UNIFYING LOCAL COMMUNICATION AND LOCAL UPDATES FOR LLM PRETRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Several methods reduce the communication cost of distributed LLM pre-training by performing multiple local optimization steps between global aggregations. However, these methods still rely on global collectives, whose duration is limited by the available bandwidth and the slowest worker. We introduce **GASLoC**, a decentralized method in which each worker performs one or more local optimizer steps, exchanges the resulting parameters with one or two randomly selected peers, and applies an outer optimizer to the pseudo-gradient computed from the received parameters. This structure avoids global collectives and allows the number of local steps to vary across workers. We establish convergence under different local step counts and show that outer momentum can accelerate consensus. Empirically, at token budgets selected according to compute optimal scaling, the strongest GASLoC variant achieves lower validation loss than the evaluated decentralized baselines when communicating after every local step. With multiple local steps between communications, it remains competitive with DiLoCo while avoiding global collectives. Under heterogeneous bandwidth, adapting the number of local steps across workers reduces end-to-end training time relative to DiLoCo.

1 INTRODUCTION

Large language models are commonly trained with data parallelism across many workers. In this setting, each worker computes gradients on a different batch, and these gradients are combined before every optimization step. This synchronization is commonly performed with All-Reduce (Li et al., 2020). Ring All-Reduce (Patarasuk & Yuan, 2009) combines gradient through reduce-scatter and distributes them through all-gather. This uses bandwidth efficiently, but each optimization step waits for all workers, so the slowest worker in the network can limit training speed. Decentralized learning based on gossip (Boyd et al., 2006; Koloskova et al., 2020) replaces global collectives with exchanges between selected peers and has been widely studied in classical optimization. This can reduce the communication and synchronization overhead associated with global collectives and allow for more robust handling of failures when only a single peer-to-peer communication fails.

Another line of work considers methods for decreasing the overall communication in model training through less frequent communication (Stich, 2019; Douillard et al., 2024; Charles et al., 2026), gradient compression (Wang et al., 2023; Sarfi et al., 2025), or communication overlap (Douillard et al., 2025). Although these methods reduce the frequency or cost of global synchronization, they still rely on *global* aggregations, whose impact can be mitigated by momentum. In particular, DiLoCo-style methods apply an outer optimizer to the aggregated updates. Empirical results show that outer optimizers with momentum improve convergence (Douillard et al., 2024), and theoretical analysis indicates that the outer learning rate can reduce sensitivity to the inner step size (Khaled et al., 2025). However, the role of outer momentum in realistic deep learning settings remains only partially understood (Agarwala, 2026).

In this work, we formalize a local optimization framework that builds on a classical connection between momentum and Chebyshev acceleration. In gossip optimization, the disagreement caused by delayed information propagation is measured by the consensus error. Classical methods accelerate its reduction by applying Chebyshev polynomials to the gossip operations (Loizou et al., 2019; Berthier et al., 2020). Equivalently, this acceleration can be implemented through a Nesterov-type momentum scheme (Scaman et al., 2017). We use this equivalence to interpret the outer momentum

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.

of local optimization methods as a mechanism for accelerating consensus. The resulting framework combines sparse gossip communication with local optimization methods like DiLoCo and achieve state-of-the-art performance among gossip based methods (Wang et al., 2025).

By interpreting the outer optimizer through the lens of communication acceleration, our analysis jointly captures the effects of both the outer and inner optimizers. For instance, Wang et al. (2025) study a decentralized adaptation of Adam under a common data distribution and heterogeneous communication and computation times, but mix models after every optimizer step at every worker. However, to our knowledge, no existing method combines blocks of multiple local steps whose length may vary across workers, randomized peer exchanges, and consensus acceleration in this setting.

To address this gap, we introduce **GASLoC**, a decentralized method in which each worker performs one or more local optimizer steps and exchanges the resulting parameters with one or two peers. An outer optimizer then processes the pseudo-gradient computed from the received parameters using momentum to accelerate consensus. Decoupling inner and outer optimization lets local steps vary across workers, allowing to account for compute or bandwidth heterogeneity.

Our contributions are:

- We extend the outer optimizer structure of FEDOPT (Reddi et al., 2021) to sparse gossip communication. With matching inner and outer settings, GASLoC recovers DiLoCo (Douillard et al., 2024) under complete communication. With randomized peer sampling, GASLoC avoids global collectives, while the expected consensus does not degrade with worker count.
- We prove a nonconvex convergence guarantee for a homogeneous objective, allowing workers to perform different numbers of local steps. We also show that outer momentum can accelerate consensus by improving the dependence of the communication complexity on the spectral gap χ to $\sqrt{\chi}$. This provides a novel interpretation of outer momentum, recently of great interest in DiLoCo literature, through the lens of communication acceleration.
- Across model , architectures and up to 32 workers, the strongest variant of GASLoC achieves lower final validation loss than existing state-of-the-art gossip methods for LLM pre-training. This is both in settings with frequent communication (single step) and infrequent communication (multi-step). We also show that in multi-step settings it can achieve performance comparable to DiLoCo while avoiding global collectives. Finally, we demonstrate under a bandwidth straggler, we can achieve substantially better wall-clock times.

2 RELATED WORK

Local optimization with global coordination. Federated learning (Konečný et al., 2016; McMahan et al., 2017) and Local SGD (Stich, 2019) reduce communication frequency by performing multiple local steps before global aggregation. Building on this structure, FEDOPT (Reddi et al., 2021) uses adaptive server optimization, SlowMo (Wang et al., 2020) adds momentum, and DiLoCo (Douillard et al., 2024) combines local AdamW with global Nesterov momentum for LLM pre-training. Khaled et al. (2025) analyze the outer learning rate, momentum, and acceleration. Recent extensions reduce the cost of the remaining global coordination by overlapping global communication with local computation, as in Streaming DiLoCo and Eager Updates (Douillard et al., 2025; Kale et al., 2025). Other methods use different local step counts under compute heterogeneity, hierarchical aggregation for geographically distributed training, or quorum updates to tolerate stragglers and failures (Liu et al., 2024; Kim et al., 2025; Douillard et al., 2026). However, they retain a global aggregation or central coordinator, which motivates decentralized optimization based on peer exchanges.

Decentralized optimization. Decentralized optimization removes global aggregation by propagating model updates through peer exchanges (Boyd et al., 2006). Stochastic Gradient Push couples every gradient step with peer mixing (Assran et al., 2019), while Koloskova et al. (2020) also cover multiple local SGD steps, pairwise gossip, several topologies, and homogeneous or heterogeneous local objectives. We instead assume a common objective and address compute or bandwidth heterogeneity through local step counts that vary across workers. With sparse exchanges, each update reaches the other workers only through several rounds, so its propagation depends on the communication topology.

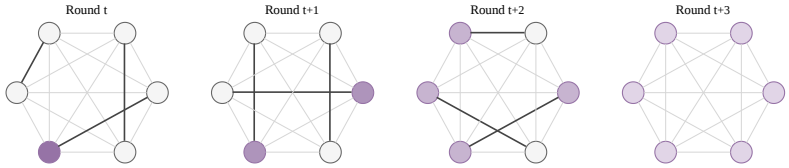


Figure 1: **Information propagation with time-varying 1-Peer gossip.** Nodes shaded in purple has received information from the initial worker. Black edges identify the active matching in the round, as it changes information propagates without global collectives. We call this communication pattern GASLoC-1-Peer.

Sparse communication topologies. Communication schemes without exact global aggregation differ in whether they retain a coordinator and how they propagate model updates. EASGD uses elastic updates to a central reference model, while RelaySum uses spanning trees to reach exact uniform averaging with a finite number of steps (Zhang et al., 2015; Vogels et al., 2021). Gossip instead removes the coordinator and communicates over sparse peer edges, which can be sampled at random or follow deterministic graph sequences (Boyd et al., 2006). Deterministic schemes such as 1-Peer exponential graphs, Alternating Exponential Rings, and k-Peer graphs obtain their mixing properties from prescribed graph cycles (Ying et al., 2021; Wang et al., 2025; Niwa et al., 2025). Our variants instead sample a fresh uniform permutation at every outer step to form an induced matching or ring, so the spectral gap of the expected communication operator does not vanish as the number of workers increases. We use uniform sampling and leave optimized edge distributions, activation rates, and fully asynchronous protocols outside our scope (Boyd et al., 2006; Nabli & Oyallon, 2025; Nabli et al., 2025). Randomization improves the expected mixing rate, but sparse exchanges still delay update propagation, which motivates consensus acceleration.

Consensus acceleration. Accelerated gossip methods reduce this delay by retaining information from previous exchanges through momentum or polynomial recursions (Loizou et al., 2019; Berthier et al., 2020). Gossip-PGA controls slow mixing through periodic global averaging, which reintroduces a global collective (Chen et al., 2021). Event driven methods instead preserve peer communication: continuized acceleration uses random communication events, DADAO separates these events from gradient computations under strong convexity, and $\mathrm{A}^2\mathrm{CiD}^2$allows local minibatch updates to proceed alongside randomized peer averaging (Even et al., 2021; Nabli & Oyallon, 2023; Nabli et al., 2023). GASLoC instead applies momentum through an outer optimizer after a block of local updates, matching the inner and outer structure used in low communication LLM training (Douillard et al., 2024).

Decentralized LLM training. The combination of decentralized communication and adaptive optimization has only recently been studied for LLM training. DAdam applies decentralized Adam to transformer training and accounts for heterogeneous communication and computation, but requires every worker to mix models after each local optimizer step (Wang et al., 2025). NoLoCo uses random pairwise outer updates together with dynamic routing between replicas of adjacent pipeline stages, which ties its communication rule to pipeline parallelism (Kolehmainen et al., 2025). Factored Gossip DiLoCo separates synchronization into parameter mixing that overlaps local computation and a blocking operation on outer updates, allowing global or peer mixing in either operation (Koneputugodage et al., 2026). GASLoC instead applies one time-varying peer mixing rule after a block of local adaptive updates, allows the block length to differ across workers, and does not depend on pipeline parallelism.

3 Method and Background

Notation. We consider n workers with local models $x_i \in \mathbb{R}^d$. We write $x = (x_1, \dots, x_n)$ for the collection of local models and denote by $\bar{x}$ their average. Let $\pi \triangleq I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ be the projector onto the disagreement subspace. We encode communication by a weighted graph Laplacian

$$\Lambda \triangleq \frac{1}{2}\sum_{(i,j)\in\mathcal{E}} \lambda_{ij}(e_i - e_j)(e_i - e_j)^\top, \qquad \text{with} \sum_{i,(ij)\in\mathcal{E}} \lambda_{ij} = 1 \text{ and } \lambda_{ij} > 0\,,$$

so that $\Lambda\mathbf{1} = 0$ and also let $\mathcal{N}_i = \{j, (ij) \in \mathcal{E}\} \cup \{i\}$. In this case, the graph connectivity is determined by $\chi \geq 1$, denoting the inverse of the smallest nonzero eigenvalue, the spectral gap (Nabli & Oyallon, 2023).

Algorithm 1 GASLoC

Input: inner steps $\{H_i\}_{i=1}^n$, inner optimizer $\mathrm{Optimizer}^{\mathrm{in}}$, outer optimizer $\mathrm{Optimizer}^{\mathrm{out}}$
for $t = 0, 1, \ldots, T-1$ **do**
  for each worker $i \in \{1, \ldots, n\}$ **in parallel do**
    $x_i^{(t,0)} \leftarrow x_i^{(t)}$
    for $h = 0, 1, \ldots, H_i - 1$ **do**
      $x_i^{(t,h+1)} \leftarrow \mathrm{Optimizer}^{\mathrm{in}}\left(x_i^{(t,h)}, \nabla f(x_i^{(t,h)})\right)$
    end for
    $y_i^{(t)} \leftarrow x_i^{(t,H_i)}$
  end for
  Sample a shared permutation $\sigma^{(t)}$ of $\{1, \ldots, n\}$
  for each worker $i \in \{1, \ldots, n\}$ **in parallel do**
    $\mathcal{N}_i \leftarrow \textsc{sample_peers}(\sigma^{(t)}, i)$ ▷ Algorithm 2
    $x_i^{(t+1)} \leftarrow \mathrm{Optimizer}^{\mathrm{out}}\left(x_i^{(t)}, \frac{1}{|\mathcal{N}_i^{(t)}|}\sum_{j \in \mathcal{N}_i^{(t)}}\left(x_i^{(t)} - y_j^{(t)}\right)\right)$
  end for
end for
Output: $\bar{x}^{(T)} = \frac{1}{n}\sum_{i=1}^n x_i^{(T)}$

Classical communication acceleration. To isolate the effect of communication, first consider consensus, how well the local models x_i agree, on a fixed connected graph. With the normalization above, the nonzero eigenvalues of Λ lie in $[1/\chi, 1]$. Writing $u_t = \pi x_t$ for the disagreement, ordinary gossip gives

$$u_{t+1} = (I - \alpha\Lambda)u_t, \qquad \|u_t\| \le (1 - \alpha/\chi)^t \|u_0\|, \qquad 0 < \alpha \le 1.$$

Thus, for a constant communication step size α, reducing disagreement by a factor ε requires $O(\chi \log(1/\varepsilon))$ communication rounds. Communication acceleration aims to reduce this number of rounds by retaining information from previous exchanges. Classical accelerated gossip uses polynomial recursions or momentum to suppress slowly mixing disagreement modes more quickly (Scaman et al., 2017; Loizou et al., 2019; Berthier et al., 2020). For example, a momentum recursion has the form

$$u_{t+1} = (I - \alpha\Lambda)u_t + \gamma(u_t - u_{t-1}).$$

With suitable parameters and initialization, classical accelerated schemes improve the fixed-graph communication complexity to $O(\sqrt{\chi}\log(1/\varepsilon))$. Each round still uses the same graph; the additional memory is stored locally. Below, we connect this consensus mechanism to outer momentum when communication rounds are separated by local optimization steps.

GASLoC. We describe GASLoC in the inner-loop setting for SGD, while Algorithm 1 gives a more general formulation for arbitrary inner/outer optimizers. In parallel, each node performs local gradient steps, and an outer loop periodically aggregates the resulting updates. The number of local steps H_i may be adapted across nodes to compensate for a straggler worker. More precisely, for each node i, the SGD version of the method is written as follows, where $\eta, \alpha, \beta > 0$ are learning rates:

$$\textbf{Inner loop} \qquad\qquad \textbf{Outer loop}$$

$$\begin{cases} x_{t,0}^i = x_t^i, \\ x_{t,h+1}^i = x_{t,h}^i - \dfrac{\beta}{H_i}\nabla f(x_{t,h}^i), \qquad h = 0, \ldots, H_i - 1, \end{cases} \qquad \begin{cases} g_t^i \triangleq x_{t,H_i}^i - x_{t,0}^i, \\ x_{t+1} = x_t + \eta g_t - \alpha\Lambda(x_t + \eta g_t). \end{cases} \tag{1}$$

When $\Lambda = \pi$, this algorithm recovers the DiLoCo and FEDOPT (Reddi et al., 2021; Douillard et al., 2024) framework exactly, and therefore can be viewed as a generalization of it. This connects DiLoCo's outer-optimizer memory to classical communication acceleration, when the graph is not complete the outer optimizer in our framework can be seen to have a precise function: accelerating concensus. Compared with DAdam (Wang et al., 2025), our method allows both multiple local steps and outer momentum. In the SGD formulation above, we introduce the following momentum update

$$x_{t+1} = x_t + \eta g_t - \alpha\Lambda(x_t + \eta g_t) + \gamma\Big[(x_t + \eta g_t) - (x_{t-1} + \eta g_{t-1})\Big], \qquad t \ge 1, \tag{2}$$

where $\gamma > 0$ is the momentum parameter. Setting $g_t = 0$ and projecting onto the disagreement subspace recovers the classical momentum recursion above. In GASLoC, momentum instead acts on the post-local-update iterates $x_t + \eta g_t$, which also contain the intervening optimization drift.

3.1 RANDOMIZED PEER-TO-PEER COMMUNICATION

Classical 1-Peer communication schemes typically rely on carefully constructed graph sequences tailored to the underlying network topology (Ying et al., 2021; Wang et al., 2025). While these schemes are attractive due to their low per-round communication cost, their mixing properties can be poor: exponential graphs yield a spectral gap scaling as $\log n$, whereas cycles lead to a spectral gap scaling as n^2, thereby slowing convergence. Such sparse communication patterns remain highly desirable in heterogeneous settings, where limiting communication and synchronization is essential. Our approach preserves this low communication cost while leveraging randomization (Nabli et al., 2023) to improve the average mixing behavior.

Randomized 1-Peer and 2-Peer communication schemes. We propose randomized peer communication to reduce synchronization costs while preserving good mixing properties. For simplicity, assume that $n = 2m$. At each communication round, we sample a permutation $\sigma : \{1, \dots, n\} \to \{1, \dots, n\}$ uniformly at random. The Laplacian of the corresponding 1-*Peer communication graph* is

$$\Lambda_\sigma^{(1)} = \frac{1}{2} \sum_{i=1}^{m} \big(e_{\sigma(2i-1)} - e_{\sigma(2i)}\big)\big(e_{\sigma(2i-1)} - e_{\sigma(2i)}\big)^\top.$$

Figure 1 illustrates an example of the randomized 1-Peer case, where each round activates a different sparse matching. We also consider a 2-*Peer communication graph*, obtained by connecting the nodes in the cyclic order induced by σ. Namely, using the convention $\sigma(n+1) = \sigma(1)$, define

$$\Lambda_\sigma^{(2)} = \frac{1}{2} \sum_{i=1}^{n} \frac{1}{2}\big(e_{\sigma(i)} - e_{\sigma(i+1)}\big)\big(e_{\sigma(i)} - e_{\sigma(i+1)}\big)^\top.$$

Algorithm 2 gives the corresponding sampling procedure for both communication schemes.

Proposition 1. *Let σ be sampled uniformly at random from the permutations of $\{1, \dots, n\}$. Then*

$$\mathbb{E}_\sigma\big[\Lambda_\sigma^{(1)}\big] = \mathbb{E}_\sigma\big[\Lambda_\sigma^{(2)}\big] = \frac{n}{2(n-1)}\pi, \quad \mathbb{E}_\sigma\big[\big(\Lambda_\sigma^{(1)}\big)^2\big] = \frac{n}{2(n-1)}\pi, \quad \mathbb{E}_\sigma\big[\big(\Lambda_\sigma^{(2)}\big)^2\big] = \frac{3n}{8(n-1)}\pi$$

Consequently, in expectation, the spectral gap of both the 1-Peer and 2-Peer communication schemes scales as 1. This contrasts with a fixed cycle graph, for which χ scales as n^2 and becomes inefficient for large n. Moreover, the lower variance of the 2-Peer scheme explains its gains over 1-Peer.

Adapting local steps to system heterogeneity. To compensate for differences in round time, Algorithm 1 allows the number of local steps H_i to vary across workers. We quantify these differences through the computation and communication times of each worker. Specifically, T_i^{comp} denotes the time required by worker i for one local step with fixed batch size. The communication time T_i^{comm} denotes the time required to send or receive one payload equal in size to the model. In the following comparison, we consider a bandwidth straggler. We assume symmetric links with full duplex communication, ignore latency and network contention, and allow peer transfers to overlap with computation.

In DiLoCo, all workers perform the same number H of local steps before participating in a global All-Reduce. A ring All-Reduce communicates $2(n-1)/n$ model equivalents per worker,which approaches 2 as n increases. Its modeled outer round duration is

$$T_{\mathrm{round}}^{\mathrm{AllReduce}} = \max_i \{H T_i^{\mathrm{comp}}\} + 2\frac{n-1}{n} \max_i \{T_i^{\mathrm{comm}}\}.$$

By contrast, GASLoC communicates with $k \in \{1, 2\}$ peers and does not require workers to participate in the same collective. Under the overlap model above, its outer round duration is

$$T_{\mathrm{round}}^{\mathbf{GASLoC},k} = \max_i \{H_i T_i^{\mathrm{comp}} + k T_i^{\mathrm{comm}}\}.$$

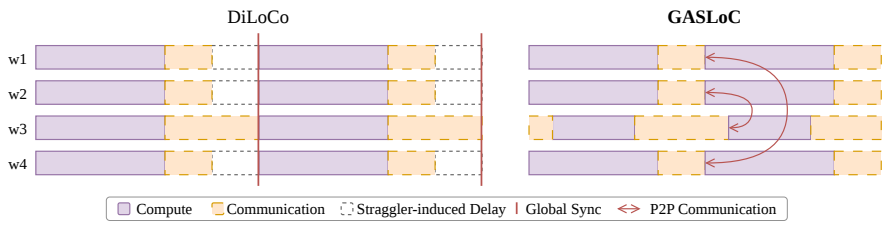


Figure 2: **Scheduling under bandwidth heterogeneity.** Left: DiLoCo performs $H = 30$ local steps per worker before a global All-Reduce delayed by the straggler. Right: GASLoC assigns $H_3 < 30$ steps to the straggler, allowing its slower peer exchange to start earlier and overlap with other workers' computation without blocking all workers.

Reducing H_i for a worker with limited bandwidth allows its communication to start earlier, thus reducing the modeled round duration.

For a prescribed maximum H, let τ be the target duration of well-connected workers that perform H steps, and choose $H_i \in \{1, \dots, H\}$ to satisfy

$$kT_i^{\text{comm}} + H_i T_i^{\text{comp}} \approx \tau.$$

A bandwidth straggler then offsets its larger T_i^{comm} with a smaller H_i, allowing its communication to start earlier while the other workers remain close to H. This rule also handles compute heterogeneity by assigning fewer steps to slower workers. However, a smaller H_i processes fewer batches per round and may require more rounds to match a fixed token budget.

Fault tolerance. Sparse communication can improve robustness to stragglers and transient failures: each synchronization round depends only on the activated peer exchanges, so a slowdown affects only nearby workers, assuming failed exchanges can be skipped or rescheduled. In contrast, All-Reduce requires every worker to join each collective step, making it more sensitive to stragglers and temporary failures.

3.2 Convergence of GASLoC

Appendix A proves the following convergence result for different communication schemes.

Proposition 2. *Let $x^\star \in \arg\min f$, and suppose that $x^\star$ is an unconstrained minimizer, so that $\nabla f(x^\star) = 0$. Suppose that, for every ξ, $F(\cdot;\xi)$ is L-smooth. Assume moreover that*

$$\mathbb{E}_\xi[\nabla F(x;\xi)] = \nabla f(x), \qquad \mathbb{E}_\xi\left[\|\nabla F(x;\xi) - \nabla f(x)\|^2\right] \le \sigma^2.$$

Assume the dynamic is given by Eq. 1, and that

$$0 < \alpha < 1, \qquad 0 < \beta \le \frac{1}{8L}, \qquad 0 < \eta \le \frac{\alpha}{3\sqrt{3}\beta\rho L}.$$

The parameter (ρ, γ) is set to $(\chi, 0)$ for standard gossip and to $(\sqrt{\chi}, (1 - \sqrt{\alpha/\chi})^2)$ for accelerated gossip. For 1-Peer, it is set to $\left(\frac{\alpha}{1-\sqrt{1-\frac{n}{n-1}\alpha+\frac{n}{2(n-1)}\alpha^2}}, 0\right)$, while for 2-Peer it is set to $\left(\frac{\alpha}{1-\sqrt{1-\frac{n}{n-1}\alpha+\frac{3n}{8(n-1)}\alpha^2}}, 0\right)$.

If the initialization satisfies $x_0^1 = \cdots = x_0^n = x_0$, then, for every $T \ge 1$,

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla f(\bar{x}_t)\|^2\right] \le \frac{4(f(x_0) - f(x^\star))}{T\eta\beta} + \frac{2L\eta\beta\sigma^2}{n^2}\sum_{i=1}^{n}\frac{1}{H_i} + 60\beta^2L^2\sigma^2.$$

This proposition shows that, by appropriately tuning the step sizes β, η, the residual terms in the convergence bound can be made arbitrarily small, which is the expected behavior for this type of method. We note that further tuning the momentum, as done in Douillard et al. (2024) is helpful to

Table 1: **Single-step decentralized training.** Validation loss with one local step on 16 replicas on both the 134M and 551M modelstrained on 2.7B and 11B tokens, respectively. At both model scales, GASLoC-2-Peer gives the lowest validation loss among decentralized methods.

	134M		551M	
AdamW DDP (*ref.*)	3.18		2.63	
Topology	**GASLoC**	DAdam	**GASLoC**	DAdam
ring	3.36	**3.33**	**2.71**	2.73
complete	3.29	**3.25**	**2.70**	2.71
1-Peer	**3.24**	3.29	**2.71**	2.72
2-Peer	**3.22**	3.28	**2.69**	2.71

Table 2: **End-to-end training time comparison.** Measured time is seconds with 8xA6000 GPUs. GPU connectivity was limited using the `tc` command to 2, 5, and 10 Gbps. We observe that 1-peer can significantly accelerate wall clock time, while 2-peer is similar to DDP but more fault tolerant.

Method	**10 Gbps**	**5 Gbps**	**2 Gbps**
AdamW DDP	7261 s	9241 s	15229 s
GASLoC-1-Peer	6408 s	7543 s	10973 s
GASLoC-2-Peer	7549 s	9837 s	16650 s

improve constants, yet not asymptotic convergence rates, which is well-known (Nabli & Oyallon, 2023). The proof relies on a Lyapunov function which combines an optimization term, $f(\bar{x}) - f(x^\star)$, that measures progress of the network average, and a consensus term that controls the disagreement between nodes. The precise form of the consensus term depends on the gossip mechanism used. The resulting bound is standard in spirit and is comparable to existing results such as (Koloskova et al., 2020). However, the analysis highlights the benefit of the outer momentum mechanism: it reduces the number of communication rounds required to achieve a given level of consensus.

4 Experiments

4.1 Experimental Setup

We evaluate GASLoC while varying the model architecture and scale, communication topology, number of local updates per round, and worker count up to 32. The main experiments assume homogeneous workers, while a separate experiment considers a worker with limited bandwidth. All comparisons use the same number of processed tokens, evaluating decentraliezd methods on the averaged model $\bar{x}^{(T)}$.

We pre-train 134M and 551M Llama-3-style decoder-only Transformers (Grattafiori et al., 2024) and OLMoE-1B-7B (Muennighoff et al., 2025) on FineWeb (Penedo et al., 2024). Following Hoffmann et al. (2022), the dense models use 2.7B and 11B tokens, respectively, while OLMoE-1B-7B uses 26B tokens for its active parameters. All models use the LLaMA-2 tokenizer (Touvron et al., 2023), sequences of 2048 tokens, and a global batch of around 2M tokens in homogeneous settings.

All methods use AdamW (Loshchilov & Hutter, 2019) locally, with peak inner learning rates tuned separately and scheduled with warmup followed by cosine decay. In the main comparisons, workers perform $H \in \{1, 30\}$ local steps between communication rounds. When applicable, we tune the outer learning rate and Nesterov momentum. Appendix B provides additional details.

Baselines. We compare with DDP (Li et al., 2020) and DiLoCo (Douillard et al., 2024) as distributed baselines. For decentralized comparison with $H = 1$, we use DAdam (Wang et al., 2025), which combines adaptive local optimization with parameter mixing between neighboring workers. We also include NoLoCo (Kolehmainen et al., 2025) as a baseline in a subset of the evaluated settings and implement it with the same randomized 1-Peer communication scheme as GASLoC-1-Peer.

DAdam mixes neighboring parameters after every local optimizer step, so it is directly comparable to our protocol only when $H = 1$. For $H > 1$, delaying this mixing changes the optimization dynamics, not just the communication frequency. We therefore introduce Local-DAdam with the same H local steps blocks. It replaces the outer pseudo-gradient with DAdam's disagreement term computed from the models before each block. See Appendix C for additional details.

4.2 Results with Homogeneous Workers

Single-step regime. In the single-step regime, communication follows every optimizer update, which typically gives the best performance. This regime retains benefits of gossip such as fault tolerance, but cannot reduce communication frequency (Lian et al., 2017; Wang et al., 2025). Ta-

Table 3: **134M multi-step decentralized training.** Validation loss after training on 2.7B tokens with $H = 30$ local steps across communication topologies and replica counts. "—" indicates incompatible method-topology combinations. Across replica counts, GASLoC gives the lowest loss for every topology.

DiLoCo (*ref.*)	8 replicas 3.30			16 replicas 3.40			32 replicas 3.53		
Topology	**GASLoC**	Loc.-DAdam	NoLoCo	**GASLoC**	Loc.-DAdam	NoLoCo	**GASLoC**	Loc.-DAdam	NoLoCo
ring	**3.32**	3.51	3.44	**3.45**	3.69	3.56	**3.81**	4.00	3.90
4-regular	**3.31**	3.50	3.43	**3.43**	3.68	3.53	**3.59**	3.95	3.70
8-regular	—	—	—	**3.42**	3.67	3.52	**3.55**	3.91	3.65
16-regular	—	—	—	—	—	—	**3.54**	3.88	3.65
complete	**3.30**	3.47	3.42	**3.40**	3.65	3.51	**3.53**	3.86	3.64
1-Peer	**3.34**	3.51	3.47	**3.49**	3.66	3.59	**3.66**	3.88	3.75
2-Peer	**3.32**	3.48	3.43	**3.45**	3.66	3.55	**3.62**	3.86	3.70

Table 4: **Decentralized training on the OLMoE-7B.** Validation loss and downstream accuracy for the MoE trained on 26B tokens with 8 replicas. We report length-normalized accuracy and its standard error for ARC-E and HellaSwag (HSwag). GASLoC achieves lower validation losses and better downstream task performance than the decentralized baselines.

(a) **One-step**

Method	**Loss**	**ARC-E**	**HSwag**
AdamW DDP (*ref.*)	2.36	49.87 ± 1.02	52.71 ± 0.49
DAdam (complete)	2.42	46.89 ± 1.02	48.87 ± 0.49
GASLoC (2-Peer)	**2.37**	$\mathbf{47.60 \pm 1.02}$	$\mathbf{52.56 \pm 0.50}$

(b) **Multi-step**

Method	**Loss**	**ARC-E**	**HSwag**
DiLoCo (*ref.*)	2.51	43.56 ± 1.02	44.77 ± 0.50
Local-DAdam (complete)	2.55	43.03 ± 1.02	42.41 ± 0.49
GASLoC (2-Peer)	**2.53**	$\mathbf{43.56 \pm 1.02}$	$\mathbf{43.14 \pm 0.49}$

ble 1 compares GASLoC and DAdam across topologies at both dense model scales. Since complete GASLoC is equivalent to DiLoCo, its gap to DDP is consistent with prior work, which found $H = 1$ to be the weakest evaluated communication interval (Charles et al., 2026). However, randomized peer communication reduces this gap, with GASLoC-2-Peer performing best among the decentralized methods. Both randomized variants also improve over complete DAdam despite communicating with only one or two peers rather than all workers. For GASLoC-2-Peer, the advantage increases to 0.06 when both methods use the same randomized pattern. For the 551M model, GASLoC improves over DAdam for every tested communication topology, with GASLoC-2-Peer remaining the strongest decentralized method. Table 4a shows that GASLoC-2-Peer also outperforms complete DAdam on the 7B MoE in validation loss and downstream performance. Under uniformly limited bandwidth, Table 2 shows that GASLoC-1-Peer completes the same training workload faster than DDP. The saved time allows it to perform additional steps within DDP's time budget, and Figure 3 shows that the corresponding improvement in final validation loss increases as bandwidth decreases. Overall, these results suggest that randomized sparse communication can mitigate the degradation observed with complete-graph updates, while remaining competitive on the challenging ring graph.

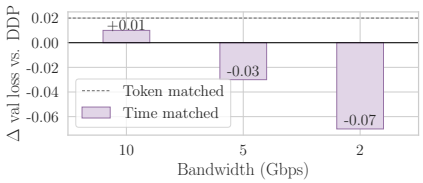


Figure 3: **Time-matched training under limited bandwidth.** Validation loss difference between GASLoC-1-Peer and DDP training the 134M on 8xA6000 GPUs and uniform bandwidth controlled using `tc`. At equal training time, GASLoC-1-Peer outperforms DDP under tighter bandwidth limits.

Multi-step regime. The $H = 30$ regime tests whether sparse communication can preserve validation loss when workers communicate less often. Each worker performs a block of local updates before communication, so the outer step must correct both local drift and disagreement between replicas. The main question is whether GASLoC can remain close to DiLoCo without global collectives. Table 3 shows that GASLoC on the complete graph matches this reference

Table 5: **551M multi-step decentralized training.** Validation loss after training on 11B tokens with $H = 30$ inner steps. GASLoC outperforms the decentralized baseline.

DiLoCo (*ref.*)	8 replicas 2.64		16 replicas 2.67	
Topology	**GASLoC**	Loc.-DAdam	**GASLoC**	Loc.-DAdam
ring	**2.68**	2.92	**2.79**	3.07
complete	**2.64**	2.84	**2.67**	2.97
1-Peer	**2.67**	2.93	**2.81**	3.05
2-Peer	**2.64**	2.91	**2.72**	3.04

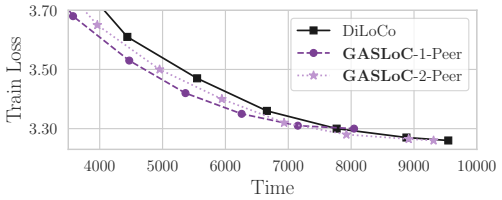


Figure 4: **Training with a bandwidth straggler.** Training loss over wall-clock time for the 134M model with 8 workers, one of which limited to 10% of the other workers. Both GASLoC variants reduce training loss faster than DiLoCo.

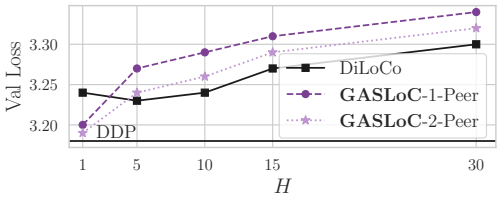


Figure 5: **Sensitivity to communication frequency.** Final validation loss for the 134M model with 8 workers for $H \in \{1, 5, 10, 15, 30\}$. Both GASLoC variants follow the same overall trend as DiLoCo as H increases.

across replica counts, as expected from their equivalence. For fixed k-regular graphs, increasing k reduces the gap from complete communication, with the ring graph ramaining close at smaller replica counts and degrading at 32. The randomized peer variants remain close to the reference, with 2-Peer communication consistently performing better than 1-Peer. By contrast, Local-DAdam performs consistently worse and does not benefit from randomized peer communication. Tables 4b and 5 show that GASLoC-2-Peer matches DiLoCo at 8 replicas and remains the closest sparse variant at 16 on the 551M model, while outperforming complete Local-DAdam in validation loss and downstream performance on the MoE. All GASLoC variants also outperform the corresponding Local-DAdam baselines, supporting the role of the outer update when local drift accumulates.

4.3 Heterogeneous Communication Setting

To evaluate the methods in the presence of a bandwidth straggler, we train the 134M model on eight A6000 GPUs, cap all links at 1 Gbps using `tc` command in linux, and limit one worker to 100 Mbps. This bandwidth range is common for distributed training over wide area networks (Wang et al., 2023; Jaghouar et al., 2024; Lidin et al., 2026). All methods process the same amount of tokens, and we measure their end-to-end training time. Following common implementations (Thérien et al., 2026), DiLoCo synchronizes its pseudo-gradient through an All-Reduce, whose duration is limited by the slowest link. GASLoC instead uses sparse randomized peer exchanges, keeping $H = 30$ for the other workers while reducing the the straggler's local step count to $H_i = 15$ for GASLoC-1-Peer and $H_i = 1$ for GASLoC-2-Peer. Because the straggler performs fewer local steps, both GASLoC variants require more outer steps to match the token budget. Even so, Figure 4 shows that GASLoC-1-Peer reduces end-to-end training time by about 16% compared to DiLoCo. GASLoC-2-Peer also reduces it about 2%, despite communicating 14% more data per round.

4.4 Sensitivity to Communication Frequency

Figure 5 compares the sensitivity of DiLoCo and the sparse variants to the number H of local steps between outer updates. At $H = 1$, both sparse variants achieve lower validation loss than DiLoCo, consistently with its reported degradation at small H (Charles et al., 2026). From $H = 5$ onward, the validation losses increase at similar rates over the evaluated values of H. Throughout this range, GASLoC-2-Peer remains closer to DiLoCo than GASLoC-1-Peer. Therefore, we can conclude that sparse communication mainly changes the loss level rather than its dependence on H in this regime.

5 Conclusion

We introduced GASLoC, a decentralized pre-training method that combines local optimizer steps, randomized peer communication, and an outer optimizer. Under complete communication, GASLoC recovers DiLoCo, while its sparse variants avoid global collectives. We establish convergence with worker-specific local step counts and show that outer momentum can accelerate consensus. Empirically, GASLoC improves over the evaluated decentralized baselines when communicating after every local step and remains close to DiLoCo when communication is less frequent. In a bandwidth-straggler experiment, GASLoC reaches a similar validation loss in less end-to-end training time. Together, these results establish GASLoC as a valid decentralized alternative to DiLoCo that avoids global collectives while retaining comparable validation loss.

AI USE STATEMENT

Generative AI tools were used to improve the clarity and grammar of the manuscript, to assist with the implementation of standard software functions, and to support code review. They were not used to formulate the research questions, develop the proposed method, design the experiments, derive the theoretical results, or interpret the empirical findings. The authors reviewed and revised every AI generated content.

REPRODUCIBILITY STATEMENT

Appendix A provides the complete proofs of our theoretical results, while Appendices B–E report the experimental protocol, implementation and baseline details, hyperparameters, and complete results.

REFERENCES

Atish Agarwala. High dimensional theory of two-phase optimizers. *arXiv preprint arXiv:2603.26954*, 2026. doi: 10.48550/arXiv.2603.26954. URL https://arxiv.org/abs/2603.26954.

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. PMLR, 2019.

Sangmin Bae, Yujin Kim, Reza Bayat, Sungnyun Kim, Jiyoun Ha, Tal Schuster, Adam Fisch, Hrayr Harutyunyan, Ziwei Ji, Aaron Courville, and Se-Young Yun. Mixture-of-recursions: Learning dynamic recursive depths for adaptive token-level computation, 2025. URL https://arxiv.org/abs/2507.10524.

Raphaël Berthier, Francis Bach, and Pierre Gaillard. Accelerated gossip in networks of given dimension using jacobi polynomial iterations. *SIAM Journal on Mathematics of Data Science*, 2 (1):24–47, 2020.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE transactions on information theory*, 52(6):2508–2530, 2006.

Zachary Charles, Gabriel Teston, Lucio M. Dery, J Keith Rush, Nova Fallen, Zachary Garrett, Arthur Szlam, and Arthur Douillard. Communication-efficient language model training scales reliably and robustly: Scaling laws for diloco. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=X4SCxcgb3O.

Yiming Chen, Kun Yuan, Yingya Zhang, Pan Pan, Yinghui Xu, and Wotao Yin. Accelerating gossip sgd with periodic global averaging. In *International Conference on Machine Learning*, pp. 1791–1802. PMLR, 2021.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018. URL https://arxiv.org/abs/1803.05457.

Arthur Douillard, Qixuan Feng, Andrei Alex Rusu, Rachita Chhaparia, Yani Donchev, Adhiguna Kuncoro, MarcAurelio Ranzato, Arthur Szlam, and Jiajun Shen. Diloco: Distributed low-communication training of language models. In *2nd Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@ICML 2024)*, 2024. URL https://openreview.net/forum?id=pICSfWkJIk.

Arthur Douillard, Yani Donchev, J Keith Rush, Satyen Kale, Zachary Charles, Gabriel Teston, Zachary Garrett, Jiajun Shen, Ross McIlroy, David Lacey, Alexandre Rame, Arthur Szlam, MarcAurelio Ranzato, and Paul R Barham. Streaming diloco with overlapping communication. In *Second*

Conference on Language Modeling, 2025. URL https://openreview.net/forum?id=yYk3zK0X6Q.

Arthur Douillard, Keith Rush, Yani Donchev, Zachary Charles, Nova Fallen, Ayush Dubey, Ionel Gog, Josef Dean, Blake Woodworth, Zachary Garrett, Nate Keating, Jenny Bishop, Henry Prior, Edouard Yvinec, Arthur Szlam, Marc'Aurelio Ranzato, and Jeff Dean. Decoupled diloco for resilient distributed pre-training, 2026. URL https://arxiv.org/abs/2604.21428.

Mathieu Even, Raphaël Berthier, Francis Bach, Nicolas Flammarion, Hadrien Hendrikx, Pierre Gaillard, Laurent Massoulié, and Adrien Taylor. Continuized accelerations of deterministic and stochastic gradient descents, and of gossip algorithms. *Advances in Neural Information Processing Systems*, 34:28054–28066, 2021.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 2024. URL https://zenodo.org/records/12608602.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei

Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 30016–30030. Curran Associates, Inc., 2022. doi: 10.52202/

068431-2176. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/c1e2faff6f588870935f114ebe04a3e5-Paper-Conference.pdf.

Sami Jaghouar, Jack Min Ong, Manveer Basra, Fares Obeid, Jannik Straube, Michael Keiblinger, Elie Bakouch, Lucas Atkins, Maziyar Panahi, Charles Goddard, Max Ryabinin, and Johannes Hagemann. Intellect-1 technical report, 2024. URL https://arxiv.org/abs/2412.01152.

Paul Janson, Edouard Oyallon, and Eugene Belilovsky. Stabilizing native low-rank LLM pretraining, 2026. URL https://arxiv.org/abs/2602.12429.

Satyen Kale, Arthur Douillard, and Yanislav Donchev. Eager updates for overlapped communication and computation in diloco. *arXiv preprint arXiv:2502.12996*, 2025.

Ahmed Khaled, Satyen Kale, Arthur Douillard, Chi Jin, Rob Fergus, and Manzil Zaheer. Understanding outer optimizers in local sgd: Learning rates, momentum, and acceleration. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 100324–100377. Curran Associates, Inc., 2025. doi: 10.52202/085713-3354. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/90ad0e850532986dff56da49bc599904-Paper-Conference.pdf.

Geon-Woo Kim, Junbo Li, Shashidhar Gandham, Omar Baldonado, Adithya Gangidi, Pavan Balaji, Zhangyang Wang, and Aditya Akella. HALoS: Hierarchical asynchronous local SGD over slow networks for geo-distributed large language model training. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 30548–30566. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/kim25y.html.

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

Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A unified theory of decentralized sgd with changing topology and local updates. In *International conference on machine learning*, pp. 5381–5393. PMLR, 2020.

Chamin P Hewa Koneputugodage, Thalaiyasingam Ajanthan, Sameera Ramasinghe, Hadi Mohaghegh Dolatabadi, Shamane Siriwardhana, Gil Avraham, Violetta Shevchenko, Karol Pajak, James Snewin, and Alexander Long. Factored gossip diloco: Reducing blocking communication within diloco. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=1MynM9hkFH.

Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016. URL https://arxiv.org/abs/1610.05492.

Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, and Soumith Chintala. Pytorch distributed: experiences on accelerating data parallel training. *Proc. VLDB Endow.*, 13(12):3005–3018, 2020. ISSN 2150-8097. doi: 10.14778/3415478.3415530. URL https://doi.org/10.14778/3415478.3415530.

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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 5336–5346. Curran Associates Inc., 2017.

Wanchao Liang, Tianyu Liu, Less Wright, Will Constable, Andrew Gu, Chien-Chin Huang, Iris Zhang, Wei Feng, Howard Huang, Junjie Wang, Sanket Purandare, Gokul Nadathur, and Stratos Idreos. Torchtitan: One-stop pytorch native solution for production ready LLM pretraining. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=SFN6Wm7YBI.

Joel Lidin, Amir Sarfi, Erfan Miahi, Quentin Anthony, Shivam Chauhan, Evangelos Pappas, Benjamin Thérien, Eugene Belilovsky, and Samuel Dare. Covenant-72b: Pre-training a 72b llm with trustless peers over-the-internet, 2026. URL https://arxiv.org/abs/2603.08163.

Bo Liu, Rachita Chhaparia, Arthur Douillard, Satyen Kale, Andrei Alex Rusu, Jiajun Shen, Arthur Szlam, and MarcAurelio Ranzato. Asynchronous local-SGD training for language modeling. In *2nd Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@ICML 2024)*, 2024. URL https://openreview.net/forum?id=mT3PdRKl40.

Nicolas Loizou, Michael Rabbat, and Peter Richtárik. Provably accelerated randomized gossip algorithms. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7505–7509, 2019. doi: 10.1109/ICASSP.2019.8683847.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu (eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 20–22 Apr 2017. URL https://proceedings.mlr.press/v54/mcmahan17a.html.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. OLMoE: Open mixture-of-experts language models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xXTkbTBmqq.

Adel Nabli and Edouard Oyallon. Dadao: Decoupled accelerated decentralized asynchronous optimization. In *International Conference on Machine Learning*, pp. 25604–25626. PMLR, 2023.

Adel Nabli and Edouard Oyallon. Decentralized asynchronous optimization with dadao allows decoupling and acceleration. *Journal of Machine Learning Research*, 26(207):1–48, 2025.

Adel Nabli, Eugene Belilovsky, and Edouard Oyallon. $\mathbf{A^2CiD^2}$: Accelerating asynchronous communication in decentralized deep learning. *Advances in Neural Information Processing Systems*, 36: 47451–47474, 2023.

Adel Nabli, Louis Fournier, Pierre ERBACHER, Louis Serrano, Eugene Belilovsky, and Edouard Oyallon. Acco: Accumulate while you communicate for communication-overlapped sharded llm training. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 69278–69305. Curran Associates, Inc., 2025.

Kenta Niwa, Yuki Takezawa, Guoqiang Zhang, and W. Bastiaan Kleijn. Revisiting 1-peer exponential graph for enhancing decentralized learning efficiency. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Pitch Patarasuk and Xin Yuan. Bandwidth optimal all-reduce algorithms for clusters of workstations. *J. Parallel Distrib. Comput.*, 69(2):117–124, 2009.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=LkFG3lB13U5.

Amir Sarfi, Benjamin Thérien, Joel Lidin, and Eugene Belilovsky. Communication efficient LLM pre-training with sparseloco. In *OPT 2025: Optimization for Machine Learning*, 2025. URL https://openreview.net/forum?id=o3fzlo9xDT.

Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *international conference on machine learning*, pp. 3027–3036. PMLR, 2017.

Sebastian U. Stich. Local SGD converges fast and communicates little. In *International Conference on Learning Representations*, 2019.

Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning*, pp. 1139–1147, 2013. URL LlamaTeam.

Benjamin Thérien, Xiaolong Huang, Aaron Defazio, Irina Rish, and Eugene Belilovsky. Muloco: Muon is a practical inner optimizer for diloco. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=OBXBXSgwcD.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Oskar van der Wal, Pietro Lesci, Max Müller-Eberstein, Naomi Saphra, Hailey Schoelkopf, Willem Zuidema, and Stella R Biderman. Polypythias: Stability and outliers across fifty language model pre-training runs. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 86180–86204, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/d611d06e3207330555fbc10810e70163-Paper-Conference.pdf.

Thijs Vogels, Lie He, Anastasiia Koloskova, Sai Praneeth Karimireddy, Tao Lin, Sebastian U Stich, and Martin Jaggi. Relaysum for decentralized deep learning on heterogeneous data. *Advances in Neural Information Processing Systems*, 34:28004–28015, 2021.

Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael Rabbat. Slowmo: Improving communication-efficient distributed sgd with slow momentum. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkxJ8REYPH.

Jue Wang, Yucheng Lu, Binhang Yuan, Beidi Chen, Percy Liang, Christopher De Sa, Christopher Re, and Ce Zhang. CocktailSGD: Fine-tuning foundation models over 500Mbps networks. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 36058–36076. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/wang23t.html.

Zesen Wang, Jiaojiao Zhang, Xuyang Wu, and Mikael Johansson. From promise to practice: Realizing high-performance decentralized training. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 26204–26249, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/41b93c59da0d0f835907fd661d419db2-Paper-Conference.pdf.

Bicheng Ying, Kun Yuan, Yiming Chen, Hanbin Hu, PAN PAN, and Wotao Yin. Exponential graph is provably efficient for decentralized deep training. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 13975–13987. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/74e1ed8b55ea44fd7dbb685c412568a4-Paper.pdf.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Sixin Zhang, Anna E Choromanska, and Yann LeCun. Deep learning with elastic averaging sgd. *Advances in neural information processing systems*, 28, 2015.

A Proof of Proposition 2

Proposition 2. *Let $x^\star \in \arg\min f$, and suppose that $x^\star$ is an unconstrained minimizer, so that $\nabla f(x^\star) = 0$. Suppose that, for every ξ, $F(\cdot;\xi)$ is L-smooth. Assume moreover that the stochastic gradients are unbiased and have variance bounded by σ^2, namely*

$$\mathbb{E}_\xi[\nabla F(x;\xi)] = \nabla f(x), \qquad \mathbb{E}_\xi\left[\|\nabla F(x;\xi) - \nabla f(x)\|^2\right] \le \sigma^2.$$

Assume that $0 < \alpha < 1, 0 < \beta \le \frac{1}{8L}$, and that the outer step size satisfies $0 < \eta \le \frac{\alpha}{3\sqrt{3\beta\rho L}}$. The constant ρ depends on the communication scheme as follows:

1. ***Standard gossip with spectral gap χ:*** *we take*
$$\rho = \chi, \qquad \gamma = 0.$$
2. ***Accelerated gossip with spectral gap χ via outer momentum:*** *we take*
$$\rho = \sqrt{\chi}, \qquad \gamma = \left(1 - \sqrt{\frac{\alpha}{\chi}}\right)^2.$$
3. ***k-Peer gossip:*** *we take*
$$\rho = \begin{cases} \dfrac{\alpha}{1 - \sqrt{1 - \frac{n}{n-1}\alpha + \frac{n}{2(n-1)}\alpha^2}}, & \text{for randomized } 1\text{-Peer communication}, \\ \dfrac{\alpha}{1 - \sqrt{1 - \frac{n}{n-1}\alpha + \frac{3n}{8(n-1)}\alpha^2}}, & \text{for randomized } 2\text{-Peer communication}, \end{cases} \qquad \gamma = 0.$$

If the initialization satisfies $x_0^1 = \cdots = x_0^n = x_0$, then, for every $T \ge 1$,

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla f(\bar{x}_t)\|^2\right] \le \frac{4(f(x_0) - f(x^\star))}{T\eta\beta} + \frac{2L\eta\beta\sigma^2}{n^2}\sum_{i=1}^{n}\frac{1}{H_i} + 60\beta^2L^2\sigma^2.$$

$$\bar{x} \triangleq \frac{1}{n}\sum_{i=1}^{n} x^i$$

denote the network average. We introduce the Bregman divergence associated with f:

$$d_f(x,y) \triangleq f(x) - f(y) - \langle \nabla f(y),\, x - y\rangle. \tag{3}$$

The following standard identities will be used repeatedly:

$$d_F(x,z) = d_F(x,y) + d_F(y,z) + \langle x - y, \nabla F(y) - \nabla f(z)\rangle, \tag{4}$$

$$d_F(x,y) \le \frac{L}{2}\|x - y\|^2, \tag{5}$$

$$d_F(x,x^\star) \ge 0. \tag{6}$$

Note that the above also holds for $f = \mathbb{E}[F]$.

For the inner loop, recall that for each node i,

$$x^i_{t,k+1} = x^i_{t,k} - \frac{\beta}{H_i}\nabla F(x^i_{t,k}; \xi^i_{t,k}), \qquad k = 0, \dots, H_i - 1, \tag{7}$$

with $x^i_{t,0} = x^i_t$. We define the accumulated local update

$$g^i_t \triangleq -\frac{\beta}{H_i}\sum_{k=0}^{H_i-1} \nabla F(x^i_{t,k}; \xi^i_{t,k}), \tag{8}$$

so that

$$x^i_{t,H_i} = x^i_t + g^i_t. \tag{9}$$

Stacking the local vectors, we write $g_t = (g^1_t, \dots, g^n_t)$, and the outer update becomes

$$x_{t+1} = x_t + \eta g_t - \alpha\Lambda(x_t + \eta g_t). \tag{10}$$

Let

$$\bar{x}_t \triangleq \frac{1}{n}\sum_{i=1}^n x^i_t, \qquad \bar{g}_t \triangleq \frac{1}{n}\sum_{i=1}^n g^i_t.$$

Since $\Lambda\mathbf{1} = 0$, averaging the outer update yields

$$\bar{x}_{t+1} = \bar{x}_t + \eta\bar{g}_t. \tag{11}$$

Applying the three-point identity for d_f with

$$x = \bar{x}_{t+1}, \qquad y = \bar{x}_t, \qquad z = x^\star,$$

we obtain

$$d_f(\bar{x}_{t+1}, x^\star) - d_f(\bar{x}_t, x^\star) = d_f(\bar{x}_{t+1}, \bar{x}_t) + \langle \bar{x}_{t+1} - \bar{x}_t, \nabla f(\bar{x}_t) - \nabla f(x^\star)\rangle \tag{12}$$

$$= d_f(\bar{x}_t + \eta\bar{g}_t, \bar{x}_t) + \eta\left\langle \bar{g}_t, \nabla f(\bar{x}_t) - \nabla f(x^\star)\right\rangle. \tag{13}$$

If $x^\star$ is an unconstrained minimizer, then $\nabla f(x^\star) = 0$, and therefore

$$d_f(\bar{x}_{t+1}, x^\star) - d_f(\bar{x}_t, x^\star) = d_f(\bar{x}_t + \eta\bar{g}_t, \bar{x}_t) + \eta\langle \bar{g}_t, \nabla f(\bar{x}_t)\rangle. \tag{14}$$

Using L-smoothness,

$$d_f(\bar{x}_t + \eta\bar{g}_t, \bar{x}_t) \le \frac{L\eta^2}{2}\|\bar{g}_t\|^2. \tag{15}$$

Then, by the bias–variance decomposition,

$$\mathbb{E}\|\bar{g}_t\|^2 = \|\mathbb{E}_t[\bar{g}_t]\|^2 + \mathbb{E}\,\|\bar{g}_t - \mathbb{E}[\bar{g}_t]\|^2 \tag{16}$$

$$= \left\|\frac{1}{n}\sum_{i=1}^n \mathbb{E}[g^i_t]\right\|^2 + \frac{1}{n^2}\sum_{i=1}^n \mathbb{E}\left\|g^i_t - \mathbb{E}[g^i_t]\right\|^2 \tag{17}$$

$$\le \beta^2\left\|\frac{1}{n}\sum_{i=1}^n \frac{1}{H_i}\sum_{k=0}^{H_i-1}\nabla f(x^i_{t,k})\right\|^2 + \frac{\beta^2\sigma^2}{n^2}\sum_{i=1}^n\frac{1}{H_i}. \tag{18}$$

Now, we will use that, taking the expectations:

$$\mathbb{E}\langle \bar{g}_t, \nabla f(\bar{x}_t)\rangle = \frac{1}{n}\sum_{i=1}^n \mathbb{E}\langle g^i_t, \nabla f(\bar{x}_t)\rangle \tag{19}$$

Here, using Lemma 2, we have:

$$\begin{aligned}\mathbb{E}\left[\left\langle g^i_t, \nabla f(\bar{x}_t)\right\rangle\right] \le -\beta\left(\frac{1}{2} - \frac{3}{2}\beta^2L^2e^2\right)\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 - \frac{\beta}{2}\mathbb{E}\|\frac{1}{H_i}\sum_{k=0}^{H_i-1}\nabla f(x^i_{t,k})\|^2 \\ + \frac{3}{2}\beta L^2e^2\mathbb{E}\|x^i_t - \bar{x}_t\|^2 + \frac{3}{2}\beta^3L^2e^2\sigma^2.\end{aligned} \tag{20}$$

Introduce $G_t^i = \frac{1}{H_i}\sum_{k=0}^{H_i-1}\nabla f(x_{t,k}^i)$. Combining the last two inequalities yields

$$\begin{aligned}
&\mathbb{E}[d_f(\bar{x}_{t+1},x^\star) - d_f(\bar{x}_t,x^\star)] \\
&\le -\eta\beta\left(\frac{1}{2} - \frac{3}{2}\beta^2L^2e^2\right)\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 \\
&\quad - \frac{\beta\eta}{2n}\sum_{i=1}^n \mathbb{E}\left\|G_t^i\right\|^2 + \frac{\eta^2L\beta^2}{2}\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n G_t^i\right\|^2 \\
&\quad + \frac{3\eta\beta L^2e^2}{2n}\sum_{i=1}^n \mathbb{E}\|x_t^i - \bar{x}_t\|^2 \\
&\quad + \frac{L\eta^2\beta^2\sigma^2}{2n^2}\sum_{i=1}^n \frac{1}{H_i} + \frac{3}{2}\eta\beta^3L^2e^2\sigma^2. \qquad (21)
\end{aligned}$$

Next, we use the communication contraction bound given by Lemma 1.

$$\Delta V_\pi \le -\frac{\alpha}{\rho}\|\pi x_t\|^2 + \frac{\rho\eta^2\beta^2}{\alpha}\|\pi G_t\|^2.$$

Introduce the Lyapunov function

$$\phi \triangleq d_f(\bar{x},x^\star) + \Omega V_\pi(x).$$

Combining the previous inequalities gives

$$\begin{aligned}
\mathbb{E}[\phi_{t+1} - \phi_t] &\le -\eta\beta\left(\frac{1}{2} - \frac{3}{2}\beta^2L^2e^2\right)\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 \\
&\quad - \frac{\beta\eta}{2n}\sum_{i=1}^n \mathbb{E}\left\|G_t^i\right\|^2 + \frac{\eta^2L\beta^2}{2}\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n G_t^i\right\|^2 \\
&\quad + \frac{3\eta\beta L^2e^2}{2n}\sum_{i=1}^n \mathbb{E}\|x_t^i - \bar{x}_t\|^2 \\
&\quad + \frac{L\eta^2\beta^2\sigma^2}{2n^2}\sum_{i=1}^n \frac{1}{H_i} + \frac{3}{2}\eta\beta^3L^2e^2\sigma^2 \\
&\quad + \Omega\left(-\frac{\alpha}{\rho}\mathbb{E}\|\pi x_t\|^2 + \frac{\rho\eta^2\beta^2}{\alpha}\mathbb{E}\|\pi G_t\|^2\right). \qquad (22)
\end{aligned}$$

Using

$$\sum_{i=1}^n \mathbb{E}\|G_t^i\|^2 = \mathbb{E}\|\pi G_t\|^2 + n\mathbb{E}\|\bar{G}_t\|^2, \qquad \bar{G}_t \triangleq \frac{1}{n}\sum_{i=1}^n G_t^i,$$

we recombine the inequalities and obtain

$$\begin{aligned}
\mathbb{E}[\phi_{t+1} - \phi_t] &\le -\eta\beta\left(\frac{1}{2} - \frac{3}{2}\beta^2L^2e^2\right)\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 \\
&\quad - \left(\frac{\beta\eta}{2} - \frac{L\beta^2\eta^2}{2}\right)\mathbb{E}\|\bar{G}_t\|^2 \\
&\quad - \left(\frac{\beta\eta}{2n} - \Omega\frac{\rho\eta^2\beta^2}{\alpha}\right)\mathbb{E}\|\pi G_t\|^2 \\
&\quad - \left(\Omega\frac{\alpha}{\rho} - \frac{3\eta\beta L^2e^2}{2n}\right)\mathbb{E}\|\pi x_t\|^2 \\
&\quad + \frac{L\eta^2\beta^2\sigma^2}{2n^2}\sum_{i=1}^n \frac{1}{H_i} + \frac{3}{2}\eta\beta^3L^2e^2\sigma^2. \qquad (23)
\end{aligned}$$

Here, we gonna pick $\Omega = \frac{3\rho\eta\beta L^2 e^2}{2\alpha n}$

so that:

$$\left(-\frac{\beta\eta}{2n} + \Omega\frac{\beta^2\rho\eta^2}{\alpha}\right) = -\frac{\beta\eta}{2n} + \frac{3\beta^2\rho^2\eta^3\beta L^2 e^2}{2\alpha^2 n} \tag{24}$$

$$= \frac{\beta\eta}{2n}\left(-1 + \frac{3\rho^2\beta^2\eta^2 L^2 e^2}{\alpha^2}\right). \tag{25}$$

Thus it is necessary so that $\frac{3\rho^2\beta^2\eta^2 L^2 e^2}{\alpha^2} \leq 1$, $\frac{3}{2}\beta^2 L^2 e^2 \leq \frac{1}{4}$ and $L\beta\eta \leq 1$

Then, we obtain

$$\mathbb{E}[\phi_{t+1} - \phi_t] \leq -\frac{\eta\beta}{4}\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 + \frac{L\eta^2\beta^2\sigma^2}{2n^2}\sum_{i=1}^{n}\frac{1}{H_i} + \frac{3}{2}\eta\beta^3 L^2 e^2\sigma^2.$$

Summing the above inequality from $t = 0$ to $T - 1$, we get

$$\sum_{t=0}^{T-1}\mathbb{E}[\phi_{t+1} - \phi_t] \leq -\frac{\eta\beta}{4}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 + \sum_{t=0}^{T-1}\left(\frac{L\eta^2\beta^2\sigma^2}{2n^2}\sum_{i=1}^{n}\frac{1}{H_i} + \frac{3}{2}\eta\beta^3 L^2 e^2\sigma^2\right) \tag{26}$$

$$= -\frac{\eta\beta}{4}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 + T\left(\frac{L\eta^2\beta^2\sigma^2}{2n^2}\sum_{i=1}^{n}\frac{1}{H_i} + \frac{3}{2}\eta\beta^3 L^2 e^2\sigma^2\right). \tag{27}$$

Since the left-hand side telescopes, we have

$$\sum_{t=0}^{T-1}\mathbb{E}[\phi_{t+1} - \phi_t] = \mathbb{E}[\phi_T] - \phi_0.$$

Therefore,

$$\mathbb{E}[\phi_T] - \phi_0 \leq -\frac{\eta\beta}{4}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 + T\left(\frac{L\eta^2\beta^2\sigma^2}{2n^2}\sum_{i=1}^{n}\frac{1}{H_i} + \frac{3}{2}\eta\beta^3 L^2 e^2\sigma^2\right). \tag{28}$$

Since $\phi_T \geq 0$, it follows that

$$-\phi_0 \leq \mathbb{E}[\phi_T] - \phi_0.$$

Combining this with the previous inequality gives

$$-\phi_0 \leq -\frac{\eta\beta}{4}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 + T\left(\frac{L\eta^2\beta^2\sigma^2}{2n^2}\sum_{i=1}^{n}\frac{1}{H_i} + \frac{3}{2}\eta\beta^3 L^2 e^2\sigma^2\right). \tag{29}$$

Rearranging, we obtain

$$\frac{\eta\beta}{4}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 \leq \phi_0 + T\left(\frac{L\eta^2\beta^2\sigma^2}{2n^2}\sum_{i=1}^{n}\frac{1}{H_i} + \frac{3}{2}\eta\beta^3 L^2 e^2\sigma^2\right). \tag{30}$$

Dividing both sides by $T\eta\beta/4$, we get

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 \leq \frac{4\phi_0}{T\eta\beta} + \frac{4}{\eta\beta}\left(\frac{L\eta^2\beta^2\sigma^2}{2n^2}\sum_{i=1}^{n}\frac{1}{H_i} + \frac{3}{2}\eta\beta^3 L^2 e^2\sigma^2\right) \tag{31}$$

$$= \frac{4(f(x_0) - f(x^*))}{T\eta\beta} + \frac{2L\eta\beta\sigma^2}{n^2}\sum_{i=1}^{n}\frac{1}{H_i} + 6\beta^2 L^2 e^2\sigma^2. \tag{32}$$

Lemma 1 (Gossip lemma)**.** ***(i) Standard gossip.*** *Assume $\gamma = 0$. Then*

$$\|\pi x_{t+1}\|^2 - \|\pi x_t\|^2 \leq -\frac{\alpha}{\chi}\|\pi x_t\|^2 + \frac{\chi}{\alpha}\|\pi y_t\|^2.$$

(ii) Random k-Peer gossip. *Assume $\gamma = 0$. For the randomized 1-Peer scheme, the bound holds with*

$$\chi_1 = \frac{\alpha}{1 - \sqrt{1 - \frac{n}{n-1}\alpha + \frac{n}{2(n-1)}\alpha^2}},$$

namely

$$\mathbb{E}_t\left[\|\pi x_{t+1}\|^2\right] - \|\pi x_t\|^2 \le -\frac{\alpha}{\chi_1}\|\pi x_t\|^2 + \frac{\chi_1}{\alpha}\|\pi y_t\|^2.$$

For the randomized 2-Peer scheme, the bound holds with

$$\chi_2 = \frac{\alpha}{1 - \sqrt{1 - \frac{n}{n-1}\alpha + \frac{3n}{8(n-1)}\alpha^2}}$$

as

$$\mathbb{E}_t\left[\|\pi x_{t+1}\|^2\right] - \|\pi x_t\|^2 \le -\frac{\alpha}{\chi_2}\|\pi x_t\|^2 + \frac{\chi_2}{\alpha}\|\pi y_t\|^2.$$

(iii) Accelerated gossip. *Assume*

$$\gamma = \left(1 - \sqrt{\frac{\alpha}{\chi}}\right)^2, \qquad 0 < \alpha \le 1,$$

and define

$$X_t := \begin{pmatrix} \pi x_t \\ \pi x_{t-1} \end{pmatrix}.$$

Then,

$$\|X_{t+1}\|^2 - \|X_t\|^2 \le -\frac{\alpha}{\sqrt{\chi}}\|X_t\|^2 + \frac{\sqrt{\chi}}{\alpha}\|\pi y_t\|^2 \le -\frac{\alpha}{\sqrt{\chi}}\|\pi x_t\|^2 + \frac{\sqrt{\chi}}{\alpha}\|\pi y_t\|^2.$$

Proof. We prove the three claims separately.

Proof of (i). Since $\Lambda \mathbf{1} = 0$, we have

$$\pi\Lambda = \Lambda\pi = \Lambda.$$

Thus, when $\gamma = 0$,

$$\pi x_{t+1} = (I - \alpha\Lambda)(\pi x_t + \pi y_t).$$

On the disagreement subspace, the eigenvalues of $I - \alpha\Lambda$ are

$$1 - \alpha\lambda_r(\Lambda), \qquad r = 2, \dots, n.$$

Since $0 < \alpha \le 1$ and

$$\lambda_r(\Lambda) \in \left[\frac{1}{\chi}, 1\right],$$

we get

$$\|(I - \alpha\Lambda)u\| \le \left(1 - \frac{\alpha}{\chi}\right)\|u\|$$

for all $u \perp \mathbf{1}$. Hence

$$\|\pi x_{t+1}\|^2 \le \left(1 - \frac{\alpha}{\chi}\right)^2 \|\pi x_t + \pi y_t\|^2.$$

Using Young's inequality,

$$\|a + b\|^2 \le (1 + \nu)\|a\|^2 + \left(1 + \frac{1}{\nu}\right)\|b\|^2,$$

with

$$\nu = \frac{1}{1 - \alpha/\chi} - 1,$$

we obtain

$$\|\pi x_{t+1}\|^2 - \|\pi x_t\|^2 \le -\frac{\alpha}{\chi}\|\pi x_t\|^2 + \frac{\chi}{\alpha}\left(1-\frac{\alpha}{\chi}\right)^2\|\pi y_t\|^2.$$

Since $\left(1-\frac{\alpha}{\chi}\right)^2 \le 1$, this gives

$$\|\pi x_{t+1}\|^2 - \|\pi x_t\|^2 \le -\frac{\alpha}{\chi}\|\pi x_t\|^2 + \frac{\chi}{\alpha}\|\pi y_t\|^2.$$

Proof of (ii). We have

$$\mathbb{E}_t\left[\|\pi x_{t+1}\|^2\right] = \mathbb{E}_t\left[\|(I-\alpha\Lambda_t)\pi(x_t+y_t)\|^2\right] \tag{33}$$

$$= (x_t+y_t)^\top \mathbb{E}_t\left[\pi - 2\alpha\Lambda_t + \alpha^2\Lambda_t^2\right](x_t+y_t). \tag{34}$$

For the 1-Peer scheme, since

$$\mathbb{E}[\Lambda_t] = \mathbb{E}[\Lambda_t^2] = \frac{n}{2(n-1)}\pi,$$

we obtain

$$\mathbb{E}_t\left[\|\pi x_{t+1}\|^2\right] \le \left(1 - \frac{n}{n-1}\alpha + \frac{n}{2(n-1)}\alpha^2\right)\|\pi x_t + \pi y_t\|^2.$$

Let

$$q_1 = 1 - \frac{n}{n-1}\alpha + \frac{n}{2(n-1)}\alpha^2.$$

Then, by Young's inequality,

$$q_1\|\pi x_t + \pi y_t\|^2 \le \sqrt{q_1}\|\pi x_t\|^2 + \frac{q_1}{1-\sqrt{q_1}}\|\pi y_t\|^2.$$

Therefore,

$$\mathbb{E}_t\left[\|\pi x_{t+1}\|^2\right] - \|\pi x_t\|^2 \le -\left(1-\sqrt{q_1}\right)\|\pi x_t\|^2 + \frac{q_1}{1-\sqrt{q_1}}\|\pi y_t\|^2.$$

Since $q_1 \le 1$, this implies

$$\mathbb{E}_t\left[\|\pi x_{t+1}\|^2\right] - \|\pi x_t\|^2 \le -\frac{\alpha}{\chi_1}\|\pi x_t\|^2 + \frac{\chi_1}{\alpha}\|\pi y_t\|^2, \qquad \chi_1 = \frac{\alpha}{1-\sqrt{q_1}}.$$

For the 2-Peer scheme, using

$$\mathbb{E}[\Lambda_t] = \frac{n}{2(n-1)}\pi, \qquad \mathbb{E}[\Lambda_t^2] = \frac{3n}{8(n-1)}\pi,$$

we obtain

$$\mathbb{E}_t\left[\|\pi x_{t+1}\|^2\right] \le \left(1 - \frac{n}{n-1}\alpha + \frac{3n}{8(n-1)}\alpha^2\right)\|\pi x_t + \pi y_t\|^2.$$

Let

$$q_2 = 1 - \frac{n}{n-1}\alpha + \frac{3n}{8(n-1)}\alpha^2.$$

Then, by Young's inequality,

$$q_2\|\pi x_t + \pi y_t\|^2 \le \sqrt{q_2}\|\pi x_t\|^2 + \frac{q_2}{1-\sqrt{q_2}}\|\pi y_t\|^2.$$

Therefore,

$$\mathbb{E}_t\left[\|\pi x_{t+1}\|^2\right] - \|\pi x_t\|^2 \le -(1-\sqrt{q_2})\|\pi x_t\|^2 + \frac{q_2}{1-\sqrt{q_2}}\|\pi y_t\|^2.$$

Since $q_2 \le 1$, this implies

$$\mathbb{E}_t\left[\|\pi x_{t+1}\|^2\right] - \|\pi x_t\|^2 \le -\frac{\alpha}{\chi_2}\|\pi x_t\|^2 + \frac{\chi_2}{\alpha}\|\pi y_t\|^2,$$

with

$$\chi_2 = \frac{\alpha}{1 - \sqrt{1 - \frac{n}{n-1}\alpha + \frac{3n}{8(n-1)}\alpha^2}}.$$

(iii) Accelerated gossip. Since $\pi\Lambda = \Lambda\pi = \Lambda$, applying π to the accelerated update gives

$$\pi x_{t+1} = \big((1+\gamma)I - \alpha\Lambda\big)\pi x_t - \gamma\pi x_{t-1} + (I - \alpha\Lambda)\pi y_t.$$

Therefore,

$$X_{t+1} = AX_t + \begin{pmatrix}(I - \alpha\Lambda)\pi y_t \\ 0\end{pmatrix}.$$

with

$$A = \begin{pmatrix}(1+\gamma)I - \alpha\Lambda & -\gamma I \\ I & 0\end{pmatrix}.$$

Let (u, v) be an eigenvector of A with eigenvalue r. Since $u = rv$, and since $\Lambda v = \lambda v$ with

$$\lambda \in \left[\frac{1}{\chi}, 1\right],$$

we get

$$r^2 - (1 - \alpha\lambda + \gamma)r + \gamma = 0.$$

Set

$$q := 1 - \sqrt{\frac{\alpha}{\chi}}, \qquad \gamma = q^2.$$

Then

$$0 \le 1 - \alpha\lambda + \gamma \le 2q,$$

so the roots have modulus at most q. Thus, we have that

$$\|AX_t\| \le (1 - \sqrt{\frac{\alpha}{\chi}})\|X_t\|.$$

Consequently, using triangular inequality

$$\|X_{t+1}\| \le q\|X_t\| + \left\|\begin{pmatrix}(I - \alpha\Lambda)\pi y_t \\ 0\end{pmatrix}\right\|.$$

Since $0 < \alpha \le 1$ and $\lambda_r(\Lambda) \in [1/\chi, 1]$, we have

$$\|(I - \alpha\Lambda)\pi y_t\| \le \|\pi y_t\|.$$

Therefore,

$$\|X_{t+1}\| \le \left(1 - \sqrt{\frac{\alpha}{\chi}}\right)\|X_t\| + \|\pi y_t\|.$$

Using Young inequality,

$$\|X_{t+1}\|^2 \le \left(1 - \sqrt{\frac{\alpha}{\chi}}\right)^2 (1+\nu)\|X_t\|^2 + (1 + \frac{1}{\nu})\|\pi y_t\|^2.$$

Choose

$$\nu = \frac{1}{1 - \sqrt{\alpha/\chi}} - 1.$$

Then

$$\left(1 - \sqrt{\frac{\alpha}{\chi}}\right)^2 (1+\nu) = 1 - \sqrt{\frac{\alpha}{\chi}},$$

and

$$1 + \frac{1}{\nu} = \frac{1}{\sqrt{\alpha/\chi}} = \sqrt{\frac{\chi}{\alpha}}.$$

Therefore,

$$\|X_{t+1}\|^2 \le \left(1 - \sqrt{\frac{\alpha}{\chi}}\right)\|X_t\|^2 + \sqrt{\frac{\chi}{\alpha}}\|\pi y_t\|^2.$$

Subtracting $\|X_t\|^2$, we obtain

$$\|X_{t+1}\|^2 - \|X_t\|^2 \le -\sqrt{\frac{\alpha}{\chi}}\|X_t\|^2 + \sqrt{\frac{\chi}{\alpha}}\|\pi y_t\|^2.$$

Since $0 < \alpha \le 1$, this also implies the looser bound

$$\|X_{t+1}\|^2 - \|X_t\|^2 \le -\frac{\alpha}{\sqrt{\chi}}\|X_t\|^2 + \frac{\sqrt{\chi}}{\alpha}\|\pi y_t\|^2.$$

□

Proposition 3. *Let σ be sampled uniformly at random from the set of permutations of $\{1, \dots, n\}$. Then*

$$\mathbb{E}_\sigma\big[\Lambda_\sigma^{(1)}\big] = \mathbb{E}_\sigma\big[\Lambda_\sigma^{(2)}\big] = \frac{n}{2(n-1)}\pi, \qquad \mathbb{E}_\sigma\big[\big(\Lambda_\sigma^{(1)}\big)^2\big] = \frac{n}{2(n-1)}\pi \qquad \mathbb{E}_\sigma\big[\big(\Lambda_\sigma^{(2)}\big)^2\big] = \frac{3n}{8(n-1)}\pi$$

Proof. We first consider the 1-Peer scheme. For any unordered pair $\{a, b\}$ with $a \ne b$, the probability that a and b are matched together in a uniformly random perfect matching is

$$\mathbb{P}\big(\{a,b\} \text{ is an edge}\big) = \frac{1}{n-1}.$$

Therefore,

$$\mathbb{E}_\sigma\big[\Lambda_\sigma^{(1)}\big] = \frac{1}{2}\sum_{1 \le a < b \le n} \mathbb{P}\big(\{a,b\} \text{ is an edge}\big)(e_a - e_b)(e_a - e_b)^\top.$$

Using the above probability gives

$$\mathbb{E}_\sigma\big[\Lambda_\sigma^{(1)}\big] = \frac{1}{2(n-1)}\sum_{1 \le a < b \le n} (e_a - e_b)(e_a - e_b)^\top.$$

The standard identity

$$\sum_{1 \le a < b \le n} (e_a - e_b)(e_a - e_b)^\top = nI - \mathbf{1}\mathbf{1}^\top = n\pi$$

then yields

$$\mathbb{E}_\sigma\big[\Lambda_\sigma^{(1)}\big] = \frac{n}{2(n-1)}\pi.$$

The proof for the 2-Peer scheme is analogous. In a uniformly random cycle induced by σ, any unordered pair $\{a, b\}$ appears as an edge with probability

$$\mathbb{P}\big(\{a,b\} \text{ is an edge}\big) = \frac{2}{n-1}.$$

Indeed, once the position of a in the cycle is fixed, the node b can occupy either of the two neighboring positions among the remaining $n - 1$ positions. Hence,

$$\mathbb{E}_\sigma\big[\Lambda_\sigma^{(2)}\big] = \frac{1}{4}\sum_{1 \le a < b \le n} \frac{2}{n-1}(e_a - e_b)(e_a - e_b)^\top.$$

Using the same identity as above, we obtain

$$\mathbb{E}_\sigma\big[\Lambda_\sigma^{(2)}\big] = \frac{n}{2(n-1)}\pi.$$

It remains to compute the second moments. We first consider the 1-Peer scheme. Recall that

$$\Lambda_\sigma^{(1)} = \frac{1}{2}\sum_{i=1}^{m} u_i u_i^\top, \qquad u_i := e_{\sigma(2i-1)} - e_{\sigma(2i)}.$$

Since the pairs form a matching, the vectors u_i have disjoint supports. Hence,

$$u_i^\top u_j = 0 \quad \text{for } i \neq j, \qquad u_i^\top u_i = 2.$$

Therefore,

$$\begin{aligned}
\left(\Lambda_\sigma^{(1)}\right)^2 &= \frac{1}{4}\sum_{i,j=1}^{m} u_i u_i^\top u_j u_j^\top \\
&= \frac{1}{4}\sum_{i,j=1}^{m} u_i (u_i^\top u_j) u_j^\top \\
&= \frac{1}{4}\sum_{i=1}^{m} 2 u_i u_i^\top \\
&= \frac{1}{2}\sum_{i=1}^{m} u_i u_i^\top \\
&= \Lambda_\sigma^{(1)}.
\end{aligned}$$

Taking expectations gives

$$\mathbb{E}_\sigma\left[(\Lambda_\sigma^{(1)})^2\right] = \mathbb{E}_\sigma\left[\Lambda_\sigma^{(1)}\right] = \frac{n}{2(n-1)}\pi.$$

We now consider the 2-Peer scheme. Let

$$v_i := e_{\sigma(i)} - e_{\sigma(i+1)}, \qquad \sigma(n+1) = \sigma(1).$$

Then

$$\Lambda_\sigma^{(2)} = \frac{1}{4}\sum_{i=1}^{n} v_i v_i^\top.$$

For a fixed cycle, $\Lambda_\sigma^{(2)}$ is one fourth of the usual Laplacian of a cycle graph. Its eigenvalues are therefore

$$\lambda_k = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi k}{n}\right)\right), \qquad k = 0, \ldots, n-1.$$

Thus,

$$\begin{aligned}
\mathrm{Tr}\left[\left(\Lambda_\sigma^{(2)}\right)^2\right] &= \frac{1}{4}\sum_{k=0}^{n-1}\left(1 - \cos\left(\frac{2\pi k}{n}\right)\right)^2 \\
&= \frac{1}{4}\sum_{k=0}^{n-1}\left[1 - 2\cos\left(\frac{2\pi k}{n}\right) + \cos^2\left(\frac{2\pi k}{n}\right)\right].
\end{aligned}$$

Using

$$\sum_{k=0}^{n-1}\cos\left(\frac{2\pi k}{n}\right) = 0, \qquad \sum_{k=0}^{n-1}\cos^2\left(\frac{2\pi k}{n}\right) = \frac{n}{2},$$

we obtain

$$\mathrm{Tr}\left[\left(\Lambda_\sigma^{(2)}\right)^2\right] = \frac{1}{4}(n + \frac{n}{2}) = \frac{3n}{8}.$$

Moreover, the distribution of the random cycle is invariant under relabeling of the nodes. Hence

$$\mathbb{E}_\sigma\left[(\Lambda_\sigma^{(2)})^2\right]$$

must act identically on all directions orthogonal to $\mathbf{1}$ and must vanish on $\mathrm{span}\{\mathbf{1}\}$. Therefore, there exists a scalar α such that

$$\mathbb{E}_\sigma\left[(\Lambda_\sigma^{(2)})^2\right] = \alpha\pi.$$

Taking traces yields

$$\frac{3n}{8} = \mathrm{Tr}\left(\mathbb{E}_\sigma\left[(\Lambda_\sigma^{(2)})^2\right]\right) = \mathrm{Tr}(\alpha\Pi) = \alpha(n-1).$$

Hence

$$\alpha = \frac{3n}{8(n-1)}.$$

Consequently,

$$\mathbb{E}_\sigma\big[(\Lambda_\sigma^{(2)})^2\big] = \frac{3n}{8(n-1)}\pi.$$

This concludes the proof. □

Lemma 2 (Gradient mismatch with stochastic gradients)**.** *Assume f is L-smooth. Let $\nabla F(x;\xi)$ be an unbiased stochastic estimator of $\nabla f(x)$, satisfying*

$$\mathbb{E}\big[\nabla F(x;\xi) \mid x\big] = \nabla f(x), \qquad \mathbb{E}\big[\|\nabla F(x;\xi) - \nabla f(x)\|^2 \mid x\big] \le \sigma^2.$$

Let $0 < \beta \le 1/L$, and define

$$g_t^i := x_{t,H}^i - x_t^i,$$

where

$$x_{t,k+1}^i = x_{t,k}^i - \frac{\beta}{H}\nabla F(x_{t,k}^i;\xi_{t,k}^i), \qquad x_{t,0}^i = x_t^i.$$

Then,

$$\mathbb{E}\big[\langle g_t^i, \nabla f(\bar{x}_t)\rangle\big] \le -\beta\left(\frac{1}{2} - \frac{3}{2}\beta^2 L^2 e^2\right)\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 - \frac{\beta}{2}\mathbb{E}\|\frac{1}{H}\sum_{k=0}^{H-1}\nabla f(x_{t,k}^i)\|^2 + \frac{3}{2}\beta L^2 e^2 \mathbb{E}\|x_t^i - \bar{x}_t\|^2 + \frac{3}{2}\beta^3 L^2 e^2 \sigma^2.$$

Proof. We want to upper-bound

$$\mathbb{E}\big[\langle g_t^i, \nabla f(\bar{x}_t)\rangle\big].$$

By the local stochastic update rule,

$$x_{t,k+1}^i = x_{t,k}^i - \frac{\beta}{H}\nabla F(x_{t,k}^i;\xi_{t,k}^i), \qquad k = 0,\dots,H-1.$$

Iterating this recursion gives

$$x_{t,H}^i = x_t^i - \frac{\beta}{H}\sum_{k=0}^{H-1}\nabla F(x_{t,k}^i;\xi_{t,k}^i).$$

Hence

$$g_t^i := -x_t^i + x_{t,H}^i = -\frac{\beta}{H}\sum_{k=0}^{H-1}\nabla F(x_{t,k}^i;\xi_{t,k}^i).$$

Assume that $\mathcal{F}_{t,k}^i$ contains all randomness before sampling $\xi_{t,k}^i$, so that $x_{t,k}^i$ and $\bar{x}_t$ are $\mathcal{F}_{t,k}^i$-measurable. Also assume conditional unbiasedness:

$$\mathbb{E}\big[\nabla F(x_{t,k}^i;\xi_{t,k}^i) \mid \mathcal{F}_{t,k}^i\big] = \nabla f_i(x_{t,k}^i).$$

Then

$$\begin{aligned}\mathbb{E}\big[\langle g_t^i, \nabla f(\bar{x}_t)\rangle\big] &= -\frac{\beta}{H}\sum_{k=0}^{H-1}\mathbb{E}\big[\langle \nabla F(x_{t,k}^i;\xi_{t,k}^i), \nabla f(\bar{x}_t)\rangle\big] \\ &= -\frac{\beta}{H}\sum_{k=0}^{H-1}\mathbb{E}\big[\langle \nabla f_i(x_{t,k}^i), \nabla f(\bar{x}_t)\rangle\big].\end{aligned}$$

In the homogeneous case $f_i = f$, this becomes

$$\mathbb{E}\big[\langle g_t^i, \nabla f(\bar{x}_t)\rangle\big] = -\frac{\beta}{H}\sum_{k=0}^{H-1}\mathbb{E}\big[\langle \nabla f(x_{t,k}^i), \nabla f(\bar{x}_t)\rangle\big].$$

Now, we obtain

$$\begin{aligned}\mathbb{E}\big[\langle g_t^i, \nabla f(\bar{x}_t)\rangle\big] &= \beta\mathbb{E}\left[\left\langle -\frac{1}{H}\sum_{k=0}^{H-1}\nabla f(x_{t,k}^i), \nabla f(\bar{x}_t)\right\rangle\right] \\ &= -\frac{\beta}{2}\mathbb{E}\|\frac{1}{H}\sum_{k=0}^{H-1}\nabla f(x_{t,k}^i)\|^2 - \frac{\beta}{2}\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 \\ &\quad + \frac{\beta}{2}\mathbb{E}\left[\|\frac{1}{H}\sum_{k=0}^{H-1}(\nabla f(x_{t,k}^i) - \nabla f(\bar{x}_t))\|^2\right].\end{aligned}$$

Using the inner-loop drift estimate from Lemma 3,

$$\begin{aligned}\mathbb{E}\|x_{t,k}^i - \bar{x}_t\|^2 &\le 3\left(1+\frac{\beta L}{H}\right)^{2k}\mathbb{E}\|x_t^i - \bar{x}_t\|^2 \\ &\quad + 3\frac{\mathbb{E}\|\nabla f(\bar{x}_t)\|^2}{L^2}\left(\left(1+\frac{\beta L}{H}\right)^k - 1\right)^2 \\ &\quad + 3\frac{\beta\sigma^2}{L}\frac{\left(1+\frac{\beta L}{H}\right)^{2k}-1}{2+\frac{\beta L}{H}},\end{aligned}$$

and summing over $k = 0, \dots, H-1$, one obtains

$$\begin{aligned}\mathbb{E}\left[\sum_{k=0}^{H-1}\|x_{t,k}^i - \bar{x}_t\|^2\right] &\le 3\mathbb{E}\|x_t^i - \bar{x}_t\|^2\sum_{k=0}^{H-1}\left(1+\frac{\beta L}{H}\right)^{2k} \\ &\quad + 3\frac{\mathbb{E}\|\nabla f(\bar{x}_t)\|^2}{L^2}\sum_{k=0}^{H-1}\left[\left(1+\frac{\beta L}{H}\right)^k - 1\right]^2 \\ &\quad + 3\frac{\beta\sigma^2}{L}\frac{1}{2+\frac{\beta L}{H}}\sum_{k=0}^{H-1}\left[\left(1+\frac{\beta L}{H}\right)^{2k} - 1\right].\end{aligned}$$

Let

$$a := 1 + \frac{\beta L}{H}.$$

Then

$$\sum_{k=0}^{H-1} a^{2k} = \frac{a^{2H}-1}{a^2-1},$$

$$\sum_{k=0}^{H-1}(a^k - 1)^2 = \frac{a^{2H}-1}{a^2-1} - 2\frac{a^H - 1}{a-1} + H,$$

and

$$\sum_{k=0}^{H-1}(a^{2k} - 1) = \frac{a^{2H}-1}{a^2-1} - H.$$

Therefore,

$$\begin{aligned}\mathbb{E}\left[\sum_{k=0}^{H-1}\|x_{t,k}^i - \bar{x}_t\|^2\right] &\le 3\frac{a^{2H}-1}{a^2-1}\mathbb{E}\|x_t^i - \bar{x}_t\|^2 \\ &\quad + 3\frac{\mathbb{E}\|\nabla f(\bar{x}_t)\|^2}{L^2}\left[\frac{a^{2H}-1}{a^2-1} - 2\frac{a^H-1}{a-1} + H\right] \\ &\quad + 3\frac{\beta\sigma^2}{L}\frac{1}{2+\frac{\beta L}{H}}\left[\frac{a^{2H}-1}{a^2-1} - H\right].\end{aligned}$$

Next, by Jensen's inequality and L-smoothness,

$$\mathbb{E}\left[\left\|\frac{1}{H}\sum_{k=0}^{H-1}\left(\nabla f(x_{t,k}^i)-\nabla f(\bar{x}_t)\right)\right\|^2\right] \le \frac{1}{H}\sum_{k=0}^{H-1}\mathbb{E}\left\|\nabla f(x_{t,k}^i)-\nabla f(\bar{x}_t)\right\|^2 \le \frac{L^2}{H}\sum_{k=0}^{H-1}\mathbb{E}\left\|x_{t,k}^i-\bar{x}_t\right\|^2.$$

Thus,

$$\mathbb{E}\left[\left\langle g_t^i, \nabla f(\bar{x}_t)\right\rangle\right] \le -\frac{\beta}{2}\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 + \frac{\beta L^2}{2H}\sum_{k=0}^{H-1}\mathbb{E}\left\|x_{t,k}^i-\bar{x}_t\right\|^2.$$

Substituting the previous drift-sum bound yields

$$\begin{aligned}\mathbb{E}\left[\left\langle g_t^i, \nabla f(\bar{x}_t)\right\rangle\right] \le& -\frac{\beta}{2}\mathbb{E}\|\frac{1}{H}\sum_{k=0}^{H-1}\nabla f(x_{t,k}^i)\|^2 - \frac{\beta}{2}\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 \\ &+ \frac{3\beta L^2}{2H}\frac{a^{2H}-1}{a^2-1}\mathbb{E}\|x_t^i-\bar{x}_t\|^2 \\ &+ \frac{3\beta}{2H}\left[\frac{a^{2H}-1}{a^2-1}-2\frac{a^H-1}{a-1}+H\right]\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 \\ &+ \frac{3\beta^2 L\sigma^2}{2H}\frac{1}{2+\frac{\beta L}{H}}\left[\frac{a^{2H}-1}{a^2-1}-H\right].\end{aligned}$$

Since

$$a-1=\frac{\beta L}{H}, \qquad a^2-1=\frac{\beta L}{H}\left(2+\frac{\beta L}{H}\right),$$

we have

$$\frac{1}{H}\frac{a^{2H}-1}{a^2-1}=\frac{1}{\beta L}\frac{a^{2H}-1}{2+\frac{\beta L}{H}}.$$

Therefore,

$$\begin{aligned}\mathbb{E}\left[\left\langle g_t^i, \nabla f(\bar{x}_t)\right\rangle\right] \le& -\frac{\beta}{2}\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 - \frac{\beta}{2}\mathbb{E}\|\frac{1}{H}\sum_{k=0}^{H-1}\nabla f(x_{t,k}^i)\|^2 \\ &+ \frac{3L}{2}\frac{\left(1+\frac{\beta L}{H}\right)^{2H}-1}{2+\frac{\beta L}{H}}\mathbb{E}\|x_t^i-\bar{x}_t\|^2 \\ &+ \frac{3\beta}{2}\left[\frac{1}{\beta L}\frac{\left(1+\frac{\beta L}{H}\right)^{2H}-1}{2+\frac{\beta L}{H}}-\frac{2}{\beta L}\left(\left(1+\frac{\beta L}{H}\right)^H-1\right)+1\right]\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 \\ &+ \frac{3\beta^2 L\sigma^2}{2}\frac{1}{2+\frac{\beta L}{H}}\left[\frac{1}{\beta L}\frac{\left(1+\frac{\beta L}{H}\right)^{2H}-1}{2+\frac{\beta L}{H}}-1\right].\end{aligned}$$

Now assume $0<\beta L\le 1$. Then

$$\left(1+\frac{\beta L}{H}\right)^H \le e^{\beta L}, \qquad \left(1+\frac{\beta L}{H}\right)^{2H} \le e^{2\beta L},$$

and

$$2+\frac{\beta L}{H}\ge 2.$$

Hence

$$\frac{\left(1+\frac{\beta L}{H}\right)^{2H}-1}{2+\frac{\beta L}{H}} \le \frac{e^{2\beta L}-1}{2} \le \beta L e^{2\beta L} \le \beta L e^2.$$

Thus,

$$\frac{3L}{2}\frac{\left(1+\frac{\beta L}{H}\right)^{2H}-1}{2+\frac{\beta L}{H}}\mathbb{E}\|x_t^i-\bar{x}_t\|^2 \le \frac{3}{2}\beta L^2 e^2 \mathbb{E}\|x_t^i-\bar{x}_t\|^2.$$

Moreover,

$$\left(\left(1+\frac{\beta L}{H}\right)^k-1\right)^2 \le \left(e^{\beta L}-1\right)^2 \le \beta^2 L^2 e^2,$$

so

$$\frac{3\beta}{2H}\sum_{k=0}^{H-1}\left[\left(1+\frac{\beta L}{H}\right)^k-1\right]^2 \mathbb{E}\|\nabla f(\bar{x}_t)\|^2 \le \frac{3}{2}\beta^3 L^2 e^2 \mathbb{E}\|\nabla f(\bar{x}_t)\|^2.$$

Similarly,

$$\frac{\left(1+\frac{\beta L}{H}\right)^{2k}-1}{2+\frac{\beta L}{H}} \le \frac{e^{2\beta L}-1}{2} \le \beta L e^2,$$

and therefore

$$\frac{3\beta^2 L\sigma^2}{2H}\sum_{k=0}^{H-1}\frac{\left(1+\frac{\beta L}{H}\right)^{2k}-1}{2+\frac{\beta L}{H}} \le \frac{3}{2}\beta^3 L^2 e^2 \sigma^2.$$

Combining these estimates, we obtain

$$\begin{aligned}\mathbb{E}\big[\langle g_t^i, \nabla f(\bar{x}_t)\rangle\big] \le& -\frac{\beta}{2}\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 + \frac{3}{2}\beta^3 L^2 e^2 \mathbb{E}\|\nabla f(\bar{x}_t)\|^2 \\ &+ \frac{3}{2}\beta L^2 e^2 \mathbb{E}\|x_t^i-\bar{x}_t\|^2 + \frac{3}{2}\beta^3 L^2 e^2\sigma^2 - \frac{\beta}{2}\mathbb{E}\|\frac{1}{H}\sum_{k=0}^{H-1}\nabla f(x_{t,k}^i)\|^2.\end{aligned}$$

Equivalently,

$$\boxed{\begin{aligned}\mathbb{E}\big[\langle g_t^i, \nabla f(\bar{x}_t)\rangle\big] \le& -\beta\left(\frac{1}{2}-\frac{3}{2}\beta^2 L^2 e^2\right)\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 - \frac{\beta}{2}\mathbb{E}\|\frac{1}{H}\sum_{k=0}^{H-1}\nabla f(x_{t,k}^i)\|^2 \\ &+ \frac{3}{2}\beta L^2 e^2 \mathbb{E}\|x_t^i-\bar{x}_t\|^2 + \frac{3}{2}\beta^3 L^2 e^2\sigma^2.\end{aligned}}$$

This proves the desired bound whenever $0 < \beta L \le 1$. □

Lemma 3 (Inner-loop drift in norm with stochastic gradients)**.** *Assume that f is L-smooth. Let $\nabla F(x;\xi)$ be an unbiased stochastic estimator of $\nabla f(x)$, with variance bounded by σ^2, i.e.,*

$$\mathbb{E}\big[\nabla F(x;\xi)\big] = \nabla f(x), \qquad \mathbb{E}\big[\|\nabla F(x;\xi)-\nabla f(x)\|^2\big] \le \sigma^2.$$

Suppose that the inner-loop iterates satisfy

$$x_{t,k+1}^i = x_{t,k}^i - \frac{\beta}{H}\nabla F(x_{t,k}^i;\xi_{t,k}^i), \qquad x_{t,0}^i = x_t^i.$$

Then, for every $k \in \{0,\dots,H\}$,

$$\mathbb{E}\big[\|x_{t,k}^i-\bar{x}_t\|^2\big|x_t\big] \le 3\left(1+\frac{\beta L}{H}\right)^{2k}\mathbb{E}\|x_t^i-\bar{x}_t\|^2 + 3\frac{\mathbb{E}\|\nabla f(\bar{x}_t)\|^2}{L^2}\left(\left(1+\frac{\beta L}{H}\right)^k-1\right)^2 + 3\frac{\beta\sigma^2}{L}\frac{\left(1+\frac{\beta L}{H}\right)^{2k}-1}{2+\frac{\beta L}{H}} \tag{35}$$

Proof. Let

$$d_{t,k}^i := x_{t,k}^i - \bar{x}_t, \qquad \eta_{t,k}^i := \nabla F(x_{t,k}^i; \xi_{t,k}^i) - \nabla f(x_{t,k}^i).$$

Then

$$d_{t,k+1}^i = d_{t,k}^i - \frac{\beta}{H}\big(\nabla f(x_{t,k}^i) + \eta_{t,k}^i\big).$$

Taking norms and using the triangle inequality gives

$$\|d_{t,k+1}^i\| \le \|d_{t,k}^i\| + \frac{\beta}{H}\|\nabla f(x_{t,k}^i)\| + \frac{\beta}{H}\|\eta_{t,k}^i\|.$$

Since f is L-smooth, its gradient is L-Lipschitz. Hence

$$\|\nabla f(x_{t,k}^i)\| \le \|\nabla f(\bar{x}_t)\| + \|\nabla f(x_{t,k}^i) - \nabla f(\bar{x}_t)\| \le \|\nabla f(\bar{x}_t)\| + L\|d_{t,k}^i\|.$$

Therefore,

$$\|d_{t,k+1}^i\| \le \left(1 + \frac{\beta L}{H}\right)\|d_{t,k}^i\| + \frac{\beta}{H}\|\nabla f(\bar{x}_t)\| + \frac{\beta}{H}\|\eta_{t,k}^i\|.$$

Taking conditional expectation with respect to the randomness at step k, we obtain

$$\|d_{t,k+1}^i\| \le \left(1 + \frac{\beta L}{H}\right)\|d_{t,k}^i\| + \frac{\beta}{H}\|\nabla f(\bar{x}_t)\| + \frac{\beta}{H}\|\eta_{t,k}^i\|.$$

Let $q = 1 + \frac{\beta L}{H}$, iterating this recursion from 0 to $k-1$ yields

$$\|d_{t,k}^i\| \le q^k\|d_{t,0}^i\| + \frac{\beta}{H}\sum_{j=0}^{k-1} q^j\|\nabla f(\bar{x}_t)\| + \frac{\beta}{H}\sum_{j=0}^{k-1} q^j\|\eta_{t,k}^i\|.$$

Since $d_{t,0}^i = x_t^i - \bar{x}_t$, we obtain

$$\|x_{t,k}^i - \bar{x}_t\| \le \left(1 + \frac{\beta L}{H}\right)^k\|x_t^i - \bar{x}_t\| + \frac{\beta}{H}\|\nabla f(\bar{x}_t)\|\sum_{j=0}^{k-1}\left(1 + \frac{\beta L}{H}\right)^j + \frac{\beta}{H}\sum_{j=0}^{k-1} q^j\|\eta_{t,k}^i\|$$

Finally, if $L > 0$, then

$$\sum_{j=0}^{k-1}\left(1 + \frac{\beta L}{H}\right)^j = \frac{\left(1 + \frac{\beta L}{H}\right)^k - 1}{\frac{\beta L}{H}}.$$

Therefore,

$$\|x_{t,k}^i - \bar{x}_t\| \le \left(1 + \frac{\beta L}{H}\right)^k\|x_t^i - \bar{x}_t\| + \frac{\beta}{H}\|\nabla f(\bar{x}_t)\|\frac{\left(1 + \frac{\beta L}{H}\right)^k - 1}{\frac{\beta L}{H}} + \frac{\beta}{H}\sum_{j=0}^{k-1} q^j\|\eta_{t,k}^i\|$$

Consequently,

$$\|x_{t,k}^i - \bar{x}_t\|^2 \le 3(1 + \frac{\beta L}{H})^{2k}\|x_t^i - \bar{x}_t\|^2 + 3\frac{\|\nabla f(\bar{x}_t)\|^2}{L^2}((1 + \frac{\beta L}{H})^k - 1)^2$$
$$+ 3\frac{\beta^2}{H}\sum_{j=0}^{k-1}(1 + \frac{\beta L}{H})^{2j}\|\eta_{t,j}^i\|^2$$

and

$$
\begin{aligned}
\mathbb{E}\|x_{t,k}^i - \bar{x}_t\|^2 &\le 3\left(1+\frac{\beta L}{H}\right)^{2k}\mathbb{E}\|x_t^i-\bar{x}_t\|^2 + 3\frac{\mathbb{E}\|\nabla f(\bar{x}_t)\|^2}{L^2}\left(\left(1+\frac{\beta L}{H}\right)^k-1\right)^2 \\
&\quad + 3\frac{\beta^2\sigma^2}{H}\sum_{j=0}^{k-1}\left(1+\frac{\beta L}{H}\right)^{2j} \\
&= 3\left(1+\frac{\beta L}{H}\right)^{2k}\mathbb{E}\|x_t^i-\bar{x}_t\|^2 + 3\frac{\mathbb{E}\|\nabla f(\bar{x}_t)\|^2}{L^2}\left(\left(1+\frac{\beta L}{H}\right)^k-1\right)^2 \\
&\quad + 3\frac{\beta^2\sigma^2}{H}\frac{\left(1+\frac{\beta L}{H}\right)^{2k}-1}{\left(1+\frac{\beta L}{H}\right)^2-1} \\
&= 3\left(1+\frac{\beta L}{H}\right)^{2k}\mathbb{E}\|x_t^i-\bar{x}_t\|^2 + 3\frac{\mathbb{E}\|\nabla f(\bar{x}_t)\|^2}{L^2}\left(\left(1+\frac{\beta L}{H}\right)^k-1\right)^2 \\
&\quad + 3\frac{\beta\sigma^2}{L}\frac{\left(1+\frac{\beta L}{H}\right)^{2k}-1}{2+\frac{\beta L}{H}}.
\end{aligned}
$$

This proves the desired bound. □

B Experimental Protocol

Data. We train all models on FineWeb (Penedo et al., 2024). We tokenize the data with the LLaMA-2 tokenizer (Touvron et al., 2023) from the Together AI repository, which has a vocabulary of 32K tokens. We pack the tokenized data into sequences of 2048 tokens for a global batch containing 2M tokens divided across the n workers. Following the approximate allocation of 20 training tokens per parameter from Hoffmann et al. (2022), we train the 134M and 551M dense models on 2.7B and 11B tokens, respectively. Since OLMoE-1B-7B has 1.3B active parameters, we train it on 26B tokens under the same allocation.

Models. We evaluate two dense Llama-3-style decoder-only Transformers (Grattafiori et al., 2024) and the sparse OLMoE-1B-7B model (Muennighoff et al., 2025). We set the vocabulary size of OLMoE-1B-7B to 32K to match the LLaMA-2 tokenizer used in all experiments. Table 6 reports the architecture of the dense models. Both dense models use causal attention, bf16 training, and depth initialization. For both dense models, we use RoPE with $\theta = 10{,}000$, a layer normalization epsilon of 10^{-5}, and an FFN dimension multiplier of 1 with dimensions rounded to a multiple of 256. We disable Flex attention for these models.

Optimization. All methods use AdamW (Loshchilov & Hutter, 2019) for local optimization. We fix the weight decay at 0.1, set the momentum coefficients to $(\beta_1, \beta_2) = (0.9, 0.95)$, and tune the peak local learning rate for each method. The learning-rate schedule uses a 10% warmup phase that starts at 1% of the peak learning rate and then follows cosine decay. For DiLoCo-style methods, each worker performs H local updates before the outer step. In DiLoCo, workers compute a pseudo-gradient by All-Reduce and apply it through an outer optimizer; in GASLoC, this synchronized step is replaced by decentralized communication over the selected topology. Both DiLoCo and GASLoC use a Nesterov

Table 6: Model configurations used in the pre-training experiments. The Q/K/V head dimension is $d_{\text{model}}/n_{\text{heads}}$.

Model	Layers	$\mathrm{d}_{\text{model}}$	Attention heads	KV heads	Q/K/V head dim.	Token Budget
134M	12	768	12	12	64	2.7 B
551M	16	1536	24	24	64	11 B

momentum optimizer (Sutskever et al., 2013) for the outer step, with learning rate and momentum tuned independently for each method.

Implementation. We implement all methods in PyTorch, use NCCL for communication, and launch each run with `torchrun`. The 134M and 551M experiments run on nodes with four NVIDIA H100 GPUs, while OLMoE experiments use eight NVIDIA H200 GPUs. Within each node, GPUs communicate through NVIDIA NVLink. Across nodes, they communicate through InfiniBand. All 134M and 551M experiments simulate multiple logical workers on each GPU. OLMoE is evaluated with eight physical workers.

Hyperparameters. We conduct the broadest hyperparameter searches on the 134M model because of its lower training cost. For DDP and DiLoCo, we sweep the peak inner learning rate from 1×10^{-3} to 8×10^{-3} in increments of 1×10^{-3}. We use DDP as the reference for $H = 1$ and DiLoCo as the reference for $H > 1$. For DAdam, Local-DAdam, NoLoCo, and GASLoC, we search within 1×10^{-3} of the optimal inner learning rate of the corresponding reference. For outer optimizer-based methods, we search the outer learning rate in $\{0.4, 0.6, 0.8, 1.0\}$ and the outer momentum in $\{0.1, 0.3, 0.5, 0.7, 0.9\}$, then refine both in increments of 0.1 around the best values. For the other methods that use an outer optimizer, we search only in a neighborhood of the values selected for the corresponding reference. For the 551M model, we tune the inner learning rate and weight decay for DDP and DiLoCo. For DiLoCo, we tune the outer learning rate and momentum in a narrow neighborhood of the best values found for the 134M model. For the remaining methods on the 551M model, we keep the weight decay fixed and apply the same neighborhood searches around the optimum of the corresponding reference. For OLMoE-1B-7B, we tune the inner learning rate for DDP and DiLoCo over $\{1 \times 10^{-3}, 1.4 \times 10^{-3}, 2 \times 10^{-3}, 2.8 \times 10^{-3}\}$ on the first 600 steps to avoid loss spikes. We reuse the selected learning rate for methods with the same communication frequency. Following the optimal values reported by Charles et al. (2026) and optimal values on smaller models, we set the outer learning rate to 0.8 and the outer momentum to 0.9 for methods with communication frequency $H = 30$. For GASLoC and Local-DAdam $H = 1$ we set the outer parameter learning rate to 0.9 and the outer momentum to 0.2. Tables 7 to 9 report the selected hyperparameters for each method, model size and replica count for $H = 1$ and $H = 30$, respectively.

Hyperparameters were tuned on a holdout validation set with 1024 sequences ($\approx$ 2M tokens), while the validation loss reported in the result tables are from an holdout set of 4096 sequences ($\approx$ 8M tokens).

B.1 1-Peer and 2-Peer Topologies

At each communication round t, we construct a sparse undirected communication graph $G_t = (V, E_t)$ where $V = \{1, \ldots, n\}$ is the set of workers and E_t is the set of active peer-to-peer exchanges for that round.

Unless otherwise stated, we assume that the underlying admissible graph is complete: every pair of workers can potentially communicate, but only a sparse subset of edges is activated at each round. To sample this subset, we first draw a random permutation $\sigma^{(t)}$ of the workers and interpret it as a cycle. Each worker then selects the peer on its right or the 2 closest peers on this random cycle. Finally, we symmetrize the selections: an undirected edge $\{i, j\}$ is included in E_t whenever either worker i selects worker j, or worker j selects worker i. Algorithm 2 summarizes this procedure.

A fresh permutation is sampled at every communication round, so the active graph changes over time. This allows workers to communicate with different peers across rounds while keeping the number of active exchanges low. The same construction can be applied to arbitrary admissible topologies by restricting the possible peer selections to the edges of the underlying graph. Figures 6 and 7 show examples of 1- and 2-Peer topologies that vary from one round to the next.

B.2 Limited Bandwidth Experiments

We run the limited bandwidth experiments using our TorchTitan based implementation of GASLoC (Liang et al., 2025). Each replica runs in a separate Linux network namespace connected through a virtual Ethernet bridge. We use `tc netem` to limit traffic in both directions

Table 7: Selected hyperparameters for runs using $H = 1$ inner steps.

Model	Method	Topology	HP	8 replicas	16 replicas	32 replicas
Llama 134M	DDP	—	η	5e-3	5e-3	5e-3
	DAdam	ring	η	5e-3	5e-3	5e-3
		complete	η	7e-3	7e-3	7e-3
		1-Peer	η	6e-3	6e-3	6e-3
		2-Peer	η	6e-3	6e-3	6e-3
	GASLoC	ring	η	8e-3	8e-3	7e-3
			η_{out}	0.7	0.7	0.7
			μ_{out}	0.3	0.2	0.2
		complete	η	6e-3	6e-3	6e-3
			η_{out}	0.7	0.7	0.7
			μ_{out}	0.2	0.1	0.1
		1-Peer	η	6e-3	6e-3	6e-3
			η_{out}	0.9	0.9	0.9
			μ_{out}	0.3	0.3	0.3
		2-Peer	η	6e-3	6e-3	6e-3
			η_{out}	0.9	0.9	0.9
			μ_{out}	0.2	0.2	0.3
Llama 551M	DDP	—	η	6e-3	6e-3	—
	DAdam	ring	η	7e-3	8e-3	—
		complete	η	7e-3	8e-3	—
		1-Peer	η	7e-3	8e-3	—
		2-Peer	η	7e-3	8e-3	—
	GASLoC	ring	η	6e-3	6e-3	—
			η_{out}	1	1	—
			μ_{out}	0.3	0.4	—
		complete	η	6e-3	6e-3	—
			η_{out}	0.7	0.7	—
			μ_{out}	0.2	0.1	—
		1-Peer	η	8e-3	8e-3	—
			η_{out}	0.9	0.9	—
			μ_{out}	0.3	0.2	—
		2-Peer	η	8e-3	8e-3	—
			η_{out}	0.9	0.9	—
			μ_{out}	0.2	0.2	—
OLMoE	DDP	—	η	2e-3	—	—
	DAdam	complete	η	2e-3	—	—
	GASLoC	2-Peer	η	2e-3	—	—
			η_{out}	0.9	—	—
			μ_{out}	0.2	—	—

Algorithm 2 1- and 2-Peer sampling

Input: permutation of the nodes $\sigma^{(t)}$, node's index i
function SAMPLE_PEERS ($\sigma^{(t)},\ i$)
  Compute $i^- \leftarrow \texttt{pred}(\sigma^{(t)},\ i)$, $i^+ \leftarrow \texttt{succ}(\sigma^{(t)},\ i)$ and $p_i \leftarrow \texttt{pos}(\sigma^{(t)},\ i)$
  if 1–Peer **then**
    return $\{i,\ i^+\}$ **if** $p_i \bmod 2 = 0$, **else** $\{i^-,\ i\}$
  else if 2–Peer **then return** $\{i^-,\ i,\ i^+\}$
end function

Table 8: Selected hyperparameters for runs Llama 134M using $H = 30$ inner steps.

Model	Method	Topology	HP	8 replicas	16 replicas	32 replicas
Llama 134M	DiLoCo	—	η	2e-3	2e-3	2e-3
			η_{out}	0.8	0.8	0.8
			μ_{out}	0.7	0.7	0.7
	Local-DAdam	ring	η	2e-3	3e-3	2e-3
			η_{out}	0.3	0.4	0.4
			μ_{out}	0.6	0.8	0.8
		complete	η	4e-3	3e-3	2e-3
			η_{out}	0.3	0.3	0.3
			μ_{out}	0.3	0.5	0.6
		1-Peer	η	2e-3	3e-3	2e-3
			η_{out}	0.4	0.4	0.4
			μ_{out}	0.6	0.6	0.4
		2-Peer	η	4e-3	3e-3	2e-3
			η_{out}	0.4	0.4	0.4
			μ_{out}	0.6	0.5	0.4
	NoLoCo	ring	η	2e-3	2e-3	2e-3
			η_{out}	0.8	0.8	0.8
			μ_{out}	0.7	0.7	0.7
			γ	0.6	0.7	0.7
		complete	η	2e-3	2e-3	2e-3
			η_{out}	0.9	0.8	0.8
			μ_{out}	0.7	0.7	0.7
			γ	0.6	0.6	0.7
		1-Peer	η	2e-3	2e-3	2e-3
			η_{out}	0.8	0.8	0.8
			μ_{out}	0.7	0.7	0.7
			γ	0.6	0.7	0.7
		2-Peer	η	2e-3	2e-3	2e-3
			η_{out}	0.8	0.8	0.8
			μ_{out}	0.7	0.7	0.7
			γ	0.6	0.6	0.7
	GASLoC	ring	η	2e-3	2e-3	2e-3
			η_{out}	0.8	0.8	0.8
			μ_{out}	0.7	0.7	0.7
		complete	η	2e-3	2e-3	2e-3
			η_{out}	0.8	0.8	0.8
			μ_{out}	0.7	0.7	0.7
		1-Peer	η	3e-3	3e-3	2e-3
			η_{out}	0.8	0.8	0.8
			μ_{out}	0.6	0.6	0.6
		2-Peer	η	3e-3	2e-3	2e-3
			η_{out}	0.7	0.8	0.7
			μ_{out}	0.7	0.7	0.7

Table 9: Selected hyperparameters for runs Llama 551M and OLMoE using $H = 30$ inner steps.

Model	Method	Topology	HP	8 replicas	16 replicas
Llama 551M	DiLoCo	—	η	4e-3	4e-3
			η_{out}	0.9	0.9
			μ_{out}	0.8	0.8
	Local-DAdam	ring	η	2e-	3e-3
			η_{out}	0.4	0.4
			μ_{out}	0.6	0.4
		complete	η	2e-3	3e-3
			η_{out}	0.4	0.2
			μ_{out}	0.6	0.4
		1-Peer	η	2e-3	2e-3
			η_{out}	0.4	0.4
			μ_{out}	0.6	0.6
		2-Peer	η	2e-3	2e-3
			η_{out}	0.4	0.4
			μ_{out}	0.6	0.7
	GASLoC	ring	η	4e-3	4e-3
			η_{out}	0.8	0.8
			μ_{out}	0.7	0.7
		complete	η	4e-3	4e-3
			η_{out}	0.9	0.9
			μ_{out}	0.8	0.8
		1-Peer	η	3e-3	3e-3
			η_{out}	0.9	0.9
			μ_{out}	0.7	0.6
		2-Peer	η	3e-3	3e-3
			η_{out}	0.9	0.9
			μ_{out}	0.8	0.8
OLMoE	DiLoCo	—	η	2e-3	—
			η_{out}	0.8	—
			μ_{out}	0.9	—
	Local-DAdam	complete	η	2e-3	—
			η_{out}	0.8	—
			μ_{out}	0.9	—
	GASLoC	2-Peer	η	2e-3	—
			η_{out}	0.8	—
			μ_{out}	0.9	—

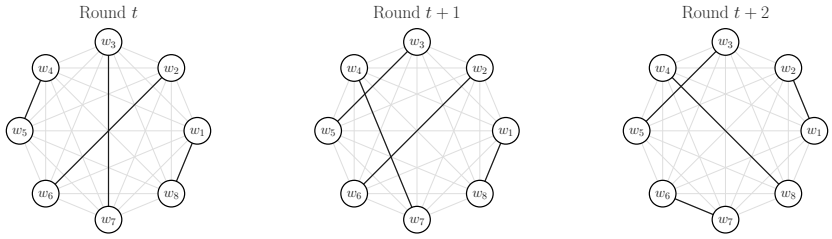


Figure 6: Time-varying 1-Peer graph

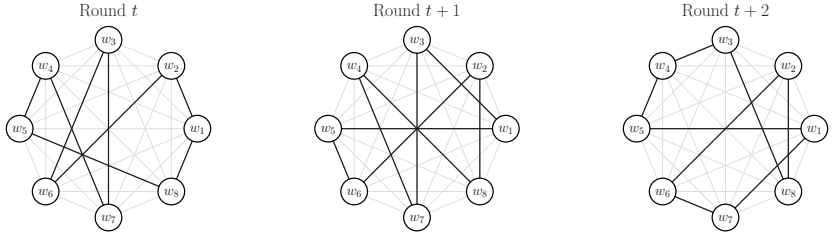


Figure 7: Time-varying 2-Peer graph

according to the bandwidth assigned to each replica. We force NCCL to use its socket transport over the limited virtual interfaces and disable all communication paths that could bypass them.

C Local-DAdam: Adapting DAdam to Multiple Local Steps

DAdam is defined in the regime where communication is performed at every optimizer step. (Wang et al., 2025). Let W be the symmetric and doubly stochastic mixing matrix of the communication graph. Writing the local optimizer update as $\mathrm{Optimizer}^{\mathrm{in}}$, one DAdam step is

$$y_i^{(t)} \leftarrow \mathrm{Optimizer}^{\mathrm{in}}\left(x_i^{(t)}, \nabla f_i(x_i^{(t)})\right), \qquad x_i^{(t+1)} \leftarrow \sum_{j\in\mathcal{N}_i} w_{ij} x_j^{(t)} + \left(y_i^{(t)} - x_i^{(t)}\right).$$

DAdam mixes the pre-update parameters $x_i^{(t)}$ and then applies the local optimizer displacement. Thus, DAdam is directly comparable to our protocol when $H = 1$, where every local optimizer step is followed by communication. For $H > 1$, however, simply delaying the mixing operation until after a block of H local updates does not recover the original DAdam algorithm: it removes the per-step gossip operation on which DAdam is based and changes the optimization dynamics.

We define Local-DAdam to compare the methods under the same infrequent communication schedule as DiLoCo (Douillard et al., 2024) and GASLoC. Starting from $x_i^{(t)}$, worker i applies H steps of $\mathrm{Optimizer}^{\mathrm{in}}$ and obtains $y_i^{(t,H)}$. It then computes the DAdam disagreement from the models at the start of the round and applies

$$x_i^{(t+1)} \leftarrow \mathrm{Optimizer}^{\mathrm{out}}\left(y_i^{(t,H)}, \sum_{i=1}^{n} w_{ij}\left(x_i^{(t)} - x_j^{(t)}\right)\right)$$

When $H = 1$ and $\mathrm{Optimizer}^{\mathrm{out}}$ is SGD with a fixed unit learning rate, this expression is the DAdam update above. For $H > 1$, the correction contains no model produced within the local block and only measures disagreement at its start. Thus, Local-DAdam does not recover the intermediate mixing of DAdam. We use it as a controlled adaptation of DAdam to infrequent communication.

D Additional Results

D.1 Multi-seed Results

We repeat the 16-replica 134M model experiments over four seeds to measure the variability of the validation loss. The mean results in Table 10 reproduce the same trends between methods as the

Table 10: **Multi-seed evaluation.** Mean validation loss and standard deviation over four seeds for the model with 134M parameters and 16 replicas.

AdamW DDP (*ref.*)	$H = 1$		DiLoCo (*ref.*)	$H = 30$	
	3.177 ± 0.004			3.404 ± 0.007	
Topology	**GASLoC**	DAdam		**GASLoC**	Local-DAdam
ring	3.364 ± 0.006	3.332 ± 0.006		3.453 ± 0.007	3.689 ± 0.006
complete	3.293 ± 0.007	3.247 ± 0.006		3.403 ± 0.008	3.651 ± 0.007
1-Peer	3.241 ± 0.009	3.292 ± 0.009		3.489 ± 0.011	3.662 ± 0.010
2-Peer	3.219 ± 0.008	3.277 ± 0.007		3.452 ± 0.009	3.657 ± 0.008

Table 11: **One-step Decentralized Methods.** Results for the 134M-parameter model, using **one local step** per communication round. We compare DDP to decentralized methods GASLoC and DAdam across 8, 16, and 32 replicas.

	8 replicas		**16 replicas**		**32 replicas**	
AdamW DDP *(ref.)*	3.18		3.18		3.18	
Topology	**GASLoC**	**DAdam**	**GASLoC**	**DAdam**	**GASLoC**	**DAdam**
ring	3.40	**3.31**	3.36	**3.33**	3.44	**3.41**
complete	3.24	**3.23**	3.29	**3.25**	3.36	**3.30**
1-Peer	**3.20**	3.25	**3.24**	3.29	**3.31**	3.37
2-Peer	**3.19**	3.25	**3.22**	3.28	**3.28**	3.34

single seed evaluation, and the largest standard deviation is only 0.011. Prior multi-seed studies report broadly consistent pretraining dynamics across model sizes and fast cross-seed convergence for larger models (van der Wal et al., 2025). These findings suggest that the larger-scale results are unlikely to exhibit greater instability. We therefore use a single seed for the broader evaluation, consistently with recent LLM pre-training studies (Douillard et al., 2024; Kolehmainen et al., 2025; Bae et al., 2025; Charles et al., 2026; Janson et al., 2026).

D.2 Complete Single-Step Results

We provide additional single-step results to complement the comparison in Section 4.2. Tables 11 and 12 extend Table 1.

Across these configurations, the qualitative behavior remains consistent with the main results. GASLoC with randomized communications gives the strongest decentralized results, with the 2-Peer variant performing best overall. In particular, in the 8-replica setting, GASLoC-2-Peer can even match the reference performance for the larger model. The 1-Peer variant, instead, remains competitive, while benefiting from an even saparser communication. In contrast, DAdam does not show the same benefits from randomized communication.

D.3 Downstream Tasks Evaluation

To test whether the validation loss differences also hold on established benchmarks, we evaluate the OLMoE-1B-7B checkpoints on four downstream tasks. ARC-Easy (ARC-E) evaluates science question answering at grade school level (Clark et al., 2018), while HellaSwag evaluates commonsense reasoning through plausible sentence completion (Zellers et al., 2019). PIQA focuses on physical commonsense reasoning (Bisk et al., 2020). This set therefore complements validation loss with measures of science knowledge, and commonsense reasoning.

We run all tasks zero shot with version 0.4.12 of the Language Model Evaluation Harness (Gao et al., 2024). The custom adapter scores continuations by log likelihood in bfloat16, using a batch size of 16 and an input length of at most 2047 tokens. For ARC-E, HellaSwag, and PIQA, we report length-normalized accuracy and its standard error.

Table 12: **One-step Decentralized Methods.** Validation loss for the 551M-parameter LLM using 8 and 16 replicas.

Method	Topology	Val Loss	
		8 replicas	16 replicas
AdamW DDP	–	2.64	2.64
DAdam	ring	2.71	2.73
GASLoC	ring	**2.66**	**2.70**
DAdam	complete	2.70	2.71
GASLoC	1-Peer	2.67	2.71
GASLoC	2-Peer	**2.64**	**2.69**

Table 13: **Downstream evaluation on OLMoE-1B-7B.** Results for checkpoints trained on 26B tokens with 8 replicas. Evaluation is zero shot. We report accuracy normalized by answer length for ARC-Easy, HellaSwag, and PIQA. All values are percentages with standard errors.

(a) **One-step**

Method	ARC-Easy	HellaSwag	PIQA
AdamW DDP (*ref.*)	49.87 ± 1.02	52.71 ± 0.49	72.36 ± 1.04
DAdam (complete)	46.89 ± 1.02	48.87 ± 0.49	71.71 ± 1.05
GASLoC (2-Peer)	$\mathbf{47.60 \pm 1.02}$	$\mathbf{52.56 \pm 0.50}$	$\mathbf{72.58 \pm 1.04}$

(b) **Multi-step**

Method	ARC-Easy	HellaSwag	PIQA
DiLoCo (*ref.*)	43.56 ± 1.01	44.77 ± 0.50	70.56 ± 1.06
Loc.-DAdam (complete)	43.03 ± 1.02	42.41 ± 0.49	69.31 ± 1.07
GASLoC (2-Peer)	$\mathbf{43.56 \pm 1.02}$	$\mathbf{43.13 \pm 0.49}$	$\mathbf{69.80 \pm 1.07}$

The compact comparison in Table 4 reports ARC-E and HellaSwag next to validation loss, while Table 13 reports all three downstream tasks. The one-step and multi-step results are shown in Table 13a and Table 13b, respectively. In the available one-step comparison, DAdam is below AdamW DDP on all four tasks, with the largest differences on ARC-E and HellaSwag.

E Additional Analysis

E.1 The Role of the Outer Optimizer

In GASLoC, the outer optimizer processes an update that contains both local drift and disagreement between replicas. With sparse communication, this disagreement cannot be removed in one round because each update reaches only a subset of workers. Momentum uses information from previous outer rounds to accelerate consensus, so its dynamics may also affect optimization under sparse communication. We therefore examine the optimizer choice and its hyperparameters for GASLoC-2-Peer, using complete communication as a reference.

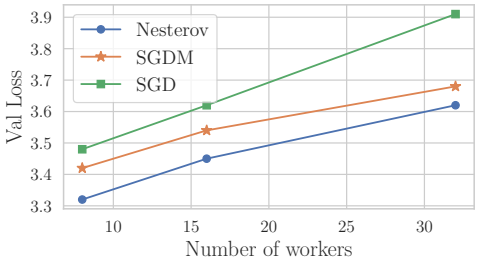
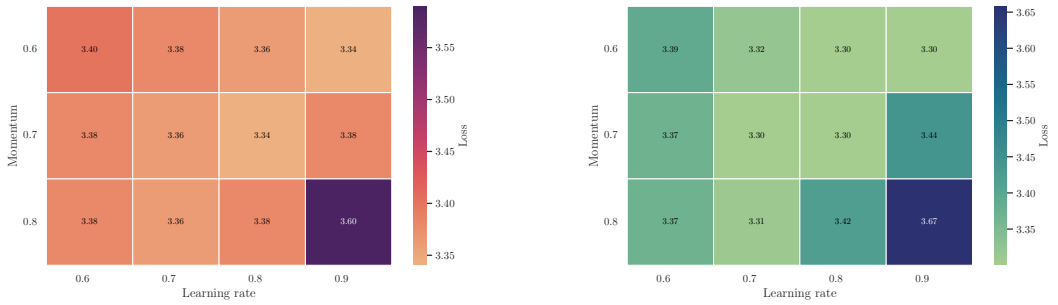


Figure 8: **Effect of the outer optimizer.** Final validation loss of GASLoC-2-Peer with SGD, SGDM, or Nesterov momentum as the number of workers increases. Both momentum methods outperform SGD at every evaluated worker count, with Nesterov momentum giving the lowest loss throughout.

E.1.1 Outer Optimizer Choice

We first compare SGD, SGD with momentum (SGDM), and Nesterov momentum as the outer optimizer of GASLoC-2-Peer. This comparison tests whether the consensus benefit described above is also reflected in validation loss under sparse communication. Figure 8 shows that both momentum methods outperform SGD across the evaluated worker counts, with Nesterov momentum giving the lowest loss and SGDM the second lowest. This ordering holds with 32 workers and matches the complete communication results of Douillard et al. (2024), so the empirical benefit of outer momentum extends to the sparse setting considered here.



(a) GASLoC-2-Peer

(b) GASLoC complete graph

Figure 9: **Outer optimizer hyperparameter sensitivity.** Validation loss across outer learning rates and momentum coefficients for the 134M model with 8 workers. GASLoC-2-Peer and GASLoC with complete communication exhibit similar regions of low loss.

E.1.2 Hyperparameter Sensitivity

Having established the benefit of momentum, we next examine whether sparse communication makes its hyperparameters harder to tune. To isolate this effect, we compare GASLoC-2-Peer with complete communication over the same hyperparameter grid. As shown in Figure 9, both methods have similar regions of low validation loss, with several configurations close to the minimum. Although the minimum loss of GASLoC-2-Peer is slightly higher, its highest loss at the grid boundary is lower. Randomized 2-Peer communication therefore does not narrow the useful hyperparameter region in this sweep.

E.2 Compute Utilization under Limited Bandwidth

The previous analyses concern optimization behavior, while limited bandwidth also affects how efficiently the available compute is used. We estimate this effect through the theoretical compute utilization

$$\frac{T_{\text{compute}}}{T_{\text{compute}} + T_{\text{comm}}},$$

where T_{compute} and T_{comm} are the estimated computation and communication times. We estimate the computation time from the model FLOP profile using a theoretical peak FP16/BF16 throughput of 4.5×10^{15} FLOPs/s per GPU and a machine FLOP utilization of 40%. The simulation then considers 16 workers with 8 GPUs each, varies their link bandwidth, and limits one straggler to 20% of that bandwidth.

Figure 10 reports the resulting estimates for a Llama-3 model with 70B parameters (Grattafiori et al., 2024). Because DDP and DiLoCo use All-Reduce, the straggler determines their communication time and delays the other workers. GASLoC instead allows this worker to perform fewer local steps, so that it starts its slower peer exchange earlier and partly offsets the limited link. As a result, both sparse variants achieve higher estimated compute utilization than the All-Reduce baselines when bandwidth is limited. GASLoC-1-Peer exceeds 80% utilization at about 10 Gbps, while GASLoC-2-Peer reaches the same level at about 20 Gbps.

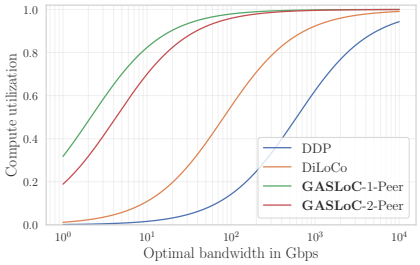


Figure 10: **Simulated compute utilization.** Theoretical compute utilization for a model with 70B parameters and 16 workers as the bandwidth of the non-stragglers varies, with one straggler limited to 20% of this bandwidth. DDP and DiLoCo are limited by this worker because they use All-Reduce, while the sparse variants let it perform fewer local steps before each peer exchange. This schedule gives the sparse variants higher estimated utilization when bandwidth is limited.