with $\chi_{\text{AV}}^k \in \mathbb{R}^{n_\chi}$. Then, we enforce the following assumptions.

Assumption 8: The origin is an equilibrium point for both (42) and (44), namely

$$f(0, \gamma, k) = 0, \quad f_{\text{AV}}(0) = 0,$$

for all $\gamma \geq 0$ and $k \in \mathbb{N}$. ■

Assumption 9: There exist $l_1, l_2, l_{\text{AV}}, \bar{\gamma}_1 > 0$ such that

$$\begin{aligned}\|f(\chi,\gamma,k) - f(\chi',\gamma,k)\| &\leq l_1 \|\chi - \chi'\| \\ \|f(\chi,\gamma,k) - f(\chi,\gamma',k)\| &\leq l_2|\gamma - \gamma'| \, \|\chi\| \\ \|f_{\text{AV}}(\chi) - f_{\text{AV}}(\chi')\| &\leq l_{\text{AV}} \|\chi - \chi'\|,\end{aligned}$$

for all $\gamma, \gamma' \in (0, \bar{\gamma}_1]$, $\chi, \chi' \in \mathcal{B}_r(0_{n_\chi})$, and $k \in \mathbb{N}$. ■

Assumption 10: Let $\Delta f : \mathbb{R}^{n_\chi} \times \mathbb{N} \to \mathbb{R}^{n_\chi}$ be defined as

$$\Delta f(\chi, k) := f(\chi, 0, k) - f_{\text{AV}}(\chi).$$

Then, there exists a nonnegative strictly decreasing function $\nu(k)$ such that $\lim_{k\to\infty} \nu(k) = 0$ and

$$\begin{aligned}\left\| \frac{1}{T} \textstyle\sum_{\tau=\bar{k}+1}^{\bar{k}+T} \Delta f(\chi, \tau) \right\| &\leq \nu(T) \|\chi\| \\ \left\| \frac{1}{T} \textstyle\sum_{\tau=\bar{k}+1}^{\bar{k}+T} \frac{\partial \Delta f(\chi,\tau)}{\partial \chi} \right\| &\leq \nu(T),\end{aligned}$$

uniformly in $\bar{k} \in \mathbb{N}$ and $\chi \in \mathcal{B}_r(0_{n_\chi})$. ■

These assumptions ensure the following result.

Theorem 2 (See [39, Thm. 2.2.3]): Consider system (42) and let Assumptions 7–10 hold. If there exists $\bar{\gamma}_1 > 0$ such that, for all $\gamma \in (0, \bar{\gamma}_1)$, the origin is an exponentially stable equilibrium point of the averaged system (44), then there exists $\bar{\gamma} \in (0, \bar{\gamma}_1)$ such that, for all $\gamma \in (0, \bar{\gamma})$, the origin is an exponentially stable equilibrium point of system (42). ■

B. Proof of Lemma 1

We point out that Assumption (3) implies that $R^\top A R$ is Schur. Hence, for all $q > 0$, there exist $\Pi_1 = \Pi_1^\top > 0$ and $\Pi_2 = \Pi_2^\top > 0$ solving the discrete-time Lyapunov equations

$$\left(R^\top A R\right)^\top \Pi_1 \left(R^\top A R\right) - \Pi_1 = -q I_{(N-1)d} \tag{46a}$$

$$\left(R^\top A R\right)^\top \Pi_2 \left(R^\top A R\right) - \Pi_2 = -I_{(N-1)d}, \tag{46b}$$

where $q > 0$ will be fixed later. Let $\Pi := \text{blkdiag}(\Pi_1, \Pi_2) \in \mathbb{R}^{2(N-1)d}$ and $U : \mathbb{R}^{2(N-1)d} \to \mathbb{R}$ be defined as

$$U(\zeta) := \zeta^\top \Pi \zeta.$$

Being U quadratic, the conditions (24a) and (24c) are satisfied. To conclude the proof, we show that (24b) is verified studying the increment of U along the trajectories of system (23), i.e.,

$$\begin{aligned}\Delta U(\zeta) &:= U(\tilde{A}\zeta + g_{\text{B}}(\tilde{\text{x}}, \zeta, \tilde{\theta})) - U(\zeta) \\ &= \zeta^\top (\tilde{A}^\top \Pi \tilde{A} - \Pi)\zeta + 2\zeta^\top \tilde{A}^\top \Pi g_{\text{B}}(\tilde{\text{x}}, \zeta, \tilde{\theta}) \\ &\quad + \Delta\hat{G}_2(\text{x}^k, \zeta_1, \theta^k)^\top \mathcal{M}(\Pi) \Delta\hat{G}_2(\text{x}^k, \zeta_1, \theta^k),\end{aligned} \tag{47}$$

where $\mathcal{M}(\Pi) := [0 \;\; R^\top M] \Pi [0 \;\; R^\top M]^\top$. By looking at the definition of $\hat{G}_2$ [cf., (14)] and $\Delta\hat{G}_2$ [cf., (21)], since $\nabla_2 \hat{f}_i$ is Lipschitz continuous (cf., Assumption 4), we have

$$\|\Delta\hat{G}_2(\tilde{\text{x}} + x^\star, \tilde{\text{w}}_\perp, \tilde{\theta} + \theta^\star)\| \leq L_\sigma \|R\tilde{\text{w}}_\perp\|, \tag{48}$$

for all $\tilde{\text{w}}_\perp \in \mathbb{R}^{(N-1)d}$. Now, we use (46) to bound (47) as

$$\begin{aligned}\Delta U(\zeta) &= -q\|\zeta_1\|^2 - \|\zeta_2\|^2 + 2\zeta^\top \tilde{A}^\top \Pi g_{\text{B}}(\tilde{\text{x}}, \zeta, \tilde{\theta}) \\ &\quad + \Delta\hat{G}_2(\text{x}, \zeta_1, \tilde{\theta})^\top (R^\top M)^\top \Pi_2 (R^\top M) \Delta\hat{G}_2(\text{x}, \zeta_1, \tilde{\theta}) \\ &\overset{\text{(a)}}{\leq} -q\|\zeta_1\|^2 - \|\zeta_2\|^2 + 2\|\zeta_2\| \left\|R^\top A R \Pi_2 R A\right\| \left\|\Delta\hat{G}_2(\text{x}, \zeta_1, \tilde{\theta})\right\| \\ &\quad + \left\|(R^\top M)^\top \Pi_2 (R^\top M)\right\| \left\|\Delta\hat{G}_2(\text{x}, \zeta_1, \tilde{\theta})\right\|^2 \\ &\overset{\text{(b)}}{\leq} -q\|\zeta_1\|^2 - \|\zeta_2\|^2 + 2L_\sigma \left\|R^\top A R \Pi_2 R A\right\| \|R\zeta_1\| \, \|\zeta_2\| \\ &\quad + \left\|(R^\top M)^\top \Pi_2 (R^\top M)\right\| L_\sigma^2 \|R\zeta_1\|^2,\end{aligned} \tag{49}$$

where in (a), we use the Cauchy–Schwarz inequality and the definition of g_{B} [cf., (22)], and in (b), we use (48). Let us define $\kappa_1 := \|(R^\top M)^\top \Pi_2 (R^\top M)\| L_\sigma^2 \|R\|^2$ and $\kappa_{12} := L_\sigma \|R^\top A R \Pi_2 R A\| \|R\|$. Then, we compactly rewrite (49) as

$$\Delta U(\zeta) \leq -\begin{bmatrix}\|\zeta_1\| \\ \|\zeta_2\|\end{bmatrix}^\top \underbrace{\begin{bmatrix}q - \kappa_1 & \kappa_{12} \\ \kappa_{12} & 1\end{bmatrix}}_{=:P_U} \begin{bmatrix}\|\zeta_1\| \\ \|\zeta_2\|\end{bmatrix}. \tag{50}$$

By the Sylvester criterion, we know that $P_U > 0$ if and only if

$$q > \kappa_1 \quad \text{and} \quad q > \kappa_1 + \kappa_{12}^2. \tag{51}$$

We set $q > \bar{q}_1 := \kappa_1 + \kappa_{12}^2$, and thus, we bound (50) as

$$\Delta U(\zeta) \leq -\lambda_{\min}(P_U) \|\zeta\|^2,$$

where $\lambda_{\min}(P_U) > 0$ denotes the smallest eigenvalue of P_U. Hence, also (24b) is achieved, and thus, the proof concludes.

C. Proof of Lemma 2

By relying on the accuracy characterization (6c), we get

$$\nabla_2 \tilde{\ell}_{\mathcal{K}}(\tilde{\text{x}}, 0) = \frac{1}{\mathcal{K}} \textstyle\sum_{k=1}^{\mathcal{K}} \text{p}_\ell(\tilde{\text{x}} + x^\star, k), \tag{52}$$

for all $\tilde{\text{x}} \in \mathbb{R}^n$. Since $\theta^\star$ is a minimizer of problem (4) for $k = \mathcal{K}$ (cf., Assumption 5), it holds $\nabla_2 \tilde{\ell}_{\mathcal{K}}(\tilde{\text{x}}, 0) = 0$, which, combined with (52), leads to $\sum_{k=1}^{\mathcal{K}} \text{p}_\ell(\tilde{\text{x}} + x^\star, k) = 0$ for all $\tilde{\text{x}} \in \mathbb{R}^n$. Thus, system (27) reduces to

$$\tilde{\theta}_{\text{AV}}^{k+1} = \tilde{\theta}_{\text{AV}}^k - \gamma \nabla_2 \tilde{\ell}_{\mathcal{K}}(\tilde{\text{x}}_{\text{AV}}, \tilde{\theta}_{\text{AV}}^k) \tag{53a}$$

$$\tilde{\text{x}}_{\text{AV}}^{k+1} = \tilde{\text{x}}_{\text{AV}}^k - \gamma \hat{G}^d(\tilde{\text{x}}_{\text{AV}}^k, \tilde{\theta}_{\text{AV}}^k). \tag{53b}$$

Now, let $W : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ be the Lyapunov function

$$W(\tilde{\theta}_{\text{AV}}, \tilde{\text{x}}_{\text{AV}}) = \alpha \left\|\tilde{\theta}_{\text{AV}}\right\|^2 + \|\tilde{\text{x}}_{\text{AV}}\|^2,$$

where $\alpha > 0$ will be fixed in the following. Let us introduce the increments $\Delta W_\theta(\tilde{\theta}_{\text{AV}}, \tilde{\text{x}}_{\text{AV}})$ and $\Delta W_{\text{x}}(\tilde{\theta}_{\text{AV}}, \tilde{\text{x}}_{\text{AV}})$ defined as

$$\begin{aligned}\Delta W_\theta(\tilde{\theta}_{\text{AV}}, \tilde{\text{x}}_{\text{AV}}) &:= \|\tilde{\theta}_{\text{AV}} - \gamma \nabla_2 \tilde{\ell}_{\mathcal{K}}(\tilde{\text{x}}_{\text{AV}}, \tilde{\theta}_{\text{AV}})\|^2 - \|\tilde{\theta}_{\text{AV}}\|^2 \\ \Delta W_{\text{x}}(\tilde{\theta}_{\text{AV}}, \tilde{\text{x}}_{\text{AV}}) &:= \|\tilde{\text{x}}_{\text{AV}} - \gamma \hat{G}^d(\tilde{\text{x}}_{\text{AV}}, \tilde{\theta}_{\text{AV}})\|^2 - \|\tilde{\text{x}}_{\text{AV}}\|^2.\end{aligned}$$

Then, by considering $(\tilde{\theta}_{\text{AV}}, \tilde{\text{x}}_{\text{AV}}) \in \mathbb{R}^m \times \mathbb{R}^n$ with $(\tilde{\theta}_{\text{AV}} + \theta^\star) \in \Theta$, the increment of W along the trajectories of system (53) is

$$\Delta W(\tilde{\theta}_{\text{AV}}, \tilde{\text{x}}_{\text{AV}}) = \alpha \Delta W_\theta(\tilde{\theta}_{\text{AV}}, \tilde{\text{x}}_{\text{AV}}) + \Delta W_{\text{x}}(\tilde{\theta}_{\text{AV}}, \tilde{\text{x}}_{\text{AV}}). \tag{54}$$

As for $\Delta W_\theta(\tilde{\theta}_{\text{AV}}, \tilde{\text{x}}_{\text{AV}})$, we expand the square norm and obtain

$$\begin{aligned}&\Delta W_\theta(\tilde{\theta}_{\text{AV}}, \tilde{\text{x}}_{\text{AV}}) \\ &= -2\gamma \nabla_2 \tilde{\ell}_{\mathcal{K}}(\tilde{\text{x}}_{\text{AV}}, \tilde{\theta}_{\text{AV}})^\top \tilde{\theta}_{\text{AV}} + \gamma^2 \left\|\nabla_2 \tilde{\ell}_{\mathcal{K}}(\tilde{\text{x}}_{\text{AV}}, \tilde{\theta}_{\text{AV}})\right\|^2 \\ &\overset{\text{(a)}}{\leq} -\gamma \frac{2\mu_\ell L_\ell}{\mu_\ell + L_\ell} \left\|\tilde{\theta}_{\text{AV}}\right\|^2 - \gamma \left(\frac{2}{\mu_\ell + L_\ell} - \gamma\right) \left\|\nabla_2 \tilde{\ell}_{\mathcal{K}}(\tilde{\text{x}}_{\text{AV}}, \tilde{\theta}_{\text{AV}})\right\|^2,\end{aligned} \tag{55}$$

where in (a), we use $\nabla_2\tilde{\ell}_{\mathcal{K}}(\tilde{\mathrm{x}}_{\mathrm{AV}},0)=0$, the μ_ℓ-strong convexity of $\tilde{\ell}_{\mathcal{K}}$ in Θ, and the L_ℓ-Lipschitz continuity of $\nabla_2\tilde{\ell}_{\mathcal{K}}$ (cf., Assumption 6). Then, we expand ΔW_{x} and obtain

$$\begin{aligned}
&\Delta W_{\mathrm{x}}(\tilde{\theta}_{\mathrm{AV}},\tilde{\mathrm{x}}_{\mathrm{AV}})\\
&=-2\gamma\hat{G}^d(\tilde{\mathrm{x}}_{\mathrm{AV}},\tilde{\theta}_{\mathrm{AV}})^\top\tilde{\mathrm{x}}_{\mathrm{AV}}+\gamma^2\left\|\hat{G}^d(\tilde{\mathrm{x}}_{\mathrm{AV}},\tilde{\theta}_{\mathrm{AV}})\right\|^2\\
&\overset{(a)}{=}-2\gamma\nabla f_\sigma(\tilde{\mathrm{x}}_{\mathrm{AV}}+x^\star)^\top\tilde{\mathrm{x}}_{\mathrm{AV}}\\
&\quad-2\gamma(\hat{G}^d(\tilde{\mathrm{x}}_{\mathrm{AV}},\tilde{\theta}_{\mathrm{AV}})-\hat{G}^d(\tilde{\mathrm{x}}_{\mathrm{AV}},0))^\top\tilde{\mathrm{x}}_{\mathrm{AV}}\\
&\quad+\gamma^2\left\|\hat{G}^d(\tilde{\mathrm{x}}_{\mathrm{AV}},\tilde{\theta}_{\mathrm{AV}})-\hat{G}^d(\tilde{\mathrm{x}}_{\mathrm{AV}},0)+\hat{G}^d(\tilde{\mathrm{x}}_{\mathrm{AV}},0)\right\|^2\\
&\overset{(b)}{=}-2\gamma\nabla f_\sigma(\tilde{\mathrm{x}}_{\mathrm{AV}}+x^\star)^\top\tilde{\mathrm{x}}_{\mathrm{AV}}+2\gamma^2\left\|\hat{G}^d(\tilde{\mathrm{x}}_{\mathrm{AV}},0)\right\|^2\\
&\quad+2\gamma\left\|\hat{G}^d(\tilde{\mathrm{x}}_{\mathrm{AV}},\tilde{\theta}_{\mathrm{AV}})-\hat{G}^d(\tilde{\mathrm{x}}_{\mathrm{AV}},0)\right\|\|\tilde{\mathrm{x}}_{\mathrm{AV}}\|\\
&\quad+2\gamma^2\left\|\hat{G}^d(\tilde{\mathrm{x}}_{\mathrm{AV}},\tilde{\theta}_{\mathrm{AV}})-\hat{G}^d(\tilde{\mathrm{x}}_{\mathrm{AV}},0)\right\|^2,
\end{aligned}\tag{56}$$

where in (a), we add $\pm\nabla f_\sigma(\tilde{\mathrm{x}}_{\mathrm{AV}}+x^\star)=\hat{G}^d(\tilde{\mathrm{x}}_{\mathrm{AV}},0)$, and (b) uses the triangle and Cauchy–Schwarz inequalities and $\nabla f_\sigma(x^\star)=\hat{G}^d(0,0)=0$. By Lipschitz continuity of $\nabla_1\hat{f}_i$, $\nabla_2\hat{f}_i$, and ϕ_i (cf., Assumptions 2 and 4), there exists $L_\theta>0$ such that

$$\|\hat{G}^d(\tilde{\mathrm{x}}_{\mathrm{AV}},\tilde{\theta}_{\mathrm{AV}})-\hat{G}^d(\tilde{\mathrm{x}}_{\mathrm{AV}},\bar{\theta}_{\mathrm{AV}})\|\le L_\theta\|\tilde{\theta}_{\mathrm{AV}}-\bar{\theta}_{\mathrm{AV}}\|,\tag{57}$$

for all $\tilde{\mathrm{x}}_{\mathrm{AV}}\in\mathbb{R}^n$ and $\tilde{\theta}_{\mathrm{AV}},\bar{\theta}_{\mathrm{AV}}\in\mathbb{R}^m$. By (57), μ_f-strong convexity of f_σ, and L_x-Lipschitz continuity of ∇f_σ, we bound (56) as

$$\begin{aligned}
&\Delta W_{\mathrm{x}}(\tilde{\theta}_{\mathrm{AV}},\tilde{\mathrm{x}}_{\mathrm{AV}})\\
&\le-\gamma\frac{2\mu_f L_x}{\mu_f+L_x}\|\tilde{\mathrm{x}}_{\mathrm{AV}}\|^2-\gamma\left(\frac{2}{\mu_f+L_x}-\gamma\right)\|\hat{G}^d(\tilde{\mathrm{x}}_{\mathrm{AV}},0)\|^2\\
&\quad+2\gamma L_\theta\|\tilde{\mathrm{x}}_{\mathrm{AV}}\|\,\|\tilde{\theta}_{\mathrm{AV}}\|+2\gamma^2L_\theta^2\|\tilde{\theta}_{\mathrm{AV}}\|^2.
\end{aligned}\tag{58}$$

Let us arbitrarily set $\beta_x\in(0,\frac{2\mu_f L_x}{\mu_f+L_x})$ and $\beta_\theta\in(0,\frac{2\mu_\ell L_\ell}{\mu_\ell+L_\ell})$ and define $\bar{\gamma}_2:=\min\{\frac{\beta_x}{\mu_f L_x},\frac{\beta_\theta}{\mu_\ell L_\ell}\}$. Then, by choosing $\gamma\in(0,\bar{\gamma}_2)$ and using (55)–(58), we bound ΔW [cf., (54)] as

$$\begin{aligned}
&\Delta W(\tilde{\theta}_{\mathrm{AV}},\tilde{\mathrm{x}}_{\mathrm{AV}})\\
&\le-\begin{bmatrix}\|\tilde{\theta}_{\mathrm{AV}}\|\\\|\tilde{\mathrm{x}}_{\mathrm{AV}}\|\end{bmatrix}^\top\underbrace{\begin{bmatrix}\alpha\gamma\beta_\theta-2\gamma^2L_\theta^2&-\gamma L_\theta\\-\gamma L_\theta&\gamma\beta_x\end{bmatrix}}_{=:P_W(\alpha)}\begin{bmatrix}\|\tilde{\theta}_{\mathrm{AV}}\|\\\|\tilde{\mathrm{x}}_{\mathrm{AV}}\|\end{bmatrix}.
\end{aligned}$$

Choose $\alpha>\bar{\alpha}:=(L_\theta^2(1+2\gamma\beta_x))/\beta_\theta\beta_x$. By the Sylvester criterion, we get $P_W(\alpha)>0$, and thus, we bound $\Delta W(\chi_{\mathrm{AV}})$ as

$$\Delta W(\chi_{\mathrm{AV}})\le-\lambda_{\min}(P_W(\alpha))\|\chi_{\mathrm{AV}}\|^2,$$

where $\lambda_{\min}(P_W(\alpha))>0$ represents the smallest eigenvalue of P_W and $\chi_{\mathrm{AV}}:=\mathrm{col}(\tilde{\theta}_{\mathrm{AV}},\tilde{\mathrm{x}}_{\mathrm{AV}})$. Thus, the exponential stability of the origin is proved (see, e.g. [40, Thm. 13.11]).

D. Proof of Lemma 3

The proof relies on Theorem 2 (cf., Appendix A). Hence, we need to check that Assumptions 7–10 are satisfied. First, Assumption 7 is guaranteed by the periodicity of $\mathrm{d}_{x,i}^k$ and $\mathrm{d}_{\sigma,i}^k$ (cf., Assumption 5). Second, Assumption 8 [i.e., the fact that the origin is an equilibrium point of both (25) and the averaged system (53)] is satisfied in light of (6c) in Assumption 6 and since $\theta^\star$ is the solution to (4) with $k=\mathcal{K}$ (cf., Assumption 5). Third, the Lipschitz properties in Assumption 9 follow from Assumptions 5 and 6. Fourth, the periodicity of $\mathrm{d}_{x,i}^k$ and $\mathrm{d}_{\sigma,i}^k$ (cf., Assumption 6) allows for satisfying Assumption 10 with $\nu(T):=\frac{L}{1+T}$, where $L>0$ depends on the Lipschitz constants of $\nabla_3\ell$ and $\nabla_3\tilde{\ell}_{\mathcal{K}}$. Finally, Lemma 2 proves that the origin is an exponentially stable equilibrium of the averaged system (53). Hence, by Theorem 2, there exist $\bar{\gamma}_1\in(0,\bar{\gamma}_2)$ and $r>0$ such that, for all $\gamma\in(0,\bar{\gamma}_1)$ and $\chi^0\in\mathcal{X}$, the origin is an exponentially stable equilibrium of (25). The proof follows by the converse Lyapunov theorem (see, e.g., [39, Thm. 2.2.1]).

E. Proof of Lemma 4

The proof is based on timescale separation. First, Lemma 1 provides a Lyapunov function proving the global exponential stability of the origin of (23), i.e., the boundary-layer system associated with (18). Second, Lemma 3 provides a Lyapunov function proving exponential stability of the origin of (28), i.e., the reduced system associated to (18). Third, the dynamics of system (18) and the equilibrium map (19) are Lipschitz continuous (see Assumptions 2 and 4–6). We thus guarantee exponential stability of $(\theta^\star,x^\star,h(\theta^\star,x^\star))$ for system (18) by slightly extending [41, Thm. II.3]. In detail, the mentioned result provides global results but requires global results in both the boundary-layer and reduced system, while Lemma 3 (i.e., the stability result of the reduced system) holds only locally. Hence, we conclude that the origin is an exponentially stable equilibrium point of system (18) for any initial condition $(\tilde{\theta}^0,\tilde{\mathrm{x}}^0,\mathrm{w}_\perp^0,\mathrm{z}_\perp^0)\in D$. The proof follows by the converse Lyapunov theorem; see, e.g., [39, Thm. 2.2.1].

References

[1] P. Giselsson and A. Rantzer, *Large-Scale and Distributed Optimization*, vol. 2227. Berlin, Germany: Springer, 2018.

[2] A. Nedić and J. Liu, "Distributed optimization for control," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 1, pp. 77–103, 2018.

[3] T. Yang et al., "A survey of distributed optimization," *Annu. Rev. Control*, vol. 47, pp. 278–305, 2019.

[4] A. Testa, G. Carnevale, and G. Notarstefano, "A tutorial on distributed optimization for cooperative robotics: From setups and algorithms to toolboxes and research directions," *Proc. IEEE*, vol. 113, no. 1, pp. 40–65, Jan. 2025.

[5] X. Li, L. Xie, and Y. Hong, "Distributed aggregative optimization over multi-agent networks," *IEEE Trans. Autom. Control*, vol. 67, no. 6, pp. 3165–3171, Jun. 2022.

[6] X. Li, X. Yi, and L. Xie, "Distributed online convex optimization with an aggregative variable," *IEEE Trans. Control Netw. Syst.*, vol. 9, no. 1, pp. 438–449, Mar. 2022.

[7] T. Wang and P. Yi, "Distributed projection-free algorithm for constrained aggregative optimization," *Int. J. Robust Nonlinear Control*, vol. 33, no. 10, pp. 5273–5288, 2023.

[8] L. Chen, G. Wen, X. Fang, J. Zhou, and J. Cao, "Achieving linear convergence in distributed aggregative optimization over directed graphs," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 54, no. 7, pp. 4529–4541, Jul. 2024.

[9] Z. Wang, D. Wang, J. Lian, H. Ge, and W. Wang, "Momentum-based distributed gradient tracking algorithms for distributed aggregative optimization over unbalanced directed graphs," *Automatica*, vol. 164, 2024, Art. no. 111596.

[10] M. Chen, D. Wang, X. Wang, Z.-G. Wu, and W. Wang, "Distributed aggregative optimization via finite-time dynamic average consensus," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 6, pp. 3223–3231, Nov./Dec. 2023.

[11] L. Chen, G. Wen, H. Liu, W. Yu, and J. Cao, "Compressed gradient tracking algorithm for distributed aggregative optimization," *IEEE Trans. Autom. Control*, vol. 69, no. 10, pp. 6576–6591, Oct. 2024.
[12] G. Carnevale and G. Notarstefano, "A learning-based distributed algorithm for personalized aggregative optimization," in *Proc. IEEE 61st Conf. Decis. Control*, 2022, pp. 1576–1581.
[13] R. Brumali, G. Carnevale, and G. Notarstefano, "A deep learning approach for distributed aggregative optimization with users' feedback," in *Proc. 6th Annu. Learn. Dyn. Control Conf.*, 2024, pp. 1552–1564.
[14] X. Luo, Y. Zhang, and M. M. Zavlanos, "Socially-aware robot planning via bandit human feedback," in *Proc. ACM/IEEE 11th Int. Conf. Cyber-Phys. Syst.*, 2020, pp. 216–225.
[15] A. Simonetto, E. Dall'Anese, J. Monteil, and A. Bernstein, "Personalized optimization with user's feedback," *Automatica*, vol. 131, 2021, Art. no. 109767.
[16] F. Fabiani, A. Simonetto, and P. J. Goulart, "Learning equilibria with personalized incentives in a class of nonmonotone games," in *Proc. 2022 Eur. Control Conf.*, 2022, pp. 2179–2184.
[17] A. M. Ospina, A. Simonetto, and E. Dall'Anese, "Time-varying optimization of networked systems with human preferences," *IEEE Trans. Control Netw. Syst.*, vol. 10, no. 1, pp. 503–515, Mar. 2023.
[18] I. Notarnicola, A. Simonetto, F. Farina, and G. Notarstefano, "Distributed personalized gradient tracking with convex parametric models," *IEEE Trans. Autom. Control*, vol. 68, no. 1, pp. 588–595, Jan. 2023.
[19] S. Liu, P.-Y. Chen, B. Kailkhura, G. Zhang, A. O. Hero III, and P. K. Varshney, "A primer on zeroth-order optimization in signal processing and machine learning: Principals, recent advances, and applications," *IEEE Signal Process. Mag.*, vol. 37, no. 5, pp. 43–54, Sep. 2020.
[20] Y. Zhang, Y. Zhou, K. Ji, and M. M. Zavlanos, "A new one-point residual-feedback oracle for black-box learning and control," *Automatica*, vol. 136, 2022, Art. no. 110006.
[21] M. Fazel, R. Ge, S. Kakade, and M. Mesbahi, "Global convergence of policy gradient methods for the linear quadratic regulator," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1467–1476.
[22] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *Proc. 16th Annu. ACM-SIAM Symp. Discrete Algorithms*, pp. 385–394, 2005.
[23] A. Saha and A. Tewari, "Improved regret guarantees for online smooth convex optimization with bandit feedback," in *Proc. 14th Int. Conf. Artif. Intell. Statist., JMLR Workshop Conf. Proc.*, 2011, pp. 636–642.
[24] X. Chen, Y. Tang, and N. Li, "Improve single-point zeroth-order optimization using high-pass and low-pass filters," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 3603–3620.
[25] Q. Xiao, Q. Ling, and T. Chen, "Lazy queries can reduce variance in zeroth-order optimization," *IEEE Trans. Signal Process.*, vol. 71, pp. 3695–3709, 2023.
[26] Y. Zhang, Y. Zhou, K. Ji, Y. Shen, and M. M. Zavlanos, "Boosting one-point derivative-free online optimization via residual feedback," *IEEE Trans. Autom. Control*, vol. 69, no. 9, pp. 6309–6316, Sep. 2024.
[27] E. Mhanna and M. Assaad, "Zero-order one-point gradient estimate in consensus-based distributed stochastic optimization," *Trans. Mach. Learn. Res.*, 2024.
[28] L. Cothren, G. Bianchin, and E. Dall'Anese, "Online optimization of dynamical systems with deep learning perception," *IEEE Open J. Control Syst.*, vol. 1, pp. 306–321, 2022.
[29] L. Cothren, G. Bianchin, S. Dean, and E. Dall'Anese, "Perception-based sampled-data optimization of dynamical systems," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 5083–5088, 2023.
[30] M. Pirrone, E. Dall'Anese, and T. W. Barton, "Data-driven optimization strategies for tunable RF systems," *IEEE Trans. Microw. Theory Techn.*, vol. 72, no. 3, pp. 1919–1931, Mar. 2024.
[31] R. Schwan, C. N. Jones, and D. Kuhn, "Stability verification of neural network controllers using mixed-integer programming," *IEEE Trans. Autom. Control*, vol. 68, no. 12, pp. 7514–7529, Dec. 2023.
[32] L. Furieri, C. L. Galimberti, and G. Ferrari-Trecate, "Neural system level synthesis: Learning over all stabilizing policies for nonlinear systems," in *Proc. IEEE 61st Conf. Decis. Control*, 2022, pp. 2765–2770.
[33] A. Martin and L. Furieri, "Learning to optimize with convergence guarantees using nonlinear system theory," *IEEE Control Syst. Lett.*, vol. 8, pp. 1355–1360, 2024.
[34] F. Fabiani and P. J. Goulart, "Reliably-stabilizing piecewise-affine neural network controllers," *IEEE Trans. Autom. Control*, vol. 68, no. 9, pp. 5201–5215, Sep. 2023.
[35] D. Lupu and I. Necoara, "Exact representation and efficient approximations of linear model predictive control laws via HardTanh type deep neural networks," *Syst. Control Lett.*, vol. 186, 2024, Art. no. 105742.
[36] J. Nubert, J. Köhler, V. Berenz, F. Allgöwer, and S. Trimpe, "Safe and fast tracking on a robot manipulator: Robust MPC and neural network control," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3050–3057, Apr. 2020.
[37] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist., JMLR Workshop Conf. Proc.*, 2010, pp. 249–256.
[38] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *J. Mach. Learn. Res.*, vol. 18, pp. 1–43, 2018.
[39] E.-W. Bai, L.-C. Fu, and S. S. Sastry, "Averaging analysis for discrete time and sampled data adaptive systems," *IEEE Trans. Circuits Syst.*, vol. 35, no. 2, pp. 137–148, Feb. 1988.
[40] W. M. Haddad and V. Chellaboina, *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach*. Princeton, NJ, USA: Princeton Univ. Press, 2008.
[41] G. Carnevale, N. Bastianello, G. Notarstefano, and R. Carli, "ADMM-tracking gradient for distributed optimization over asynchronous and unreliable networks," *IEEE Trans. Autom. Control*, vol. 70, no. 8, pp. 5160–5175, Aug. 2025.

Riccardo Brumali (Graduate Student Member, IEEE) received the Laurea degree summa cum laude in automation engineering from the University of Bologna, Bologna, Italy, in 2023. He is currently working toward the Ph.D. degree within the Ph.D. Programme in biomedical, electrical, and systems engineering with the Department of Electrical, Electronic, and Information Engineering.

His research focuses on distributed optimization.

Guido Carnevale (Member, IEEE) received the M.Sc. degree "summa cum laude" in automation engineering and the Ph.D. degree in biomedical, electrical, and systems engineering from the University of Bologna, Bologna, Italy, in 2019 and 2023, respectively.

In 2022, he was a Visiting Scholar with the University of Oxford. He is a Junior Assistant Professor with the Department of Electrical, Electronic, and Information Engineering G. Marconi, Alma Mater Studiorum Università di Bologna. His research interests include distributed optimization and games over networks and optimal control.

Giuseppe Notarstefano (Member, IEEE) received the Laurea degree "summa cum laude" in electronics engineering from the Università di Pisa, Pisa, Italy, in 2003, and the Ph.D. degree in automation and operation research from the Università di Padova, Padua, Italy, in 2007.

He was an Assistant Professor, Ricercatore, from 2007, and Associate Professor from 2016 to 2018 with the Università del Salento, Lecce, Italy. He has been a Visiting Scholar with the University of Stuttgart, University of California Santa Barbara, and University of Colorado Boulder. He is a Professor with the Department of Electrical, Electronic, and Information Engineering G. Marconi, Alma Mater Studiorum Università di Bologna, Bologna, Italy. His research interests include distributed optimization, cooperative control in complex networks, applied nonlinear optimal control, and trajectory optimization and maneuvering of aerial and car vehicles.

Dr. Notarstefano was an Associate Editor for IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, and IEEE CONTROL SYSTEMS LETTERS. He has been part of the Conference Editorial Board of IEEE Control Systems Society and EUCA. He was the recipient of the ERC Starting Grant 2014.

Open Access funding provided by 'Alma Mater Studiorum - Università di Bologna' within the CRUI CARE Agreement