TTA-GT: TIME-TO-ACCURACY ORIENTED ADAPTIVE FIRST/SECOND-ORDER GRADIENT TRACKING FOR DECENTRALIZED TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Decentralized training of deep neural networks has emerged as a promising paradigm for scalable AI development, offering the potential to overcome the scalability limitations of conventional centralized training infrastructures. Decentralized training efficiency is often measured in terms of iteration count. This metric can mis-represent wall-clock efficiency when first-order (FO) and second-order (SO) directions have different runtime costs. We study time-to-accuracy (TTA) oriented direction selection in decentralized gradient tracking (GT). Each node chooses between a low-cost FO direction and a damped SO direction. We propose TTA-GT, a node level selector that estimates a local TTA score for each candidate. The score is the product of predicted remaining iterations and per round time, which decides whether the expected future iteration saving of SO justifies its current runtime cost. Prediction affects efficiency but not correctness, with convergence guaranteed by explicit direction constraints. Under standard strongly convex and smooth assumptions, we prove linear convergence for adaptive FO/SO direction sequences whose applied directions satisfy explicit alignment and norm conditions. Experiments on synthetic quadratic problems and logistic regression on real datasets show that TTA-GT can reduce end-to-end TTA under heterogeneous environments, where nodes vary communication conditions, such as bandwidth and latency, as well as different computation capabilities. In a 300-dimensional quadratic setting, TTA-GT reduces TTA from 2.714 to 1.934, reaching a 28.7% improvement. TTA-GT can achieve significant TTA reduction when SO directions offer sufficient iteration savings, while their additional computation cost remains non-negligible.

1 INTRODUCTION

Decentralized training has emerged as an effective paradigm for scaling deep neural network training across distributed computational resources. Unlike centralized approaches based on a parameter server, decentralized architectures enable nodes to exchange information through local communication with neighboring nodes. This communication structure has been studied extensively in decentralized SGD, gossip-based training, and recent communication-efficient systems for large models (Lian et al., 2017; Assran et al., 2019; Koloskova et al., 2020; Douillard et al., 2023; Qi et al., 2025; Kolehmainen et al., 2025; Kim & Lee, 2026). In decentralized training, the overall training efficiency is jointly determined by local computation, communication overhead, and the update directions applied at individual nodes. Different update directions can provide different levels of training progress while introducing different computation and communication costs.

For decentralized training to be effective in practical machine learning systems, iteration count alone is insufficient to measure its efficiency. In heterogeneous environment, training cost and progress can vary across nodes due to differences in computation capability, communication bandwidth, latency, and local data characteristics (Tang et al., 2018b; Lian et al., 2018; Ryabinin et al., 2021; Gong et al., 2022). Existing approaches have explored topology design and communication compression to reduce system overhead (Takezawa et al., 2023; Alistarh et al., 2017; Vogels et al., 2019; Tang et al., 2018a), but the tradeoff between the cost of local updates and the resulting training progress remains insufficiently explored. For example, a first-order update typically incurs lower computation

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

costs, but may require more rounds to reach a target accuracy, whereas a second-order update may achieve faster progress at a higher per round cost. Therefore, the choice of update direction directly affects the tradeoff between optimization progress and execution cost. Optimization progress is the reduction in the gap to the target accuracy from an update, while execution cost is the computation and communication time it requires. A direction that minimizes the number of iterations may not minimize the total training time. Therefore, it is important to consider adaptive direction selection based on time to accuracy (TTA), i.e., the wall-clock time required to reach a target accuracy, in decentralized training systems so as to achieve the tradeoff between training progress and execution cost.

The challenge in the adaptive selection mechanisms for decentralized training is to adapt update directions to the evolving computation and communication regime without compromising the convergence behavior of gradient tracking. The preferred direction depends on the runtime environment, including computation capability and communication conditions such as bandwidth and latency, and optimization state rather than being fixed a priori. Selecting directions based on predicted TTA also introduces uncertainty because a direction that appears favorable according to the prediction may be poorly aligned with the descent direction, and the prediction quality may vary over time.

In this paper, we provide a comprehensive study on the adaptive direction selection for decentralized training. First, we propose TTA-GT, a first- and second-order direction selection algorithm in decentralized gradient tracking as an online decision selection based on TTA that links local update choices to distributed execution cost. We then develop KT-predictor consisting of a K-predictor that estimates the remaining training steps required to reach the target accuracy and a T-predictor that estimates the per round execution time. Furthermore, under strongly convex and smooth objectives, we establish linear convergence when the applied update directions satisfy suitable acute-angle and length conditions. We validate our method on synthetic strongly convex quadratic problems, nonconvex problems and logistic regression tasks on real datasets.

2 Related Work

Decentralized training. Early decentralized SGD and gradient-push methods show that competitive convergence is possible without a central coordinator (Lian et al., 2017; Assran et al., 2019). Subsequent work studied how topology changes, local updates, and communication patterns affect training dynamics (Koloskova et al., 2020; Di et al., 2024; Liu et al., 2024). These works establish the convergence and communication foundations of decentralized training under prescribed local update rules. However, they do not focus on the selection of the local update directions, which is an important factor for decentralized training systems.

Communication-efficient training. Communication efficiency is critical for decentralized training. Existing approaches reduce communication costs through topology design, compression, and communication-computation overlap (Takezawa et al., 2023; Alistarh et al., 2017; Vogels et al., 2019; Tang et al., 2018a). Recent methods further improve efficiency through compression-based error control, local training, or communication-computation overlap (Gao et al., 2024; Condat et al., 2025; Sun et al., 2024; Wang et al., 2024). These methods mainly optimize the communication process through changes in topology, representation, or execution scheduling. They do not consider communication and computation characteristics when deciding which local optimization direction should be applied at each node. Such characteristics are critical in large-scale decentralized training, as heterogeneous computation and communication conditions can significantly affect end-to-end training efficiency. In this work, we incorporate runtime communication and computation information into the online selection of first- and second-order directions, enabling adaptive optimization choices that better reduce end-to-end training time.

Heterogeneity-aware and adaptive methods. System heterogeneity has motivated a broad class of methods that adapt decentralized or federated optimization to heterogeneous devices and execution conditions. Moshpit SGD and FedADMM explicitly account for device or system heterogeneity (Ryabinin et al., 2021; Gong et al., 2022), while DISH allows different agents to employ different update types depending on the system regime (Niu & Wei, 2022; 2024). Related decentralized and federated methods further accommodate heterogeneous computation and communication through mechanisms such as sporadic participation or client-side scheduling (Zehtabi et al., 2025). However,

these adaptations typically rely on predefined device classes, participation patterns, or scheduling policies and remain fixed during training. They lack an online mechanism to adjust optimization directions according to the changing system conditions and optimization states. TTA-GT addresses this limitation by dynamically selecting directions during optimization, enabling adaptive training in heterogeneous and time-varying environments with different communication conditions, including bandwidth and latency variations, and different computation capabilities across nodes.

3 Problem Setup

3.1 Decentralized Gradient Tracking Model

In decentralized training, a set of n nodes collaboratively trains a model using distributed data and local computation. Each node $i \in \{1, \dots, n\}$ stores a local dataset and maintains a local model copy $x_i^t \in \mathbb{R}^d$ at iteration t. The local model is updated using information from both the local data and neighboring nodes, while each node maintains a local training objective f_i. Nodes do not communicate with a central coordinator. Instead, they exchange information with neighboring nodes over a connected communication graph represented by a mixing matrix W. The goal of decentralized training is to achieve model consensus among nodes while minimizing the global training objective over all local datasets. This objective is written as

$$\min_{x \in \mathbb{R}^d} F(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x). \tag{1}$$

Each node updates its local model x_i^t through a gradient tracking procedure which maintains a local tracking variable $s_i^t \in \mathbb{R}^d$. Let $x^t = (x_1^t, \dots, x_n^t)$ and $s^t = (s_1^t, \dots, s_n^t)$ stack these variables, and let $g(x^t)$ stack the local gradients $\nabla f_i(x_i^t)$. The algorithms studied in this paper follow the gradient tracking recursion

$$x^{t+1} = \mathcal{W} x^t - d^t, \tag{2}$$

$$s^{t+1} = \mathcal{W} s^t + g(x^{t+1}) - g(x^t), \tag{3}$$

where $\mathcal{W} := W \otimes I_d$ and $d^t = (d_1^t, \dots, d_n^t)$ denotes the collection of local update directions.

3.2 Candidate Direction Set

Equation (2) leaves the node level direction d_i^t unspecified. Standard gradient tracking fixes this direction to a scaled tracking vector. In contrast, we use this degree of freedom to model a local algorithmic choice at each node. The choice is restricted to two candidates updates $d_{i,a}^t$ that are both computable from local information, where a denotes the candidate type, with $a = \mathrm{FO}$ for the first-order direction and $a = \mathrm{SO}$ for the second-order direction.

Each node accesses a first-order candidate

$$d_{i,\mathrm{FO}}^t = \eta_1^t \tilde{s}_i^t, \tag{4}$$

where η_1^t may be fixed or normalized by a local smoothness estimate. When a second-order candidate is evaluated, node i forms

$$d_{i,\mathrm{SO}}^t = \eta_2^t \left(\nabla^2 f_i(x_i^t) + \mu I \right)^{-1} \tilde{s}_i^t, \tag{5}$$

where $\tilde{s}^t = \mathcal{W} s^t$, $\mu \geq 0$ is a damping parameter that improves the numerical stability of the Hessian inverse. This direction is referred to as a damped Hessian. These two directions differ in the quantities that determine TTA. The first-order candidate incurs lower computation costs per round, but may require more rounds to reach a target accuracy. The second-order candidate may improve the local decrease or conditioning of the update, but it introduces additional computation and may change the round time, the wall-clock time required to complete one training round, including communication and local computation.

3.3 Runtime Model and TTA Metric

The two candidate directions affect TTA through both optimization progress and execution time. A direction that gives a larger decrease in one update may still be slower to reach a target accuracy if it increases the time of each update round. Therefore, we use TTA as the primary evaluation metric, which evaluates training efficiency by the cumulative wall-clock time required to reach a target accuracy. At round t, nodes exchange information with their neighbors, form local directions, and update their local models. We refer to this communication and update cycle as one synchronized round, and model its node level execution time as

$$T_{i,a}^t = T_{i,a,\mathrm{comm}}^t + T_{i,a,\mathrm{comp}}^t, \qquad a \in \{\mathrm{FO}, \mathrm{SO}\}. \tag{6}$$

We measure computation and communication times using wall-clock times on each node. Here $T_{i,a,\mathrm{comm}}^t$ is the time spent on neighbor communication in the mixing step, and $T_{i,a,\mathrm{comp}}^t$ is the local computation time for constructing candidate a at node i. Term $T_{i,a,\mathrm{comm}}^t$ may vary with active edges, message size, bandwidth, latency, jitter, and retransmission cost, while term $T_{i,a,\mathrm{comp}}^t$ may vary across nodes and candidate types. The TTA metric accumulates these costs until the target objective accuracy is first reached. Under this metric, the preferred direction is the one that gives the smaller time to reach the target, not necessarily the one with the largest single round decrease.

We consider adaptive direction selection where each node chooses between FO and SO directions using the currently available information during optimization progress. The objective is to minimize TTA under heterogeneous environments.

4 Method: TTA-GT

4.1 Local Progress and K/T Prediction

To address the online decision selection, we propose TTA-GT, a node level online direction selection algorithm built on the decentralized gradient tracking recursion in Equations (2)–(3). The abbreviated node level decision procedure is summarized in Algorithm 1, while the complete pseudocode is provided in Appendix A. In TTA-GT, each d_i^t is selected by node i through online evaluation of candidate directions based on their predicted TTA. This mechanism decouples gradient tracking from a fixed node level direction update rule and enables adaptive direction selection. At each selection evaluation, node i decides whether to apply the first- or second-order candidate using only quantities available before the next update. This decision is made by a KT-predictor with two components. The K-predictor estimates the relative number of residual reduction steps required by each candidate. The goal is not to predict the exact future trajectory, but to provide a ranking signal for direction selection. The T-predictor estimates the execution time of one step under the current node and communication conditions. The TTA score is the product of the predicted remaining iterations K and time per iteration T, i.e.,

$$S_{i,a}^k = \widehat{K}_{i,a}^k \widehat{T}_{i,a}^k. \tag{7}$$

Since each node cannot directly access the global objective gap in a decentralized setting, we construct a locally computable residual for candidate ranking. This residual combines the magnitude of the mixed tracking direction with the disagreement between the current local model and its neighbor average. These quantities are evaluated only when the selector is triggered, not at every optimization round. Let $k = 0, 1, \ldots$ index selector evaluations, and let t_k be the corresponding optimization round. Then we have

$$\tilde{x}_i^{t_k} = \sum_{j=1}^n W_{ij} x_j^{t_k}, \qquad e_i^k := x_i^{t_k} - \tilde{x}_i^{t_k}.$$

For node i, we define

$$r_i^k = \left\|\tilde{s}_i^{t_k}\right\|^2 + \lambda_c \left\|e_i^k\right\|^2, \tag{8}$$

where $\lambda_c \geq 0$ controls the weight assigned to local disagreement. This residual is not the global objective gap. It is used only for local comparison of candidate directions.

The K-predictor estimates the future number of residual reduction steps of each candidate direction. Let $H_i^k := \nabla^2 f_i(x_i^{t_k}) + \mu I$. Holding the current neighbor mixture fixed, we define the model

residual after one step as

$$\widehat{r}_{i,a,\text{model}}^{k+1} = \left\|\tilde{s}_i^{t_k} - H_i^k d_{i,a}^{t_k}\right\|^2 + \lambda_c \left\|e_i^k - d_{i,a}^{t_k}\right\|^2. \tag{9}$$

Using a first-order Taylor approximation of the tracking signal around the current iterate, the post-step tracking term is approximated by $\tilde{s}_i^t - H_i^t d_{i,a}^t$. Holding the current neighbor mixture fixed, the corresponding disagreement term is $e_i^t - d_{i,a}^t$. It is not an exact prediction of the next decentralized state. The corresponding model ratio is the predicted residual divided by the current residual ,i.e,

$$q_{i,a,\text{model}}^k = \frac{\widehat{r}_{i,a,\text{model}}^{k+1}}{\max\{r_i^k, \epsilon_r\}}, \tag{10}$$

where ϵ_r is a sufficiently small positive number to avoid division by zero and $q_{i,a,\text{model}}^k < 1$ indicates that candidate a is predicted to contract the local residual, whereas values near or above one indicate little local progress or possible worsening.

We smooth this ratio with the previous estimate for the same candidate,

$$\widehat{q}_{i,a}^k = \begin{cases} \beta_q \widehat{q}_{i,a}^{k-1} + (1-\beta_q) q_{i,a,\text{model}}^k, & \text{if a previous estimate is available,} \\ q_{i,a,\text{model}}^k, & \text{otherwise,} \end{cases} \tag{11}$$

where the smoothing coefficient $\beta_q < 1$ controls the stability-responsiveness trade-off. Larger values give more stable estimates but slower adaptation, while smaller values react faster but are more sensitive to noise. This prevents the selector from reacting to transient residual fluctuations caused by stochastic gradients, communication variability, or asynchronous neighbor states.

Given the estimated contraction ratio, the predicted remaining number of local residual reductions is

$$\widehat{K}_{i,a}^k = \begin{cases} 0, & r_i^k \le r_{\text{tar}}, \\ K_{\max}, & \widehat{q}_{i,a}^k \le 0 \text{ or } \widehat{q}_{i,a}^k \ge 1, \\ \min\left\{K_{\max}, \left\lceil \dfrac{\log(r_{\text{tar}}/r_i^k)}{\log \widehat{q}_{i,a}^k} \right\rceil\right\}, & 0 < \widehat{q}_{i,a}^k < 1, \end{cases} \tag{12}$$

where r_{tar} is the target residual. The estimate $\widehat{K}_{i,a}^k$ is a local ranking component, since the residual in Equation (8) is local and future nodes may change their modes.

The T-predictor estimates the per step execution time of each candidate direction at node i. We define it as

$$\widehat{T}_{i,a}^k = \widehat{T}_{i,a,\text{comm}}^k + \widehat{T}_{i,a,\text{comp}}^k, \qquad \widehat{T}_{i,a,\text{comp}}^k \leftarrow \beta_T \widehat{T}_{i,a,\text{comp}}^{k-1} + (1-\beta_T)\tau_{i,a}^k. \tag{13}$$

Here $\widehat{T}_{i,a,\text{comm}}^k$ denotes the estimated value of $T_{i,a,\text{comm}}^k$, while $\widehat{T}_{i,a,\text{comp}}^k$ is the smoothed estimate of $T_{i,a,\text{comp}}^k$. Term $\tau_{i,a}^k$ is the measured computation time of candidate a at the k-th evaluation, and $\beta_T \in [0, 1)$ is the smoothing coefficient. When the first timing sample becomes available, we initialize $\widehat{T}_{i,a,\text{comp}}^k = \tau_{i,a}^k$. If a timing estimate is unavailable, the corresponding score is set to $+\infty$ until a warm-up or probe sample is collected.

4.2 Validity-Constrained FO/SO Selection

The TTA score (7) ranks candidates according to predicted efficiency, but it does not guarantee that a candidate update preserves the descent geometry required by gradient tracking. Therefore, we introduce validity constraints before allowing the second-order candidate to participate in TTA selection. The resulting update may point away from the mixed tracking direction, or its magnitude may be too large for the current iterate. We therefore constrain the second-order candidate before using it in the TTA comparison. The candidate must have a well-defined descent geometry, retain a sufficient component along the tracked direction, and have a bounded step length. It enters the score comparison only when it satisfies

$$\nabla^2 f_i(x_i^{t_k}) + \mu I \succ 0, \tag{14}$$

$$\left\langle d_{i,\text{SO}}^{t_k}, \tilde{s}_i^{t_k} \right\rangle \ge \theta_- \left\|\tilde{s}_i^{t_k}\right\|^2, \tag{15}$$

$$\left\| d_{i,\text{SO}}^{t_k} \right\| \le \theta_+ \left\|\tilde{s}_i^{t_k}\right\|. \tag{16}$$

Equation (14) requires the damped Hessian to be positive definite. Geometrically, its local quadratic model has positive curvature in every direction, so the inverse in Equation (5) is well defined and does not reverse a direction because of a negative eigenvalue. Under the strongly convex assumptions, this condition is automatic for $\mu \geq 0$, but it is retained for numerical robustness and for approximate curvature estimates.

Equation (15) controls the projection of the second-order direction onto the mixed tracking direction. Specifically, $\left\langle d_{i,\mathrm{SO}}^{t_k}, \tilde{s}_i^{t_k} \right\rangle / \left\| \tilde{s}_i^{t_k} \right\|$ is the signed projection of $d_{i,\mathrm{SO}}^{t_k}$ onto $\tilde{s}_i^{t_k}$. The lower bound requires this projection to be at least $\theta_- \left\| \tilde{s}_i^{t_k} \right\|$, which prevents the update from becoming orthogonal or oppositely oriented to the direction used by gradient tracking.

Equation (16) bounds the length of the second-order direction by $\theta_+ \left\| \tilde{s}_i^{t_k} \right\|$. It limits the radius of the admissible update region and prevents a curvature estimate from producing an excessively large local move. Together, the last two conditions restrict the candidate to a bounded region with a sufficient component along $\tilde{s}_i^{t_k}$, so the TTA rule can adapt the direction without accepting an update that is geometrically inconsistent with gradient tracking. If the second-order candidate violates any of these conditions, the node falls back to the first-order direction.

After these checks, node i allows the second-order candidate to enter the mode only if its predicted score to the local residual target improves on the first-order candidate by a margin, i.e.,

$$S_{i,\mathrm{SO}}^k < (1-\delta) S_{i,\mathrm{FO}}^k, \tag{17}$$

where $\delta > 0$ reduces switching caused by noisy predictions. The operational use of this margin is described by the node level state mechanism below.

4.3 Node Level Online Selection Mechanism

The score in Equation (7) enables decentralized direction selection using each node's residual, candidate timings, and communication estimates. Each node maintains a persistent mode $m_i^t \in \{\mathrm{FO}, \mathrm{SO}\}$, which denotes the update mode used in round t at node i, as well as the last switch time $\tau_{i,\mathrm{sw}}$, the residual rate estimates $\widehat{r}_{i,a,\mathrm{model}}^{k+1}$, and timing estimates $\widehat{T}_{i,a}^k$. Let $H, P \in \mathbb{N}$ and $R \in \mathbb{N}_0$ be the number of warm-up rounds, the probe interval and the cooldown length, respectively. During warm-up rounds $t = 0, \ldots, H-1$, each node evaluates and times both candidates to initialize their timing and residual rate estimates, while the applied mode remains FO. After warm-up, a probe is performed at rounds satisfying $t \geq H$ and $(t-H) \bmod P = 0$. A probe refreshes the validity, timing, and residual rate estimates of the second-order candidate even when the current mode is FO, allowing the node to reassess the benefit of switching. Cooldown allows a mode change only when $t - \tau_{i,\mathrm{sw}} \geq R$. When a mode change occurs, $\tau_{i,\mathrm{sw}}$ is updated to the current round t. Otherwise, it remains unchanged. The elapsed time of warm-up and probe rounds is counted in the same TTA accumulator as ordinary rounds.

With these mechanisms, node i selects or maintains SO mode only when the second-order candidate is valid and satisfies the margin condition in Equation (17). Once cooldown has elapsed, violation of either requirement causes the node to switch to FO, so switching is allowed in both directions. If cooldown has not elapsed, the current mode is retained. The effects and practical choices of these hyperparameters are examined in Appendix D.

5 Convergence Analysis

Unlike conventional gradient tracking analyses that assume a fixed update direction, TTA-GT generates a time-varying and node-dependent direction sequence. Therefore, it is important to analyze whether adaptive direction switching preserves the convergence behavior of gradient tracking. The analysis only uses two properties of the applied directions, i.e., sufficient alignment with the mixed tracking variable and a uniform length bound. Thus, the result covers both fixed first-order updates and adaptive first/second-order switching whenever the actually applied directions satisfy the same direct thresholds.

Assumption 1 (Local strong convexity and smoothness)**.** *For each node i, the local objective f_i is m_i-strongly convex and ℓ_i-smooth, i.e.,*

$$m_i I \preceq \nabla^2 f_i(x) \preceq \ell_i I \qquad \textit{for all } x.$$

Algorithm 1 Node level online TTA-GT update at node i.

Input: Initial states x_i^0, $s_i^0 = \nabla f_i(x_i^0)$, mixing matrix W, probe round P, parameters.
Output: Local models x_i^T.
Initialize $m_i^0 \leftarrow \mathrm{FO}$.
$\tilde{x}_i^t = \sum_j W_{ij} x_j^t$, $\tilde{s}_i^t = \sum_j W_{ij} s_j^t$ and $d_{i,\mathrm{FO}}^t = \eta_1^t \tilde{s}_i^t$.
for $t = 0, \ldots, T-1$ **do**
  if $t \mod P == 0$ **then**
    Compute mixed variables: $\tilde{x}_i^t = \sum_j W_{ij} x_j^t, \quad \tilde{s}_i^t = \sum_j W_{ij} s_j^t$
    Generate candidate directions: $d_{i,\mathrm{FO}}^t = \eta_1^t \tilde{s}_i^t$, $d_{i,\mathrm{SO}}^t = \eta_2^t (\nabla^2 f_i(x_i^t) + \mu I)^{-1} \tilde{s}_i^t$
    Estimate TTA scores: $S_{i,a} = K_{i,a} T_{i,a}, \quad a \in \{\mathrm{FO}, \mathrm{SO}\}$
    if $d_{i,\mathrm{SO}}^t$ satisfies validity constraints and $S_{i,\mathrm{SO}} < (1-\delta) S_{i,\mathrm{FO}}$ **then**
      $m_i^{t+1} \leftarrow \mathrm{SO}$
    else
      $m_i^{t+1} \leftarrow \mathrm{FO}$
    end if
    Apply selected direction: $d_i^t = d_{i,m_i^{t+1}}^t$
  end if
  Update: $x_i^{t+1} = \sum_j W_{ij} x_j^t - d_i^t$ $s_i^{t+1} = \sum_j W_{ij} s_j^t + \nabla f_i(x_i^{t+1}) - \nabla f_i(x_i^t)$
end for

Assumption 2 (Communication matrix)**.** *The matrix W is symmetric and doubly stochastic, and the communication graph is connected.*

Let $J := \frac{1}{n} \mathbf{1}\mathbf{1}^\top \otimes I_d$, denotes the orthogonal projection onto the consensus subspace of the stacked node variables. Equivalently, $\sigma := \|\mathcal{W} - J\|_2 < 1$. The condition $\sigma < 1$ is the mixing property used to contract disagreement after each communication step.

Assumption 3 (Accepted direction conditions)**.** *There exist constants $0 < \theta_- \leq \theta_+ < +\infty$ such that every accepted direction satisfies*

$$\left\langle d_i^t, \tilde{s}_i^t \right\rangle \geq \theta_- \left\| \tilde{s}_i^t \right\|^2, \tag{18}$$

$$\left\| d_i^t \right\| \leq \theta_+ \left\| \tilde{s}_i^t \right\|, \tag{19}$$

for all i and t.

Equation (18) has a direct geometric meaning. The quantity $\langle d_i^t, \tilde{s}_i^t \rangle / \|\tilde{s}_i^t\|$ is the signed projection of the applied direction onto the mixed tracking direction. The inequality requires this projection to be at least $\theta_- \|\tilde{s}_i^t\|$, so the update has a sufficiently large component in the descent direction used by gradient tracking. Equation (19) bounds the total length of the applied direction by $\theta_+ \|\tilde{s}_i^t\|$. Thus, the direction cannot be arbitrarily long relative to the tracked direction. Together, the two conditions restrict d_i^t to a bounded, descent aligned region around $\tilde{s}_i^t$. For first-order directions, the accepted direction conditions can be guaranteed by choosing an appropriate learning rate. Since a first-order update is a positive scaling of $\tilde{s}_i^t$, the angle condition holds naturally. In particular, when $\tilde{s}_i^t \neq 0$, Assumption 3 implies

$$\cos \angle(d_i^t, \tilde{s}_i^t) = \frac{\langle d_i^t, \tilde{s}_i^t \rangle}{\|d_i^t\| \, \|\tilde{s}_i^t\|} \geq \frac{\theta_-}{\theta_+},$$

so the applied direction forms a uniformly acute angle with $\tilde{s}_i^t$.

Theorem 1 (Linear convergence of Algorithm 1)**.** *Under Assumptions 1–3 and the parameter conditions in Appendix F, the iterates generated by Algorithm 1 converge linearly with rate ρ, where $m = \frac{1}{n} \sum_i m_i$,*

$$\rho = 1 - \frac{m\theta_-}{8} \in (0, 1).$$

Algorithm 1 shows that adaptive direction selection preserves the linear convergence of gradient tracking as long as the selected directions satisfy the accepted direction conditions. Existing decentralized gradient tracking methods achieve linear convergence, $\|x^t - x^\star\| \leq C \rho_{GT}^t$, where

$\rho_{GT} \in (0, 1)$ depends on the strong convexity parameter, smoothness constant, and network connectivity. Theorem 1 shows TTA-GT preserves the same linear convergence. This result allows nodes to switch between first- and second-order directions without requiring a fixed update rule. This flexibility is important in heterogeneous decentralized training environment, where local curvature, computation cost, communication bottlenecks, and optimization state can vary across nodes.

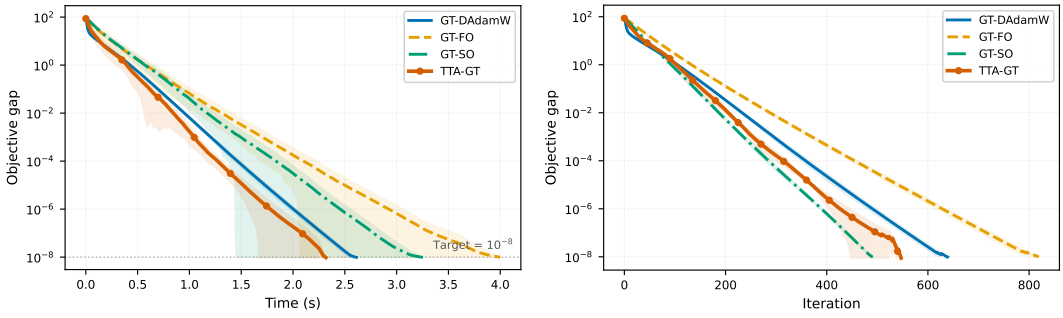
6 EXPERIMENTS

We evaluate the effectiveness of TTA selection framework for reducing end-to-end training time in decentralized gradient tracking. We conduct both strongly convex and nonconvex experiments with heterogeneous computation and communication conditions. The experiments consider practical sources of runtime variation, including communication heterogeneity caused by bandwidth and latency, computation heterogeneity arising from unequal processing capabilities across nodes, and different problem dimensions.

We compare with fixed direction policies to isolate the benefit of adaptive selection. GT-FO and GT-SO are used in the strongly convex experiments, while GT-FO and GT-DAdamW are used in the nonconvex experiments. The comparison methods are related to standard gradient tracking methods (Nedić et al., 2017; Qu & Li, 2018), distributed Newton type methods, hybrid first- and second-order methods (Shamir et al., 2014; Wang et al., 2018; Niu & Wei, 2024; 2022), and decentralized AdamW (Loshchilov & Hutter, 2019).

6.1 STRONGLY CONVEX EXPERIMENTS

We first evaluate TTA-GT on synthetic strongly convex quadratic objectives. Since $x^\star$ is unique and can be calculated, the target is defined by the exact objective gap. We consider different network topologies, problem dimensions (d), and heterogeneous computation and communication settings to examine TTA. The complete configuration list and additional results are provided in Appendix B.



(a) Objective gap vs. time (s).

(b) Objective gap vs. iteration.

Figure 1: Objective gap against TTA and iteration count in the homogeneous ring topology and $d = 100$ setting.

As shown in Figure 1, TTA-GT reaches the target in 1.934s with 569.2 iterations, while GT-SO requires 2.714s despite converging with 510.2 iterations. The result shows that a method may require more rounds while still reaching the target earlier in time.

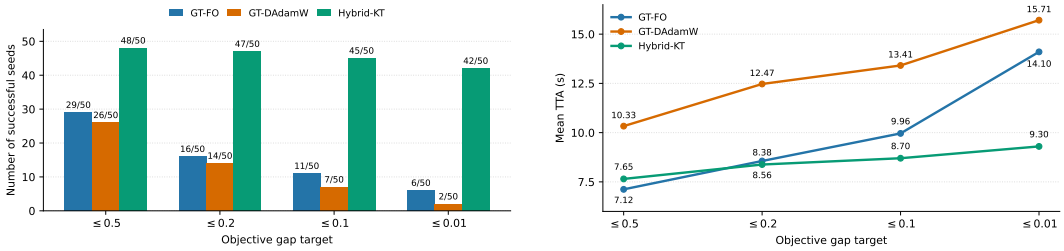
Therefore, compared with the iteration counts, TTA is a more accurate metric that captures the tradeoff between optimization progress and the execution cost. Table 1 reports the end-to-end TTA under representative settings that vary dimension and system cost structure. In all experiments,TTA-GT outperforms the others. When $d = 300$, TTA-GT reduces TTA from 2.714s (GT-SO) to 1.934s, reaching a 28.7% reduction. Under computational heterogeneity, TTA-GT reduces TTA from 2.517s to 2.143s. Conversely, Additional results ($d = 20$) in Appendix B show that low dimensional setting provides a boundary case in which GT-SO (0.504s) is faster than TTA-GT (0.554s) because additional second-order computation cost is small.

Table 1: TTA comparison under strongly convex quadratic objectives ($\epsilon = 10^{-8}$)

Setting	GT-DAdamW	GT-FO	GT-SO	TTA-GT
Ring, $d = 100$, homogeneous	2.570	3.281	2.932	**1.949**
Ring, $d = 100$, link bandwidth heterogeneity	13.196	16.859	10.670	**10.139**
Ring, $d = 100$, computational heterogeneity	3.082	3.937	2.517	**2.143**
Ring, $d = 300$, homogeneous	6.786	8.626	2.714	**1.934**

6.2 Nonconvex Experiments[1]

We further extend TTA-based selection framework to Hybrid-KT, where GT-FO and GT-DAdamW are used as two candidate update rules, to examine whether the framework remains effective without explicit second-order information. We evaluate it under a heterogeneous computation and communication setting with an 8-node two-cluster network topology.



(a) Target attainment rate vs. objective gap target

(b) Mean TTA vs. objective gap target

Figure 2: Nonconvex target attainment and TTA comparison.

Figure 2a compares the number of runs that reach different objective gap targets within the fixed budget. Hybrid-KT reaches 48/50, 47/50, 45/50, and 42/50 successful runs for increasingly strict objective gap targets ($\leq$0.5, $\leq$0.2, $\leq$0.1, and $\leq$0.01), substantially exceeding GT-FO and GT-DAdamW, whose success counts decrease to 6/50 and 2/50 at the strictest target, respectively. This indicates that adaptive direction selection is more effective when sustained optimization progress is required. Figure 2b further shows the aggregated TTA over all runs, including failed runs through timeout penalties. Hybrid-KT achieves competitive TTA at loose targets and provides larger improvements for stricter targets. When the objective gap thresholds are 0.2, 0.1, and 0.01, Hybrid-KT achieves mean TTA values of 8.38s, 8.70s, and 9.30s, respectively.

In the nonconvex setting, different update directions may exhibit different behaviors across training stages. The adaptive switching mechanism allows Hybrid-KT to adjust its update strategy according to the current optimization state, potentially avoiding stagnation caused by a fixed update rule and improving the probability of reaching stricter objective targets.

7 Conclusions

This paper studies the adaptive direction selection in decentralized gradient tracking from the perspective of TTA. We show that a direction requiring fewer iterations to reach a target accuracy does not necessarily achieve a shorter TTA, since its additional computation or communication overhead may dominate the overall training cost. We proposed TTA-GT, a node level adaptive framework that selects directions according to their predicted TTA while maintaining convergence through explicit direction constraints. Experiments demonstrate that the optimal update strategy is not determined solely by convergence speed. Instead, the preferred direction depends on the interaction between optimization progress and execution costs, and TTA-GT effectively adapts this tradeoff under heterogeneous decentralized training environment.

[1]This experiment is intended as an empirical assessment and does not extend the theoretical convergence guarantee to nonconvex objectives.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and grammar checking. We have not used generative AI tools for generating scientific claims, mathematical proofs, experimental results, or algorithmic contributions. Additionally, we used generative AI tools for minor writing improvements and organizing parts of the manuscript. All AI-assisted edits were carefully reviewed and revised by the authors. The authors verified the correctness of all technical statements, mathematical derivations, experimental descriptions, and references independently. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, volume 30, pp. 1709–1720, 2017.

Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Mike Rabbat. Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning*, pp. 344–353, 2019.

Laurent Condat, Artavazd Maranjyan, and Peter Richtárik. LoCoDL: Communication-efficient distributed learning with local training and compression. In *International Conference on Learning Representations*, 2025.

Hao Di, Haishan Ye, Xiangyu Chang, Guang Dai, and Ivor Tsang. Double stochasticity gazes faster: Snap-shot decentralized stochastic gradient tracking methods. In *International Conference on Machine Learning*, pp. 10765–10791, 2024.

Arthur Douillard, Qixuan Feng, Andrei A. Rusu, Rachita Chhaparia, Yani Donchev, Adhiguna Kuncoro, Marc'Aurelio Ranzato, Arthur Szlam, and Jiajun Shen. DiLoCo: Distributed low-communication training of language models, 2023. arXiv preprint arXiv:2311.08105.

Yuan Gao, Rustem Islamov, and Sebastian Stich. EControl: Fast distributed optimization with compression and error control. In *International Conference on Learning Representations*, 2024.

Yonghai Gong, Yichuan Li, and Nikolaos M. Freris. FedADMM: A robust federated deep learning framework with adaptivity to system heterogeneity. In *IEEE International Conference on Data Engineering*, pp. 2575–2587, 2022.

Jihun Kim and Namhoon Lee. SeedFlood: A step toward scalable decentralized training of LLMs, 2026. arXiv preprint arXiv:2602.18181.

Jari Kolehmainen, Nikolay Blagoev, John Donaghy, Oğuzhan Ersoy, and Christopher Nies. NoLoCo: No-all-reduce low communication training method for large models, 2025. arXiv preprint arXiv:2506.10911.

Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U. Stich. A unified theory of decentralized SGD with changing topology and local updates. In *International Conference on Machine Learning*, pp. 5381–5393, 2020.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5330–5340, 2017.

Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous decentralized parallel stochastic gradient descent. In *International Conference on Machine Learning*, pp. 3043–3052, 2018.

Yuxing Liu, Lesi Chen, and Luo Luo. Decentralized convex finite-sum optimization with better dependence on condition numbers. In *International Conference on Machine Learning*, pp. 30807–30841, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Angelia Nedić, Alex Olshevsky, and Wei Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633, 2017.

Xiaochun Niu and Ermin Wei. DISH: A distributed hybrid primal-dual optimization framework to utilize system heterogeneity. In *IEEE Conference on Decision and Control*, pp. 6503–6510, 2022.

Xiaochun Niu and Ermin Wei. DISH: A distributed hybrid optimization method leveraging system heterogeneity. *IEEE Transactions on Signal Processing*, 72:4007–4021, 2024.

Ji Qi, WenPeng Zhu, Li Li, Ming Wu, YingJun Wu, Wu He, Xun Gao, Jason Zeng, and Michael Heinrich. DiLoCoX: A low-communication large-scale training framework for decentralized cluster, 2025. arXiv preprint arXiv:2506.21263.

Guannan Qu and Na Li. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3):1245–1260, 2018.

Max Ryabinin, Eduard Gorbunov, Vsevolod Plokhotnyuk, and Gennady Pekhimenko. Moshpit SGD: Communication-efficient decentralized training on heterogeneous unreliable devices. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18195–18211, 2021.

Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate Newton-type method. In *International Conference on Machine Learning*, pp. 1000–1008, 2014.

Weigao Sun, Zhen Qin, Weixuan Sun, Shidi Li, Dong Li, Xuyang Shen, Yu Qiao, and Yiran Zhong. CO2: Efficient distributed training with full communication-computation overlap. In *International Conference on Learning Representations*, 2024.

Yuki Takezawa, Ryoma Sato, Han Bao, Kenta Niwa, and Makoto Yamada. Beyond exponential graph: Communication-efficient topologies for decentralized learning via finite-time convergence. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. Communication compression for decentralized training. In *Advances in Neural Information Processing Systems*, volume 31, pp. 7663–7673, 2018a.

Hanlin Tang, Xiangru Lian, Ming Yan, Ce Zhang, and Ji Liu. D^2: Decentralized training over decentralized data. In *International Conference on Machine Learning*, pp. 4848–4856, 2018b.

Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. PowerSGD: Practical low-rank gradient compression for distributed optimization. In *Advances in Neural Information Processing Systems*, volume 32, pp. 14236–14245, 2019.

Guanhua Wang, Heyang Qin, Sam Ade Jacobs, Connor Holmes, Samyam Rajbhandari, Olatunji Ruwase, Feng Yan, Lei Yang, and Yuxiong He. ZeRO++: Extremely efficient collective communication for giant model training. In *International Conference on Learning Representations*, 2024.

Shusen Wang, Farbod Roosta-Khorasani, Peng Xu, and Michael W. Mahoney. GIANT: Globally improved approximate Newton method for distributed optimization. In *Advances in Neural Information Processing Systems*, volume 31, pp. 2338–2348, 2018.

Shahryar Zehtabi, Dong-Jun Han, Rohit Parasnis, Seyyedali Hosseinalipour, and Christopher Brinton. Decentralized sporadic federated learning: A unified algorithmic framework with convergence guarantees. In *International Conference on Learning Representations*, 2025.

A Complete TTA-GT Pseudocode

This section gives the full node level TTA-GT procedure, including predictor initialization, warm-up and probe rounds, second-order validity checks, cooldown, and mode switching logic.

Algorithm 1 Complete node level online TTA-GT update at node i.

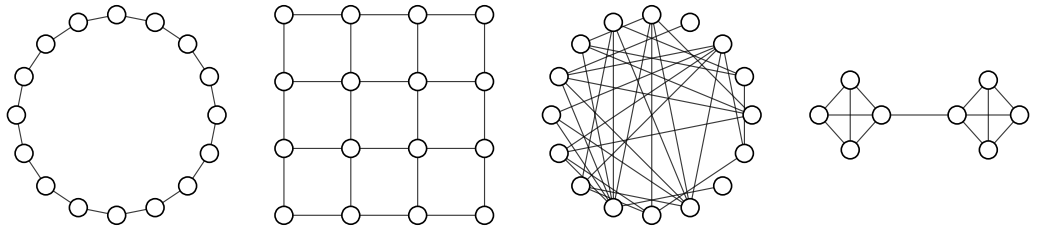
Input: Initial states x_i^0, $s_i^0 = \nabla f_i(x_i^0)$, mixing matrix W; step sizes η_1^t, η_2^t; regularization μ; thresholds θ_-, θ_+; warm-up length H; probe interval P; cooldown R; margin δ; and total number of rounds T.
Output: Final local models $\{x_i^T\}_{i=1}^n$, selected modes $\{m_i^t\}$, and accumulated TTA.
Initialize $m_i^0 \leftarrow \mathrm{FO}$, $\tau_{i,\mathrm{sw}} \leftarrow 0$, and all timing and residual-rate estimates as unavailable.
$\tilde{x}_i^t = \sum_j W_{ij} x_j^t$, $\tilde{s}_i^t = \sum_j W_{ij} s_j^t$ and $d_{i,\mathrm{FO}}^t = \eta_1^t \tilde{s}_i^t$.
A round is warm-up if $t < H$ and is a probe if $t \geq H$ and $(t - H) \bmod P = 0$.
if t is a warm-up or probe round, or $m_i^t = \mathrm{SO}$ **then**
    $d_{i,\mathrm{SO}}^t = \eta_2^t (\nabla^2 f_i(x_i^t) + \mu I)^{-1} \tilde{s}_i^t$ and measure its time.
end if
Compute $S_{i,a}^k = \widehat{K}_{i,a}^k \widehat{T}_{i,a}^k$, $a \in \{\mathrm{FO}, \mathrm{SO}\}$.
if SO is evaluated and either $\nabla^2 f_i(x_i^t) + \mu I \not\succ 0$, $\langle d_{i,\mathrm{SO}}^t, \tilde{s}_i^t \rangle < \theta_- \|\tilde{s}_i^t\|^2$, or $\|d_{i,\mathrm{SO}}^t\| > \theta_+ \|\tilde{s}_i^t\|$ **then**
    $v_{i,\mathrm{SO}}^t \leftarrow$ false and $S_{i,\mathrm{SO}}^k \leftarrow +\infty$.
end if
if $t < H$ **then**
    $\widehat{m}_i^{t+1} \leftarrow \mathrm{FO}$.
else if $t - \tau_{i,\mathrm{sw}} < R$ **then**
    $\widehat{m}_i^{t+1} \leftarrow m_i^t$.
else if $v_{i,\mathrm{SO}}^t$ and $S_{i,\mathrm{SO}}^k < (1 - \delta) S_{i,\mathrm{FO}}^k$ **then**
    $\widehat{m}_i^{t+1} \leftarrow \mathrm{SO}$.
else
    $\widehat{m}_i^{t+1} \leftarrow \mathrm{FO}$.
end if
if $\widehat{m}_i^{t+1} \neq m_i^t$ **then**
    $m_i^{t+1} \leftarrow \widehat{m}_i^{t+1}$ and $\tau_{i,\mathrm{sw}} \leftarrow t$.
else
    $m_i^{t+1} \leftarrow m_i^t$ and $\tau_{i,\mathrm{sw}}$ remains unchanged.
end if
$d_i^t = d_{i,m_i^{t+1}}^t$, $x_i^{t+1} = \sum_j W_{ij} x_j^t - d_i^t$, $s_i^{t+1} = \sum_j W_{ij} s_j^t + \nabla f_i(x_i^{t+1}) - \nabla f_i(x_i^t)$.

B Strongly Convex Experimental Details

The strongly convex experiments use synthetic quadratic objectives with known optimizer $x^\star$. Each configuration has 16 nodes and dimension $d = 100$, and is evaluated on five seeds. We vary the communication topology among ring, grid, and Erdős–Rényi graphs (Figure 3a-3c), the dimension among 20, 100, and 300, and the system conditions across three heterogeneity profiles. H1 varies link bandwidth, H2 varies node compute speed, and H3 varies both. Within each seed, all methods share the same objective, topology, initialization, bandwidth realization, and compute multipliers.

The target is the first iterate satisfying $F(\bar{x}^t) - F(x^\star) \leq 10^{-8}$. TTA is accumulated along the execution time. All methods use the same timing procedure. Hyperparameters are fixed across the five seeds, and the full iteration and computation/communication decomposition is retained in the underlying experiment records.

As shown in Figure 2, TTA-GT and GT-SO in H1 require nearly the same number of rounds (489.6 versus 488.6), but TTA-GT reduces computation time from 3.338 s to 2.793 s by using a lower-cost mixture of first- and second-order updates. Communication time increases slightly from 7.333 s to 7.347 s, while total TTA decreases from 10.670 s to 10.139 s, a 4.97% improvement. In H2, where



(a) Ring (b) Grid (c) Erdős–Rényi (d) Two-cluster bottleneck

Figure 3: Communication topology used in the experiments.

Table 2: Complete TTA comparison under strongly convex quadratic objectives($\epsilon = 10^{-8}$)

Setting	GT-DAdamW	GT-FO	GT-SO	TTA-GT
Ring, $d = 100$, H0	2.570	3.281	2.932	**1.949**
Grid, $d = 100$, H0	3.071	3.783	2.255	**2.099**
Erdős–Rényi, H0	3.765	3.758	2.427	**2.259**
Ring, $d = 100$, H1	13.196	16.859	10.670	**10.139**
Ring, $d = 100$, H2	3.082	3.937	2.517	**2.143**
Ring, $d = 100$, H3	13.884	17.738	13.908	**13.608**
Ring, $d = 20$, H0	1.849	2.126	**0.504**	0.554
Ring, $d = 300$, H0	6.786	8.626	2.714	**1.934**

node compute speeds vary, TTA-GT reduces TTA from 2.517 s for GT-SO to 2.143 s, corresponding to a 14.9% improvement.

H3 exhibits the same tradeoff with a smaller gain. TTA-GT requires 7.2 more rounds than GT-SO, while reducing computation time from 3.900 s to 3.453 s. Communication time increases from 10.008 s to 10.155 s, resulting in a total TTA reduction from 13.908 s to 13.608 s (2.16%).

C Nonconvex Experimental Details

We use the following nonconvex synthetic objective,

$$F(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x), \qquad f_i(x) = \frac{1}{2}\|x - c_i\|^2 + r\sum_{j=1}^{d}[1 - \cos(s_{ij}x_j + \phi_{ij})] + \sum_{j=1}^{d} a_{ij}x_j^4,$$

with $n = 8$, $d = 100$, cluster gap $g = 4$, heterogeneity $h = 3$, and nonconvex perturbation strength $r = 3$. The node centers are generated as

$$c_i = \sigma_i g u + \epsilon_i, \qquad \sigma_i \in \{-1, +1\}, \qquad \epsilon_i \sim \mathcal{N}(0, h^2 I).$$

The communication graph is a two-cluster bottleneck topology (Figure 3d) with one inter-cluster bridge. Communication occurs every eight local steps with 5 Mbps bandwidth and 20 ms latency. We use 10 validation seeds and 50 test seeds. The maximum nominal budget is 400 steps.

GT-FO uses $\eta_{\mathrm{FO}} = 0.01$, GT-DAdamW uses $\eta_{\mathrm{AdamW}} = 0.01$, and Hybrid-KT uses $\eta_{\mathrm{FO}} = 0.005$ and $\eta_{\mathrm{AdamW}} = 0.01$. The Hybrid-KT selector uses margin 0.1 with zero exit margin and zero cooldown. All configurations are selected independently on the validation seeds.

Objective gaps are computed relative to the lowest objective value among trajectories with the same seed. We evaluate target attainment over 50 seeds and report the mean TTA over all runs. A run that reaches the target within the initial 400-step budget uses its observed reaching time. For runs that do not reach the target within 400 steps, we continue the run for an additional 400 steps. If the target is reached during this extension, the actual reaching time is used. Otherwise, the run is considered unsuccessful and assigned the full 800-step runtime in the aggregate TTA calculation.

Table 3: Nonconvex high accuracy results on 50 seeds.

	Objective gap target			
Method	≤ 0.5	≤ 0.2	≤ 0.1	≤ 0.01
Runs reaching target within 400 steps				
GT-FO	29/50	16/50	11/50	6/50
GT-DAdamW	26/50	14/50	7/50	2/50
Hybrid-KT	**48/50**	**47/50**	**45/50**	**42/50**
Aggregate mean TTA (s)				
GT-FO	**7.12**	8.56	9.96	14.10
GT-DAdamW	10.33	12.47	13.41	15.71
Hybrid-KT	7.65	**8.38**	**8.70**	**9.30**

Table 4: Sensitivity of β_T under different runtime conditions.

Runtime condition	$(\max - \min)/\mathrm{TTA}_{\beta_T=0.9}$
Static network	1.8%
Dynamic network	4.9%

As shown in Table 3, Hybrid-KT reaches the target in 48/50, 47/50, 45/50, and 42/50 runs for objective gaps of ≤ 0.5, ≤ 0.2, ≤ 0.1, and ≤ 0.01, respectively. Compared with GT-FO, it improves the strictest-target success count from 6/50 to 42/50, while reducing the mean TTA at ≤ 0.01 from 14.10s to 9.30s (34.0% reduction). Since mean TTA includes unsuccessful runs through timeout penalties, these improvements reflect both higher target attainment probability and faster convergence on the complete seed set.

D Hyperparameter Ablation

We evaluate the sensitivity of TTA-GT to the parameters governing prediction smoothing, residual construction, and adaptation. The experiments use the same strongly convex synthetic setting with a 16-node ring and the target $F(\bar{x}^t) - F(x^\star) \leq 10^{-8}$. We vary one parameter at a time around the default configuration

$$\lambda_c = 1, \qquad \beta_q = 0.95, \qquad \beta_T = 0.9, \qquad H = 5, \qquad P = 50, \qquad R = 5.$$

Results are reported as the relative change in TTA from the default. The purpose of these ablations is to examine whether each parameter is necessary for effective adaptation and to identify a practical range of values.

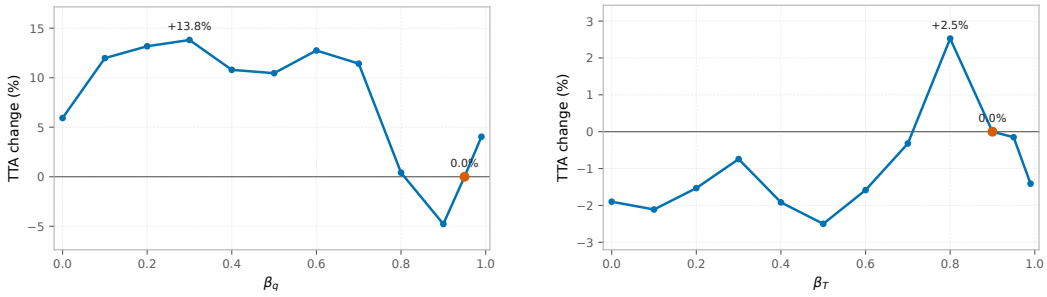
D.1 Predictor Smoothing Coefficients

Figure 4 shows sweeps of the two smoothing coefficients. The parameter β_q controls the responsiveness of the progress predictor. Small values lead to more reactive switching and worse TTA, while larger values improve stability. We therefore use $\beta_q = 0.95$ as the default.The selector is less sensitive to β_T in the static heterogeneous-ring setting because the runtime profile changes little during training. The sweep varies β_T while fixing the implementation-level KTV2 discount at 0.9, isolating the effect of time smoothing.

We further evaluate β_T under static and dynamic environment with time-varying communication and computation conditions. Table 4 reports the resulting TTA ranges, showing limited sensitivity under stable runtime and moderately larger variation under dynamic computation.

D.2 Adaptation Control and Residual Construction

igure 5a shows that the warm-up budget H controls the number of early second-order samples used to initialize the predictor. Removing warm-up hurts TTA, increasing it by 14.1%, indicating that the



(a) β_q sweep.

(b) β_T sweep.

Figure 4: Sweeps of β_q and β_T

selector benefits from early measurements before relying on score comparisons. A short warm-up is sufficient, while increasing H beyond the default provides no additional benefit, with $H = 10$ increasing TTA by 2.5%.

Figure 5b shows that frequent probing increases overhead and switch activity, while very sparse probing can make estimates stale. In particular, reducing P from 50 to 10 increases TTA by 9.1%, whereas $P = 25$ changes TTA by only 2.6%.

As shown in Figure 5c, the cooldown period R prevents immediate mode reversals after a switch. Small cooldown values behave similarly, while a large R degrades TTA by delaying mode changes. For example, $R = 20$ increases TTA by 12.8%, whereas $R \leq 10$ changes TTA by at most 2.6%. Thus, we use $P = 50$ and $R = 5$ as defaults.

As shown in Figure 5d, the consensus residual weight λ_c controls the relative importance of local optimization progress and consensus disagreement in r_i^k. This is the most sensitive parameter among the tested choices. Setting $\lambda_c = 0$ removes consensus disagreement from the residual and substantially worsens TTA by 24.3%, while an excessively large value, $\lambda_c = 10$, increases TTA by 7.2%. Moderate nonzero values perform much more stably, with $\lambda_c = 0.1$ differing from the default by only -0.6%, supporting the default choice $\lambda_c = 1$.

D.3 Probe–Cooldown Interaction

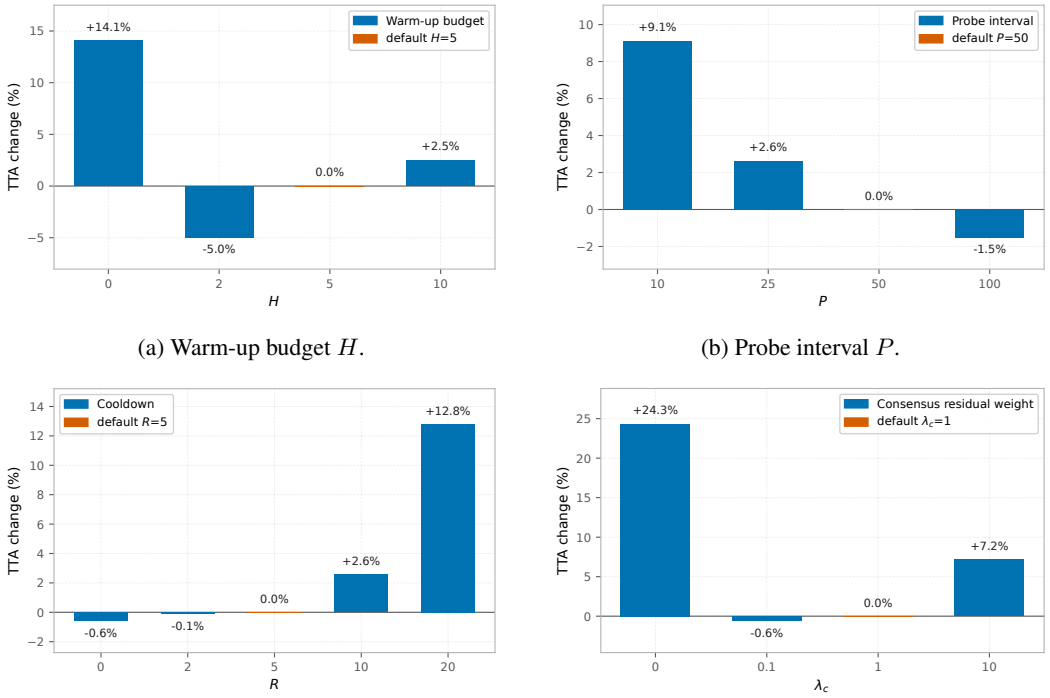
Because probing and cooldown jointly affect adaptation, we perform a two-dimensional sweep over P and R/P. Figure 6 shows that short probe intervals increase switching and overhead. For $P = 10$, the selector makes about 88–91 switches and TTA increases by 8.9%–13.7% across the tested cooldown ratios. Large ratios such as $R/P = 1$ make cooldown overly restrictive, increasing TTA by 32.2% at $P = 50$ and 56.3% at $P = 100$. In contrast, $R/P = 0.1$ remains stable for moderate probe intervals, with TTA changes of 2.3%, 0.0%, and 2.3% for $P = 25$, 50, and 100, respectively.

Cooldown remains effective when $R < P$, as different cooldown ratios still produce different TTA and switching behavior. Its effect becomes strongest as R approaches P, where adaptation becomes increasingly restricted.

E Logistic Regression Results On Real Datasets

We further evaluate the direction selector on the public LIBSVM logistic-regression dataset `a9a`. Each experiment uses 16 nodes, a ring topology, and the H3 heterogeneous network profile. The data are split either iid or by a Dirichlet partition. The target is the first iterate satisfying objective gap 10^{-4}. Table 5 reports the mean TTA to this target.

The real-data results support the same qualified conclusion as the main experiments. On `a9a`, TTA-GT is fastest under iid and moderately heterogeneous partitions. It reduces TTA from 49.218 to 2.323 in the iid partition and from 42.264 to 2.695 under Dirichlet $\alpha = 1$, relative to GT-FO. It also reduces TTA by about 1.9–2.4$\times$ relative to GT-SO in these two settings. Under the more heterogeneous `a9a` partition, GT-SO and TTA-GT are nearly tied. The real-data experiments therefore do



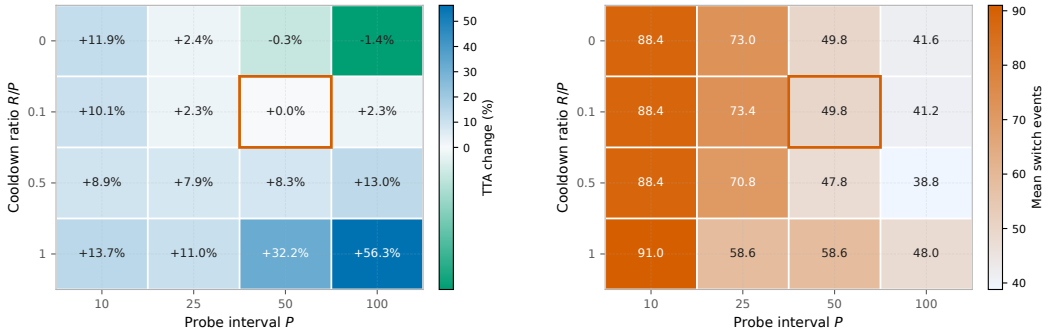
(a) Warm-up budget H.

(b) Probe interval P.

(c) Cooldown period R.

(d) Consensus residual weight λ_c.

Figure 5: TTA change of warm-up, probing, cooldown, and consensus weighting relative to the default setting.



(a) TTA change.

(b) Mean switch events.

Figure 6: Two-dimensional probe/cooldown ablation.

not suggest that adaptation is always preferable. They show that selection based on TTA is most useful when curvature information can reduce iterations but is not always worth its computation cost.

F Explicit Convergence Constants

Let $m = \frac{1}{n}\sum_i m_i$, $L_F = \frac{1}{n}\sum_i \ell_i$, and $\ell = \max_i \ell_i$. The constants m, L_F, and ℓ separate the three roles of curvature in the proof. Strong convexity controls the descent of the averaged model, the average smoothness appears in the objective recursion, and the worst case local smoothness controls disagreement and tracking errors.

Table 5: Real-data logistic-regression TTA on the LIBSVM `a9a` dataset.

Dataset	Partition	Dim.	GT-FO	GT-SO	TTA-GT	GT-FO / TTA-GT	GT-SO / TTA-GT
a9a	iid	124	49.218	5.549	**2.323**	21.19	2.39
a9a	Dirichlet $\alpha = 1$	124	42.264	5.171	**2.695**	15.68	1.92
a9a	Dirichlet $\alpha = 0.1$	124	54.305	**9.822**	9.914	5.48	0.99

Let

$$\Gamma := \frac{1+\sigma^2}{2}, \qquad \Delta := 1-\Gamma = \frac{1-\sigma^2}{2}, \qquad \kappa_\sigma := \frac{1+\sigma^2}{1-\sigma^2},$$

and choose Lyapunov weights $\lambda_B, \lambda_D > 0$ such that

$$\lambda_D \geq \frac{4}{\Delta}\frac{\sigma^2}{n}\left(\frac{\theta_+^2}{2\theta_-} + \frac{\theta_-}{4} + \frac{3L_F\theta_+^2}{2}\right),$$
$$\lambda_B \geq \frac{4}{\Delta}\left[\frac{\ell^2}{n}\left(\frac{\theta_-}{4} + \frac{2\theta_+^2}{\theta_-} + \frac{3L_F\theta_+^2}{2}\right) + 3\lambda_D\kappa_\sigma\ell^2(1+\sigma+\theta_+\ell)^2\right]. \tag{20}$$

Set $\alpha := m\theta_-/8$ and suppose that

$$\theta_+\ell \leq 1, \qquad 3L_F\theta_+^2 \leq \frac{\theta_-}{16}, \qquad \alpha \leq \frac{\Delta}{4}, \qquad 3\kappa_\sigma\ell^2\theta_+^2 \leq \frac{\Delta}{4}, \tag{21}$$

$$6\kappa_\sigma nL_F\theta_+^2(\lambda_B + \ell^2\lambda_D) \leq \frac{\alpha}{2}, \qquad 3\lambda_B\kappa_\sigma\sigma^2\theta_+^2 \leq \frac{\lambda_D\Delta}{4}. \tag{22}$$

These conditions ensure that the descent from strong convexity controls the additional terms introduced by mixing, tracking, and variable local directions.

G Proof of Theorem 1

This appendix provides the detailed proof of the linear convergence theorem. The proof follows a Lyapunov analysis of three coupled quantities: the optimization error, the consensus error, and the gradient tracking error. The adaptive first-/second-order selection mechanism only affects the proof through the accepted direction conditions

$$\langle d_i^t, \tilde{s}_i^t\rangle \geq \theta_-\|\tilde{s}_i^t\|^2, \qquad \|d_i^t\| \leq \theta_+\|\tilde{s}_i^t\|.$$

Therefore, the analysis applies to any direction sequence satisfying these conditions, including the TTA-GT switching rule.

G.1 Gradient Tracking Invariant and Auxiliary Errors

By Equation (3)and W's stochasticity, we obtain

$$\bar{s}^{t+1} = \bar{s}^t + \frac{1}{n}\sum_{i=1}^n \left[\nabla f_i(x_i^{t+1}) - \nabla f_i(x_i^t)\right].$$

Combining the initialization $s_i^0 = \nabla f_i(x_i^0)$, By induction, we obtain

$$\bar{s}^t = \frac{1}{n}\sum_{i=1}^n \nabla f_i(x_i^t). \tag{23}$$

Let

$$h^t := \nabla F(\bar{x}^t), \qquad q^t := \bar{s}^t - h^t.$$

By Equation (23) and local smoothness,

$$q^t = \frac{1}{n}\sum_{i=1}^n \left[\nabla f_i(x_i^t) - \nabla f_i(\bar{x}^t)\right],$$

therefore,

$$\left\|q^t\right\| \leq \frac{\ell}{n}\sum_{i=1}^{n}\left\|x_i^t - \bar{x}^t\right\| \leq \frac{\ell}{\sqrt{n}}\sqrt{B_t}. \tag{24}$$

Denote

$$e^t := \tilde{s}^t - Js^t = (\mathcal{W} - J)s^t.$$

then

$$\left\|e^t\right\| \leq \sigma\sqrt{D_t}. \tag{25}$$

The dimensionally consistent form is

$$\tilde{s}^t = \mathbf{1} \otimes (h^t + q^t) + e^t. \tag{26}$$

G.2 Direction Term Estimation

Let

$$\gamma_- := \theta_-, \qquad \gamma_+ := \theta_+.$$

The acute-angle direction condition is

$$\left\langle d_i^t, \tilde{s}_i^t \right\rangle \geq \gamma_- \left\|\tilde{s}_i^t\right\|^2, \qquad \left\|d_i^t\right\| \leq \gamma_+ \left\|\tilde{s}_i^t\right\|.$$

By

$$\bar{d}^t := \frac{1}{n}\sum_{i=1}^{n} d_i^t, \qquad \bar{x}^{t+1} = \bar{x}^t - \bar{d}^t,$$

We next estimate $\left\langle h^t, \bar{d}^t \right\rangle$ and $\left\|\bar{d}^t\right\|^2$.

First, by

$$\tilde{s}^t = Js^t + e^t = \mathbf{1} \otimes \bar{s}^t + e^t$$

and $e^t \perp \mathrm{range}(J)$, we have

$$\frac{1}{n}\left\|\tilde{s}^t\right\|^2 = \left\|\bar{s}^t\right\|^2 + \frac{1}{n}\left\|e^t\right\|^2.$$

Furthermore,

$$\begin{aligned}
\left\langle \bar{s}^t, \bar{d}^t \right\rangle &= \frac{1}{n}\sum_{i=1}^{n}\left\langle \bar{s}^t, d_i^t \right\rangle \\
&= \frac{1}{n}\sum_{i=1}^{n}\left\langle \tilde{s}_i^t - e_i^t, d_i^t \right\rangle \\
&\geq \frac{\gamma_-}{n}\left\|\tilde{s}^t\right\|^2 - \frac{\gamma_+}{n}\left\|e^t\right\|\left\|\tilde{s}^t\right\|.
\end{aligned}$$

By Young's inequality

$$\frac{\gamma_+}{n}\left\|e^t\right\|\left\|\tilde{s}^t\right\| \leq \frac{\gamma_-}{2n}\left\|\tilde{s}^t\right\|^2 + \frac{\gamma_+^2}{2\gamma_- n}\left\|e^t\right\|^2,$$

we obtain

$$\left\langle \bar{s}^t, \bar{d}^t \right\rangle \geq \frac{\gamma_-}{2}\left\|\bar{s}^t\right\|^2 - \frac{\gamma_+^2}{2\gamma_- n}\left\|e^t\right\|^2. \tag{27}$$

On the other hand,

$$\left\|\bar{d}^t\right\| \leq \frac{\gamma_+}{n}\sum_{i=1}^{n}\left\|\tilde{s}_i^t\right\| \leq \gamma_+\left\|\bar{s}^t\right\| + \frac{\gamma_+}{\sqrt{n}}\left\|e^t\right\|.$$

By $h^t = \bar{s}^t - q^t$, By Equation (27), we obtain

$$\begin{aligned}
\left\langle h^t, \bar{d}^t \right\rangle &= \left\langle \bar{s}^t, \bar{d}^t \right\rangle - \left\langle q^t, \bar{d}^t \right\rangle \\
&\geq \frac{\gamma_-}{2}\left\|\bar{s}^t\right\|^2 - \frac{\gamma_+^2}{2\gamma_- n}\left\|e^t\right\|^2 - \gamma_+\left\|q^t\right\|\left\|\bar{s}^t\right\| - \frac{\gamma_+}{\sqrt{n}}\left\|q^t\right\|\left\|e^t\right\|.
\end{aligned}$$

Using Young's inequality again,

$$\gamma_+ \left\|q^t\right\| \left\|\bar{s}^t\right\| \leq \frac{\gamma_-}{4} \left\|\bar{s}^t\right\|^2 + \frac{\gamma_+^2}{\gamma_-} \left\|q^t\right\|^2,$$

$$\frac{\gamma_+}{\sqrt{n}} \left\|q^t\right\| \left\|e^t\right\| \leq \frac{\gamma_+^2}{\gamma_-} \left\|q^t\right\|^2 + \frac{\gamma_-}{4n} \left\|e^t\right\|^2,$$

Thus

$$\left\langle h^t, \bar{d}^t \right\rangle \geq \frac{\gamma_-}{4} \left\|\bar{s}^t\right\|^2 - \frac{2\gamma_+^2}{\gamma_-} \left\|q^t\right\|^2 - \left(\frac{\gamma_+^2}{2\gamma_- n} + \frac{\gamma_-}{4n}\right) \left\|e^t\right\|^2.$$

Moreover, since $\bar{s}^t = h^t + q^t$, we have

$$\left\|\bar{s}^t\right\|^2 \geq \frac{1}{2} \left\|h^t\right\|^2 - \left\|q^t\right\|^2.$$

therefore,

$$\left\langle h^t, \bar{d}^t \right\rangle \geq \frac{\gamma_-}{8} \left\|h^t\right\|^2 - \left(\frac{\gamma_-}{4} + \frac{2\gamma_+^2}{\gamma_-}\right) \left\|q^t\right\|^2 - \left(\frac{\gamma_+^2}{2\gamma_- n} + \frac{\gamma_-}{4n}\right) \left\|e^t\right\|^2. \tag{28}$$

Substituting Equation (24) and (25), we obtain

$$\left\langle h^t, \bar{d}^t \right\rangle \geq \frac{\gamma_-}{8} \left\|h^t\right\|^2 - \frac{\ell^2}{n} \left(\frac{\gamma_-}{4} + \frac{2\gamma_+^2}{\gamma_-}\right) B_t - \frac{\sigma^2}{n} \left(\frac{\gamma_+^2}{2\gamma_-} + \frac{\gamma_-}{4}\right) D_t. \tag{29}$$

At the same time, By

$$\left\|\bar{d}^t\right\| \leq \gamma_+ \left(\left\|h^t\right\| + \left\|q^t\right\| + \frac{1}{\sqrt{n}} \left\|e^t\right\|\right)$$

and $(a+b+c)^2 \leq 3(a^2+b^2+c^2)$, we obtain

$$\left\|\bar{d}^t\right\|^2 \leq 3\gamma_+^2 \left\|h^t\right\|^2 + \frac{3\gamma_+^2 \ell^2}{n} B_t + \frac{3\gamma_+^2 \sigma^2}{n} D_t. \tag{30}$$

G.3 Recursion of the Three Error Terms

Optimization Error Recursion. By the L_F-smoothness,

$$F(\bar{x}^{t+1}) \leq F(\bar{x}^t) - \left\langle h^t, \bar{d}^t \right\rangle + \frac{L_F}{2} \left\|\bar{d}^t\right\|^2.$$

Substituting Equation (29) and (30), we obtain

$$A_{t+1} \leq A_t - \left(\frac{\gamma_-}{8} - \frac{3L_F\gamma_+^2}{2}\right) \left\|h^t\right\|^2 + \frac{\ell^2}{n} \left(\frac{\gamma_-}{4} + \frac{2\gamma_+^2}{\gamma_-} + \frac{3L_F\gamma_+^2}{2}\right) B_t + \frac{\sigma^2}{n} \left(\frac{\gamma_+^2}{2\gamma_-} + \frac{\gamma_-}{4} + \frac{3L_F\gamma_+^2}{2}\right) D_t.$$

By the m-strong convexity,

$$\left\|h^t\right\|^2 = \left\|\nabla F(\bar{x}^t)\right\|^2 \geq 2mA_t.$$

Hence

$$\begin{aligned} A_{t+1} \leq & \left(1 - \frac{m\gamma_-}{4} + 3mL_F\gamma_+^2\right) A_t + \frac{\ell^2}{n} \left(\frac{\gamma_-}{4} + \frac{2\gamma_+^2}{\gamma_-} + \frac{3L_F\gamma_+^2}{2}\right) B_t \\ & + \frac{\sigma^2}{n} \left(\frac{\gamma_+^2}{2\gamma_-} + \frac{\gamma_-}{4} + \frac{3L_F\gamma_+^2}{2}\right) D_t. \end{aligned} \tag{31}$$

Consensus Error Recursion. By Equation (2), left-multiplying $I - J$, we obtain

$$x^{t+1} - Jx^{t+1} = (\mathcal{W} - J)x^t - (I - J)d^t.$$

therefore,

$$\sqrt{B_{t+1}} \leq \sigma \sqrt{B_t} + \left\|d^t\right\|.$$

Moreover, by Equation (16) and Equation (26),

$$\left\|d^t\right\| \leq \gamma_+ \left\|\tilde{s}^t\right\| \leq \gamma_+ \left(\sqrt{n}\left\|h^t\right\| + \ell\sqrt{B_t} + \sigma\sqrt{D_t}\right).$$

For any $a, b \geq 0$, we have

$$(\sigma a + b)^2 \leq \Gamma a^2 + \kappa_\sigma b^2.$$

Combining

$$\left\|h^t\right\|^2 = \left\|\nabla F(\bar{x}^t)\right\|^2 \leq 2L_F A_t,$$

we obtain

$$\begin{aligned} B_{t+1} \leq & 6\kappa_\sigma\gamma_+^2 nL_F A_t + \left(\Gamma + 3\kappa_\sigma\gamma_+^2\ell^2\right) B_t \\ & + 3\kappa_\sigma\gamma_+^2\sigma^2 D_t. \end{aligned} \tag{32}$$

Gradient Tracking Error Recursion. By Equation (3), left-multiplying $I - J$, we obtain

$$s^{t+1} - Js^{t+1} = (\mathcal{W} - J)s^t + (I - J)\big(g(x^{t+1}) - g(x^t)\big).$$

Since each ∇f_i is ℓ_i-Lipschitz and $\ell = \max_i \ell_i$, we have

$$\sqrt{D_{t+1}} \leq \sigma\sqrt{D_t} + \ell\left\|x^{t+1} - x^t\right\|.$$

Moreover,

$$x^{t+1} - x^t = (\mathcal{W} - I)x^t - d^t,$$

and

$$\left\|(\mathcal{W} - I)x^t\right\| = \left\|(\mathcal{W} - I)(x^t - Jx^t)\right\| \leq (1 + \sigma)\sqrt{B_t}.$$

therefore,

$$\left\|x^{t+1} - x^t\right\| \leq (1 + \sigma + \gamma_+\ell)\sqrt{B_t} + \gamma_+\sqrt{n}\left\|h^t\right\| + \gamma_+\sigma\sqrt{D_t}.$$

Using again

$$(\sigma a + b)^2 \leq \Gamma a^2 + \kappa_\sigma b^2$$

and $\|h^t\|^2 \leq 2L_F A_t$, we obtain

$$D_{t+1} \leq 6\kappa_\sigma\ell^2\gamma_+^2 nL_F A_t + 3\kappa_\sigma\ell^2(1 + \sigma + \gamma_+\ell)^2 B_t + \left(\Gamma + 3\kappa_\sigma\ell^2\gamma_+^2\sigma^2\right) D_t. \tag{33}$$

G.4 Matrix Recursion and Coefficient Bounds

Let

$$E_t := \begin{bmatrix} A_t \\ B_t \\ D_t \end{bmatrix}.$$

By Equation (31), (32) and(33) we obtain

$$E_{t+1} \leq \mathcal{R}E_t.$$

where $\mathcal{R}$ is Equation (31)–(33) right-hand side with respect to (A_t, B_t, D_t) coefficient matrix. We do not introduce additional names for its entries; instead, we substitute $\gamma_- = \theta_-$ and $\gamma_+ = \theta_+$ directly. We have

$$\Gamma + 3\kappa_\sigma\theta_+^2\ell^2 \leq \Gamma + \frac{\Delta}{4}, \qquad \Gamma + 3\kappa_\sigma\ell^2\theta_+^2\sigma^2 \leq \Gamma + \frac{\Delta}{4}.$$

At the same time

$$3mL_F\theta_+^2 \leq \frac{m\theta_-}{16} = \frac{\alpha}{2}.$$

Thus,

$$1 - \frac{m\theta_-}{4} + 3mL_F\theta_+^2 \leq 1 - \frac{3\alpha}{2}. \tag{34}$$

G.5 Lyapunov Weight Selection and Contraction Verification

Define the weighted Lyapunov function

$$V_t = A_t + \lambda_B B_t + \lambda_D D_t.$$

The matrix recursion yields

$$V_{t+1} \le \Psi_A A_t + \Psi_B B_t + \Psi_D D_t,$$

where

$$\Psi_A := 1 - \frac{m\theta_-}{4} + 3mL_F\theta_+^2 + 6\lambda_B\kappa_\sigma\theta_+^2 nL_F + 6\lambda_D\kappa_\sigma\ell^2\theta_+^2 nL_F,$$
$$\Psi_B := \frac{\ell^2}{n}\left(\frac{\theta_-}{4} + \frac{2\theta_+^2}{\theta_-} + \frac{3L_F\theta_+^2}{2}\right) + \lambda_B\left(\Gamma + 3\kappa_\sigma\theta_+^2\ell^2\right) + 3\lambda_D\kappa_\sigma\ell^2(1+\sigma+\theta_+\ell)^2,$$
$$\Psi_D := \frac{\sigma^2}{n}\left(\frac{\theta_+^2}{2\theta_-} + \frac{\theta_-}{4} + \frac{3L_F\theta_+^2}{2}\right) + 3\lambda_B\kappa_\sigma\theta_+^2\sigma^2 + \lambda_D\left(\Gamma + 3\kappa_\sigma\ell^2\theta_+^2\sigma^2\right).$$

We next prove

$$\Psi_A \le \rho, \qquad \Psi_B \le \rho\lambda_B, \qquad \Psi_D \le \rho\lambda_D, \qquad \rho := 1-\alpha.$$

Verification $\Psi_A \le \rho$. By Equation (34) and (22),

$$6\kappa_\sigma nL_F\theta_+^2(\lambda_B + \ell^2\lambda_D) \le \frac{\alpha}{2}.$$

Therefore,

$$\Psi_A \le 1 - \frac{3\alpha}{2} + \frac{\alpha}{2} = \rho.$$

Verification $\Psi_B \le \rho\lambda_B$. This inequality is equivalent to

$$\frac{\ell^2}{n}\left(\frac{\theta_-}{4} + \frac{2\theta_+^2}{\theta_-} + \frac{3L_F\theta_+^2}{2}\right) + 3\lambda_D\kappa_\sigma\ell^2(1+\sigma+\theta_+\ell)^2 \le \lambda_B\left(\rho - \Gamma - 3\kappa_\sigma\theta_+^2\ell^2\right).$$

By Equation (21),

$$\rho - \Gamma - 3\kappa_\sigma\theta_+^2\ell^2 \ge \Delta - \alpha - \frac{\Delta}{4} \ge \frac{\Delta}{2}.$$

therefore,

$$\lambda_B\left(\rho - \Gamma - 3\kappa_\sigma\theta_+^2\ell^2\right) \ge \frac{\lambda_B\Delta}{2}.$$

On the other hand,By the weight choice Equation (20),

$$\frac{\ell^2}{n}\left(\frac{\theta_-}{4} + \frac{2\theta_+^2}{\theta_-} + \frac{3L_F\theta_+^2}{2}\right) + 3\lambda_D\kappa_\sigma\ell^2(1+\sigma+\theta_+\ell)^2 \le \frac{\lambda_B\Delta}{4} \le \frac{\lambda_B\Delta}{2}.$$

Hence

$$\frac{\ell^2}{n}\left(\frac{\theta_-}{4} + \frac{2\theta_+^2}{\theta_-} + \frac{3L_F\theta_+^2}{2}\right) + 3\lambda_D\kappa_\sigma\ell^2(1+\sigma+\theta_+\ell)^2 \le \lambda_B\left(\rho - \Gamma - 3\kappa_\sigma\theta_+^2\ell^2\right),$$

i.e.

$$\Psi_B \le \rho\lambda_B.$$

Verification $\Psi_D \le \rho\lambda_D$. This inequality is equivalent to

$$\frac{\sigma^2}{n}\left(\frac{\theta_+^2}{2\theta_-} + \frac{\theta_-}{4} + \frac{3L_F\theta_+^2}{2}\right) + 3\lambda_B\kappa_\sigma\theta_+^2\sigma^2 \le \lambda_D\left(\rho - \Gamma - 3\kappa_\sigma\ell^2\theta_+^2\sigma^2\right).$$

Similarly, By Equation (21),

$$\rho - \Gamma - 3\kappa_\sigma\ell^2\theta_+^2\sigma^2 \ge \frac{\Delta}{2}.$$

Thus,

$$\lambda_D \left(\rho - \Gamma - 3\kappa_\sigma \ell^2 \theta_+^2 \sigma^2\right) \geq \frac{\lambda_D \Delta}{2}.$$

On the other hand,By Equation (20) and (22),

$$\frac{\sigma^2}{n}\left(\frac{\theta_+^2}{2\theta_-} + \frac{\theta_-}{4} + \frac{3L_F\theta_+^2}{2}\right) \leq \frac{\lambda_D \Delta}{4}, \qquad 3\lambda_B \kappa_\sigma \sigma^2 \theta_+^2 \leq \frac{\lambda_D \Delta}{4}.$$

therefore,

$$\frac{\sigma^2}{n}\left(\frac{\theta_+^2}{2\theta_-} + \frac{\theta_-}{4} + \frac{3L_F\theta_+^2}{2}\right) + 3\lambda_B \kappa_\sigma \sigma^2 \theta_+^2 \leq \frac{\lambda_D \Delta}{2} \leq \lambda_D \left(\rho - \Gamma - 3\kappa_\sigma \ell^2 \theta_+^2 \sigma^2\right),$$

i.e.

$$\Psi_D \leq \rho \lambda_D.$$

In summary, under Equation (20)–(22) under

$$V_{t+1} \leq \rho A_t + \rho \lambda_B B_t + \rho \lambda_D D_t = \rho V_t.$$

By $\alpha \leq \Delta/4 < 1$ we have $\rho = 1 - \alpha \in (0, 1)$.By induction, we obtain

$$V_t \leq \rho^t V_0.$$

By $A_t, B_t, D_t \geq 0$, we have

$$A_t \leq \rho^t V_0, \qquad B_t \leq \frac{\rho^t V_0}{\lambda_B}, \qquad D_t \leq \frac{\rho^t V_0}{\lambda_D}.$$

This proves the main theorem.