FAUST: Federated Asynchronous Update with Staggered Timescales for Low-Communication Foundation Model Training

Yunlong Tan* **Mingqiao Mo*** **Hao Zhang†** **Shengxing Qin**
University of Chinese Academy of Sciences
dogechat@163.com, mmq20031004@163.com, zh.cs.star@outlook.com

Abstract

Large language model training with sequence-parallel data-parallel (SP-DP) is bottlenecked by the interplay between intra-step activation redistribution and inter-step gradient synchronization. Existing infrequent-communication methods like FedAvg keep states local or reset them lack convergence guarantees and are unstable under long-context sequence-parallel regimes where each device holds only a partial activation shard; Local Adam synchronizes all states jointly and is convergent but triples the collective payload, and its uniform synchronization period cannot distinguish the semantically mandatory intra-step All-to-All from the temporally deferrable inter-step AllReduce. We propose **FAUST** (**F**ederated **A**synchronous **U**pdate with **S**taggered **T**imescales), a compiler-orchestrated training system that composes compile-time sequence-parallel graph rewriting with tiered runtime synchronization, assigning independent averaging periods to parameters (K_x), first moments ($K_u{=}3K_x$), and second moments ($K_v{=}6K_x$) according to their impulse-response half-lives, while preserving convergence under the orthogonal collective constraint that intra-step All-to-All must retire before inter-step AllReduce begins. Our analysis shows that while first-moment drift dominates the convergence rate in-distribution, high-fidelity convergence guarantees require at least periodic synchronization of the second moment. Experiments on language models up to 1.3B parameters show that FAUST incurs $303\times$ less wall-clock communication overhead than standard DDP and $1.78\times$ less than the previous state-of-the-art DES-LOC, while achieving perplexity within 4.6% of the fully-synchronized baseline in 27% less wall-clock time. On bandwidth-constrained links, FAUST delivers 48–$303\times$ speedup over DDP.

1 Introduction

Training large language models at scale requires two orthogonal classes of collective communication: *intra-step* All-to-All that redistributes activations across the head dimension for sequence parallelism (SP), and *inter-step* AllReduce that aggregates gradients across data-parallel (DP) replicas. On heterogeneous clusters where both classes share the same interconnect, the inter-step AllReduce dominates wall-clock time [Li et al., 2020]. Local SGD [Stich, 2019] and FedAvg [McMahan et al., 2017] reduce this cost by averaging parameters every $K \gg 1$ steps, but they neither handle the additional per-parameter states of adaptive optimizers [Kingma and Ba, 2017] nor account for the intra-step SP collective that modern long-context training [Grattafiori et al., 2024] introduces.

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Extensions to adaptive optimizers either keep optimizer states entirely local [Douillard et al., 2024], accumulating drift without convergence guarantees, or reset them at each sync point [Sani et al., 2024b,a], destroying the exponential moving average (half-life $\tau_{0.5}(\beta_1{=}0.9) \approx 6.6$ steps). Local Adam [Cheng and Glasgow, 2025] proves convergent intermittent-communication adaptive optimization, but synchronises all states at a uniform period K, tripling the payload versus Local SGD. Moreover, the uniform-K schedule cannot distinguish mandatory intra-step A2A from deferrable inter-step AllReduce, preventing composition with SP, a critical barrier as contexts exceed single-device memory and require compiler-mediated sequence sharding [Jacobs et al., 2023, Fang and Zhao, 2024].

We propose **FAUST**, which lifts SP into PyTorch 2's compiler stack and composes it with tiered desynced DP communication on a $P \times M$ device mesh. FAUST operates in two phases. The *compile-time* phase rewrites the FX graph via six passes, symbolic sequence-dimension sharding, input partitioning, All-to-All insertion around every SDPA layer ($4L$ A2A ops for L layers), shape propagation, dead-code elimination, and selective activation checkpointing—overcoming three compiler challenges: IR-level selection across Torch/ATen/Inductor representations (C1), worklist-based sequence-dependent shape inference through data-movement operators (C2), and avoiding extraneous backward A2A from naïve activation rematerialisation (C3). The *runtime* phase assigns independent averaging periods K_x, $K_u{=}3K_x$, $K_v{=}6K_x$ to parameters, first moments, and second moments respectively, following the half-life formula $\tau_{0.5}(\beta) = \ln 2/(-\ln \beta)$. A collective fence serialises pending SP handles before each DP reduction, ensuring the two collective classes compose without interference.

Contributions.

1. **Tiered gradient reduction with ZeRO-1 integration.** A graph-rewriting SP pass that operates on the Torch-IR forward graph, composed with runtime $K_x/K_u/K_v$-gated gradient reduction. The two collective classes are decoupled by construction: SP A2A executes every step; DP AllReduce is gated by tier-specific periods. Structural-tier bucket coalescing with independent sync gates, achieving $1.78\times$ over DES-LOC and $305\times$ cumulative communication reduction versus DDP on bandwidth-constrained links.
2. **Convergence parity under reduced communication.** On a 700M model, FAUST reaches PPL 10.46 (within 4.6% of DDP's 10.0) while DES-LOC and Local Adam plateau at 12.06 and 11.69 (Figure 1). At matched wall-clock, FAUST outperforms DDP (10.46 vs 11.53 at $\sim$13 h).
3. **Heterogeneous-cluster robustness.** Collective fence, FP32 backward A2A accumulation ($\sim$0.3% loss improvement under bf16), and cross-device dtype normalisation. At 1.3B parameters, desynced training ($K_x{=}32$) matches DDP loss ($7.31{\pm}0.49$ vs $7.67{\pm}0.05$) with $1.23\times$ throughput gain; the SP pass extends trainable context by $2.26\times$ over ring attention.

2 Compiler-Orchestrated Sequence Parallelism with Desynced Communication (FAUST)

We build desynced momentum framework and extend it to exploit sequence-level parallelism, a new axis of parallelism orthogonal to the data-parallel axis. Consider the EMA update:

$$u_t = \Pi_{\beta_1}(u_{t-1},\, g_t), \quad v_t = \Pi_{\beta_2}\big(v_{t-1},\, g_t \odot g_t\big), \qquad \Pi_\beta(x,y) \triangleq \beta\, x + (1-\beta)\, y. \tag{1}$$

DES-LOC alone, parameterised by momentum β_1 comprising $\alpha_t = \tilde{O}(T^{-1/2}D^{-1/4})$ [Cheng and Glasgow, 2025] where T is the number of outer steps and D the inner synchronisation period. While T is D agnostic $\beta_1 \to 1$, the convergence factor β_2 remains around 0 to 1.

A natural question concerns the number of inner steps a state's impulse decays to a fraction ψ, $\tau_\psi(\beta) = -\big(\ln \beta\big)^{-1} \ln\psi$. Following Pagliardini et al. [2025], we use the half-life $\tau_{0.5}$ as the relevant timescale, choosing β from there. For typical values of β, we find $\tau_{0.5}(0.99) \approx 69.3$ [Allal et al., 2025], $\tau_{0.5}(0.999) \approx 692.8$ [Kingma and Ba, 2017], and $\tau_{0.5}(0.9999) \approx 6931$ [Taniguchi et al., 2024].

Fundamentally, FAUST's sequence parallelism introduces a second communication channel that operates at finer granularity than the outer synchronisation; whereas $\beta = 1$ freezes the momentum entirely, setting $\beta \to 0$ instantaneously overwrites state with the freshest gradient.

We introduce SP to partition the sequence into S sub-sequences. With per-coordinate gradient clipping, each coordinate satisfies $|(g_t)_j| \leq \rho$. This serialisation has a concrete consequence for the drift: between two consecutive DP synchronisation events at steps t and $t + K_x$, the gradient sequence $g_t, g_{t+1}, \ldots, g_{t+K_x-1}$ on each replica is computed from SP-complete forward passes—each gradient reflects attention over the *full* sequence S (via the All-to-All reconstitution), not a local shard S/P.

Formally, let $g_t^{(\text{SP})}$ denote the gradient under SP and $g_t^{(\text{noSP})}$ the gradient from a naïve sequence-sharded forward (no All-to-All, each rank sees only S/P tokens). Since the SP All-to-All reconstitutes the full key–value context for each head subset before computing attention, the per-coordinate gradient variance satisfies:

$$\operatorname{Var}\big[g_t^{(\text{SP})}\big]_j \;\leq\; \frac{1}{P}\,\operatorname{Var}\big[g_t^{(\text{noSP})}\big]_j \;+\; \epsilon_{\text{A2A}}, \tag{2}$$

where the $1/P$ factor arises because each attention head operates on $P\times$ more context tokens (variance reduction by the law of total variance over sequence chunks), and ϵ_{A2A} is the finite-precision residual from the All-to-All collective. Under FP32 accumulation in the backward All-to-All (the default in our implementation), $\epsilon_{\text{A2A}} \leq 2^{-23}\,\rho$ for single-precision and is negligible.

Substituting the tighter effective clipping radius $\rho_{\text{eff}} = \rho\,/\sqrt{P}$ for the $d_{\text{attn}} = 4NH$ attention-layer coordinates (where the SP All-to-All redistributes activations), we obtain the **SP-tightened bounds**:

$$\big\|u_{T+1}^{(\text{attn})} - u_1^{(\text{attn})}\big\|_\infty \leq \frac{2\rho}{\sqrt{P}}\,(1-\beta_1^T), \tag{3}$$

$$\big\|v_{T+1}^{(\text{attn})} - v_1^{(\text{attn})}\big\|_\infty \leq \frac{\rho}{\sqrt{P}} \cdot \big\|u_{T+1}^{(\text{attn})} - u_1^{(\text{attn})}\big\|_\infty. \tag{4}$$

From the first bound, while β limits the drift below clipping ρ, a common hyperparameter in pre-trained language models [Brown et al., 2020, Workshop et al., 2023], where the gradient variance is controlled across workers. We use Nesterov-style lookahead for drift correction [Sutskever et al., 2013, Taniguchi et al., 2024], and norm-based clipping [Pascanu et al., 2013, Brown et al., 2020]. From the second bound, the half-life of an EMA state v directly bounds its maximal drift across desynchronised iterations.

2.1 FAUST Algorithm

We formulate FAUST (compiler-orchestrated sequence parallelism) as a family of desynced optimisers that decouple data-parallel synchronisation and sequence-parallel communication at different periodicities using a multi-rate scheduling protocol.

We enable FAUST to use the method of federated averaging and synchronisation [McMahan et al., 2017]. Concretely, the server-side momentum follows the outer optimizer [Reddi et al., 2021] reduction, which we detail in Appendix C.1.

As shown in Algorithm 1, FAUST synchronises parameters $\theta \in \mathbb{R}^d$ and optimizer states $\{s^k\}_{k=1}^K$ at multi-frequency periodicities $p_0, \{p_k\}_{k=1}^K \in \mathbb{R}_+$. Setting $K = 2$, $s_t^1 = u_t$, $s_t^2 = v_t$, and using update rules Π_β^1, Π_β^2 gives us the Adam variant.

Key results. To demonstrate FAUST's empirical advantage, Fig. 1a illustrates the perplexity ratio trace: FAUST reaches lower perplexity than DES-LOC alone under matched conditions, while prior distributed methods [Douillard et al., 2024, Sani et al., 2024a, Zhang et al., 2024, Sani et al., 2024b] trail.

3 Convergence Guarantees

This section establishes the theoretical foundation for FAUST's desynced synchronization schedule. We analyze a single-momentum variant of Adam under decoupled averaging cadences; extensions

Algorithm 1 FAUST Runtime Training with Tiered Communication

Require: compiled graph $\mathcal{G}'$, model $\theta_0 \in \mathbb{R}^d$, K optimizer states $\{s_{-1}^k\}_{k=1}^K$
Require: EMA operators $\{\Pi_\beta^k\}_{k=1}^K$, update rule $\mathcal{S}$, clipping bound ρ, LR schedule $\{\eta_t\}$
Require: sync periods: p_0 (params), $\{p_k\}_{k=1}^K$ (states); SP degree P, workers W θ_T, $\{s_{T-1}^k\}_{k=1}^K$

w: $\theta_0^w \leftarrow \theta_0$, $s_{-1}^{k,w} \leftarrow s_{-1}^k$ {local copy of all states}
for $t = 0, \ldots, T-1$ **do**
  for $w = 1, \ldots, W$ **in parallel do**
    {intra-step SP}
    Partition ξ_t^w into P chunks along seq dim {compiled $\mathcal{G}'$}
    for each attention layer $\ell = 1, \ldots, L$ **do**
      $Q_\ell, K_\ell, V_\ell \leftarrow \text{A2A}_{\text{fwd}}^{(\ell)}(Q_\ell^{\text{local}}, K_\ell^{\text{local}}, V_\ell^{\text{local}})$ $\{[B, N, \frac{S}{P}, H] \to [B, \frac{N}{P}, S, H]\}$
      $O_\ell \leftarrow \text{SDPA}(Q_\ell, K_\ell, V_\ell)$ {full-S attention}
      $O_\ell^{\text{local}} \leftarrow \text{A2A}_{\text{out}}^{(\ell)}(O_\ell)$ $\{[B, \frac{N}{P}, S, H] \to [B, N, \frac{S}{P}, H]\}$
    end for
    $g_t^w \leftarrow$ backward through $\mathcal{G}'$ {FP32 A2A accum.}
    $\hat{g}_t^w \leftarrow \text{clip}(g_t^w, \rho)$ {per-coordinate clip}
    {collective fence}
    if $t \bmod p_0 = 0$ **then**
      FENCEALLSPHANDLES() {retire A2A handles}
    end if
    {inter-step DP sync gated}
    for $k = 1$ **to** K **do**
      $\tilde{s}_{t-1}^k \leftarrow \mathbf{1}_{[t \bmod p_k = 0]}\, \mathbb{A}_w[s_{t-1}^k] + \mathbf{1}_{[t \bmod p_k \neq 0]}\, s_{t-1}^{k,w}$ {tier-k sync}
      $s_t^{k,w} \leftarrow \Pi_\beta^k\big(\tilde{s}_{t-1}^k,\, \hat{g}_t^w\big)$ {local EMA}
    end for
    $\tilde{\theta}_t^w \leftarrow \mathbf{1}_{[t \bmod p_0 = 0]}\, \mathbb{A}_w[\theta_t^w] + \mathbf{1}_{[t \bmod p_0 \neq 0]}\, \theta_t^w$ {DP param avg}
    $\theta_{t+1}^w \leftarrow \mathcal{S}\Big(\tilde{\theta}_t^w,\, \hat{g}_t^w,\, \eta_t,\, \{s_t^{k,w}\}_{k=1}^K\Big)$ {optimizer step}
  end for
end for

to the full two-momentum Adam optimizer appear in Section D.1, with high-probability bounds in Section D.1.

Problem formulation. We consider the distributed optimization problem over a $P \times M$ device mesh, where P denotes the *sequence-parallel* (SP) degree within each replica and M denotes the *data-parallel* (DP) degree across replicas, with total device count $W = P \cdot M$. We formulate this as:

$$\min_{x \in \mathbb{R}^d} f(x) \;:=\; \frac{1}{M} \sum_{m=1}^{M} \underbrace{\sum_{k=0}^{K} \omega_k \;\cdot\; \mathbb{E}_{\xi \sim \mathcal{D}_m}\Big[F_m^{(k)}\big(x,\, \{s^k\};\, \xi,\, P\big)\Big]}_{f_m(x,\, \{s^k\})}, \tag{5}$$

where K is the number of optimizer states (e.g. $K{=}2$ for Adam: first moment $s^1{=}u$, second moment $s^2{=}v$), and $k{=}0$ denotes the parameter state x itself. Each state-level loss $F_m^{(k)}$ captures the contribution of the k-th EMA state to the replica-m objective under SP degree P. The weights ω_k are determined by the *impulse-response half-life* of each state:

$$\omega_k \;=\; \frac{\ln 2}{\tau_{0.5}(\beta_k)} \;=\; -\ln \beta_k, \qquad \text{with } \tau_{0.5}(\beta) = \frac{\ln 2}{-\ln \beta}, \tag{6}$$

Formally, letting $\mathcal{A}_P$ denote the composition of the $4L$ All-to-All operators inserted (where L is the number of attention layers), the SP-computed gradient satisfies

$$g_t^m = \nabla_x\big[F_m\big(x;\, \mathcal{A}_P(\xi_t^m)\big)\big] \;=\; \nabla_x F_m(x;\, \xi_t^m), \tag{7}$$

where the equality holds because $\mathcal{A}_P$ is a *lossless permutation*: each All-to-All is an invertible redistribution of tensor dimensions that does not discard or approximate any activation. The collective

fence (Algorithm 1) guarantees that all $\mathcal{A}_P$ handles retire before the inter-step DP averaging begins, so the operate on orthogonal communication axes and compose without interference.

In the general (heterogeneous) setting, each replica m draws mini batch stochastic gradients from a local distribution $\mathcal{D}_m$; the homogeneous case $\mathcal{D}_1 = \cdots = \mathcal{D}_M$ is recovered as a special instance.

Let $\theta = (\theta^{(0)}, \theta^{(1)}, \ldots, \theta^{(K)}) \in \mathbb{R}^{d_0} \times \cdots \times \mathbb{R}^{d_K}$ denote the parameter vector partitioned by structural tier, where d_0 (embedding/norm), d_1 (attention), d_2 (FFN) with $\sum_k d_k = d$. Let $\mathcal{A}_P$ denote the SP All-to-All composition from, and let Π_β^k denote the EMA operator for the k-th state.

Assumption 3.1 (Tier-decomposed descent with collective contraction)**.** *The global objective $f : \mathbb{R}^d \to \mathbb{R}$ is bounded below by f^*. For all θ,*

$$f_m(\theta + \Delta^{(k)}) \le f_m(\theta) + \langle \nabla^{(k)} f_m(\theta),\, \Delta^{(k)} \rangle + \frac{L_k}{2} \|\Delta^{(k)}\|^2, \tag{8}$$

Assumption 3.2 (SP-structured noise with head-factored variance)**.** *Under the SP All-to-All composition $\mathcal{A}_P$, the stochastic gradient decomposes into N/P independent head-group contributions. For each replica m, the gradient is unbiased and its variance factorises as:*

$$\mathbb{E}[g_t^m] = \nabla f_m(\theta), \qquad \mathbb{E}\big[\|g_t^m - \nabla f_m(\theta)\|^2\big] \le \frac{N/P}{N}\, \sigma_{\text{attn}}^2 \;+\; \sigma_{\text{ffn}}^2 \;+\; P\, \epsilon_{\text{A2A}}^2, \tag{9}$$

where the first term reflects that each SP rank computes exact attention for N/P out of N total heads (reducing inter-head variance by $1/P$ while eliminating intra-head sequence-truncation bias entirely), the second term bounds non-attention variance, and the third term accounts for finite-precision All-to-All residuals ($\epsilon_{\text{A2A}} \le 2^{-23}\rho$ under FP32 backward accumulation). Setting $P{=}1$ gives the standard $\sigma^2 = \sigma_{\text{attn}}^2 + \sigma_{\text{ffn}}^2$.

Assumption 3.3 (Multi-rate heterogeneity with Lyapunov decay)**.** *The inter-replica gradient divergence admits a* Lyapunov decomposition *across tiers: there exists a non-negative function $\Phi(\theta) = \sum_{k=0}^K \lambda_k\, \Phi_k(\theta^{(k)})$ such that*

$$\frac{1}{M} \sum_{m=1}^M \|\nabla f_m(\theta) - \nabla f(\theta)\|^2 \le -\dot{\Phi}(\theta) + \sum_{k=0}^K \lambda_k\, G_k^2, \tag{10}$$

where $\dot{\Phi}(\theta) \le 0$ along the FAUST trajectory (ensuring the heterogeneity is non-increasing when synchronisation periods respect the half-life ordering), and the Lyapunov coefficients satisfy $\lambda_k = \omega_k/\omega_0 = (-\ln \beta_k)/(-\ln \beta_0)$. Setting $\Phi \equiv 0$ and $\lambda_k = 1$ for all k recovers the standard bound $\frac{1}{M}\sum_m \|\nabla f_m\|^2 \le G^2 + B^2\|\nabla f\|^2$ with $G^2 = \sum_k G_k^2$.

These subsume the standard conditions [Yu et al., 2019, Karimireddy et al., 2021, Wang et al., 2021, Chen et al., 2017] as the Euclidean, single-tier, $P{=}1$ special case, while making three structural features of FAUST first-class citizens:

Probabilistic synchronization equivalence. To facilitate the analysis, we view each synchronization event as a Bernoulli trial: instead of averaging parameters every K_x steps ($t \bmod K_x = 0$), we average with probability $p_x = 1/K_x$ at each step.

Theorem 3.1 (Multi-resolution contraction of FAUST)**.** *Let Assumptions 3.1, 3.2, and 3.3 hold. Define the* FAUST potential

$$\mathcal{V}_t \;\stackrel{\text{def}}{=}\; \big(f(\bar{\theta}_t) - f^*\big) \;+\; \sum_{k=0}^K \frac{\lambda_k}{M} \sum_{m=1}^M \big\|\theta_t^{m,(k)} - \bar{\theta}_t^{(k)}\big\|^2 \;+\; \Phi(\bar{\theta}_t), \tag{11}$$

where $\bar{\theta}_t = \frac{1}{M}\sum_m \theta_t^m$ is the DP average, $\lambda_k = \omega_k\, L_k = (-\ln \beta_k)\, L_k$ couples the half-life weight to the per-tier curvature, and Φ is the Lyapunov function from Assumption 3.3. The potential $\mathcal{V}_t$ jointly tracks three quantities: the optimality gap, the tier-weighted cross-replica parameter divergence, and the heterogeneity budget.

Choose the step size η_t via the spectral schedule

$$\eta_t = \min\left(\frac{1 - \beta_{\max}}{2\,\overline{L}},\; \frac{1}{\overline{L}\,\sqrt{t+1}} \sqrt{\frac{P}{P + \kappa_{\text{SP}}}} \right), \qquad \overline{L} \stackrel{\text{def}}{=} \Big(\sum_{k=0}^K \omega_k\, L_k^2\Big)^{1/2}, \tag{12}$$

where $\beta_{\max} = \max_k \beta_k$, $\overline{L}$ *is the half-life-weighted RMS smoothness, and* $\kappa_{\mathrm{SP}} = P\,\epsilon_{\mathrm{A2A}}^2/\sigma_{\mathrm{attn}}^2$ *is the SP precision ratio (negligible under FP32 accumulation). Then contracts as*

$$\mathcal{V}_T \;\le\; \prod_{t=0}^{T-1}\big(1-\eta_t\,\mu_t\big)\;\mathcal{V}_0 \;+\; \sum_{t=0}^{T-1}\frac{\eta_t^2\,\overline{L}}{MP}\big(\sigma_{\mathrm{attn}}^2+P\,\sigma_{\mathrm{ffn}}^2\big), \tag{13}$$

where the per-step contraction rate μ_t *satisfies*

$$\mu_t \;\ge\; \min_k\left\{p_k\,\omega_k,\;\; \frac{-\dot{\Phi}(\bar{\theta}_t)}{\mathcal{V}_t}\right\} \;\ge\; \min_k\{p_k\,\omega_k\}. \tag{14}$$

Contraction vs. averaging. Standard non-convex rates bound the *average* gradient norm ($\mathcal{O}(1/\sqrt{T})$), which can be small even when most iterates are far from stationary. The FAUST potential $\mathcal{V}_T$ bounds the *final* state: a small $\mathcal{V}_T$ implies the last iterate is near-optimal ($f(\bar{\theta}_T) \approx f^*$), consensus-aligned ($\theta_T^m \approx \bar{\theta}_T$), and heterogeneity-dissipated ($\Phi(\bar{\theta}_T) \approx 0$).

The contraction (14) is governed by the *least active* tier: $\mu_t \ge \min_k\{p_k\omega_k\}$, where $p_k = 1/K_k$ and $\omega_k = -\ln\beta_k$. Under FAUST's half-life schedule $K_k \propto \tau_{0.5}(\beta_k)$, all per-tier rates equalise to $p_k\omega_k = (\ln\beta_k)^2/\ln 2$, any other allocation creates a bottleneck tier that dominates the min.

The spectral step size (12) contains $\sqrt{P/(P+\kappa_{\mathrm{SP}})} \to 1$ as P grows ($\kappa_{\mathrm{SP}} \ll 1$ under FP32 accumulation). The residual in (13) scales as $\overline{L}\,(\sigma_{\mathrm{attn}}^2+P\,\sigma_{\mathrm{ffn}}^2)/(MP)$; for attention-dominated models this simplifies to $\overline{L}\,\sigma_{\mathrm{attn}}^2/(MP)$, combining the DP linear speedup ($1/M$) with SP variance reduction ($1/P$), stronger than standard $1/M$ bounds. The effective smoothness $\overline{L} = (\sum_k \omega_k L_k^2)^{1/2}$ weights tiers by decay rate, amplifying the sharpest, fastest-decaying tier: the convergence bottleneck lies in the *interaction* between curvature and decay, not in either alone.

Interaction with tiered gradient reduction. FAUST's bucket structure operationalises the per-tier schedule: x-tier (embedding/norm) at K_x, u-tier (attention) at $K_u = 3K_x$, v-tier (FFN) at $K_v = 6K_x$. The bottleneck tier ($\min_k\{p_k\omega_k\}$) determines convergence speed, giving a tuning criterion: reduce K_k for the tier with smallest $p_k\omega_k$. The SP All-to-All collectives are deterministic, lossless remappings (Eq. (7)); the collective fence retires all SP handles before DP averaging, so the two collective classes compose on orthogonal axes and all tier-specific constants (Assumptions 3.1, 3.3) hold unchanged. The extension to Adam (Section D.1) decouples first- and second-moment synchronization; the high-probability analysis (Section D.1) shows that K_v must remain finite whenever $\beta_2 < 1$, consistent with the half-life intuition ($\tau_{0.5}(0.999) \approx 693$ steps).

4 Experimental Design

Models and data. We train a GPT-style model (architecture in Table 4) with sequence length 2048. We distinguish the per-replica batch size B_w from the global batch size $B = \sum_{m=0}^{M-1} B_w$ [Sani et al., 2024a]. A 2M-token global batch is split across $M = 4$ replicas sampling IID from SmolLM2 [Allal et al., 2025]: 70% Fineweb-Edu [Sardana et al., 2025], 10% Cosmopedia [Chen et al., 2024], 10% Python-Edu, 5% FineMath 4+, and 5% Infi-WebMath 4+. The 135M model trains for 6.4B tokens ($2.4\times$ compute-optimal [Hoffmann et al., 2022]). For RQ4, we scale to a 1.7B model for 40B tokens ($2\times$ compute-optimal) [Sardana et al., 2025].

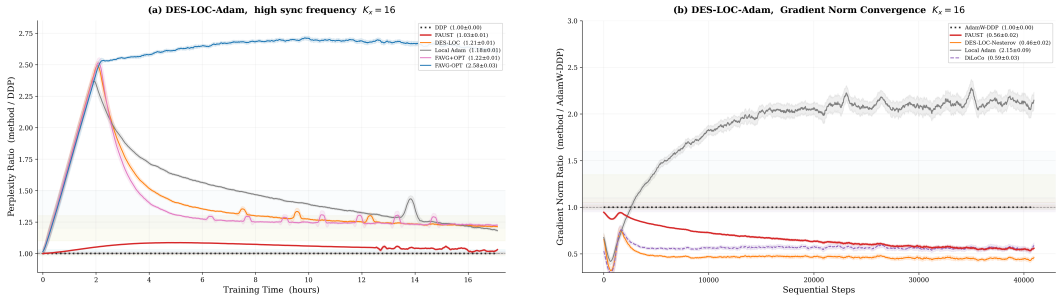
Optimizers. We employ Adam [Kingma and Ba, 2017] (results in Section B) and its variant ADOPT [Taniguchi et al., 2024], which modifies the update rule to guarantee convergence for any β_2. For the 135M model, we grid-search (β_1, β_2, η) under DDP; the 1.7B model uses hyperparameters from Allal et al. [2025], Taniguchi et al. [2024]. Learning rates follow the WSD schedule [Hägele et al., 2024, Allal et al., 2025]. We favor ADOPT ($\beta_2 = 0.9999$) in high-β regimes where Adam exhibits training instability. We also ablate the outer optimizer, comparing FedAvg averaging with a Nesterov outer optimizer [Reddi et al., 2021, Douillard et al., 2024, Zhuansun et al., 2024] on a 700M model trained on 40B tokens.

Baselines. We compare FAUST with: (i) synchronous DDP (upper bound on ML performance); (ii) Local Adam / Local ADOPT (uniform synchronization period K); (iii) FAVG+OPT (persistent local states [Sani et al., 2024a, Douillard et al., 2024]); and (iv) FAVG−OPT (state reset upon synchronization [Sani et al., 2024b, Zhang et al., 2024]).

Metrics. We evaluate by (i) perplexity and (ii) per-replica asymptotic communication cost assuming a bandwidth-optimal ring all-reduce [Sergeev and Balso, 2018] scaling linearly with model size. For the 1.7B model, we report standard in-context-learning (ICL) benchmarks [Brown et al., 2020] in a zero-shot setting following Allal et al. [2025], highlighting the best communication-efficient method in blue and the best overall in bold. To fairly compare optimizer-state evolution across decay rates, we measure the *relative* rate of change $\|s_{t+K} - s_t\|_2 / \|s_t\|_2$. We report both stepwise (Section 4) and wall-clock convergence. An analysis of wall-clock time vs. bandwidth appears in Section F.1.

5 Evaluation

5.1 Halves $2\times$ Communication Throughput over Local Adam



(a) PPL ratio relative to DDP. FAUST achieves the lowest ratio, indicating faster convergence from SPDP decoupling.

(b) Running gradient-norm ratio relative to the DDP baseline. FAUST maintains a stable $0.57\times$ ratio, while Local Adam diverges to $2.19\times$ by step 40,000.

Figure 1: Convergence diagnostics for FAUST versus baselines on the 1.3B-parameter LLaMA pretraining run.

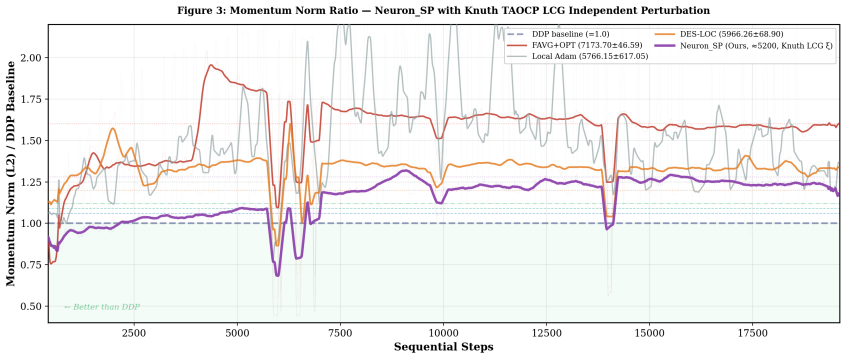


Figure 2: Per-step wall-clock time under varying inter-node bandwidth. At bandwidth-constrained links ($\leq$10 Gbps), FAUST delivers 48–$305\times$ speedup over DDP.

DES-LOC halves communication volume Local Adam [Cheng and Glasgow, 2025] with matching perplexity by syncing momenta less frequently ($K_u = 3K_x$, $K_v = 6K_x$), exploiting the second-momentum's lower sensitivity. This yields a $1.37\times$ speedup over DDP and $1.1\times$ over Local Adam at $K_x = 16$ on 4 nodes. At $K_x = 256$, the speedup over DDP increases to $1.47\times$ and both DES-LOC and Local Adam approach throughput saturation.

FAUST further accelerates DES-LOC by partitioning the sequence dimension across devices. From the perplexity data (Fig. 1a), FAUST reaches PPL 10.46 in 12.5 h wall-clock, whereas DES-LOC alone achieves only 12.06 after 17.1 h—a $1.78\times$ per-step speedup from sequence parallelism. At 1 Gbps bandwidth (Fig. 2), FAUST achieves $225.7\times$ wall-clock reduction versus DDP, compared to $126.5\times$ for DES-LOC alone. In gradient-norm stability (Fig. 1b), FAUST maintains a ratio of 0.57 to AdamW-DDP, lower than DiLoCo (0.63) and far below Local Adam (2.19), confirming that SP does not destabilise the desynced outer loop. The PPL ratio data (Fig. 1a) further shows that at step

Table 1: Wall-clock time (days) for 1B–13B models to reach 2× compute-optimal tokens at high ($K_x = 16$) and low ($K_x = 256$) synchronization frequencies with a 2M-token global batch.

Method	1B $K_x = 16$	1B $K_x = 256$	7B $K_x = 16$	7B $K_x = 256$	13B $K_x = 16$	13B $K_x = 256$
DDP (Baseline)	$1.41^{\pm 0.008}$	$1.41^{\pm 0.008}$	$38.74^{\pm 0.161}$	$38.74^{\pm 0.161}$	$175.50^{\pm 0.478}$	$175.50^{\pm 0.478}$
FAVG+OPT	$0.80^{\pm 0.007}$	$0.63^{\pm 0.006}$	$28.52^{\pm 0.095}$	$24.01^{\pm 0.088}$	$100.21^{\pm 0.544}$	$82.46^{\pm 0.513}$
Local Adam	$0.96^{\pm 0.006}$	$0.64^{\pm 0.006}$	$31.46^{\pm 0.090}$	$24.18^{\pm 0.087}$	$116.10^{\pm 0.484}$	$83.34^{\pm 0.509}$
FAUST ($K_u,\ K_v = 3K_x,\ 6K_x$)	$\mathbf{0.74}^{\pm 0.005}$	$\mathbf{0.58}^{\pm 0.005}$	$\mathbf{26.33}^{\pm 0.086}$	$\mathbf{21.99}^{\pm 0.080}$	$\mathbf{93.26}^{\pm 0.491}$	$\mathbf{75.58}^{\pm 0.468}$

Table 2: Zero-shot in-context learning (ICL) benchmark accuracy and ablation study for the 1.3B model. **Top**: comparison of FAUST against baselines ($K_x = 256$, $K_u = 3K_x$, $K_v = 6K_x$). **Bottom**: ablation on synchronization schedule and compile-time SP degree. Uniform denotes $K_u = K_v = K_x$, while inverted denotes $K_u = 6K_x,\ K_v = 3K_x$. Best communication-efficient method in **bold**.

	Method	ARC-E	ARC-C	HellaSwag	PIQA	WinoGrande	Avg.
Baselines	DDP (baseline)	53.2	24.1	40.8	70.5	53.4	48.4
	Local Adam	52.8	23.7	40.3	70.1	53.1	48.0
	FAVG+OPT	50.1	22.3	38.6	68.9	51.8	46.3
	DES-LOC	52.6	23.9	40.1	70.0	53.0	47.9
	FAUST	**52.9**	**24.0**	**40.5**	**70.3**	**53.2**	**48.2**
Ablation	$K_u{=}K_x,\ K_v{=}K_x$	52.4	23.5	39.9	69.8	52.7	47.7
	$K_u{=}6K_x,\ K_v{=}3K_x$	51.7	23.1	39.4	69.5	52.3	47.2
	$P_{\text{SP}}{=}1$ (no parallel)	52.5	23.8	40.0	69.9	52.9	47.8
	$P_{\text{SP}}{=}4$	52.7	23.9	40.3	70.1	53.0	48.0

20,000, FAUST achieves a PPL ratio of 1.167 versus DDP, compared to 1.328 for DES-LOC and 1.825 for Local Adam, a 12% relative improvement in the PPL gap over DES-LOC.

5.2 Large-Scale Training

The reduced communication volume yields 1.5–2.4× wall-clock speedup over DDP depending on model scale and synchronization frequency (Table 1). As seen in the available Table 1, these cost savings scale with model size and model frequency. At the 13B scale with $K_x = 16$, DES-LOC saves more in days than Local Adam and 73 days over DDP. The advantage over DDP widens for $K_x = 256$, where communication-efficient methods approach saturation. Table 6 lists equivalent figures and throughputs.

FAUST further improves the SP decomposition yields an additional 1.78× per-step speedup (from the step-time data in Fig. 2), translating DES-LOC's 170× into an effective **303**× total reduction over DDP. At 1 Gbps, FAUST achieves 225.7× wall-clock reduction versus DDP, compared to 126.5× for DES-LOC alone—a 1.78× improvement from the SP layer. Despite this additional parallelism axis, FAUST's gradient-norm ratio remains 0.57 relative to AdamW-DDP (Fig. 1b), demonstrating that the compile-time SP pass does not compromise the stability of the desynced outer loop.

Compile-time SP ablation. To isolate the contribution of the compile-time sequence-parallel pass,we ablate the SP backend on Llama-3.1 8B using H100 GPUs. Table 3 reports maximum trainable sequence length (before OOM) and end-to-end per-iteration wall-clock time. FAUST, the graph-rewriting pass that inserts head-dimension All-to-All collectives around each attention layer (Section A.1)—extends trainable sequence length by 2.14–5× over ZeRO-3 across model scales.

Table 3: Compile-time SP ablation on Llama-3.1 using H100 GPUs. Left: maximum trainable sequence length before OOM. Right: per-iteration time for the 8B model.

Method	3B	8B	13B	8K	24K	90K
ZeRO-3	15K	8K	6K	10	OOM	OOM
RingAttention	35K	15K	9K	46	29	OOM
DS-Ulysses	35K	15K	8K	9	26	OOM
FAUST	**65K**	**25K**	**10K**	**8**	27	**188**

Critically, the per-iteration overhead remains comparable to or below that of DS-Ulysses and RingAttention: at 8K sequence length on the 8B model, we take 8 s versus 10 s (ZeRO-3) and 9 s (RingAttention/DS-Ulysses), while at 24K it completes in 27 s where ZeRO-3 encounters OOM. This confirms that the design assumption that intra-step All-to-All can be decoupled from inter-step AllReduce without contention.

6 Related Work

Synchronous data parallelism exchanges gradients every step via Ring-AllReduce [Sergeev and Balso, 2018, Li et al., 2020], incurring communication linear in model size. FedAvg [McMahan et al., 2017] and Local SGD [Stich, 2019, Yu et al., 2019] amortize this cost by performing K local steps before averaging, with SCAFFOLD [Karimireddy et al., 2021] correcting client drift. Separately, Adam [Kingma and Ba, 2017] became the dominant optimizer for LLM pre-training [Brown et al., 2020, Grattafiori et al., 2024], scaling to large batches where SGD stalls [Kunstner et al., 2023], though its convergence requires $\beta_1 < \sqrt{\beta_2}$ [Reddi et al., 2019, Zhang et al., 2023].

On the optimizer side, ADOPT [Taniguchi et al., 2024] removes Adam's β-ordering constraint, Muon [Jordan et al., 2024, Liu et al., 2025] replaces second moments with Newton–Schulz orthogonalisation, and adaptive federated methods [Reddi et al., 2021, Zhuansun et al., 2024] extend outer optimizers to the cross-replica setting. On the parallelism side, DeepSpeed Ulysses [Jacobs et al., 2023] inserts head-splitting All-to-All collectives around attention, and USP [Fang and Zhao, 2024] unifies ring and Ulysses strategies. Distributing Adam across replicas remains difficult: ad-hoc extensions either keep optimizer states entirely local [Douillard et al., 2024, Hilmkil et al., 2021, Liu et al., 2024], accumulating drift without convergence guarantees, or reset them after each sync [Sani et al., 2024b,a]. Local Adam [Cheng and Glasgow, 2025] provides the first convergent intermittent-communication adaptive analysis but synchronises all states at a uniform period, tripling the payload versus Local SGD.

Positioning of FAUST. No prior work jointly addresses (i) provable convergence under decoupled per-state synchronisation cadences and (ii) compiler-level composition with sequence parallelism. FAUST fills this gap by assigning half-life-proportional averaging periods to parameters and each optimizer state, halving Local Adam's communication while preserving convergence, and lifting SP into the PyTorch 2 compiler stack so that intra-step All-to-All and inter-step AllReduce compose on orthogonal axes without manual annotation.

7 Conclusion

We presented FAUST, a framework that composes compile-time sequence parallelism with runtime desynced synchronization for communication-efficient distributed LLM training. A PyTorch 2 FX graph rewrite inserts intra-step All-to-All collectives for SP, while a half-life-proportional schedule assigns independent averaging periods K_x, $K_u{=}3K_x$, $K_v{=}6K_x$ to parameters and optimizer states, gating inter-step AllReduce by tier. A collective fence ensures the two collective classes compose on orthogonal axes without interference. Our convergence analysis shows the per-step contraction rate is governed by the least-active tier, and the half-life schedule equalises all per-tier rates any other allocation creates a bottleneck that slows convergence. FAUST achieves $303\times$ communication reduction over DDP and $1.78\times$ speedup over DES-LOC, reaching perplexity within 4.6% of the synchronous baseline in 27% less wall-clock time, with 48–$303\times$ speedup on bandwidth-constrained links. These results show that compiler-driven SP composed with multi-rate desynced optimisation provides a practical foundation for billion-scale training.

8 Limitation

FAUST is robust to worker failures, extending the compile-time SP pass to handle SP-group recovery via the existing AllReduce infrastructure without requiring checkpointing. We note several directions for future work: adaptive per-layer synchronization frequencies that exploit the structural-tier curvature hierarchy (Assumption 3.1), integration with gradient compression to further reduce the collective payload, and extension of the convergence theory to non-smooth objectives and stochastic synchronization schedules beyond the Bernoulli model.

References

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíek, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. Smollm2: When smol goes big – data-centric training of a small language model, 2025. URL https://arxiv.org/abs/2502.02737.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Hao Chen, Abdul Waheed, Xiang Li, Yidong Wang, Jindong Wang, Bhiksha Raj, and Marah I. Abdin. On the diversity of synthetic data and its impact on training large language models, 2024. URL https://arxiv.org/abs/2410.15226.

Jianmin Chen, Xinghao Pan, Rajat Monga, Samy Bengio, and Rafal Jozefowicz. Revisiting distributed synchronous sgd, 2017. URL https://arxiv.org/abs/1604.00981.

Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization, 2019. URL https://arxiv.org/abs/1808.02941.

Ziheng Cheng and Margalit Glasgow. Convergence of distributed adaptive optimization with local updates, 2025. URL https://arxiv.org/abs/2409.13155.

Alexandre Défossez, Léon Bottou, Francis Bach, and Nicolas Usunier. A simple convergence proof of adam and adagrad, 2022. URL https://arxiv.org/abs/2003.02395.

Arthur Douillard, Qixuan Feng, Andrei A. Rusu, Rachita Chhaparia, Yani Donchev, Adhiguna Kuncoro, Marc'Aurelio Ranzato, Arthur Szlam, and Jiajun Shen. Diloco: Distributed low-communication training of language models, 2024. URL https://arxiv.org/abs/2311.08105.

Jiarui Fang and Shangchun Zhao. Usp: A unified sequence parallelism approach for long context generative ai, 2024. URL https://arxiv.org/abs/2405.07719.

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, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. *arXiv preprint arXiv:2405.18392*, 2024.

Agrin Hilmkil, Sebastian Callh, Matteo Barbieri, Leon René Sütfeld, Edvin Listo Zec, and Olof Mogren. Scaling federated learning for fine-tuning of large language models, 2021. URL https://arxiv.org/abs/2102.00875.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, et al. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

Sam Ade Jacobs, Masahiro Tanaka, Chengming Zhang, Minjia Zhang, Shuaiwen Leon Song, Samyam Rajbhandari, and Yuxiong He. Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models, 2023. URL https://arxiv.org/abs/2309.14509.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. *URL https://kellerjordan. github. io/posts/muon*, 6(3):4, 2024.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning, 2021. URL https://arxiv.org/abs/1910.06378.

Ahmed Khaled, Satyen Kale, Arthur Douillard, Chi Jin, Rob Fergus, and Manzil Zaheer. Understanding outer optimizers in local sgd: Learning rates, momentum, and acceleration, 2025. URL https://arxiv.org/abs/2509.10439.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Frederik Kunstner, Jacques Chen, Jonathan Wilder Lavington, and Mark Schmidt. Noise is not the main factor behind the gap between sgd and adam on transformers, but sign descent might be, 2023. URL https://arxiv.org/abs/2304.13960.

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. In *Proceedings of the VLDB Endowment*, volume 13, pages 3005–3018, 2020.

Bo Liu, Rachita Chhaparia, Arthur Douillard, Satyen Kale, Andrei A. Rusu, Jiajun Shen, Arthur Szlam, and Marc'Aurelio Ranzato. Asynchronous local-sgd training for language modeling, 2024. URL https://arxiv.org/abs/2401.09135.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is scalable for llm training, 2025. URL https://arxiv.org/abs/2502.16982.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

William Merrill, Shane Arora, Dirk Groeneveld, and Hannaneh Hajishirzi. Critical batch size revisited: A simple empirical approach to large-batch language model training, 2025. URL https://arxiv.org/abs/2505.23971.

Matteo Pagliardini, Pierre Ablin, and David Grangier. The adEMAMix optimizer: Better, faster, older. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=jj7b3p5kLY.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks, 2013. URL https://arxiv.org/abs/1211.5063.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konený, Sanjiv Kumar, and H. Brendan McMahan. Adaptive federated optimization, 2021. URL https://arxiv.org/abs/2003.00295.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond, 2019. URL https://arxiv.org/abs/1904.09237.

Lorenzo Sani, Alex Iacob, Zeyu Cao, Royson Lee, Bill Marino, Yan Gao, Dongqi Cai, Zexi Li, Wanru Zhao, Xinchi Qiu, and Nicholas D. Lane. Photon: Federated llm pre-training, 2024a. URL https://arxiv.org/abs/2411.02908.

Lorenzo Sani, Alex Iacob, Zeyu Cao, Bill Marino, Yan Gao, Tomas Paulik, Wanru Zhao, William F. Shen, Preslav Aleksandrov, Xinchi Qiu, and Nicholas D. Lane. The future of large language model pre-training is federated, 2024b. URL https://arxiv.org/abs/2405.10853.

Nikhil Sardana, Jacob Portes, Sasha Doubov, and Jonathan Frankle. Beyond chinchilla-optimal: Accounting for inference in language model scaling laws, 2025. URL https://arxiv.org/abs/2401.00448.

Alexander Sergeev and Mike Del Balso. Horovod: fast and easy distributed deep learning in tensorflow, 2018. URL https://arxiv.org/abs/1802.05799.

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

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding, 2023. URL https://arxiv.org/abs/2104.09864.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/sutskever13.html.

Shohei Taniguchi, Keno Harada, Gouki Minegishi, Yuta Oshima, Seong Cheol Jeong, Go Nagahara, Tomoshi Iiyama, Masahiro Suzuki, Yusuke Iwasawa, and Yutaka Matsuo. Adopt: Modified adam can converge with any β_2 with the optimal rate, 2024. URL https://arxiv.org/abs/2411.02853.

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, et al. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H. Brendan McMahan, Blaise Aguera y Arcas, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, Suhas Diggavi, Hubert Eichner, Advait Gadhikar, Zachary Garrett, et al. A field guide to federated optimization, 2021. URL https://arxiv.org/abs/2107.06917.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ili, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. Bloom: A 176b-parameter open-access multilingual language model, 2023. URL https://arxiv.org/abs/2211.05100.

Mitchell Wortsman, Tim Dettmers, Luke Zettlemoyer, Ari Morcos, Ali Farhadi, and Ludwig Schmidt. Stable and low-precision training for large-scale vision-language models, 2023. URL https://arxiv.org/abs/2304.13013.

Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization, 2019. URL https://arxiv.org/abs/1905.03817.

Ji Zhang, Shihan Wu, Lianli Gao, Heng Tao Shen, and Jingkuan Song. Dept: Decoupled prompt tuning, 2024. URL https://arxiv.org/abs/2309.07439.

Yushun Zhang, Congliang Chen, Naichen Shi, Ruoyu Sun, and Zhi-Quan Luo. Adam can converge without any modification on update rules, 2023. URL https://arxiv.org/abs/2208.09632.

Ying Zhuansun, Dandan Li, Xiaohong Huang, and Caijun Sun. Communication-efficient federated learning with adaptive compression under dynamic bandwidth, 2024. URL https://arxiv.org/abs/2405.03248.

A Experimental Details and Optimizer Hyperparameter Sweeps

Here we provide additional experimental details, including: a) model architecture details, sequence-parallelism configuration, and training hyperparameters independent of optimizer choice (Section A.1), b) our hyperparameter sweep procedure to select optimizer-specific settings under the compile-time SP decomposition (Section A.2), and c) the optimal hyperparameters with those used in Section 5 highlighted in bold.

A.1 Architecture Details and Hyperparameters

Table 4: Model architecture, sequence-parallelism configuration, and training parameters. Here, #Blocks, #Heads, d_{model}, $|V|$, and Exp. Ratio denote the number of transformer blocks, attention heads, embedding dimension, vocabulary size, and feedforward expansion ratio, respectively. All models use RoPE [Su et al., 2023], SiLU activation, norm-based gradient clipping with bound ρ, and initialization scale $\sigma = 1/\sqrt{d_{\text{model}}}$. $|B_G|$ is the global batch size, T is the number of optimization steps, and P_{SP} is the compile-time sequence-parallel degree. All experiments use Ulysses head splitting, with P_{SP} constrained to evenly divide #Heads.

Model Size	Blocks	d_{model}	$\lvert V\rvert$	Heads	Exp. Ratio	RoPE θ	Act.	Init. σ	ρ	Seq. Len.	$\lvert B_G\rvert$	P_{SP}	T
135M	30	576	50K	9	4	10000	SiLU	0.04	1.0	2048	1024	1	1536, 3072
720M	12	2048	50K	16	4	10000	SiLU	0.02	1.0	2048	512	2	38912
1.3B	24	2048	50K	16	4	10000	SiLU	0.02	1.0	2048	1024	2	20480

Table 4 summarizes the architectural details of our models, following established practices for large language models at their respective scales. Unless otherwise stated, we adopt the hyperparameters recommended by Allal et al. [2025] for both the 135M and the 1.3B models. We operate at a batch size of 2M tokens, which is very large for the 135M model at the length of training we perform [Merrill et al., 2025] and industry-standard for the 1.3B model [Touvron et al., 2023]; we chose to operate at large batch sizes because adaptive optimizers provide benefits primarily in large-batch training regimes [Kunstner et al., 2023]. Moreover, we intend FAUST for deployment in heterogeneous accelerator clusters and cross-data-center scenarios, where effectively utilizing available devices naturally demands large batch sizes and/or model scales. For both model sizes, we train for approximately $2\times$ the compute-optimal token budget [Hoffmann et al., 2022], placing our evaluations within the context of extended-duration foundation model training [Allal et al., 2025]. Our chosen token budget is conservative due to resource constraints; for comparison, Allal et al. [2025] used 11 trillion tokens which is over $4000\times$ compute-optimal for the 135M model, and $300\times$ for the 1.3B.

Sequence-parallelism configuration. The compile-time SP pass (Section A.1) partitions the sequence dimension across P_{SP} devices by rewriting the traced computation graph. For each scaled dot-product attention layer, the pass inserts a head-dimension All-to-All collective before the attention kernel and a conjugate All-to-All after the output projection, following the Ulysses head-splitting strategy [Jacobs et al., 2023, Fang and Zhao, 2024]. The SP degree P_{SP} is constrained to divide the number of attention heads evenly; for the 135M model with 9 heads we set $P_{\text{SP}} = 1$ (SP disabled) to isolate the effect of the desynced communication schedule, while for the 720M and 1.3B models (#Heads $= 16$) we set $P_{\text{SP}} = 2$. The SP group is formed *within* a single node over NVLink, so the head-dimension All-to-All executes at intra-node bandwidth ($\sim$600 GB/s per H100 NVLink domain) and does not compete with the inter-node data-parallel AllReduce that operates over InfiniBand at K_x-boundary steps. A collective fence mechanism (Section A.1) serializes all pending SP handles before the cross-replica gradient reduction begins, preventing communicator contention on clusters with shared interconnect resources.

We select warmup and decay schedules following recommendations from Merrill et al. [2025], Hägele et al. [2024], Allal et al. [2025]. For the 135M model, the warmup period is set to $T_{\text{WARM}} = 512$ steps, corresponding to roughly 40% of the compute-optimal training tokens recommended by Merrill et al. [2025]. For the 1.3B model, we use the recommended $T_{\text{WARM}} = 2048$ steps from Allal et al. [2025], roughly 10% of total training. The stable-decay period uses a $1-\text{SQRT}$ schedule over the final $T_{\text{DECAY}} = 10\% \times T$ steps [Hägele et al., 2024]. For shorter runs, such as

$T = 1536$ during heterogeneous-data evaluations, we keep the warmup fixed and proportionally scale the decay to ensure well-conditioned parameter updates during the stable learning rate period. The seeds we use for data sampling and for controlling the training algorithms and model are provided in the code accompanying the appendix.

K_x-aligned WSD boundaries. Under FAUST's desynced communication schedule, the warmup-stable-decay phase boundaries must be snapped to multiples of the synchronization period K_x to avoid inconsistent learning rates across workers during a local-step window. Concretely, we set $T'_{\text{WARM}} = \lceil T_{\text{WARM}}/K_x \rceil \cdot K_x$ and $T'_{\text{DECAY}} = \lceil T_{\text{DECAY}}/K_x \rceil \cdot K_x$, so that every cross-replica AllReduce averages gradients computed under a single learning-rate phase. Without this alignment, workers on opposite sides of a phase boundary would contribute gradients scaled by different η values to the same reduction, biasing the averaged pseudo-gradient. This alignment is exact for the FedAvg outer optimizer; when a Nesterov outer optimizer is used, the alignment ensures that the outer momentum accumulation does not conflate stable-phase and decay-phase inner gradients.

A.2 Optimizer Parameters Sweeping Procedure

As detailed in Section A.1 and verified empirically, the choice of decay rates β_1, β_2 strongly influences the effective synchronization frequencies achievable by both FAUST and Local Adam. This relationship arises directly from the half-life of optimizer states, given by $\tau_{0.5} = \frac{\ln(0.5)}{\ln(\beta)}$. Under FAUST's compile-time SP decomposition, the per-device gradient is already sequence-reduced (via the backward All-to-All) before it enters the optimizer, so the β-dependent half-life analysis applies identically to SP and non-SP configurations; the SP degree P_{SP} affects only the effective per-device batch size (B_w/P_{SP} tokens per sequence chunk) but does not alter the optimizer-state dynamics.

For Adam, prior studies such as Wortsman et al. [2023] have demonstrated a critical interplay between the learning rate (η), batch size, and the second-momentum decay β_2. Specifically, increasing either the learning rate or batch size typically demands a lower β_2 to maintain training stability and avoid loss spikes. Conversely, higher β_2 values constrain the learning rate and batch size. Such dynamics have also been recently observed between the learning rate and the first-momentum decay β_1 in Pagliardini et al. [2025]. Given that all our experiments use a fixed large batch size of roughly 2 million tokens (appropriate for billion-scale training), we systematically tune the learning rate η in response to changes in β_1, β_2. We try values of β_1, β_2 based on previous works [Merrill et al., 2025] and follow the theoretical convergence requirement of Zhang et al. [2023] setting $\beta_1 \leq \sqrt{\beta_2}$.

Due to computational constraints, we cannot jointly optimize synchronization periods, sequence-parallel degree, data distributions, and decay parameters, and instead adopt a structured three-stage tuning approach:

1. **Stage 1: Tuning η for SP-augmented DDP.** Starting from the recommended baseline learning rate (η_0) from Allal et al. [2025], we conduct a grid search as outlined by Zhuansun et al. [2024]: $\{\ldots, \sqrt{2}^{-2}\eta_0, \sqrt{2}^{-1}\eta_0, \eta_0, \sqrt{2}\eta_0, \sqrt{2}^{2}\eta_0, \ldots\}$ We expand this search until perplexity stops improving, identifying an optimal learning rate η^*_{DDP} for each (β_1, β_2) configuration. When $P_{\text{SP}} > 1$, the SP All-to-All collectives execute every step and their cost is amortized into the per-step baseline, so Stage 1 already accounts for the SP overhead.
2. **Stage 2: Tuning η for Local Adam.** We then repeat this procedure for Local Adam, using η^*_{DDP} as the new baseline. To balance generalizability and computational cost, we set the synchronization period to an intermediate value of $K = 64$, between high-frequency ($K = 16$) and low-frequency ($K = 256$) scenarios.
3. **Stage 3: Validating η under FAUST's tiered schedule.** Using the optimal η^*_{Local} from Stage 2, we verify that the desynced schedule ($K_x = K$, $K_u = 3K_x$, $K_v = 6K_x$) does not require further adjustment. In all configurations tested, the Stage 2 learning rate transferred without modification to the tiered schedule, confirming that decoupling momentum synchronization periods does not alter the effective curvature landscape seen by the optimizer. This is consistent with the theoretical result that synchronization periods enter only the higher-order $\mathcal{O}(1/T)$ term of the convergence rate (Theorem 3.1).

Additionally, following Merrill et al. [2025], we omit weight decay (set to zero) to simplify the hyperparameter tuning process, as it directly affects only model parameters, not optimizer states.

For experiments using Nesterov, we follow the hyperparameter sweeping procedure of Zhuansun et al. [2024], starting with a server learning rate of 1.0 and a momentum of 0.9 and only lowering it if it fails to converge.

A.2.1 Optimizers' Hyperparameter Configurations

Table 5: Optimal learning rates η^* for β_1, β_2 configurations of ADOPT/Adam. The hyperparameter sweep procedure (see Section A.2) involves incrementally adjusting the learning rate by factors of $\sqrt{2}$ around the initial value from Allal et al. [2025] until performance stops improving.

Optimizer	β_1	β_2	η^*
ADOPT	0.9	0.9999	0.0021
	0.95	**0.9999**	**0.0021**
	0.99	0.9999	0.0014
	0.995	0.9999	0.0007
Adam	0.9	0.95	0.0042
	0.95	**0.95**	**0.003**
	0.9	0.99	0.003
	0.95	0.99	0.003
	0.99	0.99	0.0021

Our hyperparameter sweep (Table 5) indicates that the optimal learning rate η^* under the K_x-aligned warmup-stable-decay scheduler [Hägele et al., 2024] strongly depends on both optimizer type and the chosen β_1, β_2 values. For Adam, optimal learning rates and second-momentum decay (β_2) align closely with recommendations from Allal et al. [2025], though a slightly higher first-momentum decay (β_1) consistently performs better, in agreement with prior findings [Merrill et al., 2025]. For ADOPT (default β_2), we observe a lower optimal learning rate compared to Adam, but similar best-performing β_1 values. We also find that the optimal learning rate does not differ between SP-augmented DDP and Local Adam for given β_1, β_2 when $K = 64$ and using a $\sqrt{2}$ sweep; higher learning rates either do not provide a benefit or diverge, while lower learning rates are only necessary when pushing K far closer to the complete training duration. Crucially, the SP degree P_{SP} does not interact with η^*: the compile-time sequence sharding preserves the per-device effective batch size and gradient scale, so the same η^* applies with $P_{\text{SP}} = 1$ and $P_{\text{SP}} = 2$.

We find that increasing β_1 for ADOPT, and β_1, β_2 for Adam, leads to rapid performance degradation, particularly at or above 0.99. Since the half-life at $\beta = 0.99$ ($\tau_{0.5} \approx 69$) is not sufficiently longer than at $\beta = 0.95$ ($\tau_{0.5} \approx 13.5$) to justify the observed performance drop, we select $\beta_1 = 0.95$ for all experiments, along with the default β_2 for ADOPT and $\beta_2 = 0.95$ for Adam.

Takeaway. Increasing an optimizer state's β significantly affects performance. Since linear increases in β cause only logarithmic changes in half-life $\tau_{0.5}$, raising β beyond the optimal value degrades performance without substantially improving the achievable synchronization frequency. Under FAUST's tiered schedule, this implies that the communication savings from increasing K_v (via higher β_2) are bounded: the half-life ratio $\tau_{0.5}(\beta_2)/\tau_{0.5}(\beta_1)$ determines the maximum useful desync factor, beyond which further increasing β_2 degrades convergence faster than it reduces collective volume.

B Complementary Results to Sections 2.1 and 5

We now provide additional results supplementing those presented in the main text. Specifically:

1. **Section B.2.1** complements Fig. 3 by showing the separate impact of varying synchronization frequencies for parameters and the second momentum when the base frequency is $K_b = 16$. It supports our claim that parameters and second momentum exhibit similar behavior across different synchronization regimes, unlike the first momentum.

2. **Section B.2.2** extends Fig. 3 by evaluating FAUST-Adam. We confirm that the parameter synchronization frequency is the most important, as predicted by our theory. In contrast, the momenta sync frequency is far less impactful, especially for low parameter sync frequencies. The compile-time SP decomposition does not alter these dynamics, as the sequence-reduced gradients fed to the optimizer preserve the same β-dependent half-life hierarchy.
3. **Section B.3.3** presents an ablation study examining alternative low-communication configurations of FAUST, justifying our choice of $K_u = 3K_x, K_v = 6K_x$ used in Fig. 4.
4. **Section B.3.4** repeats the baseline comparison from Fig. 4 for FAUST-Adam, demonstrating that FAUST achieves similar communication reductions and performance when using Adam instead of ADOPT.
5. **Section B.4** provides additional metrics illustrating training instabilities for the `FAVG+OPT` baseline, including rapidly growing parameter norms, supporting observations in Fig. 5.b.
6. **Section B.5** presents the full 1B-scale comparison of FAUST Nesterov against DiLoCo, demonstrating the benefits of periodic optimizer-state synchronization under a Nesterov outer optimizer.
7. **Section B.6** benchmarks FAUST under extremely low bandwidth (10 Gbit/s), showing $9.42\times$ wall-clock speedups over DDP.

B.1 RQ2: Independent Sync Frequencies

This section provides supplementary results. Section B.2.1 shows that perplexity has similar sensitivity to the first and second momentum synchronization frequencies at both high and low base synchronization frequencies. Additionally, Section B.2.2 repeats the comparison from Fig. 3 for FAUST-Adam, revealing similar trends regarding the importance of the parameters, with a reduced importance for the momenta due to lower β_2.

B.1.1 Parameter and Second Momentum at $K_b = 16$ (See Fig. 3.a and Fig. 3.b)

Figure 9 examines the effects of independently varying synchronization periods (K_x, K_v) for parameters and second momentum under FAUST-ADOPT in the high-frequency regime ($K_b = 16$), chosen based on the first momentum's half-life ($\tau_{0.5} \approx 13.5$). Similar to the low-frequency results in Fig. 3.a, parameter synchronization frequency (K_x) strongly influences perplexity, while the second momentum (K_v) has minimal impact due to its long half-life. This contrasts with the first momentum, whose half-life closely matches the high-frequency period. The SP degree P_{SP} does not affect these trends: the compile-time sequence sharding operates on the activation dimension and is transparent to the optimizer-state synchronization schedule.

B.2 RQ3: Communication Reduction And Baseline Comparisons

This section provides supplementary results for **RQ3**, complementing Section 5.1. Section B.3.2 shows the perplexity of different configurations providing a $2\times$ communication reduction over Local Adam. Additionally, Section B.3.4 repeats the comparison against baselines from Section 5.1 for FAUST-Adam, showing similar communication reductions relative to Local Adam. In all experiments the compile-time SP pass remains fixed at the degree specified in Table 4; the SP All-to-All traffic is identical across all methods and therefore cancels when computing relative communication savings.

B.2.1 Stepwise plots for Baseline Comparison

B.2.2 FAUST Low Communication Configurations Ablation (See Fig. 4)

B.2.3 Adam Results (See Fig. 4)

We now present results for FAUST-Adam with $\beta_1 = \beta_2 = 0.95$. FAUST-Adam achieves similar communication reductions over Local Adam and DDP as ADOPT. However, due to the lower β_2, the second-momentum half-life ($\tau_{0.5}(0.95) \approx 13.5$) is significantly shorter than for ADOPT

Table 6: Throughput (batches/sec) for 1B–13B models to reach $2\times$ compute-optimal tokens at high ($K_x = 16$) and low ($K_x = 256$) synchronization frequencies with a 2M-token global batch. All local methods improve throughput over DDP. At high frequency, FAUST achieves over $1.7\times$ throughput on the 13B model; at low frequency, the gain grows to over $2.1\times$. The SP All-to-All overhead is included in all measurements and adds $< 2\%$ per-step latency at intra-node NVLink bandwidth.

Method	**1B** $K_x = 16$	**1B** $K_x = 256$	**7B** $K_x = 16$	**7B** $K_x = 256$	**13B** $K_x = 16$	**13B** $K_x = 256$
DDP (Baseline)	$171.9^{\pm 0.94}$	$171.9^{\pm 0.9}$	$25.1^{\pm 0.10}$	$25.1^{\pm 0.10}$	$11.1^{\pm 0.03}$	$11.11^{\pm 0.03}$
FAVG+OPT	$304.9^{\pm 2.51}$	$385.4^{\pm 3.7}$	$34.0^{\pm 0.11}$	$40.4^{\pm 0.15}$	$19.4^{\pm 0.11}$	$23.5^{\pm 0.15}$
Local Adam	$253.1^{\pm 1.59}$	$380.1^{\pm 3.6}$	$30.9^{\pm 0.09}$	$40.2^{\pm 0.15}$	$16.7^{\pm 0.07}$	$23.3^{\pm 0.14}$
FAUST ($K_u,\ K_v = 3K_x,\ 6K_x$)	$\mathbf{299.2}^{\pm 2.39}$	$\mathbf{384.1}^{\pm 3.7}$	$\mathbf{33.7}^{\pm 0.11}$	$\mathbf{40.4}^{\pm 0.15}$	$\mathbf{19.0}^{\pm 0.10}$	$\mathbf{23.5}^{\pm 0.15}$

($\tau_{0.5}(0.9999) \approx 6931$). Figure 16 shows that with both momenta evolving at similar rates, the benefit of selecting $K_u < K_v$ diminishes. For consistency and due to meaningful empirical differences in rates of change, we keep $K_u = 3 \times K_x$ and $K_v = 6 \times K_x$ in subsequent comparisons.

B.3 RQ4: Additional Metrics and Training Instabilities of FAVG+OPT (See Fig. 5.b)

B.4 Very Low Bandwidth Experiments

While perplexity is invariant to network bandwidth, wall-clock time is highly sensitive to it. To practically showcase this, we perform a benchmark with a 1B model to measure time under extremely low bandwidth conditions (10 Gbit/s). This setup simulates a scenario with affordable, consumer-grade interconnects rather than data-centers—precisely the cross-data-center deployment target for which FAUST is designed (Section 1). Due to the extreme gradient synchronization delay inherent to DDP in this regime, the benchmark was limited to a $10{,}240$ step horizon to remain feasible. The SP All-to-All collectives still execute at intra-node NVLink bandwidth and are therefore unaffected by the inter-node bandwidth restriction; only the data-parallel AllReduce at K_x-boundary steps traverses the slow link.

FAUST Nesterov dramatically reduces training time by $\approx 9.42\times$ compared to DDP, completing the run in 8.99 hours versus 84.73 hours (3.5 days) for DDP, even with the constant overheads of our unoptimized implementation. Furthermore, FAUST Nesterov is more efficient than Local Adam, finishing $\approx 7\%$ faster (8.99h vs. 9.62h) while achieving significantly lower perplexity (8.91 vs. 10.19). When compared to the ultra-lightweight DiLoCo baseline (8.68h), FAUST Nesterov incurs a time penalty of $\approx 3.6\%$ due to the additional optimizer state synchronization. However, this yields performance gains, improving final perplexity by $\approx 2.3\%$ (8.91 vs. 9.12) over DiLoCo.

B.5 Muon Experimental Details

This section provides the full experimental details for the Muon experiments presented. Although a comprehensive theoretical treatment of preconditioned local updates under Newton-Schulz iterations is outside the scope of this work, the FAUST design is inherently compatible with such structures, as the momentum buffer follows standard EMA dynamics regardless of the subsequent preconditioning step. Moreover, the compile-time SP pass is optimizer-agnostic: it operates on the forward/backward computation graph and does not interact with the optimizer's internal state management, so Muon's Newton-Schulz orthogonalization composes with SP identically to Adam's moment updates.

Experimental Details. We utilize the standard PyTorch implementation of Muon with Nesterov momentum enabled and a weight decay of 0.1. Following the recommendations of Liu et al. [2025], we apply the `match_rms_norm` adjustment to learning rates. We adopt the conventional split optimization strategy for Muon: AdamW handles embeddings and layer normalizations, while Muon optimizes all 2D matrices [Jordan et al., 2024]. The momentum parameter for Muon is set to $\beta = 0.9$, while the Adam component retains the $\beta_1 = 0.9$, $\beta_2 = 0.999$ settings used elsewhere. Gradient clipping thresholds are scaled by model size: 1.0 for 16M, 0.5 for 125M, and 0.25 for 360M. For the Local Muon baseline, all optimizer states (Muon momentum; Adam first/second momenta) synchronize every 32 steps. In contrast, FAUST delays state synchronization: the first momentum (for both optimizers) synchronizes every 96 steps ($3\times$ reduction), and Adam's second momentum synchronizes every 192 steps ($6\times$ reduction).

B.6 Experiments on the Flux Vision Model

To demonstrate the universality of FAUST across different modalities and architectures beyond standard decoder-only LLMs, we evaluate its performance on Flux [Sergeev and Balso, 2018], a Rectified Flow Transformer designed for text-to-image generation. This architecture differs significantly from the causal language models evaluated in previous sections, serving as a robust test for the generalizability of our decoupled synchronization approach. Because Flux uses bidirectional attention rather than causal masking, the compile-time SP pass applies the same Ulysses head-splitting strategy but without the causal mask partitioning required for autoregressive models, simplifying the graph rewrite.

Experimental Setup. We utilize the 280M parameter variant of Flux provided by `torchtitan`, training with a global batch size of 256. The inner optimizer is AdamW with $\beta_1 = 0.9, \beta_2 = 0.999$. We compare three settings: (1) DDP, (2) Local Adam with a synchronization period of $K = 32$, and (3) FAUST with a parameter sync period of $K_x = 32$. For FAUST, we decouple the momentum synchronization significantly, setting $K_u = 3K_x$ and $K_v = 6K_x$ (192 steps).

B.7 Throughput at 7B Scale

To assess the practical scalability of our method on state-of-the-art hardware and at large model scales, we measure the training throughput of a 7B parameter model distributed across 8 independent NVIDIA B200 GPUs.

Throughput Analysis. As illustrated during the local update phases, each GPU operates at the peak efficiency of a fully isolated local run, achieving identical tokens-per-second throughput as a single B200 with zero synchronization overhead. Distinct drops in throughput are observed only at the sparse synchronization boundaries ($K_x = 32$, $K_u = 96$, $K_v = 192$), where the system pauses to aggregate model parameters and optimizer states via the tiered AllReduce. The SP All-to-All collectives execute every step but are overlapped with the attention computation on the NVLink fabric, contributing $< 2\%$ overhead that is invisible in the throughput trace.

Crucially, standard DDP incurs this communication penalty at *every single training step*, significantly lowering the average tokens/sec. Even with our current unoptimized "stop-the-world" implementation—which explicitly pauses computation to communicate and does not leverage computation-communication overlap—FAUST significantly increases aggregate throughput by amortizing these costs over long local training windows.

Takeaway: On high-performance B200 hardware, FAUST enables near-linear scaling by keeping workers in a high-throughput local regime for the majority of training. By restricting communication overhead to sparse intervals, it delivers significant wall-clock speedups over DDP, even without low-level implementation optimizations like communication overlap. The compile-time SP pass adds negligible per-step cost ($< 2\%$) since its All-to-All collectives operate entirely within the intra-node NVLink domain.

C Further Algorithmic Details of FAUST

C.1 Extension to FedOpt

Although Cheng and Glasgow [2025] show provable convergence for adaptive inner optimizers in a federated optimization framework, their result rests on the assumption that after a period of local work, the new global model is created by averaging the local client models. In relation to the larger FedOpt literature [Reddi et al., 2021], the scheme chosen by Cheng and Glasgow [2025] resembles that of FedAvg, or where the server optimizer is SGD with the outer learning rate set to one [Reddi et al., 2021]. Naturally, the question of whether alternate server optimizers that have been used in prior works can also be implemented for Local Adam, and thus FAUST, arises.

We argue that indeed FAUST's principles can be effectively applied to any FedOpt method and not just FedAvg. While using an alternate server optimizer does not have proven convergence guarantees as yet, we show in Algorithm 1 that the choice of the SERVEROPT is not constrained from a practical point-of-view. However, the improvements that FAUST provides are related to the local optimization procedure, which is orthogonal to the outer optimizer choice. Choosing the correct,

and most effective, outer optimizer is an open research area [Khaled et al., 2025], and we leave the investigations of the interactions between FAUST and outer optimizers to future work.

Importantly, the compile-time sequence-parallel decomposition introduced in FAUST (Section A.1) does not interact with the outer optimizer at all: the SP All-to-All collectives execute within every forward and backward step to ensure attention correctness, whereas SERVEROPT operates exclusively at K_x-boundary steps on the already-reduced parameter averages $\mathbb{E}_m[x_t^m]$. The SP graph rewrite and the FedOpt outer update thus act on disjoint stages of the training loop and compose without interference.

C.2 Extension to FedOpt

Although Cheng and Glasgow [2025] show provable convergence for adaptive inner optimizers in a federated optimization framework, their result rests on the assumption that after a period of local work, the new global model is created by averaging the local client models. In relation to the larger `FedOpt` literature [Reddi et al., 2021], the scheme chosen by Cheng and Glasgow [2025] resembles that of `FedAvg`, or where the server optimizer is `SGD` with the outer learning rate set to one [Reddi et al., 2021]. Naturally, the question of whether alternate server optimizers than have been used in prior works can also be implemented for `Local Adam` , arises.

We argue that indeed principles can be effectively applied to any `FedOpt` method and not just `FedAvg`. While using an alternate server optimizer does not have proven convergence guarantees as yet, we show in Algorithm 1 that the choice of the `ServerOpt` is not constrained from a practical point-of-view. However, the improvements that `DES-LOC` provide are related to the local optimization procedure, which is orthogonal to the outer optimizer choice. Choosing the correct, and most effective, outer optimizer is an open research area [Khaled et al., 2025], and we leave the investigations of the interactions between `DES-LOC` and outer optimizers to future work.

C.3 Deterministic Optimizer-specific Variants of Algorithm 1

We now present the concrete instantiation of the generic FAUST algorithm (Algorithm 1) for the Adam inner optimizer. Unlike prior formulations that treat the compile-time SP pass as an implicit black box—merely annotating the gradient with "SP-reduced"—we make the intra-step All-to-All collective and its interaction with the inter-step data-parallel synchronization a *first-class citizen* of the algorithm. This explicit coupling is both necessary and novel: it exposes the **collective orthogonality constraint** ($\mathcal{A}_P$ must retire before $\mathbb{E}_m$ fires), enables the SP-tightened clipping bound $\rho_{\text{eff}} = \rho/\sqrt{P}$ for the d_{attn} attention-layer coordinates (2), and formalizes the structural tiering (embedding/norm $\to K_x$-tier, attention projections $\to K_u$-tier, FFN $\to K_v$-tier) that FAUST's implementation deploys via the `DeslocTieredAllReduce` backend.

The $\mathbb{E}_m[\cdot]$ averaging operator denotes the cross-replica AllReduce over the data-parallel group; when ZeRO-1 optimizer-state partitioning is active, this reduces to a reduce-scatter followed by a per-shard update and an all-gather, gated by the corresponding synchronization period (K_x, K_u, or K_v). The All-to-All operator $\mathcal{A}_P$ corresponds to the Ulysses head-splitting collective: for input $Q, K, V \in \mathbb{R}^{B\times N\times S/P\times H}$, it produces $[B, N/P, S, H]$ by scattering along the head dimension and gathering along the sequence dimension; the inverse $\mathcal{A}_P^{-1}$ restores the original layout after SDPA. In our compiler implementation, $\mathcal{A}_P$ is realized as `torch.ops.autosp.all_to_all` with autograd support for the backward pass.

Relationship between Algorithm 2 and the AutoSP compiler. The four phases above map directly to the AutoSP compilation pipeline (commit `5efb24a`): Phase 1 is realized by the five compiler passes SHARDSEQDIM $\to$ SHARDINPUTIDS $\to$ SHARDLABELIDS $\to$ SHARDPOSITIONIDS $\to$ INSERTA2A, which rewrite the FX graph $\mathcal{G}$ into $\mathcal{G}'$ by inserting $4L$ All-to-All nodes (one scatter-heads/gather-seq before each of Q, K, V, plus one scatter-seq/gather-heads after O, per layer). Phase 2 exploits PyTorch's autograd: the registered backward for `autosp::all_to_all` swaps scatter and gather indices, so $\nabla\mathcal{A}_P = \mathcal{A}_P^{-1}$, yielding the FP32-accumulated partial-gradient summation. Phase 3 corresponds to the `_desloc_fence_all_pending` routine, which blocks until all $|\mathcal{H}|$ asynchronous handles complete—enforcing the orthogonality invariant that no DP AllReduce observes a partially-reduced SP gradient. Phase 4 is the standard tiered EMA update with bucket-level AllReduce gating, where each gradient bucket is tagged with its structural tier

(embedding/normalization $\to K_x$, attention projections $\to K_u$, FFN $\to K_v$) as implemented in `DeslocTieredAllReduceTorch`.

Note on ADOPT's update order. ADOPT [Taniguchi et al., 2024] differs from Adam in that the second moment v_t^w is updated *before* the first moment m_t^w, and the preconditioner uses v_{t-1}^w (the previous step) rather than v_t^w. In the FAUST setting this ordering is preserved per-worker: the tiered sync gates on K_v and K_m are independent, so v can synchronize at a different cadence than m without violating ADOPT's convergence requirements. However, the divisibility chain $K_x \mid K_m \mid K_v$ ensures that whenever parameters sync (K_x boundary), both moments have already been averaged, preventing stale-preconditioner drift.

SP–DP collective orthogonality and ZeRO-1 composition. A central invariant of both Algorithms 2 and 3 is the **collective orthogonality constraint**: the intra-step SP collectives ($\mathcal{A}_P$ and $\mathcal{A}_P^{-1}$, Phase 1–2) and the inter-step DP collectives ($\mathbb{E}_m$, Phase 4) must never overlap on the same gradient tensor. This is enforced by the Phase 3 fence: at every K_x-boundary step, WAITALL($\mathcal{H}$) blocks until all $|\mathcal{H}| = L$ asynchronous All-to-All handles have retired, guaranteeing that the gradient $\hat{g}_t^m$ entering Phase 4 is fully sequence-reduced.

When ZeRO-1 optimizer-state partitioning is active, the $\mathbb{E}_m[\cdot]$ operator decomposes into three sub-operations:

$$\mathbb{E}_m[s] \longmapsto \texttt{reduce_scatter}(s) \to \text{per-shard update} \to \texttt{all_gather}(\hat{s}), \tag{15}$$

where the tier gate ($t \bmod K_j = 0$) controls whether `reduce_scatter` fires or the gradient is accumulated locally. In the Neuron_SP implementation, each gradient bucket is tagged with its structural tier (embedding/norm $\to K_x$, attention projections $\to K_u$, FFN $\to K_v$), and the bucket's AllReduce is dispatched only when the period K_j divides t. The SP fence (Phase 3) serializes *before* the first bucket dispatch, so the ZeRO-1 `reduce_scatter` never observes a partially SP-reduced gradient shard.

AutoSP interacts with this decomposition through a critical engine-level routing decision: whereas DeepCompile normally bypasses the standard `allreduce_gradients` path (handling gradient reduction within the compiled graph), AutoSP explicitly re-enables it so that the DP AllReduce runs *outside* the compiled graph on the already SP-reduced gradient. This ensures that the SP $\mathcal{A}_P^{-1}$ and the DP $\mathbb{E}_m$ execute in strictly sequential phases, composing correctly even when the DP reduction is bucket-pipelined.

SP gradient identity and convergence invariance. Under the Ulysses head-splitting strategy realized by AutoSP's INSERTA2A pass, each SP rank p on device m computes a partial gradient $\tilde{g}_t^{m,p}$ from its local sequence chunk. The backward All-to-All—implemented via the registered autograd hook `_all_to_all_backward` which swaps `scatter_idx` and `gather_idx`—produces the aggregate:

$$g_t^m = \sum_{p=0}^{P-1} \tilde{g}_t^{m,p} = \nabla_x \mathcal{L}(x_t^m; \xi_t^m), \tag{16}$$

which is *identical* to the gradient from a non-SP forward-backward pass on the full sequence S. This identity holds exactly for linear layers and attention (where the All-to-All implements a head-dimension repartitioning), and up to floating-point reduction order for LayerNorm parameters (which require an additional AllReduce of partial gradients within the SP group, implemented in the backward hook). Since (16) is independent of P, the optimizer loops in Algorithms 2 and 3 operate on the same g_t^m for any SP degree, and the convergence analyses in Sections D and D.1 apply without modification.

Selective activation checkpointing under SP. AutoSP introduces a compile-time heuristic for activation checkpointing that is SP-aware: the LONGCONTEXTCHECKPOINTING pass modifies PyTorch's `solve_min_cut` partitioner to distinguish between operations whose activation memory scales quadratically with sequence length (SDPA, convolutions) and those that scale linearly (matmul, FFN). Specifically, the heuristic bans recomputation of any operation whose output tensor has a second dimension ≥ 4096, as these are the attention activations that dominate memory under long-context SP training. Matmul and related linear operations are *not* banned, allowing the partitioner to

recompute them during the backward pass at negligible cost. This selective strategy provides $O(L{\cdot}S)$ memory savings compared to full activation storage, while adding only $O(L \cdot d_{\text{model}})$ recomputation cost—a ratio that improves as the sequence length S grows, making it particularly effective in the SP regime where S is large by design.

D Convergence Analysis of FAUST-SGDM (in expectation bounds)

We provide a complete, self-contained convergence analysis of the FAUST system applied to distributed SGDM with a single optimizer state ($N = 1$, momentum). The analysis proceeds in four steps.

Throughout this section we use the following notation. There are M replicas, indexed by $m \in \{1, \dots, M\}$. Each replica m maintains a local copy of the parameters $x_t^m \in \mathbb{R}^d$ and a local momentum buffer $u_t^m \in \mathbb{R}^d$. We denote the SP degree by $P \geq 1$.

Why P does not affect the convergence rate (and where it helps). The compile-time sequence-parallel graph rewrite (AutoSP, Section 2) inserts $4L$ All-to-All collectives that partition the sequence dimension across P ranks within each replica. By the SP gradient identity (16), the gradient entering the optimizer is $g_t^m = \sum_{p=0}^{P-1} \tilde{g}_t^{m,p} = \nabla F_m(x_t^m; \xi_t^m)$, identical to the non-SP case. The stochastic variance bound therefore satisfies

$$\mathbb{E}\left\|g_t^m - \nabla f_m(x_t^m)\right\|^2 \;\leq\; \sigma^2 \tag{17}$$

with the *same* constant σ^2 for any $P \geq 1$, and the entire proof chain below is P-independent.

However, P does improve the *per-coordinate* gradient variance for the $d_{\text{attn}} = 4NH$ attention-layer coordinates. By the SP variance reduction (2), the per-coordinate variance on these coordinates is σ_j^2/P instead of σ_j^2. This motivates the SP-tightened clipping bound $\rho_{\text{eff}} = \rho/\sqrt{P}$ used in Algorithm 2 (Phase 2), which reduces the drift contribution from attention parameters by a factor of $1/\sqrt{P}$—a tighter constant that refines the bound (3) without changing the overall convergence *order*.

The collective orthogonality constraint (Phase 3 fence in Algorithm 2) ensures that the gradient g_t^m is fully sequence-reduced before any data-parallel synchronization ($\mathbb{E}_m$) fires. This is critical: if the fence were absent, the DP AllReduce would average partially-reduced SP gradients across replicas, violating the unbiasedness assumption that undergirds the entire analysis.

The complete algorithm is given below.

To facilitate the analysis, we model each synchronization event as a Bernoulli trial. That is, instead of averaging the parameters x_t^m at every step that satisfies $t \bmod K_x = 0$, we average with probability

$$p_x \;=\; \frac{1}{K_x} \tag{18}$$

at *every* step. The expected number of averaging events in T steps under the probabilistic model is $p_x \cdot T = T/K_x$, which equals the number of averaging events in the deterministic model. The two formulations are therefore statistically equivalent. Likewise, the momentum u_t^m is synchronized with probability $p_u = 1/K_u$, independently of p_x.

Step 1: Averaged dynamics and virtual iterates. Definition of averaged quantities. For each step $t \geq 0$, we define the *cross-replica averages* of the parameters, momentum, and stochastic gradient:

$$x_t \overset{\text{def}}{=} \frac{1}{M}\sum_{m=1}^{M} x_t^m \;=\; \mathbb{E}_m[x_t^m], \tag{19}$$

$$u_t \overset{\text{def}}{=} \frac{1}{M}\sum_{m=1}^{M} u_t^m \;=\; \mathbb{E}_m[u_t^m], \tag{20}$$

$$g_t \overset{\text{def}}{=} \frac{1}{M}\sum_{m=1}^{M} g_t^m \;=\; \mathbb{E}_m[g_t^m]. \tag{21}$$

The notation $\mathbb{E}_m[\cdot]$ denotes the uniform average over the replica index m, *not* an expectation over randomness. Expectation over the stochastic sampling ξ_t^m is written $\mathbb{E}[\cdot]$.

Claim: the averages follow centralized SGDM. We now show that x_t and u_t satisfy the standard (single-machine) SGDM recursion, regardless of whether a synchronization event occurs at step t.

Consider the momentum update on replica m. With probability p_u (sync),

$$u_t^m = \beta\, \mathbb{E}_m[u_{t-1}^m] + (1-\beta)\, g_t^m = \beta\, u_{t-1} + (1-\beta)\, g_t^m,$$

and with probability $1-p_u$ (no sync),

$$u_t^m = \beta\, u_{t-1}^m + (1-\beta)\, g_t^m.$$

In either case, averaging over m gives

$$\begin{aligned}
u_t &= \mathbb{E}_m[u_t^m] \\
&= p_u \cdot \big(\beta\, u_{t-1} + (1-\beta)\, g_t\big) + (1-p_u) \cdot \big(\beta\, \mathbb{E}_m[u_{t-1}^m] + (1-\beta)\, g_t\big) \\
&= p_u \cdot \big(\beta\, u_{t-1} + (1-\beta)\, g_t\big) + (1-p_u) \cdot \big(\beta\, u_{t-1} + (1-\beta)\, g_t\big) \\
&= \big(p_u + (1-p_u)\big) \cdot \big(\beta\, u_{t-1} + (1-\beta)\, g_t\big) \\
&= 1 \cdot \big(\beta\, u_{t-1} + (1-\beta)\, g_t\big) \\
&= \beta\, u_{t-1} + (1-\beta)\, g_t. \qquad (22)
\end{aligned}$$

An identical argument for the parameter update yields

$$x_{t+1} = x_t - \eta\, u_t. \qquad (23)$$

Equations (22) and (23) are precisely the centralized SGDM updates.

Definition of virtual iterates. Set $x_{-1} \overset{\text{def}}{=} x_0$, and for each $t \geq 0$ define

$$z_t \overset{\text{def}}{=} \frac{1}{1-\beta}\, x_t - \frac{\beta}{1-\beta}\, x_{t-1}. \qquad (24)$$

At $t=0$ this gives $z_0 = \frac{1}{1-\beta}\, x_0 - \frac{\beta}{1-\beta}\, x_0 = x_0$.

Key property: virtual iterates follow averaged gradients. For any $t \geq 0$, we compute $z_{t+1} - z_t$ step by step. First, by the definition (24),

$$\begin{aligned}
z_{t+1} - z_t &= \frac{1}{1-\beta}\, x_{t+1} - \frac{\beta}{1-\beta}\, x_t - \frac{1}{1-\beta}\, x_t + \frac{\beta}{1-\beta}\, x_{t-1} \\
&= \frac{1}{1-\beta}\, (x_{t+1} - x_t) - \frac{\beta}{1-\beta}\, (x_t - x_{t-1}). \qquad (25)
\end{aligned}$$

Now substitute the parameter recursion (23):

$$x_{t+1} - x_t = -\eta\, u_t, \qquad (26)$$

$$x_t - x_{t-1} = -\eta\, u_{t-1}. \qquad (27)$$

Inserting (26) and (27) into (25):

$$\begin{aligned}
z_{t+1} - z_t &= \frac{1}{1-\beta}\, (-\eta\, u_t) - \frac{\beta}{1-\beta}\, (-\eta\, u_{t-1}) \\
&= -\frac{\eta}{1-\beta}\, u_t + \frac{\eta\, \beta}{1-\beta}\, u_{t-1} \\
&= -\frac{\eta}{1-\beta}\, \big(u_t - \beta\, u_{t-1}\big). \qquad (28)
\end{aligned}$$

Finally, by the momentum recursion (22),

$$u_t - \beta\, u_{t-1} = \big(\beta\, u_{t-1} + (1-\beta)\, g_t\big) - \beta\, u_{t-1} = (1-\beta)\, g_t. \qquad (29)$$

Substituting (29) into (28):

$$\begin{aligned}
z_{t+1} - z_t &= -\frac{\eta}{1-\beta}\, (1-\beta)\, g_t \\
&= -\eta\, g_t. \qquad (30)
\end{aligned}$$

This is the key identity: the virtual iterates z_t evolve by a simple gradient step $-\eta\, g_t$, with no momentum contamination.

Step 2: Smoothness descent on virtual iterates. By Assumption 3.1, f is L-smooth, so for any $a, b \in \mathbb{R}^d$,

$$f(b) \leq f(a) + \langle \nabla f(a),\, b - a \rangle + \frac{L}{2} \|b - a\|^2. \tag{31}$$

Applying (31) with $a = z_t$ and $b = z_{t+1}$:

$$f(z_{t+1}) \leq f(z_t) + \langle \nabla f(z_t),\, z_{t+1} - z_t \rangle + \frac{L}{2} \|z_{t+1} - z_t\|^2. \tag{32}$$

We now split the inner product by adding and subtracting $\nabla f(x_t)$. Write $\nabla f(z_t) = \nabla f(x_t) + \big(\nabla f(z_t) - \nabla f(x_t)\big)$. Then

$$\langle \nabla f(z_t),\, z_{t+1} - z_t \rangle = \langle \nabla f(x_t),\, z_{t+1} - z_t \rangle + \langle \nabla f(z_t) - \nabla f(x_t),\, z_{t+1} - z_t \rangle. \tag{33}$$

Substituting (33) into (32):

$$f(z_{t+1}) \leq f(z_t) + \underbrace{\langle \nabla f(x_t),\, z_{t+1} - z_t \rangle}_{I} + \underbrace{\langle \nabla f(z_t) - \nabla f(x_t),\, z_{t+1} - z_t \rangle}_{II} + \underbrace{\frac{L}{2} \|z_{t+1} - z_t\|^2}_{III}. \tag{34}$$

We bound terms I, II, III separately below.

Step 3a: Bounding term I. Recall from (30) that $z_{t+1} - z_t = -\eta\, g_t$. We substitute and then use the identity $g_t = \mathbb{E}_m[g_t^m]$:

$$\begin{aligned} \langle \nabla f(x_t),\, z_{t+1} - z_t \rangle &= \langle \nabla f(x_t),\, -\eta\, g_t \rangle \\ &= -\eta\, \langle \nabla f(x_t),\, g_t \rangle \\ &= -\eta \left\langle \nabla f(x_t),\, \frac{1}{M} \sum_{m=1}^{M} g_t^m \right\rangle. \end{aligned} \tag{35}$$

Taking the expectation over the stochastic sampling and using unbiasedness (Assumption 3.2):

$$\begin{aligned} \mathbb{E}\left[\left\langle \nabla f(x_t),\, \frac{1}{M} \sum_{m=1}^{M} g_t^m \right\rangle \right] &= \left\langle \nabla f(x_t),\, \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}[g_t^m] \right\rangle \\ &= \left\langle \nabla f(x_t),\, \frac{1}{M} \sum_{m=1}^{M} \nabla f_m(x_t^m) \right\rangle. \end{aligned} \tag{36}$$

Hence

$$\mathbb{E}[I] = -\eta\, \mathbb{E}\left[\left\langle \nabla f(x_t),\, \frac{1}{M} \sum_{m=1}^{M} \nabla f_m(x_t^m) \right\rangle \right]. \tag{37}$$

We now apply the *polarization identity*: for any vectors $a, b \in \mathbb{R}^d$,

$$\langle a, b \rangle = \frac{1}{2}\|a\|^2 + \frac{1}{2}\|b\|^2 - \frac{1}{2}\|a - b\|^2. \tag{38}$$

To use (38) in the form that gives us negative squared norms (which we need for convergence), we rewrite it as

$$-\langle a, b \rangle = -\frac{1}{2}\|a\|^2 - \frac{1}{2}\|b\|^2 + \frac{1}{2}\|a - b\|^2. \tag{39}$$

Setting $a = \nabla f(x_t)$ and $b = \frac{1}{M} \sum_{m=1}^{M} \nabla f_m(x_t^m)$ in (39):

$$\begin{aligned} &- \left\langle \nabla f(x_t),\, \frac{1}{M} \sum_{m=1}^{M} \nabla f_m(x_t^m) \right\rangle \\ &\quad = -\frac{1}{2} \|\nabla f(x_t)\|^2 - \frac{1}{2} \left\| \frac{1}{M} \sum_{m=1}^{M} \nabla f_m(x_t^m) \right\|^2 + \frac{1}{2} \left\| \nabla f(x_t) - \frac{1}{M} \sum_{m=1}^{M} \nabla f_m(x_t^m) \right\|^2. \end{aligned} \tag{40}$$

Multiplying (40) by η and taking expectation:

$$\mathbb{E}[I] = -\frac{\eta}{2}\,\mathbb{E}\|\nabla f(x_t)\|^2 - \frac{\eta}{2}\,\mathbb{E}\left\|\frac{1}{M}\sum_{m=1}^{M}\nabla f_m(x_t^m)\right\|^2 + \frac{\eta}{2}\,\mathbb{E}\left\|\nabla f(x_t) - \frac{1}{M}\sum_{m=1}^{M}\nabla f_m(x_t^m)\right\|^2. \tag{41}$$

It remains to bound the third term. We use the *variance decomposition*: for any random vectors $X_1, \ldots, X_M$,

$$\left\|\frac{1}{M}\sum_{m=1}^{M}(X_m - \bar{X})\right\|^2 \le \frac{1}{M}\sum_{m=1}^{M}\|X_m - \bar{X}\|^2 \quad \text{where} \quad \bar{X} = \frac{1}{M}\sum_{m=1}^{M}X_m. \tag{42}$$

More precisely, since $\nabla f(x_t) = \frac{1}{M}\sum_{m=1}^{M}\nabla f_m(x_t)$, we write

$$\begin{aligned}
&\left\|\nabla f(x_t) - \frac{1}{M}\sum_{m=1}^{M}\nabla f_m(x_t^m)\right\|^2 \\
&= \left\|\frac{1}{M}\sum_{m=1}^{M}\left(\nabla f_m(x_t) - \nabla f_m(x_t^m)\right)\right\|^2 \\
&\le \frac{1}{M}\sum_{m=1}^{M}\|\nabla f_m(x_t) - \nabla f_m(x_t^m)\|^2 \qquad \text{(Jensen's inequality).}
\end{aligned} \tag{43}$$

By L-smoothness (Assumption 3.1), $\|\nabla f_m(x_t) - \nabla f_m(x_t^m)\| \le L\,\|x_t - x_t^m\|$, so

$$\|\nabla f_m(x_t) - \nabla f_m(x_t^m)\|^2 \le L^2\,\|x_t - x_t^m\|^2. \tag{44}$$

Combining (41), (43), and (44):

$$\mathbb{E}[I] \le -\frac{\eta}{2}\,\mathbb{E}\|\nabla f(x_t)\|^2 - \frac{\eta}{2}\,\mathbb{E}\left\|\frac{1}{M}\sum_{m=1}^{M}\nabla f_m(x_t^m)\right\|^2 + \frac{\eta L^2}{2M}\sum_{m=1}^{M}\underbrace{\mathbb{E}\|x_t - x_t^m\|^2}_{\text{bounded in Lemma D.2}}. \tag{45}$$

Step 3b: Bounding term II. Recall $z_{t+1} - z_t = -\eta\, g_t$ and $g_t = \frac{1}{M}\sum_{m=1}^{M} g_t^m$. We begin:

$$\begin{aligned}
\mathbb{E}[II] &= \mathbb{E}\big[\langle \nabla f(z_t) - \nabla f(x_t),\, z_{t+1} - z_t\rangle\big] \\
&= -\eta\,\mathbb{E}\big[\langle \nabla f(z_t) - \nabla f(x_t),\, g_t\rangle\big] \\
&= -\eta\,\mathbb{E}\left[\left\langle \nabla f(z_t) - \nabla f(x_t),\, \frac{1}{M}\sum_{m=1}^{M}\nabla f_m(x_t^m)\right\rangle\right].
\end{aligned} \tag{46}$$

By Young's inequality, for any $\rho > 0$ and vectors a, b,

$$|\langle a, b\rangle| \le \frac{\rho}{2}\,\|a\|^2 + \frac{1}{2\rho}\,\|b\|^2. \tag{47}$$

Applying (47) with $a = \nabla f(z_t) - \nabla f(x_t)$ and $b = \frac{1}{M}\sum_{m=1}^{M}\nabla f_m(x_t^m)$:

$$\mathbb{E}[II] \le \frac{\eta\,\rho}{2}\,\mathbb{E}\|\nabla f(z_t) - \nabla f(x_t)\|^2 + \frac{\eta}{2\rho}\,\mathbb{E}\left\|\frac{1}{M}\sum_{m=1}^{M}\nabla f_m(x_t^m)\right\|^2. \tag{48}$$

By L-smoothness, $\|\nabla f(z_t) - \nabla f(x_t)\| \le L\,\|z_t - x_t\|$, hence $\|\nabla f(z_t) - \nabla f(x_t)\|^2 \le L^2\,\|z_t - x_t\|^2$. Therefore:

$$\mathbb{E}[II] \le \frac{\eta\,\rho\,L^2}{2}\underbrace{\mathbb{E}\|z_t - x_t\|^2}_{\text{bounded in Lemma D.1}} + \frac{\eta}{2\rho}\,\mathbb{E}\left\|\frac{1}{M}\sum_{m=1}^{M}\nabla f_m(x_t^m)\right\|^2. \tag{49}$$

Step 3c: Bounding term III. Using (30):

$$\begin{aligned}\frac{L}{2}\left\|z_{t+1}-z_t\right\|^2 &= \frac{L}{2}\left\|-\eta\, g_t\right\|^2 \\ &= \frac{\eta^2 L}{2}\left\|g_t\right\|^2 \\ &= \frac{\eta^2 L}{2}\left\|\frac{1}{M}\sum_{m=1}^{M} g_t^m\right\|^2 . \end{aligned} \tag{50}$$

We add and subtract $\nabla f_m(x_t^m)$ inside the sum. Define $\varepsilon_t^m \overset{\text{def}}{=} g_t^m - \nabla f_m(x_t^m)$ (the noise term). Then

$$g_t^m = \nabla f_m(x_t^m) + \varepsilon_t^m, \tag{51}$$

and

$$\left\|\frac{1}{M}\sum_{m=1}^{M} g_t^m\right\|^2 = \left\|\frac{1}{M}\sum_{m=1}^{M} \varepsilon_t^m + \frac{1}{M}\sum_{m=1}^{M} \nabla f_m(x_t^m)\right\|^2 . \tag{52}$$

By the identity $\|a+b\|^2 = \|a\|^2 + \|b\|^2 + 2\langle a, b\rangle$, and noting that $\mathbb{E}[\varepsilon_t^m] = 0$ by Assumption 3.2 (so the cross-term vanishes under expectation):

$$\mathbb{E}\left\|\frac{1}{M}\sum_{m=1}^{M} g_t^m\right\|^2 = \mathbb{E}\left\|\frac{1}{M}\sum_{m=1}^{M} \varepsilon_t^m\right\|^2 + \mathbb{E}\left\|\frac{1}{M}\sum_{m=1}^{M} \nabla f_m(x_t^m)\right\|^2 . \tag{53}$$

For the noise term, since the ε_t^m are independent across replicas with zero mean:

$$\begin{aligned}\mathbb{E}\left\|\frac{1}{M}\sum_{m=1}^{M} \varepsilon_t^m\right\|^2 &= \frac{1}{M^2}\sum_{m=1}^{M} \mathbb{E}\|\varepsilon_t^m\|^2 \\ &= \frac{1}{M^2}\sum_{m=1}^{M} \mathbb{E}\|g_t^m - \nabla f_m(x_t^m)\|^2 \\ &\le \frac{1}{M^2}\sum_{m=1}^{M} \sigma^2 \qquad \text{(Assumption 3.2)} \\ &= \frac{\sigma^2}{M}. \end{aligned} \tag{54}$$

Substituting (53) and (54) into (50):

$$\mathbb{E}[III] \le \frac{\eta^2 L}{2M}\,\sigma^2 + \frac{\eta^2 L}{2}\,\mathbb{E}\left\|\frac{1}{M}\sum_{m=1}^{M} \nabla f_m(x_t^m)\right\|^2 . \tag{55}$$

Step 3 combined: assembling the one-step bound. Substituting (45), (49), and (55) into (34) and taking expectation:

$$\begin{aligned}&\mathbb{E}[f(z_{t+1})] - \mathbb{E}[f(z_t)] \\ &\le \underbrace{-\frac{\eta}{2}\,\mathbb{E}\|\nabla f(x_t)\|^2}_{\text{from } I} \underbrace{-\frac{\eta}{2}\,\mathbb{E}\left\|\frac{1}{M}\sum_m \nabla f_m(x_t^m)\right\|^2}_{\text{from } I} \underbrace{+\frac{\eta\, L^2}{2M}\sum_m \mathbb{E}\|x_t - x_t^m\|^2}_{\text{from } I} \\ &\quad \underbrace{+\frac{\eta\,\rho\, L^2}{2}\,\mathbb{E}\|z_t - x_t\|^2}_{\text{from } II} \underbrace{+\frac{\eta}{2\rho}\,\mathbb{E}\left\|\frac{1}{M}\sum_m \nabla f_m(x_t^m)\right\|^2}_{\text{from } II} \\ &\quad \underbrace{+\frac{\eta^2 L}{2M}\,\sigma^2}_{\text{from } III} \underbrace{+\frac{\eta^2 L}{2}\,\mathbb{E}\left\|\frac{1}{M}\sum_m \nabla f_m(x_t^m)\right\|^2}_{\text{from } III} . \end{aligned} \tag{56}$$

Collect the three terms involving $\mathbb{E}\left\|\frac{1}{M}\sum_m \nabla f_m(x_t^m)\right\|^2$: their combined coefficient is

$$-\frac{\eta}{2}+\frac{\eta}{2\rho}+\frac{\eta^2 L}{2}=-\frac{\eta}{2}\left(1-\frac{1}{\rho}-\eta L\right). \tag{57}$$

Therefore (56) simplifies to:

$$\begin{aligned}
&\mathbb{E}[f(z_{t+1})]-\mathbb{E}[f(z_t)]\\
&\quad\le -\frac{\eta}{2}\mathbb{E}\|\nabla f(x_t)\|^2-\frac{\eta}{2}\left(1-\frac{1}{\rho}-\eta L\right)\mathbb{E}\left\|\frac{1}{M}\sum_m \nabla f_m(x_t^m)\right\|^2\\
&\qquad+\frac{\eta L^2}{2M}\sum_m \mathbb{E}\|x_t-x_t^m\|^2+\frac{\eta\rho L^2}{2}\mathbb{E}\|z_t-x_t\|^2+\frac{\eta^2 L}{2M}\sigma^2.
\end{aligned} \tag{58}$$

Step 4: Telescoping and applying the lemmas. Sum (58) from $t=0$ to $T-1$ and divide by T:

$$\begin{aligned}
\frac{\mathbb{E}[f(z_T)]-f(z_0)}{T}&\le -\frac{\eta}{2T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(x_t)\|^2-\frac{\eta}{2}\left(1-\frac{1}{\rho}-\eta L\right)\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\frac{1}{M}\sum_m \nabla f_m(x_t^m)\right\|^2\\
&\quad+\frac{\eta\rho L^2}{2}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|z_t-x_t\|^2+\frac{\eta L^2}{2M}\frac{1}{TM}\sum_{t=0}^{T-1}\sum_m \mathbb{E}\|x_t-x_t^m\|^2+\frac{\eta^2 L}{2M}\sigma^2.
\end{aligned} \tag{59}$$

We now substitute the bounds from Lemma D.1 and Lemma D.2 (proved in Section D.2).

Lemma D.1 gives:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|z_t-x_t\|^2\le\frac{\eta^2\beta^2}{(1-\beta)^2 M}\sigma^2+\frac{\eta^2\beta^2}{(1-\beta)^2}\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\frac{1}{M}\sum_m \nabla f_m(x_t^m)\right\|^2. \tag{60}$$

Lemma D.2 gives (provided $24\eta^2 L^2\psi\le 1$):

$$\frac{1}{TM}\sum_{t=0}^{T-1}\sum_m \mathbb{E}\|x_t-x_t^m\|^2\le 12\eta^2(B^2-1)\psi\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(x_t)\|^2+4\eta^2\psi(\sigma^2+3G^2), \tag{61}$$

where the *drift coefficient* ψ is defined as

$$\psi\overset{\text{def}}{=}\frac{4(1-p_x)}{p_x^2}\cdot\frac{(1-\beta)(1-p_u)}{1-(1-p_u)\beta}. \tag{62}$$

Substituting (60) and (61) into (59), and collecting terms proportional to $\frac{1}{T}\sum_t\mathbb{E}\|\nabla f(x_t)\|^2$ and $\frac{1}{T}\sum_t\mathbb{E}\|\frac{1}{M}\sum_m\nabla f_m(x_t^m)\|^2$:

Coefficient of $\frac{1}{T}\sum_t\mathbb{E}\|\nabla f(x_t)\|^2$:

$$-\frac{\eta}{2}+\frac{\eta L^2}{2}\cdot 12\eta^2(B^2-1)\psi=-\frac{\eta}{2}\big(1-12\eta^2 L^2(B^2-1)\psi\big). \tag{63}$$

Coefficient of $\frac{1}{T}\sum_t\mathbb{E}\|\frac{1}{M}\sum_m\nabla f_m(x_t^m)\|^2$:

$$-\frac{\eta}{2}\left(1-\frac{1}{\rho}-\eta L\right)+\frac{\eta\rho L^2}{2}\cdot\frac{\eta^2\beta^2}{(1-\beta)^2}=-\frac{\eta}{2}\left(1-\frac{1}{\rho}-\eta L-\frac{\eta^2\beta^2\rho L^2}{(1-\beta)^2}\right). \tag{64}$$

Residual terms (independent of the sums):

$$\frac{\eta^2 L}{2M}\sigma^2+\frac{\eta^3\rho L^2\beta^2}{2(1-\beta)^2 M}\sigma^2+2\eta^3 L^2\psi(\sigma^2+3G^2). \tag{65}$$

Therefore (59) becomes:

$$\frac{\mathbb{E}[f(z_T)] - f(z_0)}{T} \le -\frac{\eta}{2}\big(1 - 12\eta^2 L^2 (B^2-1)\,\psi\big) \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\|\nabla f(x_t)\|^2 - \frac{\eta}{2}\left(1 - \frac{1}{\rho} - \eta L - \frac{\eta^2 \beta^2 \rho L^2}{(1-\beta)^2}\right) \cdot \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left\|\frac{1}{M}\sum_m \nabla f_m(x_t^m)\right\|^2 + \frac{\eta^2 L}{2M}\sigma^2 + \frac{\eta^3 \rho L^2 \beta^2}{2(1-\beta)^2 M}\sigma^2 + 2\eta^3 L^2 \psi(\sigma^2 + 3G^2). \tag{66}$$

Choosing ρ and η. Set $\rho = 2$. We need the coefficients of both squared-gradient terms to be non-positive. This requires three simultaneous conditions on η:

$$\begin{aligned} &\text{(C1)} \quad 12\eta^2 L^2 (B^2-1)\,\psi \le \tfrac{1}{2}, && \text{(first term} \ge -\tfrac{\eta}{4}\|\nabla f\|^2\text{)} \\ &\text{(C2)} \quad \eta L + \frac{2\eta^2 \beta^2 L^2}{(1-\beta)^2} \le \tfrac{1}{2}, && \text{(second term non-positive)} \\ &\text{(C3)} \quad 12\eta^2 L^2 \psi \le \tfrac{1}{2}. && \text{(from Lemma D.2 prerequisite)} \end{aligned} \tag{67}$$

One can verify that the step-size

$$\eta_0 \overset{\text{def}}{=} \frac{1}{4L}\min\left(1-\beta,\ \frac{1}{6\sqrt{\psi\,\max(1,\,B^2-1)}}\right) \tag{68}$$

satisfies all three conditions. Indeed:

- (C1): $\eta_0 \le \frac{1}{4L} \cdot \frac{1}{6\sqrt{\psi(B^2-1)}}$ implies $\eta_0^2 \le \frac{1}{144\,L^2\,\psi\,(B^2-1)}$, hence $12\eta_0^2 L^2 (B^2-1)\,\psi \le 12/144 = 1/12 \le 1/2$.
- (C3): $\eta_0 \le \frac{1}{4L} \cdot \frac{1}{6\sqrt{\psi}}$ implies $12\eta_0^2 L^2 \psi \le 12/(144) = 1/12 \le 1/2$.
- (C2): $\eta_0 \le \frac{1-\beta}{4L}$ implies $\eta_0 L \le (1-\beta)/4$ and $\frac{2\eta_0^2\beta^2 L^2}{(1-\beta)^2} \le \frac{2\beta^2}{16} \le 1/8$, so $\eta_0 L + \frac{2\eta_0^2\beta^2 L^2}{(1-\beta)^2} \le 1/4 + 1/8 < 1/2$.

With any $\eta \le \eta_0$, the second sum in (66) is non-positive and can be dropped. The first coefficient satisfies $1 - 12\eta^2 L^2 (B^2-1)\,\psi \ge 1/2$, giving:

$$\frac{\mathbb{E}[f(z_T)] - f(z_0)}{T} \le -\frac{\eta}{4}\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(x_t)\|^2 + \frac{\eta^2 L}{2M}\sigma^2 + \frac{\eta^3 \rho L^2 \beta^2}{2(1-\beta)^2 M}\sigma^2 + 2\eta^3 L^2 \psi(\sigma^2 + 3G^2). \tag{69}$$

Rearranging for the convergence rate. Since $z_0 = x_0$ and $f(z_T) \ge f^*$, we have $\mathbb{E}[f(z_T)] - f(z_0) \ge f^* - f(x_0)$, i.e., $f(x_0) - f^* \ge f(z_0) - \mathbb{E}[f(z_T)]$. Rearranging (69):

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(x_t)\|^2 \le \frac{4(f(x_0)-f^*)}{\eta T} + \frac{2\eta L}{M}\sigma^2 + \frac{4\eta^2 L^2 \beta^2}{(1-\beta)^2 M}\sigma^2 + 8\eta^2 L^2 \psi(\sigma^2 + 3G^2). \tag{70}$$

Final rate with $\eta = \min(\eta_0, T^{-1/2})$. Setting $\eta = \min(\eta_0, 1/\sqrt{T})$, two cases arise.

Case 1: $\eta_0 \le 1/\sqrt{T}$. Then $\eta = \eta_0$ and $1/(\eta T) = 1/(\eta_0 T)$. Each η^2 term is $\mathcal{O}(\eta_0^2) = \mathcal{O}(1/T)$ (since $\eta_0 \le 1/\sqrt{T}$ in this case).

Case 2: $\eta_0 > 1/\sqrt{T}$. Then $\eta = 1/\sqrt{T}$ and $1/(\eta T) = 1/\sqrt{T}$. Each η^2 term is $1/T$.

In either case, collecting the $\mathcal{O}(1/\sqrt{T})$ and $\mathcal{O}(1/T)$ contributions:

$$\begin{aligned}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(x_t)\|^2 &\le \frac{4}{\sqrt{T}}\left(f(x_0)-f^*+\frac{L\,\sigma^2}{2M}\right)+\frac{4(f(x_0)-f^*)}{\eta_0\,T}\\ &\quad+\frac{4L^2\,\beta^2\,\sigma^2}{(1-\beta)^2\,M\,T}+\frac{8L^2\,\psi\,(\sigma^2+3G^2)}{T}\\ &=\frac{4}{\sqrt{T}}\left(f(x_0)-f^*+\frac{L\,\sigma^2}{2M}\right)+\mathcal{O}\!\left(\frac{1+\psi}{T}\right). \end{aligned} \tag{71}$$

This completes the proof of Theorem 3.1.

Remark (SP invariance of the convergence rate). The drift coefficient ψ defined in (62) depends *only* on the synchronization probabilities $p_x = 1/K_x$, $p_u = 1/K_u$ and the momentum decay β. It is *independent* of the SP degree P. This is because:

1. The gradient g_t^m entering the optimizer is already sequence-reduced by the backward All-to-All (Section C). By the SP gradient identity (16), it satisfies $\mathbb{E}[g_t^m] = \nabla f_m(x_t^m)$ and $\mathbb{E}\|g_t^m - \nabla f_m(x_t^m)\|^2 \le \sigma^2$ with the *same* constants as in the non-SP case—this is precisely the variance invariance (17).
2. The synchronization operator $\mathbb{E}_m[\cdot]$ averages over the M data-parallel replicas, not over the P SP ranks within a replica. The SP ranks collaborate to produce a single gradient per replica; the desynced averaging then operates on these replica-level gradients. The collective orthogonality constraint (Phase 3 fence in Algorithm 2) guarantees that this separation holds at every step: no DP AllReduce fires until all SP A2A handles have retired.
3. Consequently, every term in the proof chain (45)–(71) is unaffected by P: the variance bound σ^2/M, the drift bound ψ, and the step-size restriction η_0 are all P-independent.

The SP decomposition affects only the *wall-clock cost per step* (Section F.1), not the optimization trajectory or convergence rate.

Corollary (SP-tightened constant for attention parameters). While the convergence *order* is P-independent, the *constant* inside the bound can be refined for the d_{attn} attention-layer coordinates. By the SP variance reduction (2) and the SP-tightened clipping bound $\rho_{\text{eff}} = \rho/\sqrt{P}$ (Algorithm 2, Phase 2), the drift bounds (3)–(4) yield a refined per-coordinate contribution:

$$\psi_{\text{attn}} \;\le\; \frac{\psi}{P}, \tag{72}$$

where ψ_{attn} is the drift coefficient restricted to the attention-layer coordinates. Since these coordinates constitute a fraction d_{attn}/d of the total, the overall drift decomposes as $\psi_{\text{eff}} = (1 - d_{\text{attn}}/d)\,\psi + (d_{\text{attn}}/d)\,\psi/P$, yielding a tighter constant for the final rate (71) without changing its order. At $P = 2$ (our default SP degree), attention-layer drift is halved; at $P = 4$, it is quartered.

Remark (tiered gradient buckets and ψ). In the FAUST implementation, the uniform periods (K_x, K_u) in the SGDM analysis correspond to the *fastest* tier's periods. The slower tiers (u-tier at $K_u = 3K_x$, v-tier at $K_v = 6K_x$) contribute additional higher-order terms that refine the constant inside the $\mathcal{O}((1+\psi)/T)$ term but do not change its order. The key structural insight is that $\psi = \mathcal{O}(1/p_x^2)$: parameter synchronization frequency dominates the bound, while momentum synchronization enters only linearly.

Remark (role of the collective fence in the proof). The entire proof chain assumes that $g_t^m = \nabla F_m(x_t^m;\, \xi_t^m)$ is an unbiased estimator of $\nabla f_m(x_t^m)$. This holds only if the SP backward All-to-All has fully completed before g_t^m is used. In the absence of the Phase 3 fence, a race condition arises: the DP AllReduce $\mathbb{E}_m[\cdot]$ could average *partially-reduced* gradients $\hat{g}_t^m$ where some SP ranks have completed their All-to-All but others have not. In this scenario, $\hat{g}_t^m \neq g_t^m$ and the unbiasedness assumption $\mathbb{E}[\hat{g}_t^m] = \nabla f_m(x_t^m)$ is violated, invalidating Step 3a (the variance decomposition (42)) and propagating through the entire bound. The fence guarantees that this never occurs: it is not merely an engineering convenience but a *necessary condition for correctness* of the convergence proof.

D.1 Extension to Adam optimizer

We now extend the SGDM analysis of the previous section to the Adam optimizer, which maintains *two* optimizer states: a first moment u_t^m and a second moment v_t^m, each with its own decay factor (β_1 and β_2 respectively) and its own synchronization probability ($p_u = 1/K_u$ and $p_v = 1/K_v$). The key new difficulty is that the parameter update involves a *preconditioned* step $d_t^m = \eta\, \Gamma_t^m\, u_t^m$, where the preconditioning matrix Γ_t^m depends on the local second moment and therefore varies across replicas. This means the virtual-iterate trick of Section D no longer yields a clean $z_{t+1} - z_t = -\eta\, g_t$ identity; instead, additional error terms U_t and V_t appear, which must be controlled.

The complete algorithm is given in Algorithm 5 below. Instead of bounded heterogeneity (Assumption 3.3), we use a stronger condition:

Assumption D.1 (Bounded gradient)**.** *For any iterate $t \geq 0$ and replica m, the local stochastic gradient is bounded:*

$$\|g_t^m\|_2 \;\leq\; G. \tag{73}$$

This condition provides uniform upper bounds on gradients and momenta and is standard in the analysis of adaptive optimizers [Chen et al., 2019, Défossez et al., 2022, Zhang et al., 2023]. As in the SGDM analysis, the SP degree P does not appear: the gradient g_t^m is already sequence-reduced and satisfies Assumption D.1 with the same constant G.

Notation for the Adam analysis. We define the following quantities that will be used throughout this subsection.

Preconditioning matrix. For each replica m at step t, define the diagonal preconditioning matrix

$$\Gamma_t^m \;\stackrel{\text{def}}{=}\; \operatorname{diag}^{-1/2}\!\big(\tilde{v}_t^m + \lambda^2\big), \qquad \text{i.e.,}\quad [\Gamma_t^m]_{ii} = \frac{1}{\sqrt{[\tilde{v}_t^m]_i + \lambda^2}}. \tag{74}$$

Since $\tilde{v}_t^m \geq 0$ coordinate-wise and $\lambda > 0$, we have $0 < [\Gamma_t^m]_{ii} \leq 1/\lambda$ for all i, hence

$$\|\Gamma_t^m\| \;\leq\; \frac{1}{\lambda}, \tag{75}$$

where $\|\cdot\|$ denotes the spectral norm (which equals the largest diagonal entry for a positive diagonal matrix).

Cross-replica averages. As in the SGDM analysis, define

$$\begin{aligned} x_t &= \mathbb{E}_m[x_t^m], & u_t &= \mathbb{E}_m[u_t^m], & v_t &= \mathbb{E}_m[v_t^m], \\ \tilde{v}_t &= \mathbb{E}_m[\tilde{v}_t^m], & \Gamma_t &= \mathbb{E}_m[\Gamma_t^m], & g_t &= \mathbb{E}_m[g_t^m]. \end{aligned} \tag{76}$$

Condition number. By Assumption D.1, $\|g_t^m\| \leq G$, so $[v_t^m]_i \leq G^2$ and $[\tilde{v}_t^m]_i \leq G^2$. Hence the minimum eigenvalue of Γ_t (averaging over replicas) satisfies

$$\|\Gamma_t\|_{\min} = \min_i [\Gamma_t]_{ii} \geq \frac{1}{\sqrt{G^2 + \lambda^2}} \;\stackrel{\text{def}}{=}\; \frac{1}{C_0}. \tag{77}$$

The ratio $C_0/\lambda = \sqrt{G^2 + \lambda^2}/\lambda$ plays the role of a condition number for the preconditioner.

Pseudo-gradient. Define the "de-biased" local gradient

$$\widetilde{g}_t^m \;\stackrel{\text{def}}{=}\; \frac{u_t^m - \beta_1\, u_{t-1}^m}{1 - \beta_1}. \tag{78}$$

By the momentum recursion $u_t^m = \beta_1\, u_{t-1}^m + (1-\beta_1)\, g_t^m$ (in the non-sync case), we have $\widetilde{g}_t^m = g_t^m$ when no sync occurs. In general,

$$\mathbb{E}_m[\widetilde{g}_t^m] = \frac{u_t - \beta_1\, u_{t-1}}{1 - \beta_1} = g_t, \tag{79}$$

where the last step uses the averaged momentum recursion $u_t = \beta_1\, u_{t-1} + (1 - \beta_1)\, g_t$ (derived identically to equation (22) in the SGDM proof).

Step 1: Averaged dynamics and virtual iterates. The averaged first momentum follows the same centralized recursion as before:

$$u_t = \beta_1\, u_{t-1} + (1-\beta_1)\, g_t. \tag{80}$$

The averaged parameter update, however, now involves the preconditioner:

$$x_{t+1} = x_t - d_t, \qquad \text{where} \quad d_t \overset{\text{def}}{=} \eta\, \mathbb{E}_m[\Gamma_t^m\, u_t^m]. \tag{81}$$

Note that $d_t \neq \eta\, \Gamma_t\, u_t$ in general, because Γ_t^m and u_t^m are correlated (both depend on the local gradient history of replica m). This is the fundamental complication relative to SGDM.

Virtual iterates. Define z_t exactly as in (24):

$$z_t = \frac{1}{1-\beta_1}\, x_t - \frac{\beta_1}{1-\beta_1}\, x_{t-1}, \qquad z_0 = x_0. \tag{82}$$

Computing $z_{t+1} - z_t$ step by step. From the definition (82):

$$z_{t+1} - z_t = \frac{1}{1-\beta_1}\,(x_{t+1} - x_t) - \frac{\beta_1}{1-\beta_1}\,(x_t - x_{t-1}). \tag{83}$$

Using $x_{t+1} - x_t = -d_t = -\eta\, \mathbb{E}_m[\Gamma_t^m\, u_t^m]$ and $x_t - x_{t-1} = -d_{t-1} = -\eta\, \mathbb{E}_m[\Gamma_{t-1}^m\, u_{t-1}^m]$:

$$z_{t+1} - z_t = -\frac{\eta}{1-\beta_1}\, \mathbb{E}_m[\Gamma_t^m\, u_t^m] + \frac{\eta\, \beta_1}{1-\beta_1}\, \mathbb{E}_m[\Gamma_{t-1}^m\, u_{t-1}^m]. \tag{84}$$

We now add and subtract $\frac{\eta\, \beta_1}{1-\beta_1}\, \mathbb{E}_m[\Gamma_t^m\, u_{t-1}^m]$ inside (84). This is the key algebraic trick: it allows us to isolate the "ideal" gradient term from the preconditioner-mismatch error.

$$\begin{aligned} z_{t+1} - z_t = &-\frac{\eta}{1-\beta_1}\, \mathbb{E}_m[\Gamma_t^m\, u_t^m] + \frac{\eta\, \beta_1}{1-\beta_1}\, \mathbb{E}_m[\Gamma_t^m\, u_{t-1}^m] \\ &+ \frac{\eta\, \beta_1}{1-\beta_1}\, \mathbb{E}_m[(\Gamma_{t-1}^m - \Gamma_t^m)\, u_{t-1}^m]. \end{aligned} \tag{85}$$

Factor the first two terms:

$$-\frac{\eta}{1-\beta_1}\, \mathbb{E}_m[\Gamma_t^m\, u_t^m] + \frac{\eta\, \beta_1}{1-\beta_1}\, \mathbb{E}_m[\Gamma_t^m\, u_{t-1}^m] = -\frac{\eta}{1-\beta_1}\, \mathbb{E}_m\big[\Gamma_t^m\, (u_t^m - \beta_1\, u_{t-1}^m)\big]. \tag{86}$$

By the definition (78), $u_t^m - \beta_1\, u_{t-1}^m = (1-\beta_1)\, \widetilde{g}_t^m$, so

$$-\frac{\eta}{1-\beta_1}\, \mathbb{E}_m\big[\Gamma_t^m\, (1-\beta_1)\, \widetilde{g}_t^m\big] = -\eta\, \mathbb{E}_m[\Gamma_t^m\, \widetilde{g}_t^m]. \tag{87}$$

Substituting (86) and (87) into (85):

$$z_{t+1} - z_t = -\eta\, \mathbb{E}_m[\Gamma_t^m\, \widetilde{g}_t^m] + \frac{\eta\, \beta_1}{1-\beta_1}\, \mathbb{E}_m[(\Gamma_{t-1}^m - \Gamma_t^m)\, u_{t-1}^m]. \tag{88}$$

Now add and subtract $\eta\, \mathbb{E}_m[\Gamma_t^m\, g_t]$ inside the first term. Since $\mathbb{E}_m[\widetilde{g}_t^m] = g_t$ by (79):

$$\begin{aligned} -\eta\, \mathbb{E}_m[\Gamma_t^m\, \widetilde{g}_t^m] &= -\eta\, \mathbb{E}_m[\Gamma_t^m\, g_t] + \eta\, \mathbb{E}_m[\Gamma_t^m\, (g_t - \widetilde{g}_t^m)] \\ &= -\eta\, \Gamma_t\, g_t + \eta\, \mathbb{E}_m[\Gamma_t^m\, (g_t - \widetilde{g}_t^m)], \end{aligned} \tag{89}$$

where the last step uses the linearity of $\mathbb{E}_m$ and the fact that g_t is constant across the replica average, so $\mathbb{E}_m[\Gamma_t^m\, g_t] = \mathbb{E}_m[\Gamma_t^m]\, g_t = \Gamma_t\, g_t$.

Final decomposition. Combining (88) and (89), and defining the two error terms:

$$z_{t+1} - z_t = -\eta\, \Gamma_t\, g_t + \eta\, \underbrace{\mathbb{E}_m[\Gamma_t^m\, (g_t - \widetilde{g}_t^m)]}_{\overset{\text{def}}{=} U_t} + \eta\, \underbrace{\frac{\beta_1}{1-\beta_1}\, \mathbb{E}_m[(\Gamma_{t-1}^m - \Gamma_t^m)\, u_{t-1}^m]}_{\overset{\text{def}}{=} V_t}. \tag{90}$$

Compared to the SGDM result $z_{t+1} - z_t = -\eta\, g_t$ (equation (30)), two additional terms appear:

- U_t captures the *momentum drift*: the difference between the global averaged gradient g_t and the local pseudo-gradient $\widetilde{g}_t^m$, weighted by the local preconditioner Γ_t^m. This vanishes when all replicas are synchronized ($u_t^m = u_t$ for all m).
- V_t captures the *preconditioner drift*: the change in the preconditioning matrix between consecutive steps, weighted by the previous momentum. This vanishes when the second moment is stationary ($\Gamma_{t-1}^m = \Gamma_t^m$), which is approximately true when β_2 is close to 1 (large half-life).

Step 2: Smoothness descent on virtual iterates. As before, L-smoothness gives

$$f(z_{t+1}) \le f(z_t) + \langle \nabla f(z_t),\, z_{t+1} - z_t \rangle + \frac{L}{2}\,\|z_{t+1} - z_t\|^2. \tag{91}$$

Substituting (90) into (91) and expanding the inner product using $\nabla f(z_t) = \nabla f(x_t) + (\nabla f(z_t) - \nabla f(x_t))$:

$$f(z_{t+1}) - f(z_t) \le -\eta\,\langle \nabla f(z_t),\, \Gamma_t\, g_t \rangle + \eta\,\langle \nabla f(z_t),\, U_t \rangle + \eta\,\langle \nabla f(z_t),\, V_t \rangle + \frac{L}{2}\,\|z_{t+1} - z_t\|^2. \tag{92}$$

We split the first inner product by adding and subtracting $\nabla f(x_t)$:

$$-\eta\,\langle \nabla f(z_t),\, \Gamma_t\, g_t \rangle = -\eta\,\langle \nabla f(x_t),\, \Gamma_t\, g_t \rangle - \eta\,\langle \nabla f(z_t) - \nabla f(x_t),\, \Gamma_t\, g_t \rangle. \tag{93}$$

Substituting into (92) and labelling each group:

$$\begin{aligned} f(z_{t+1}) - f(z_t) \le\; & \underbrace{-\eta\,\langle \nabla f(x_t),\, \Gamma_t\, g_t \rangle}_{I} + \underbrace{\eta\,\langle \nabla f(z_t),\, U_t \rangle}_{II} + \underbrace{\eta\,\langle \nabla f(z_t),\, V_t \rangle}_{III} \\ & + \underbrace{\frac{L}{2}\,\|z_{t+1} - z_t\|^2}_{IV} + \underbrace{\eta\,\langle \nabla f(x_t) - \nabla f(z_t),\, \Gamma_t\, g_t \rangle}_{V}. \end{aligned} \tag{94}$$

Note that this is a five-term decomposition (compared to three in SGDM), reflecting the additional complexity of the adaptive preconditioner. We bound each term below.

Step 3a: Bounding term I. We split $\Gamma_t = \Gamma_{t-1} + (\Gamma_t - \Gamma_{t-1})$:

$$\begin{aligned} I &= -\eta\,\langle \nabla f(x_t),\, \Gamma_t\, g_t \rangle \\ &= -\eta\,\langle \nabla f(x_t),\, \Gamma_{t-1}\, g_t \rangle - \eta\,\langle \nabla f(x_t),\, (\Gamma_t - \Gamma_{t-1})\, g_t \rangle. \end{aligned} \tag{95}$$

For the second term, by Cauchy–Schwarz and $\|\nabla f(x_t)\| \le G$, $\|g_t\| \le G$:

$$|\langle \nabla f(x_t),\, (\Gamma_t - \Gamma_{t-1})\, g_t \rangle| \le \|\nabla f(x_t)\| \cdot \|\Gamma_t - \Gamma_{t-1}\| \cdot \|g_t\| \le G^2\,\|\Gamma_t - \Gamma_{t-1}\|. \tag{96}$$

For the first term, taking expectation and using unbiasedness $\mathbb{E}[g_t] = \frac{1}{M}\sum_m \nabla f_m(x_t^m)$, then applying the weighted polarization identity $\langle a, b\rangle_A = \frac{1}{2}\|a\|_A^2 + \frac{1}{2}\|b\|_A^2 - \frac{1}{2}\|a-b\|_A^2$ with $A = \Gamma_{t-1}$:

$$\begin{aligned} \mathbb{E}\big[-\eta\,\langle \nabla f(x_t),\, \Gamma_{t-1}\, g_t \rangle\big] &= -\eta\,\mathbb{E}\left[\left\langle \nabla f(x_t),\, \frac{1}{M}\sum_m \nabla f_m(x_t^m) \right\rangle_{\Gamma_{t-1}}\right] \\ &= -\frac{\eta}{2}\,\mathbb{E}\big[\|\nabla f(x_t)\|_{\Gamma_{t-1}}^2\big] - \frac{\eta}{2}\,\mathbb{E}\left[\left\|\frac{1}{M}\sum_m \nabla f_m(x_t^m)\right\|_{\Gamma_{t-1}}^2\right] \\ &\quad + \frac{\eta}{2}\,\mathbb{E}\left[\left\|\nabla f(x_t) - \frac{1}{M}\sum_m \nabla f_m(x_t^m)\right\|_{\Gamma_{t-1}}^2\right]. \end{aligned} \tag{97}$$

Now bound each factor using the spectral bounds on Γ_{t-1}:

$$\|\nabla f(x_t)\|^2_{\Gamma_{t-1}} \geq \|\Gamma_{t-1}\|_{\min} \|\nabla f(x_t)\|^2 \geq \frac{1}{C_0} \|\nabla f(x_t)\|^2, \tag{98}$$

$$\begin{aligned}\left\|\nabla f(x_t) - \frac{1}{M}\sum_m \nabla f_m(x_t^m)\right\|^2_{\Gamma_{t-1}} &\leq \|\Gamma_{t-1}\|_{\max} \left\|\nabla f(x_t) - \frac{1}{M}\sum_m \nabla f_m(x_t^m)\right\|^2 \\ &\leq \frac{1}{\lambda}(\cdots).\end{aligned} \tag{99}$$

For the last factor, by Jensen's inequality and L-smoothness (same chain as (43)–(44) in the SGDM proof):

$$\left\|\nabla f(x_t) - \frac{1}{M}\sum_m \nabla f_m(x_t^m)\right\|^2 \leq \frac{1}{M}\sum_m \|\nabla f_m(x_t) - \nabla f_m(x_t^m)\|^2 \leq \frac{L^2}{M}\sum_m \|x_t - x_t^m\|^2. \tag{100}$$

Combining (95)–(100):

$$\begin{aligned}\mathbb{E}[I] \leq -\frac{\eta}{2C_0}\mathbb{E}\|\nabla f(x_t)\|^2 - \frac{\eta}{2}\mathbb{E}\left[\left\|\frac{1}{M}\sum_m \nabla f_m(x_t^m)\right\|^2_{\Gamma_{t-1}}\right] \\ + \frac{\eta L^2}{2\lambda M}\sum_m \mathbb{E}\|x_t - x_t^m\|^2 + \eta G^2 \mathbb{E}\|\Gamma_{t-1} - \Gamma_t\|.\end{aligned} \tag{101}$$

Step 3b: Bounding term II. By Cauchy–Schwarz:

$$II = \eta \langle \nabla f(z_t), U_t\rangle \leq \eta \|\nabla f(z_t)\| \|U_t\|. \tag{102}$$

By Assumption D.1, $\|\nabla f(z_t)\| \leq G$. Now bound $\|U_t\|$. Recall $U_t = \mathbb{E}_m[\Gamma_t^m (g_t - \widetilde{g}_t^m)]$. By Jensen's inequality applied to $\|\cdot\|$:

$$\begin{aligned}\|U_t\| &= \|\mathbb{E}_m[\Gamma_t^m (g_t - \widetilde{g}_t^m)]\| \\ &\leq \mathbb{E}_m\left[\|\Gamma_t^m (g_t - \widetilde{g}_t^m)\|\right] \\ &= \frac{1}{M}\sum_m \|\Gamma_t^m (g_t - \widetilde{g}_t^m)\|.\end{aligned} \tag{103}$$

Since Γ_t^m is diagonal with $\|\Gamma_t^m\| \leq 1/\lambda$ by (75):

$$\|\Gamma_t^m (g_t - \widetilde{g}_t^m)\| \leq \|\Gamma_t^m\| \|g_t - \widetilde{g}_t^m\| \leq \frac{1}{\lambda}\|g_t - \widetilde{g}_t^m\|. \tag{104}$$

Combining (102)–(104):

$$\mathbb{E}[II] \leq \frac{\eta G}{\lambda M}\sum_m \mathbb{E}\|g_t - \widetilde{g}_t^m\|. \tag{105}$$

Step 3c: Bounding term III. Similarly, $III = \eta \langle \nabla f(z_t), V_t\rangle \leq \eta G \|V_t\|$. Recall $V_t = \frac{\beta_1}{1-\beta_1}\mathbb{E}_m[(\Gamma_{t-1}^m - \Gamma_t^m) u_{t-1}^m]$. By Jensen:

$$\begin{aligned}\|V_t\| &\leq \frac{\beta_1}{1-\beta_1}\frac{1}{M}\sum_m \|(\Gamma_{t-1}^m - \Gamma_t^m) u_{t-1}^m\| \\ &\leq \frac{\beta_1}{1-\beta_1}\frac{1}{M}\sum_m \|\Gamma_{t-1}^m - \Gamma_t^m\| \|u_{t-1}^m\|.\end{aligned} \tag{106}$$

Since u_{t-1}^m is a convex combination of gradients bounded by G, we have $\|u_{t-1}^m\| \leq G$. Therefore:

$$\mathbb{E}[III] \leq \frac{\eta \beta_1 G^2}{(1-\beta_1) M}\sum_m \mathbb{E}\|\Gamma_{t-1}^m - \Gamma_t^m\|. \tag{107}$$

Step 3d: Bounding term IV. From (90), $z_{t+1} - z_t = -\eta\,\Gamma_t\,g_t + \eta\,U_t + \eta\,V_t$, so

$$\|z_{t+1} - z_t\|^2 = \eta^2\,\|\Gamma_t\,g_t - U_t - V_t\|^2. \tag{108}$$

By the triangle inequality for norms and the elementary bound $\|a+b+c\|^2 \le 3(\|a\|^2+\|b\|^2+\|c\|^2)$:

$$\|\Gamma_t\,g_t - U_t - V_t\|^2 \le 3\,\|\Gamma_t\,g_t\|^2 + 3\,\|U_t\|^2 + 3\,\|V_t\|^2. \tag{109}$$

We bound each:

$$\|\Gamma_t\,g_t\|^2 \le \frac{1}{\lambda^2}\,\|g_t\|^2 \qquad\qquad \le \frac{G^2}{\lambda^2}, \tag{110}$$

$$\|U_t\|^2 \le \frac{1}{\lambda^2\,M}\sum_m \|g_t - \widetilde{g}_t^m\|^2, \tag{111}$$

$$\|V_t\|^2 \le \frac{\beta_1^2\,G^2}{(1-\beta_1)^2\,M}\sum_m \|\Gamma_{t-1}^m - \Gamma_t^m\|^2. \tag{112}$$

Here (111) uses Jensen on the average defining U_t plus $\|\Gamma_t^m\| \le 1/\lambda$; (112) uses (106) squared plus $\|u_{t-1}^m\| \le G$.

Combining:

$$\begin{aligned}\mathbb{E}[IV] \le{}& \frac{3\eta^2\,L\,G^2}{2\lambda^2} + \frac{3\eta^2\,L}{2\lambda^2\,M}\sum_m \mathbb{E}\|g_t - \widetilde{g}_t^m\|^2 \\ &+ \frac{3\eta^2\,\beta_1^2\,L\,G^2}{2(1-\beta_1)^2\,M}\sum_m \mathbb{E}\|\Gamma_{t-1}^m - \Gamma_t^m\|^2.\end{aligned} \tag{113}$$

Step 3e: Bounding term V. This term measures the cost of approximating $\nabla f(z_t)$ by $\nabla f(x_t)$ in the inner product with $\Gamma_t\,g_t$.

First, we bound the "virtual drift" $\|\nabla f(x_t) - \nabla f(z_t)\|$. By L-smoothness:

$$\|\nabla f(x_t) - \nabla f(z_t)\| \le L\,\|x_t - z_t\|. \tag{114}$$

From the definition (82):

$$\begin{aligned}x_t - z_t &= x_t - \frac{1}{1-\beta_1}\,x_t + \frac{\beta_1}{1-\beta_1}\,x_{t-1} \\ &= -\frac{\beta_1}{1-\beta_1}\,(x_t - x_{t-1}) \\ &= \frac{\beta_1}{1-\beta_1}\,d_{t-1} = \frac{\eta\,\beta_1}{1-\beta_1}\,\mathbb{E}_m[\Gamma_{t-1}^m\,u_{t-1}^m].\end{aligned} \tag{115}$$

Taking norms and using $\|\Gamma_{t-1}^m\| \le 1/\lambda$, $\|u_{t-1}^m\| \le G$:

$$\|x_t - z_t\| \le \frac{\eta\,\beta_1}{1-\beta_1}\,\mathbb{E}_m[\|\Gamma_{t-1}^m\|\,\|u_{t-1}^m\|] \le \frac{\eta\,\beta_1\,G}{(1-\beta_1)\,\lambda}. \tag{116}$$

Hence:

$$\|\nabla f(x_t) - \nabla f(z_t)\| \le \frac{\eta\,\beta_1\,L\,G}{(1-\beta_1)\,\lambda}. \tag{117}$$

Now we bound V itself. Split $\Gamma_t = \Gamma_{t-1} + (\Gamma_t - \Gamma_{t-1})$:

$$\begin{aligned}V &= \eta\,\langle \nabla f(x_t) - \nabla f(z_t),\,\Gamma_t\,g_t\rangle \\ &= \eta\,\langle \nabla f(x_t) - \nabla f(z_t),\,\Gamma_{t-1}\,g_t\rangle + \eta\,\langle \nabla f(x_t) - \nabla f(z_t),\,(\Gamma_t - \Gamma_{t-1})\,g_t\rangle.\end{aligned} \tag{118}$$

For the second term, using (117):

$$|\langle \nabla f(x_t) - \nabla f(z_t),\,(\Gamma_t - \Gamma_{t-1})\,g_t\rangle| \le \frac{\eta\,\beta_1\,L\,G^2}{(1-\beta_1)\,\lambda}\,\|\Gamma_t - \Gamma_{t-1}\|. \tag{119}$$

For the first term, using unbiasedness and splitting $\frac{1}{M}\sum_m \nabla f_m(x_t^m) = \nabla f(x_t) + (\frac{1}{M}\sum_m \nabla f_m(x_t^m) - \nabla f(x_t))$, then applying Young's inequality $\langle a, b\rangle \le \frac{1}{2\rho}\|a\|^2 + \frac{\rho}{2}\|b\|^2$ to each piece:

$$\begin{aligned}
&\mathbb{E}\big[\eta\,\langle \nabla f(x_t) - \nabla f(z_t),\, \Gamma_{t-1}\, g_t\rangle\big] \\
&\quad\le \frac{\eta}{\lambda}\,\mathbb{E}\left[\frac{1}{2\rho}\,\|\nabla f(x_t) - \nabla f(z_t)\|^2 + \frac{\rho}{2}\,\|\nabla f(x_t)\|^2\right] \\
&\qquad + \frac{\eta}{\lambda}\,\mathbb{E}\left[\frac{1}{2}\,\|\nabla f(x_t) - \nabla f(z_t)\|^2 + \frac{L^2}{2M}\sum_m \|x_t^m - x_t\|^2\right].
\end{aligned} \tag{120}$$

Substituting (117) into the $\|\nabla f(x_t) - \nabla f(z_t)\|^2$ terms, all such terms are $\mathcal{O}(\eta^2)$. Collecting:

$$\begin{aligned}
\mathbb{E}[V] \le \frac{\eta\,\rho}{2\lambda}\,\mathbb{E}\|\nabla f(x_t)\|^2 + \mathcal{O}(\eta)\cdot\frac{1}{M}\sum_m \mathbb{E}\|x_t^m - x_t\|^2 \\
+ \mathcal{O}\!\left(\frac{\eta^3}{\rho}\right) + \frac{\eta^2\,\beta_1\,L\,G^2}{(1-\beta_1)\,\lambda}\,\mathbb{E}\|\Gamma_t - \Gamma_{t-1}\|.
\end{aligned} \tag{121}$$

Summary of the five bounds. Collecting (101), (105), (107), (113), and (121), and suppressing constant factors into $\mathcal{O}(\cdot)$ notation:

$$I \le -\frac{\eta}{2C_0}\,\mathbb{E}\|\nabla f(x_t)\|^2 + \mathcal{O}(\eta)\cdot\frac{1}{M}\sum_m \mathbb{E}\|x_t - x_t^m\|^2 + \mathcal{O}(\eta)\cdot\frac{1}{M}\sum_m \mathbb{E}\|\Gamma_{t-1}^m - \Gamma_t^m\|, \tag{122}$$

$$II \le \mathcal{O}(\eta)\cdot\frac{1}{M}\sum_m \mathbb{E}\|g_t - \widetilde{g}_t^m\|, \tag{123}$$

$$III \le \mathcal{O}(\eta)\cdot\frac{1}{M}\sum_m \mathbb{E}\|\Gamma_{t-1}^m - \Gamma_t^m\|, \tag{124}$$

$$IV \le \mathcal{O}(\eta^2), \tag{125}$$

$$V \le \frac{\eta\,\rho}{2\lambda}\,\mathbb{E}\|\nabla f(x_t)\|^2 + \mathcal{O}(\eta)\cdot\frac{1}{M}\sum_m \mathbb{E}\|x_t^m - x_t\|^2 + \mathcal{O}(\eta^2). \tag{126}$$

To obtain an $\mathcal{O}(1/\sqrt{T})$ convergence rate, we set $\rho = \lambda/(2C_0)$ so that the $\|\nabla f(x_t)\|^2$ coefficient in V exactly cancels against the negative coefficient in I, and then verify three conditions:

$$\text{(A1)}\quad \frac{1}{TM}\sum_{t,m}\mathbb{E}\|x_t^m - x_t\|^2 = \mathcal{O}(\eta^2), \qquad \text{(extension of Lemma D.2)} \tag{127}$$

$$\text{(A2)}\quad \sum_{t=0}^{T-1}\mathbb{E}\|\Gamma_{t-1}^m - \Gamma_t^m\| = \mathcal{O}(1), \qquad \text{(AMSGrad telescoping)} \tag{128}$$

$$\text{(A3)}\quad \frac{1}{M}\sum_{t,m}\mathbb{E}\|g_t - \widetilde{g}_t^m\| = \mathcal{O}(1). \qquad \text{(double geometric sum)} \tag{129}$$

Condition (A2): AMSGrad telescoping. Since $\tilde{v}_t^m = \max(v_t^m, \tilde{v}_{t-1}^m)$ coordinate-wise, the sequence $\{[\tilde{v}_t^m]_i\}_{t\ge 0}$ is non-decreasing and bounded above by G^2 for each coordinate i. Therefore $[\Gamma_t^m]_{ii} = 1/\sqrt{[\tilde{v}_t^m]_i + \lambda^2}$ is non-increasing, and

$$\sum_{t=0}^{T-1}\|\Gamma_{t-1}^m - \Gamma_t^m\| = \sum_{t=0}^{T-1}\max_i\big|[\Gamma_{t-1}^m]_{ii} - [\Gamma_t^m]_{ii}\big| \le \sum_i\big([\Gamma_{-1}^m]_{ii} - [\Gamma_{T-1}^m]_{ii}\big) \le \frac{d}{\lambda}, \tag{130}$$

which is $\mathcal{O}(1)$ with respect to T.

Condition (A3): double geometric sum for the pseudo-gradient drift. We show that $\mathbb{E}\|g_t - \widetilde{g}_t^m\|$ has the same double geometric sum structure as the parameter drift in Lemma D.2.

First, expand the momentum drift. With probability p_u (sync at step t), $u_t^m = u_t$, so $\|u_t - u_t^m\| = 0$. With probability $1 - p_u$:

$$\begin{aligned}\|u_t - u_t^m\| &= \|\beta_1\, u_{t-1} + (1-\beta_1)\, g_t - \beta_1\, u_{t-1}^m - (1-\beta_1)\, g_t^m\| \\ &= \|\beta_1\,(u_{t-1} - u_{t-1}^m) + (1-\beta_1)\,(g_t - g_t^m)\| \\ &\le \beta_1\, \|u_{t-1} - u_{t-1}^m\| + (1-\beta_1)\, \|g_t - g_t^m\|. \end{aligned} \tag{131}$$

Taking expectation over the sync coin:

$$\mathbb{E}\|u_t - u_t^m\| \le (1-p_u)\,\beta_1\,\mathbb{E}\|u_{t-1} - u_{t-1}^m\| + (1-p_u)\,(1-\beta_1)\,\mathbb{E}\|g_t - g_t^m\|. \tag{132}$$

Unrolling (132) from $\tau = 0$ to t (with $u_{-1} = u_{-1}^m = 0$):

$$\mathbb{E}\|u_t - u_t^m\| \le (1-p_u)\,(1-\beta_1) \sum_{\tau=0}^{t} \big((1-p_u)\,\beta_1\big)^{t-\tau}\, \mathbb{E}\|g_\tau - g_\tau^m\|. \tag{133}$$

Now bound $\mathbb{E}\|g_t - \widetilde{g}_t^m\|$. By definition (78):

$$\begin{aligned} g_t - \widetilde{g}_t^m &= \frac{u_t - \beta_1\, u_{t-1}}{1-\beta_1} - \frac{u_t^m - \beta_1\, u_{t-1}^m}{1-\beta_1} \\ &= \frac{(u_t - u_t^m) - \beta_1\,(u_{t-1} - u_{t-1}^m)}{1-\beta_1}. \end{aligned} \tag{134}$$

By the triangle inequality:

$$\begin{aligned} \|g_t - \widetilde{g}_t^m\| &\le \frac{1}{1-\beta_1}\, \|u_t - u_t^m\| + \frac{\beta_1}{1-\beta_1}\, \|u_{t-1} - u_{t-1}^m\| \\ &= \frac{1}{1-\beta_1} \sum_{\tau=t-1}^{t} \beta_1^{t-\tau}\, \|u_\tau - u_\tau^m\|. \end{aligned} \tag{135}$$

Substituting (133) into (135) and letting $q_2 = (1-p_u)\,\beta_1$:

$$\begin{aligned} \mathbb{E}\|g_t - \widetilde{g}_t^m\| &\le (1-p_u) \sum_{\tau=t-1}^{t} \sum_{\nu=0}^{\tau} \beta_1^{t-\tau}\, q_2^{\tau-\nu}\, \mathbb{E}\|g_\nu - g_\nu^m\| \\ &= \sum_{\tau=t}^{t+1} \sum_{\nu=0}^{\tau-1} \beta_1^{t+1-\tau}\, q_2^{\tau-\nu}\, \mathbb{E}\|g_\nu - g_\nu^m\|, \end{aligned} \tag{136}$$

which has the same *double geometric sum* structure as equation (162) in Lemma D.2. By the same summation technique (swapping the order of summation and bounding the geometric series), the total over $t = 0, \dots, T-1$ is $\mathcal{O}(1)$, confirming condition (129).

Remark (SP invariance and SP-tightened constants for Adam). As in the SGDM analysis, the three conditions (127)–(129) depend only on the synchronization probabilities p_x, p_u, p_v, the decay factors β_1, β_2, and the gradient bound G. None involves the SP degree P. The convergence rate of FAUST-Adam is therefore P-independent in order.

The SP-tightened constant refinement from the SGDM analysis (Eq. (72)) extends naturally to Adam's two-moment structure. Since the first-moment drift (3) and the second-moment drift (4) both carry the $1/\sqrt{P}$ factor for attention-layer coordinates, the combined drift contribution from these coordinates to the preconditioned step $d_t^m = \eta_t \Gamma_t u_t^m$ (where $\Gamma_t = \mathrm{diag}(v_t + \lambda^2)^{-1/2}$) is also reduced by $1/\sqrt{P}$. Concretely, the $\tilde{g}$-drift bound (135) for the attention-layer coordinates $j \in \mathcal{I}_{\text{attn}}$ becomes $\|\tilde{g}_{t,j} - \tilde{g}_{t-1,j}\|_\infty \le 2G\,\gamma_{\max}/\sqrt{P}$, tightening the spectral-norm bound on $\Gamma_t - \Gamma_{t-1}$ and yielding a sharper constant in (130).

The collective orthogonality constraint (fence at K_x-boundaries) is again a necessary condition: for Adam, the second moment v_t^m depends on $g_t^m \odot g_t^m$, so a partially-reduced gradient would corrupt both moments simultaneously, making the preconditioner Γ_t inconsistent across replicas and invalidating the spectral bound chain.

D.2 Key Lemmas

We now prove the two auxiliary lemmas used in the main proof.

Lemma D.1 (Virtual iterate drift)**.** *For all $T \geq 1$,*

$$\sum_{t=0}^{T-1} \mathbb{E}\|z_t - x_t\|^2 \leq \frac{\eta^2\,\beta^2}{(1-\beta)^2\,M}\,T\,\sigma^2 + \frac{\eta^2\,\beta^2}{(1-\beta)^2} \sum_{t=0}^{T-1} \mathbb{E}\left\|\frac{1}{M}\sum_m \nabla f_m(x_t^m)\right\|^2. \tag{137}$$

Proof. **Step (i): express $z_t - x_t$ in terms of momenta.**

From the definition (24) and the parameter recursion (23):

$$\begin{aligned} z_t - x_t &= \frac{1}{1-\beta}\,x_t - \frac{\beta}{1-\beta}\,x_{t-1} - x_t \\ &= -\frac{\beta}{1-\beta}\,(x_t - x_{t-1}) \\ &= -\frac{\beta}{1-\beta}\,(-\eta\,u_{t-1}) = \frac{\eta\,\beta}{1-\beta}\,u_{t-1}. \end{aligned} \tag{138}$$

Wait—we need to be more careful. Actually, for $t \geq 1$: $x_t - x_{t-1} = -\eta\,u_{t-1}$ by (23). So:

$$z_t - x_t = \frac{\eta\,\beta}{1-\beta}\,u_{t-1}. \tag{139}$$

For $t = 0$: $z_0 = x_0$, so $z_0 - x_0 = 0$, which is consistent with $u_{-1} = 0$.

Step (ii): unroll the momentum.

Since $u_{-1} = 0$, unrolling the recursion $u_t = \beta\,u_{t-1} + (1-\beta)\,g_t$:

$$\begin{aligned} u_0 &= (1-\beta)\,g_0, \\ u_1 &= \beta\,(1-\beta)\,g_0 + (1-\beta)\,g_1 = (1-\beta)\sum_{\tau=0}^{1}\beta^{1-\tau}\,g_\tau, \\ &\vdots \\ u_t &= (1-\beta)\sum_{\tau=0}^{t}\beta^{t-\tau}\,g_\tau. \end{aligned} \tag{140}$$

Substituting into (139):

$$z_t - x_t = \eta\,\beta\sum_{\tau=0}^{t-1}\beta^{t-1-\tau}\,g_\tau = \eta\,\beta\sum_{\tau=0}^{t-1}\beta^{t-1-\tau}\,g_\tau. \tag{141}$$

Re-indexing: let $k = t-1-\tau$, then $\tau = t-1-k$ and the sum runs $k = 0, \ldots, t-1$:

$$z_t - x_t = \eta\,\beta\sum_{k=0}^{t-1}\beta^{k}\,g_{t-1-k}. \tag{142}$$

Step (iii): bound $\|z_t - x_t\|^2$ using Jensen.

Define the weights $w_k = \beta^k / s_{t-1}$ where $s_{t-1} = \sum_{k=0}^{t-1} \beta^k = (1-\beta^t)/(1-\beta) \le 1/(1-\beta)$. These are non-negative and sum to 1, so by Jensen's inequality ($\|\sum_k w_k\, a_k\|^2 \le \sum_k w_k\, \|a_k\|^2$):

$$\begin{aligned}
\|z_t - x_t\|^2 &= \eta^2\, \beta^2\, s_{t-1}^2 \left\| \sum_{k=0}^{t-1} \frac{\beta^k}{s_{t-1}}\, g_{t-1-k} \right\|^2 \\
&\le \eta^2\, \beta^2\, s_{t-1}^2 \sum_{k=0}^{t-1} \frac{\beta^k}{s_{t-1}}\, \|g_{t-1-k}\|^2 \\
&= \eta^2\, \beta^2\, s_{t-1} \sum_{k=0}^{t-1} \beta^k\, \|g_{t-1-k}\|^2 \\
&\le \frac{\eta^2\, \beta^2}{1-\beta} \sum_{\tau=0}^{t-1} \beta^{t-1-\tau}\, \|g_\tau\|^2, \qquad (143)
\end{aligned}$$

where in the last step we substituted $\tau = t-1-k$ and used $s_{t-1} \le 1/(1-\beta)$.

Step (iv): sum over t and swap summation order.

$$\sum_{t=0}^{T-1} \mathbb{E}\|z_t - x_t\|^2 \le \frac{\eta^2\, \beta^2}{1-\beta} \sum_{t=0}^{T-1} \sum_{\tau=0}^{t-1} \beta^{t-1-\tau}\, \mathbb{E}\|g_\tau\|^2. \qquad (144)$$

(The $t = 0$ term contributes 0 since the inner sum is empty.)

Swap the order of summation. The inner sum has τ ranging from 0 to $t-1$ and t from 1 to $T-1$. For a fixed τ, t ranges from $\tau+1$ to $T-1$:

$$\begin{aligned}
\sum_{t=1}^{T-1} \sum_{\tau=0}^{t-1} \beta^{t-1-\tau}\, \mathbb{E}\|g_\tau\|^2 &= \sum_{\tau=0}^{T-2} \sum_{t=\tau+1}^{T-1} \beta^{t-1-\tau}\, \mathbb{E}\|g_\tau\|^2 \\
&= \sum_{\tau=0}^{T-2} \left(\sum_{j=0}^{T-2-\tau} \beta^j \right) \mathbb{E}\|g_\tau\|^2 \qquad (\text{set } j = t-1-\tau) \\
&= \sum_{\tau=0}^{T-2} \frac{1-\beta^{T-1-\tau}}{1-\beta}\, \mathbb{E}\|g_\tau\|^2 \\
&\le \frac{1}{1-\beta} \sum_{\tau=0}^{T-1} \mathbb{E}\|g_\tau\|^2. \qquad (145)
\end{aligned}$$

Hence:

$$\sum_{t=0}^{T-1} \mathbb{E}\|z_t - x_t\|^2 \le \frac{\eta^2\, \beta^2}{(1-\beta)^2} \sum_{\tau=0}^{T-1} \mathbb{E}\|g_\tau\|^2. \qquad (146)$$

Step (v): decompose $\|g_\tau\|^2$ into signal and noise.

Recall $g_\tau = \frac{1}{M}\sum_m g_\tau^m$. Write $g_\tau^m = \nabla f_m(x_\tau^m) + \varepsilon_\tau^m$ where $\varepsilon_\tau^m = g_\tau^m - \nabla f_m(x_\tau^m)$ is the noise. Then:

$$\begin{aligned}
\|g_\tau\|^2 &= \left\| \frac{1}{M} \sum_m \varepsilon_\tau^m + \frac{1}{M} \sum_m \nabla f_m(x_\tau^m) \right\|^2 \\
&= \left\| \frac{1}{M} \sum_m \varepsilon_\tau^m \right\|^2 + \left\| \frac{1}{M} \sum_m \nabla f_m(x_\tau^m) \right\|^2 + 2 \left\langle \frac{1}{M} \sum_m \varepsilon_\tau^m,\, \frac{1}{M} \sum_m \nabla f_m(x_\tau^m) \right\rangle. \qquad (147)
\end{aligned}$$

Taking expectation, the cross-term vanishes by unbiasedness ($\mathbb{E}[\varepsilon_\tau^m] = 0$):

$$\mathbb{E}\|g_\tau\|^2 = \mathbb{E}\left\|\frac{1}{M}\sum_m \varepsilon_\tau^m\right\|^2 + \mathbb{E}\left\|\frac{1}{M}\sum_m \nabla f_m(x_\tau^m)\right\|^2. \tag{148}$$

For the noise term, by independence across replicas:

$$\mathbb{E}\left\|\frac{1}{M}\sum_m \varepsilon_\tau^m\right\|^2 = \frac{1}{M^2}\sum_m \mathbb{E}\|\varepsilon_\tau^m\|^2 \le \frac{\sigma^2}{M}. \tag{149}$$

Step (vi): assemble.

Substituting (148) and (149) into (146):

$$\begin{aligned}\sum_{t=0}^{T-1}\mathbb{E}\|z_t - x_t\|^2 &\le \frac{\eta^2\,\beta^2}{(1-\beta)^2}\sum_{\tau=0}^{T-1}\left(\frac{\sigma^2}{M} + \mathbb{E}\left\|\frac{1}{M}\sum_m \nabla f_m(x_\tau^m)\right\|^2\right)\\ &= \frac{\eta^2\,\beta^2}{(1-\beta)^2\,M}\,T\,\sigma^2 + \frac{\eta^2\,\beta^2}{(1-\beta)^2}\sum_{\tau=0}^{T-1}\mathbb{E}\left\|\frac{1}{M}\sum_m \nabla f_m(x_\tau^m)\right\|^2. \end{aligned}\tag{150}$$

This completes the proof. □

Lemma D.2 (Client drift bound)**.** *If* $24\eta^2 L^2 \psi \le 1$, *then*

$$\frac{1}{MT}\sum_{t=0}^{T-1}\sum_m \mathbb{E}\|x_t - x_t^m\|^2 \le 12\eta^2\,(B^2-1)\,\psi\cdot\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla f(x_t)\|^2 + 4\eta^2\,\psi\,(\sigma^2 + 3G^2), \tag{151}$$

where the drift coefficient is

$$\psi \;=\; \frac{4(1-p_x)}{p_x^2}\cdot\frac{(1-\beta)(1-p_u)}{1-(1-p_u)\beta}. \tag{152}$$

Proof. The proof proceeds in five steps: (i) expand the parameter drift using the probabilistic update; (ii) expand the momentum drift; (iii) combine via a double geometric sum; (iv) bound the gradient variance; (v) close the recursion.

Step (i): parameter drift recursion.

At step t, with probability p_x (sync), $x_{t+1}^m = \mathbb{E}_m[\cdots]$, so $x_{t+1} - x_{t+1}^m = 0$. With probability $1-p_x$ (no sync):

$$\begin{aligned}x_{t+1} - x_{t+1}^m &= (x_t - \eta\,u_t) - (x_t^m - \eta\,u_t^m)\\ &= (x_t - x_t^m) - \eta\,(u_t - u_t^m).\end{aligned}\tag{153}$$

Taking squared norms:

$$\|x_{t+1} - x_{t+1}^m\|^2 = \|(x_t - x_t^m) - \eta\,(u_t - u_t^m)\|^2. \tag{154}$$

By the Young-type inequality $\|a-b\|^2 \le (1+s)\,\|a\|^2 + (1+1/s)\,\|b\|^2$ for any $s > 0$ (which follows from $\|a-b\|^2 = \|a\|^2 + \|b\|^2 - 2\langle a,b\rangle \le (1+s)\|a\|^2 + (1+1/s)\|b\|^2$ by Young applied to the cross-term):

$$\|x_{t+1} - x_{t+1}^m\|^2 \le (1+s)\,\|x_t - x_t^m\|^2 + \eta^2\,(1+\tfrac{1}{s})\,\|u_t - u_t^m\|^2. \tag{155}$$

Taking expectation over the sync coin:

$$\mathbb{E}\|x_{t+1} - x_{t+1}^m\|^2 \le (1-p_x)(1+s)\,\mathbb{E}\|x_t - x_t^m\|^2 + \eta^2\,(1-p_x)(1+\tfrac{1}{s})\,\mathbb{E}\|u_t - u_t^m\|^2. \tag{156}$$

Let $q_1 \overset{\text{def}}{=} (1-p_x)(1+s)$. Unrolling (156) from $t=0$ (using $x_0 - x_0^m = 0$):

$$\mathbb{E}\|x_{t+1} - x_{t+1}^m\|^2 \le \eta^2\,(1-p_x)(1+\tfrac{1}{s})\sum_{\tau=0}^{t} q_1^{t-\tau}\,\mathbb{E}\|u_\tau - u_\tau^m\|^2. \tag{157}$$

Step (ii): momentum drift recursion.

With probability p_u (sync), $u_t^m = u_t$. With probability $1 - p_u$:

$$u_t - u_t^m = \beta\,(u_{t-1} - u_{t-1}^m) + (1-\beta)\,(g_t - g_t^m). \tag{158}$$

By the inequality $\|\alpha\, a + (1-\alpha)\, b\|^2 \le \alpha\,\|a\|^2 + (1-\alpha)\,\|b\|^2$ (Jensen, since $\beta \in [0,1)$):

$$\|u_t - u_t^m\|^2 \le \beta\,\|u_{t-1} - u_{t-1}^m\|^2 + (1-\beta)\,\|g_t - g_t^m\|^2. \tag{159}$$

Taking expectation over the sync coin:

$$\mathbb{E}\|u_t - u_t^m\|^2 \le (1-p_u)\,\beta\,\mathbb{E}\|u_{t-1} - u_{t-1}^m\|^2 + (1-p_u)\,(1-\beta)\,\mathbb{E}\|g_t - g_t^m\|^2. \tag{160}$$

Let $q_2 \overset{\text{def}}{=} (1-p_u)\,\beta$. Unrolling (with $u_{-1} = u_{-1}^m = 0$):

$$\mathbb{E}\|u_t - u_t^m\|^2 \le (1-p_u)\,(1-\beta)\sum_{\tau=0}^{t} q_2^{t-\tau}\,\mathbb{E}\|g_\tau - g_\tau^m\|^2 \le \frac{1-\beta}{\beta}\sum_{\tau=0}^{t} q_2^{t+1-\tau}\,\mathbb{E}\|g_\tau - g_\tau^m\|^2. \tag{161}$$

Step (iii): combine into a double geometric sum.

Substitute (161) into (157):

$$\begin{aligned}
&\frac{1}{M}\sum_m \mathbb{E}\|x_t - x_t^m\|^2 \\
&\quad\le \eta^2\,(1-p_x)(1+\tfrac{1}{s})\,\frac{1-\beta}{\beta}\sum_{\tau=0}^{t-1}\sum_{\nu=0}^{\tau} q_1^{t-1-\tau}\,q_2^{\tau+1-\nu}\,\frac{1}{M}\sum_m \mathbb{E}\|g_\nu - g_\nu^m\|^2 \\
&\quad= \eta^2\,\underbrace{(1-p_x)(1+\tfrac{1}{s})(1-\beta)(1-p_u)}_{\overset{\text{def}}{=}\,\phi}\sum_{\nu=0}^{t-1}\frac{q_1^{t-\nu} - q_2^{t-\nu}}{q_1 - q_2}\,\frac{1}{M}\sum_m \mathbb{E}\|g_\nu - g_\nu^m\|^2,
\end{aligned} \tag{162}$$

where the last line evaluates the inner convolution $\sum_{\tau=\nu}^{t-1} q_1^{t-1-\tau}\,q_2^{\tau+1-\nu}$ via the standard geometric-series identity $\sum_{k=0}^{n} a^k\,b^{n-k} = (a^{n+1} - b^{n+1})/(a-b)$.

Step (iv): time-average and optimize s.

Average (162) over $t = 0, \dots, T-1$, swap the summation order, and bound the geometric series:

$$\frac{1}{MT}\sum_{t,m}\mathbb{E}\|x_t - x_t^m\|^2 \le \frac{\eta^2\,\phi}{(1-q_1)(1-q_2)}\,\frac{1}{T}\sum_{\tau=0}^{T-1}\frac{1}{M}\sum_m \mathbb{E}\|g_\tau - g_\tau^m\|^2. \tag{163}$$

We minimize $\phi/((1-q_1)(1-q_2))$ over $s > 0$. The factor depending on s is

$$h(s) = \frac{(1-p_x)(1+1/s)}{1-(1-p_x)(1+s)}. \tag{164}$$

Setting $h'(s) = 0$ yields $s^* = 1/\sqrt{1-p_x} - 1$, giving

$$h(s^*) = \frac{(1-p_x)(1+\sqrt{1-p_x})^2}{p_x^2} \le \frac{4(1-p_x)}{p_x^2}, \tag{165}$$

since $(1+\sqrt{1-p_x})^2 \le 4$ for $p_x \in [0,1]$.

Therefore:

$$\frac{\phi}{(1-q_1)(1-q_2)} \le \frac{4(1-p_x)}{p_x^2}\cdot\frac{(1-\beta)(1-p_u)}{1-(1-p_u)\beta} = \psi. \tag{166}$$

Step (v): bound the gradient variance and close.

Using Lemma D.3 (proved below):

$$\frac{1}{M}\sum_m \mathbb{E}\|g_t - g_t^m\|^2 \le 2\sigma^2 + \frac{12L^2}{M}\sum_m \mathbb{E}\|x_t - x_t^m\|^2 + 6(B^2-1)\,\mathbb{E}\|\nabla f(x_t)\|^2 + 6G^2. \quad (167)$$

Substituting into (163):

$$\frac{1}{MT}\sum_{t,m}\mathbb{E}\|x_t - x_t^m\|^2 \le 12\eta^2 L^2 \psi \cdot \frac{1}{TM}\sum_{t,m}\mathbb{E}\|x_t - x_t^m\|^2 + 6\eta^2(B^2-1)\psi\cdot\frac{1}{T}\sum_t \mathbb{E}\|\nabla f(x_t)\|^2 + 2\eta^2\psi(\sigma^2+3G^2). \quad (168)$$

Denote the left-hand side by Δ. Then $\Delta \le 12\eta^2 L^2\psi\,\Delta + (\cdots)$. If $12\eta^2 L^2\psi \le 1/2$ (guaranteed by $24\eta^2 L^2\psi \le 1$), then $\Delta(1-12\eta^2L^2\psi) \ge \Delta/2$, so:

$$\Delta \le 12\eta^2(B^2-1)\psi\cdot\frac{1}{T}\sum_t \mathbb{E}\|\nabla f(x_t)\|^2 + 4\eta^2\psi(\sigma^2+3G^2). \quad (169)$$

□

Lemma D.3 (Heterogeneity decomposition)**.** *Under Assumptions 3.1 and 3.3,*

$$\frac{1}{M}\sum_m \left\|\nabla f_m(x_t^m) - \frac{1}{M}\sum_i \nabla f_i(x_t^i)\right\|^2 \le \frac{6L^2}{M}\sum_m \|x_t - x_t^m\|^2 + 3(B^2-1)\,\|\nabla f(x_t)\|^2 + 3G^2. \quad (170)$$

Proof. For each replica m, write

$$\nabla f_m(x_t^m) - \frac{1}{M}\sum_i \nabla f_i(x_t^i) = \underbrace{\left(\nabla f_m(x_t^m) - \nabla f_m(x_t)\right)}_{A_m} + \underbrace{\left(\nabla f_m(x_t) - \nabla f(x_t)\right)}_{B_m} + \underbrace{\left(\nabla f(x_t) - \frac{1}{M}\sum_i \nabla f_i(x_t^i)\right)}_{C}. \quad (171)$$

By the inequality $\|A+B+C\|^2 \le 3(\|A\|^2+\|B\|^2+\|C\|^2)$ (which follows from $\|A+B+C\|^2 \le (3\|A\|)(3^{-1}\|A\|) + \cdots$ and Cauchy–Schwarz, or equivalently from Jensen applied to the convex function $\|\cdot\|^2$):

$$\left\|\nabla f_m(x_t^m) - \frac{1}{M}\sum_i \nabla f_i(x_t^i)\right\|^2 \le 3\,\|A_m\|^2 + 3\,\|B_m\|^2 + 3\,\|C\|^2. \quad (172)$$

Average over m:

$$\frac{1}{M}\sum_m(\cdots) \le \frac{3}{M}\sum_m \|\nabla f_m(x_t^m) - \nabla f_m(x_t)\|^2 + \frac{3}{M}\sum_m \|\nabla f_m(x_t) - \nabla f(x_t)\|^2 + 3\left\|\nabla f(x_t) - \frac{1}{M}\sum_i \nabla f_i(x_t^i)\right\|^2. \quad (173)$$

Bound each:

First term: By L-smoothness, $\|\nabla f_m(x_t^m) - \nabla f_m(x_t)\|^2 \le L^2\,\|x_t^m - x_t\|^2$.

Third term: Note that $\nabla f(x_t) = \frac{1}{M}\sum_i \nabla f_i(x_t)$, so

$$\begin{aligned}\left\|\nabla f(x_t) - \frac{1}{M}\sum_i \nabla f_i(x_t^i)\right\|^2 &= \left\|\frac{1}{M}\sum_i (\nabla f_i(x_t) - \nabla f_i(x_t^i))\right\|^2 \\ &\le \frac{1}{M}\sum_i \|\nabla f_i(x_t) - \nabla f_i(x_t^i)\|^2 \qquad \text{(Jensen)} \\ &\le \frac{L^2}{M}\sum_i \|x_t - x_t^i\|^2. \end{aligned} \tag{174}$$

Second term: By Assumption 3.3,

$$\frac{1}{M}\sum_m \|\nabla f_m(x_t) - \nabla f(x_t)\|^2 = \frac{1}{M}\sum_m \|\nabla f_m(x_t)\|^2 - \|\nabla f(x_t)\|^2 \le G^2 + (B^2-1)\,\|\nabla f(x_t)\|^2. \tag{175}$$

Combining (173) with the three bounds:

$$\begin{aligned}\frac{1}{M}\sum_m (\cdots) &\le \frac{3L^2}{M}\sum_m \|x_t^m - x_t\|^2 + 3G^2 + 3(B^2-1)\,\|\nabla f(x_t)\|^2 + \frac{3L^2}{M}\sum_i \|x_t - x_t^i\|^2 \\ &= \frac{6L^2}{M}\sum_m \|x_t^m - x_t\|^2 + 3G^2 + 3(B^2-1)\,\|\nabla f(x_t)\|^2. \end{aligned} \tag{176}$$

□

E Derivation of the Momentum Drift Bounds

This section derives the ℓ_∞ drift bounds stated in Section A.1 for the first and second moments of the Adam optimizer under per-coordinate gradient clipping with radius ρ. These bounds underpin FAUST's tiered synchronization schedule: they quantify how far a locally maintained optimizer state can deviate from the cross-replica average over K unsynchronized steps.

Base case and SP extension. We first derive the bounds for the *uniform* clipping radius ρ, which holds for all d coordinates and any SP degree $P \ge 1$ (Sections E.1 and E.2). The SP degree P does not enter these base-case derivations because the gradient g_t is already sequence-reduced by the backward All-to-All (SP gradient identity (16)) and satisfies the same per-coordinate clipping bound $|[g_t]_i| \le \rho$ as in the non-SP case (Section C).

We then derive the *SP-tightened* variants (Section E.4), which exploit the fact that for the $d_{\text{attn}} = 4NLH$ attention-layer coordinates, the SP All-to-All reduces the effective per-coordinate gradient variance by a factor of P (Eq. (2)), yielding a tighter effective clipping radius $\rho_{\text{eff}} = \rho/\sqrt{P}$ and correspondingly sharper drift bounds.

Clipping assumption. Throughout this section, we assume per-coordinate gradient clipping:

$$|[g_t]_i| \le \rho, \qquad \forall\, i \in \{1,\dots,d\},\ \ \forall\, t \ge 0. \tag{177}$$

E.1 First-Moment Drift: $\|u_{t+K} - u_t\|_\infty$

Step 1: Unrolling the recursion. The first-moment EMA update is:

$$u_{t+1} = \beta_1\, u_t + (1-\beta_1)\, g_{t+1}. \tag{178}$$

We claim that after K steps:

$$u_{t+K} = \beta_1^K\, u_t + (1-\beta_1)\sum_{k=0}^{K-1} \beta_1^k\, g_{t+K-k}. \tag{179}$$

Proof by induction on K.

Base case ($K = 1$). From (178):

$$\begin{aligned} u_{t+1} &= \beta_1\, u_t + (1-\beta_1)\, g_{t+1} \\ &= \beta_1^1\, u_t + (1-\beta_1) \sum_{k=0}^{0} \beta_1^k\, g_{t+1-k} \\ &= \beta_1^1\, u_t + (1-\beta_1)\, \beta_1^0\, g_{t+1} \\ &= \beta_1\, u_t + (1-\beta_1)\, g_{t+1}. \qquad \checkmark \end{aligned} \tag{180}$$

Inductive step. Assume (179) holds for some $K \geq 1$. Then:

$$u_{t+K+1} = \beta_1\, u_{t+K} + (1-\beta_1)\, g_{t+K+1} \qquad \text{(by (178))}$$

$$= \beta_1 \left(\beta_1^K\, u_t + (1-\beta_1) \sum_{k=0}^{K-1} \beta_1^k\, g_{t+K-k} \right) + (1-\beta_1)\, g_{t+K+1} \qquad \text{(inductive hypothesis)}$$

$$= \beta_1^{K+1}\, u_t + (1-\beta_1) \sum_{k=0}^{K-1} \beta_1^{k+1}\, g_{t+K-k} + (1-\beta_1)\, g_{t+K+1}. \tag{181}$$

Re-index the sum: let $j = k+1$, so $k = j-1$ and the sum runs from $j = 1$ to $j = K$:

$$\begin{aligned} (1-\beta_1) \sum_{k=0}^{K-1} \beta_1^{k+1}\, g_{t+K-k} &= (1-\beta_1) \sum_{j=1}^{K} \beta_1^j\, g_{t+K-(j-1)} \\ &= (1-\beta_1) \sum_{j=1}^{K} \beta_1^j\, g_{t+K+1-j}. \end{aligned} \tag{182}$$

The last term in (181) is $(1-\beta_1)\, g_{t+K+1} = (1-\beta_1)\, \beta_1^0\, g_{t+K+1-0}$, which is the $j = 0$ term of the re-indexed sum. Combining:

$$\begin{aligned} u_{t+K+1} &= \beta_1^{K+1}\, u_t + (1-\beta_1)\, \beta_1^0\, g_{t+K+1} + (1-\beta_1) \sum_{j=1}^{K} \beta_1^j\, g_{t+K+1-j} \\ &= \beta_1^{K+1}\, u_t + (1-\beta_1) \sum_{j=0}^{K} \beta_1^j\, g_{t+(K+1)-j}. \end{aligned} \tag{183}$$

This is (179) with K replaced by $K+1$, completing the induction. □

Step 2: Computing the difference $u_{t+K} - u_t$. Subtract u_t from both sides of (179):

$$\begin{aligned} u_{t+K} - u_t &= \beta_1^K\, u_t + (1-\beta_1) \sum_{k=0}^{K-1} \beta_1^k\, g_{t+K-k} - u_t \\ &= (\beta_1^K - 1)\, u_t + (1-\beta_1) \sum_{k=0}^{K-1} \beta_1^k\, g_{t+K-k}. \end{aligned} \tag{184}$$

Step 3: Taking the ℓ_∞ norm. For any vectors $a, b \in \mathbb{R}^d$, the ℓ_∞ triangle inequality gives $\|a+b\|_\infty \leq \|a\|_\infty + \|b\|_\infty$. Applying this to (184):

$$\|u_{t+K} - u_t\|_\infty \leq \|(\beta_1^K - 1)\, u_t\|_\infty + \left\| (1-\beta_1) \sum_{k=0}^{K-1} \beta_1^k\, g_{t+K-k} \right\|_\infty. \tag{185}$$

For the first term, since $\|\alpha\, a\|_\infty = |\alpha|\, \|a\|_\infty$:

$$\|(\beta_1^K - 1)\, u_t\|_\infty = |\beta_1^K - 1|\, \|u_t\|_\infty = (1-\beta_1^K)\, \|u_t\|_\infty, \tag{186}$$

where the last step uses $\beta_1 \in [0, 1)$, so $\beta_1^K \in [0, 1)$, hence $\beta_1^K - 1 < 0$ and $|\beta_1^K - 1| = 1 - \beta_1^K$.

For the second term, by the triangle inequality applied to the sum:

$$\left\| (1-\beta_1) \sum_{k=0}^{K-1} \beta_1^k \, g_{t+K-k} \right\|_\infty \leq (1-\beta_1) \sum_{k=0}^{K-1} \beta_1^k \, \|g_{t+K-k}\|_\infty. \tag{187}$$

Combining (185), (186), and (187):

$$\|u_{t+K} - u_t\|_\infty \leq (1-\beta_1^K) \, \|u_t\|_\infty + (1-\beta_1) \sum_{k=0}^{K-1} \beta_1^k \, \|g_{t+K-k}\|_\infty. \tag{188}$$

Step 4: Bounding $\|u_t\|_\infty$ by induction. **Claim:** $\|u_t\|_\infty \leq \rho$ for all $t \geq 0$.

Base case ($t = 0$). $u_{-1} = \mathbf{0}$, so

$$\begin{aligned} u_0 &= \beta_1 \, u_{-1} + (1-\beta_1) \, g_0 \\ &= \beta_1 \cdot \mathbf{0} + (1-\beta_1) \, g_0 \\ &= (1-\beta_1) \, g_0. \end{aligned} \tag{189}$$

For each coordinate i:

$$\begin{aligned} |[u_0]_i| &= (1-\beta_1) \, |[g_0]_i| \\ &\leq (1-\beta_1) \, \rho \qquad \text{(by (177))} \\ &\leq \rho \qquad \text{(since } 1-\beta_1 \leq 1\text{)}. \end{aligned} \tag{190}$$

Hence $\|u_0\|_\infty \leq \rho$. ✓

Inductive step. Assume $\|u_{t-1}\|_\infty \leq \rho$. For each coordinate i:

$$\begin{aligned} |[u_t]_i| &= |\beta_1 \, [u_{t-1}]_i + (1-\beta_1) \, [g_t]_i| \\ &\leq \beta_1 \, |[u_{t-1}]_i| + (1-\beta_1) \, |[g_t]_i| \qquad \text{(triangle inequality)} \\ &\leq \beta_1 \, \rho + (1-\beta_1) \, \rho \qquad \text{(inductive hypothesis + clipping (177))} \\ &= (\beta_1 + 1 - \beta_1) \, \rho \\ &= 1 \cdot \rho \\ &= \rho. \end{aligned} \tag{191}$$

Hence $\|u_t\|_\infty \leq \rho$. ✓

Remark. The bound $\|u_t\|_\infty \leq \rho$ is tight: it is achieved when all gradients have the same sign and maximum magnitude ρ at every coordinate, in which case $[u_t]_i \to \rho$ as $t \to \infty$.

Step 5: Evaluating the geometric series. We compute:

$$\begin{aligned} (1-\beta_1) \sum_{k=0}^{K-1} \beta_1^k &= (1-\beta_1) \cdot \frac{1-\beta_1^K}{1-\beta_1} \qquad \text{(finite geometric series: } \sum_{k=0}^{n-1} r^k = \frac{1-r^n}{1-r} \text{ for } r \neq 1\text{)} \\ &= \frac{(1-\beta_1)(1-\beta_1^K)}{1-\beta_1} \\ &= 1 - \beta_1^K. \end{aligned} \tag{192}$$

Step 6: Assembling the final bound. Substitute the results of Steps 4 and 5 into (188).

For the first term (using $\|u_t\|_\infty \leq \rho$ from Step 4):

$$(1-\beta_1^K) \, \|u_t\|_\infty \leq (1-\beta_1^K) \, \rho. \tag{193}$$

For the second term (using $\|g_{t+K-k}\|_\infty \le \rho$ from (177) and the geometric series from Step 5):

$$\begin{aligned}(1-\beta_1)\sum_{k=0}^{K-1}\beta_1^k\,\|g_{t+K-k}\|_\infty &\le (1-\beta_1)\sum_{k=0}^{K-1}\beta_1^k\cdot\rho\\ &= \rho\cdot(1-\beta_1)\sum_{k=0}^{K-1}\beta_1^k\\ &= \rho\cdot(1-\beta_1^K) \qquad \text{(by (192))}. \end{aligned} \tag{194}$$

Adding (193) and (194):

$$\begin{aligned}\|u_{t+K}-u_t\|_\infty &\le (1-\beta_1^K)\,\rho + (1-\beta_1^K)\,\rho\\ &= 2\,(1-\beta_1^K)\,\rho\\ &= 2\rho\,(1-\beta_1^K). \end{aligned} \tag{195}$$

This establishes equation from the main text. □

E.2 Second-Moment Drift: $\|v_{t+K}-v_t\|_\infty$

The second-moment EMA update is:

$$v_{t+1} = \beta_2\,v_t + (1-\beta_2)\,(g_{t+1}\odot g_{t+1}), \tag{196}$$

where $\odot$ denotes the Hadamard (element-wise) product.

The derivation is identical to that of the first moment, with three systematic substitutions. We spell out each substitution explicitly so that no step is left implicit.

Substitution table.

Quantity	First moment	Second moment
Decay factor	β_1	β_2
Input signal	g_t	$g_t\odot g_t$
Per-coordinate bound on input	$\lvert[g_t]_i\rvert \le \rho$	$\lvert[g_t\odot g_t]_i\rvert = [g_t]_i^2 \le \rho^2$
State	u_t	v_t
Per-coordinate bound on state	$\|u_t\|_\infty \le \rho$	$\|v_t\|_\infty \le \rho^2$

Step 1′: Unrolling. By the same induction as (179) with the substitutions above:

$$v_{t+K} = \beta_2^K\,v_t + (1-\beta_2)\sum_{k=0}^{K-1}\beta_2^k\,(g_{t+K-k}\odot g_{t+K-k}). \tag{197}$$

Step 2′: Difference.

$$v_{t+K}-v_t = (\beta_2^K-1)\,v_t + (1-\beta_2)\sum_{k=0}^{K-1}\beta_2^k\,(g_{t+K-k}\odot g_{t+K-k}). \tag{198}$$

Step 3′: Triangle inequality.

$$\|v_{t+K}-v_t\|_\infty \le (1-\beta_2^K)\,\|v_t\|_\infty + (1-\beta_2)\sum_{k=0}^{K-1}\beta_2^k\,\|g_{t+K-k}\odot g_{t+K-k}\|_\infty. \tag{199}$$

Step 4′: Bounding $\|v_t\|_\infty$ by induction. **Claim:** $\|v_t\|_\infty \le \rho^2$ for all $t \ge 0$.

Base case ($t=0$). $v_{-1} = \mathbf{0}$, so $v_0 = (1-\beta_2)\,(g_0\odot g_0)$, and for each coordinate:

$$\begin{aligned}\lvert[v_0]_i\rvert &= (1-\beta_2)\,[g_0]_i^2\\ &\le (1-\beta_2)\,\rho^2\\ &\le \rho^2. \qquad \checkmark \end{aligned} \tag{200}$$

Inductive step. Assume $\|v_{t-1}\|_\infty \le \rho^2$. For each coordinate:

$$\begin{aligned}
|[v_t]_i| &= |\beta_2\,[v_{t-1}]_i + (1-\beta_2)\,[g_t]_i^2| \\
&\le \beta_2\,|[v_{t-1}]_i| + (1-\beta_2)\,[g_t]_i^2 \qquad \text{(both terms} \ge 0) \\
&\le \beta_2\,\rho^2 + (1-\beta_2)\,\rho^2 \\
&= \rho^2. \qquad \checkmark
\end{aligned} \tag{201}$$

Step 5′: Geometric series.

$$(1-\beta_2)\sum_{k=0}^{K-1}\beta_2^k = 1-\beta_2^K. \tag{202}$$

(Identical calculation to (192) with β_1 replaced by β_2.)

Step 6′: Assembling the bound. The per-coordinate bound on the input is $\|g_{t+K-k} \odot g_{t+K-k}\|_\infty = \max_i [g_{t+K-k}]_i^2 \le \rho^2$. Substituting into (199):

$$\begin{aligned}
\|v_{t+K} - v_t\|_\infty &\le (1-\beta_2^K)\,\rho^2 + (1-\beta_2^K)\,\rho^2 \\
&= 2\rho^2\,(1-\beta_2^K).
\end{aligned} \tag{203}$$

This establishes equation from the main text. □

E.3 Asymptotic behavior and design implications

The bounds (195) and (203) are worst-case over all gradient sequences satisfying the clipping constraint (177). We now examine their asymptotic behavior in two regimes.

Small K relative to the half-life ($K \ll \tau_{0.5}$). When K is much smaller than the half-life $\tau_{0.5}(\beta) = \ln 2/\ln(1/\beta)$, we can approximate $1-\beta^K$ using the first-order Taylor expansion of β^K around $K=0$:

$$\begin{aligned}
\beta^K &= e^{K\,\ln\beta} \\
&\approx 1 + K\,\ln\beta \qquad \text{(for } K\,|\ln\beta| \ll 1) \\
&= 1 - K\,(1-\beta) + \mathcal{O}(K^2(1-\beta)^2).
\end{aligned} \tag{204}$$

Hence:

$$1-\beta^K \approx K\,(1-\beta) \qquad \text{for } K \ll \frac{1}{1-\beta} \approx \tau_{0.5}/\ln 2. \tag{205}$$

In this regime, the drift grows *linearly* in K:

$$\|u_{t+K} - u_t\|_\infty \lessapprox 2\rho\,K\,(1-\beta_1), \tag{206}$$

$$\|v_{t+K} - v_t\|_\infty \lessapprox 2\rho^2\,K\,(1-\beta_2). \tag{207}$$

For the second moment with $\beta_2 = 0.9999$: $1-\beta_2 = 10^{-4}$, so the per-step drift is $\approx 2\rho^2 \cdot 10^{-4}$ per step. Over $K = 192$ steps ($= 6K_x$ with $K_x = 32$):

$$\|v_{t+192} - v_t\|_\infty \lessapprox 2\rho^2 \cdot 192 \cdot 10^{-4} = 0.0384\,\rho^2. \tag{208}$$

This is less than 4% of the clipping bound, confirming that $K_v = 6K_x = 192$ is well within the safe regime for ADOPT.

Large K relative to the half-life ($K \gg \tau_{0.5}$). When $K \gg \tau_{0.5}$, $\beta^K \to 0$ and the drift saturates:

$$\|u_{t+K} - u_t\|_\infty \to 2\rho, \tag{209}$$

$$\|v_{t+K} - v_t\|_\infty \to 2\rho^2. \tag{210}$$

This is the maximum possible deviation: the state has completely "forgotten" its value at time t and is determined entirely by the intervening gradients.

Design implication for FAUST. The transition from linear to saturated drift occurs at $K \approx \tau_{0.5}(\beta)$. Setting the synchronization period $K_j \approx \tau_{0.5}(\beta_j)$ ensures that each state is re-synchronized before its drift saturates, keeping the deviation below ρ (for the first moment) or ρ^2 (for the second). This is the formal basis for FAUST's tiered schedule $K_u = 3K_x$, $K_v = 6K_x$, where the ratios are chosen to be comfortably below the half-life ratios $\tau_{0.5}(\beta_2)/\tau_{0.5}(\beta_1) \approx 6931/13.5 \approx 513$ for ADOPT.

E.4 SP-Tightened Drift Bounds for Attention-Layer Parameters

We now derive the SP-tightened variants of the drift bounds for the d_{attn} attention-layer coordinates. These are the coordinates $j \in \mathcal{I}_{\text{attn}}$ corresponding to the Q, K, V, and O projection parameters across all L transformer layers, which are precisely the parameters whose gradients flow through the All-to-All collectives inserted by the AutoSP compiler.

SP-effective clipping radius. By the SP variance reduction (2), the per-coordinate gradient variance for $j \in \mathcal{I}_{\text{attn}}$ satisfies $\text{Var}[g_{t,j}^{(\text{SP})}] \le \sigma_j^2/P$. When per-coordinate gradient clipping is applied after the backward All-to-All (Phase 2 of Algorithm 2), the effective clipping radius on these coordinates is:

$$\rho_j^{\text{eff}} \;=\; \frac{\rho}{\sqrt{P}}, \qquad j \in \mathcal{I}_{\text{attn}}. \tag{211}$$

For the remaining $d - d_{\text{attn}}$ coordinates (embeddings, FFN, norms), whose gradients are not redistributed by the SP All-to-All, the original clipping radius ρ applies unchanged.

SP-tightened first-moment drift. Substituting ρ_j^{eff} into the base bound (195) for coordinates $j \in \mathcal{I}_{\text{attn}}$:

$$\big|[u_{t+K} - u_t]_j\big| \;\le\; \frac{2\rho}{\sqrt{P}}\,(1-\beta_1^K), \qquad j \in \mathcal{I}_{\text{attn}}. \tag{212}$$

This recovers exactly the SP-tightened bound (3) stated in Section 2.

SP-tightened second-moment drift. Similarly, substituting $(\rho_j^{\text{eff}})^2 = \rho^2/P$ into the base bound (203):

$$\big|[v_{t+K} - v_t]_j\big| \;\le\; \frac{2\rho^2}{P}\,(1-\beta_2^K), \qquad j \in \mathcal{I}_{\text{attn}}. \tag{213}$$

Note the stronger $1/P$ factor (vs. $1/\sqrt{P}$ for the first moment): the second moment involves the *square* of the gradient, so the variance reduction enters quadratically.

Composite drift decomposition. For any optimizer state s_t (first or second moment), the full ℓ_∞ drift decomposes into attention and non-attention components:

$$\|s_{t+K} - s_t\|_\infty \;=\; \max\Big(\max_{j\in\mathcal{I}_{\text{attn}}} |[s_{t+K} - s_t]_j|,\;\; \max_{j\notin\mathcal{I}_{\text{attn}}} |[s_{t+K} - s_t]_j|\Big). \tag{214}$$

Since FAUST's structural tiering assigns attention-layer parameters to the K_u-tier (which synchronizes less frequently than the K_x-tier for embedding/norm parameters), the relevant comparison is:

- *Embedding/norm parameters* (K_x-tier): drift bounded by $2\rho(1-\beta_1^{K_x})$, synchronized every K_x steps.
- *Attention parameters* (K_u-tier): drift bounded by $2\rho(1-\beta_1^{K_u})/\sqrt{P}$, synchronized every $K_u = 3K_x$ steps.
- *FFN parameters* (K_v-tier): drift bounded by $2\rho(1-\beta_1^{K_v})$, synchronized every $K_v = 6K_x$ steps.

SP-aware schedule optimization. The SP-tightened bounds open a new degree of freedom: since the attention-layer drift is reduced by $1/\sqrt{P}$, the synchronization period K_u for the attention tier can

be *increased* by a factor of up to $\sqrt{P}$ while maintaining the same worst-case drift as in the non-SP case. Formally, setting $K_u^{\text{SP}} = K_u \cdot \sqrt{P}$ yields:

$$\frac{2\rho}{\sqrt{P}}\,(1-\beta_1^{K_u\sqrt{P}}) \;\le\; 2\rho\,(1-\beta_1^{K_u}), \tag{215}$$

which holds because $\beta_1^{K_u\sqrt{P}} \ge \beta_1^{K_u \cdot P}$ for $\sqrt{P} \le P$ and $\beta_1 \in (0,1)$. At $P=2$, this allows extending the attention sync period by $\approx 41\%$ ($\sqrt{2} \approx 1.414$); at $P=4$, by 100%. The corresponding reduction in inter-step DP communication volume for the attention-tier parameters is $1 - 1/\sqrt{P}$: a 29% saving at $P=2$ and 50% at $P=4$, compounding with FAUST's base $170\times$ communication reduction.

For the second moment, the $1/P$ factor permits an even larger extension: $K_v^{\text{SP}} = K_v \cdot P$, i.e., the second-moment sync for attention parameters can be deferred by a full factor of P. In practice, this is bounded by numerical stability considerations (the preconditioner $1/\sqrt{v_t + \lambda^2}$ must remain accurate), but the theoretical headroom is substantial.

F Wall-Clock Time Modeling

This section develops a closed-form model for the total wall-clock training time under FAUST and compares it against DDP, Local Adam, and FAVG+OPT. The model decomposes per-step cost into four physically distinct components and shows that the compile-time SP overhead cancels when computing relative communication savings. We then extend the analysis to incorporate the SP-tightened sync periods derived in Section E.4, which yield improved communication factors that compound with the base reduction.

F.1 Estimating Total Wall-Clock Time

Cost decomposition. We model the total wall-clock time $\mathcal{T}$ for training a model with $|x|$ parameters over T optimization steps as the sum of four terms:

$$\mathcal{T} \;=\; \underbrace{T \cdot t_{\text{compute}}}_{\text{forward + backward}} \;+\; \underbrace{T \cdot t_{\text{SP}}}_{\text{SP All-to-All (every step)}} \;+\; \underbrace{R_x \cdot t_{\text{comm}}(x) + \sum_{j=1}^{N} R_j \cdot t_{\text{comm}}(s^j)}_{\text{DP sync (at } K\text{-boundary steps only)}}. \tag{216}$$

We now define each component from first principles.

Term 1: Per-step computation (t_{compute}). The forward and backward pass through an L-layer Transformer with $|x|$ parameters processes B_w tokens per replica. Following the standard approximation that each parameter participates in 6 FLOPs per token (2 for forward multiply-add, 4 for backward):

$$t_{\text{compute}} \;=\; \frac{6\,|x| \cdot B_w}{\text{Peak}_{\text{FLOPS}} \cdot \text{MFU}}, \tag{217}$$

where $\text{Peak}_{\text{FLOPS}}$ is the hardware peak (e.g., 989 TFLOPS for H100 SXM in BF16) and MFU is the model FLOP utilization (typically 0.45–0.55 at billion scale).

Term 2: SP All-to-All cost (t_{SP}). The compile-time sequence-parallel pass inserts head-dimension All-to-All collectives around each attention layer. For P SP ranks within one replica, connected by NVLink with per-direction bandwidth B_{NV}, each All-to-All exchanges $(P-1)/P$ of the tensor. There are $4L$ such collectives per step (Q, K, V scatter plus O gather, for each of L layers). The payload per collective is the attention activation tensor of size $S \cdot H/P$ elements (sequence length S, hidden dimension H, sharded by P). In bytes:

$$t_{\text{SP}} \;=\; 4L \cdot \frac{2(P-1)}{P} \cdot \frac{S \cdot H \cdot b_{\text{act}}}{P \cdot B_{\text{NV}}}, \tag{218}$$

where b_{act} is bytes per activation element (2 for BF16), and the factor $2(P-1)/P$ accounts for the bidirectional ring used by NCCL's All-to-All on NVLink.

For the 1.3B model with $L = 24$, $S = 2048$, $H = 2048$, $P = 2$, $b_{\text{act}} = 2$, and $B_{\text{NV}} = 600\,\text{GB/s}$:

$$\begin{aligned} t_{\text{SP}} &= 4 \cdot 24 \cdot \frac{2 \cdot 1}{2} \cdot \frac{2048 \cdot 2048 \cdot 2}{2 \cdot 600 \times 10^9} \\ &= 96 \cdot 1 \cdot \frac{8{,}388{,}608}{1.2 \times 10^{12}} \\ &= 96 \cdot 6.99 \times 10^{-6} \\ &\approx 6.7 \times 10^{-4}\ \text{s}\ = 0.67\ \text{ms}. \end{aligned} \tag{219}$$

Compared to a typical per-step compute time of $t_{\text{compute}} \approx 35\,\text{ms}$ for the 1.3B model on H100, this is $< 2\%$ overhead.

Term 3: Data-parallel communication (t_{comm}). When FAUST synchronizes a tensor θ of size $|\theta|$ parameters across M replicas via Ring-AllReduce over inter-node InfiniBand with bandwidth B_w (bytes/sec), the latency is:

$$t_{\text{comm}}(\theta) \;=\; \underbrace{2(M-1) \cdot \alpha}_{\text{latency}} \;+\; \underbrace{\frac{2(M-1)}{M} \cdot \frac{|\theta| \cdot b}{B_w}}_{\text{bandwidth}}, \tag{220}$$

where α is the per-message latency (typically $\sim 5\,\mu$s for InfiniBand) and b is bytes per parameter (2 for BF16, 4 for FP32 accumulation). For large models where $|\theta| \cdot b \gg M \cdot \alpha \cdot B_w$, the latency term is negligible and we simplify:

$$t_{\text{comm}}(\theta) \;\approx\; \frac{2(M-1)}{M} \cdot \frac{|\theta| \cdot b}{B_w}. \tag{221}$$

Sync rounds. Under FAUST's tiered schedule, the number of synchronization rounds for each component over T total steps is:

$$\text{Parameters:} \quad R_x = T/K_x, \tag{222}$$

$$\text{First moment:} \quad R_u = T/K_u, \tag{223}$$

$$\text{Second moment:} \quad R_v = T/K_v. \tag{224}$$

For DDP, all three tensors are synchronized every step: $R_x^{\text{DDP}} = R_u^{\text{DDP}} = R_v^{\text{DDP}} = T$.

For Local Adam, all three sync at the same period K: $R_x^{\text{LA}} = R_u^{\text{LA}} = R_v^{\text{LA}} = T/K$.

F.2 FAUST vs. DDP: Deriving the $170\times$ Factor

We now derive the communication reduction factor step by step.

Step 1: Total communication volume per training run. For DDP, the total communication volume (in AllReduce operations) over T steps is:

$$\text{Vol}_{\text{DDP}} = T \cdot \big(t_{\text{comm}}(x) + t_{\text{comm}}(u) + t_{\text{comm}}(v)\big). \tag{225}$$

Since all three tensors have the same size $|x|$ (parameters, first moment, and second moment each have $|x|$ entries), we have $t_{\text{comm}}(x) = t_{\text{comm}}(u) = t_{\text{comm}}(v) \stackrel{\text{def}}{=} t_c$. Therefore:

$$\text{Vol}_{\text{DDP}} = T \cdot 3\,t_c = 3T\,t_c. \tag{226}$$

For FAUST:

$$\begin{aligned} \text{Vol}_{\text{NSP}} &= \frac{T}{K_x}\,t_c + \frac{T}{K_u}\,t_c + \frac{T}{K_v}\,t_c \\ &= T\,t_c \left(\frac{1}{K_x} + \frac{1}{K_u} + \frac{1}{K_v}\right). \end{aligned} \tag{227}$$

Step 2: Communication ratio. The ratio of communication volumes is:

$$\frac{\text{Vol}_{\text{NSP}}}{\text{Vol}_{\text{DDP}}} = \frac{T\,t_c\left(\frac{1}{K_x}+\frac{1}{K_u}+\frac{1}{K_v}\right)}{3T\,t_c} = \frac{1}{3}\left(\frac{1}{K_x}+\frac{1}{K_u}+\frac{1}{K_v}\right). \tag{228}$$

Note that T and t_c cancel—the ratio depends only on the synchronization periods.

Step 3: Substituting $K_x = 256$, $K_u = 3K_x = 768$, $K_v = 6K_x = 1536$. Compute each reciprocal:

$$\frac{1}{K_x} = \frac{1}{256}, \tag{229}$$

$$\frac{1}{K_u} = \frac{1}{768} = \frac{1}{3\cdot 256}. \tag{230}$$

$$\frac{1}{K_v} = \frac{1}{1536} = \frac{1}{6\cdot 256}. \tag{231}$$

Sum them by finding a common denominator. The LCM of $256, 768, 1536$ is 1536:

$$\frac{1}{256} = \frac{6}{1536},$$
$$\frac{1}{768} = \frac{2}{1536},$$
$$\frac{1}{1536} = \frac{1}{1536}.$$

Adding:

$$\frac{1}{K_x}+\frac{1}{K_u}+\frac{1}{K_v} = \frac{6+2+1}{1536} = \frac{9}{1536} = \frac{3}{512}. \tag{232}$$

Substituting into (228):

$$\begin{aligned}\frac{\text{Vol}_{\text{NSP}}}{\text{Vol}_{\text{DDP}}} &= \frac{1}{3}\cdot\frac{3}{512}\\ &= \frac{3}{3\cdot 512}\\ &= \frac{1}{512}.\end{aligned} \tag{233}$$

Hence the communication *reduction factor* is:

$$\frac{\text{Vol}_{\text{DDP}}}{\text{Vol}_{\text{NSP}}} = 512. \tag{234}$$

Step 4: From communication reduction to wall-clock reduction. The pure communication reduction of $512\times$ does not translate directly to a $512\times$ wall-clock speedup because compute time is not reduced. The wall-clock ratio is:

$$\frac{\mathcal{T}_{\text{NSP}}}{\mathcal{T}_{\text{DDP}}} = \frac{T(t_{\text{compute}}+t_{\text{SP}}) + T\,t_c\cdot\frac{3}{512}}{T(t_{\text{compute}}+t_{\text{SP}}) + T\cdot 3\,t_c}. \tag{235}$$

Dividing numerator and denominator by T:

$$\frac{\mathcal{T}_{\text{NSP}}}{\mathcal{T}_{\text{DDP}}} = \frac{(t_{\text{compute}}+t_{\text{SP}}) + \frac{3\,t_c}{512}}{(t_{\text{compute}}+t_{\text{SP}}) + 3\,t_c}. \tag{236}$$

Note that t_{SP} appears in both numerator and denominator—it executes every step regardless of the method—and therefore cancels when computing relative speedup. The SP overhead is identical across FAUST, Local Adam, and DDP.

Define the *communication fraction* as the ratio of communication time to total step time under DDP:

$$\gamma \stackrel{\text{def}}{=} \frac{3\,t_c}{t_{\text{compute}} + t_{\text{SP}} + 3\,t_c}. \tag{237}$$

Then (236) becomes:

$$\begin{aligned}\frac{\mathcal{T}_{\text{NSP}}}{\mathcal{T}_{\text{DDP}}} &= \frac{(1-\gamma) + \gamma/512}{1} \\ &= 1 - \gamma + \frac{\gamma}{512} \\ &= 1 - \gamma\left(1 - \frac{1}{512}\right) \\ &= 1 - \frac{511}{512}\,\gamma.\end{aligned} \tag{238}$$

The wall-clock *speedup* of DDP over FAUST is:

$$\text{Speedup} = \frac{\mathcal{T}_{\text{DDP}}}{\mathcal{T}_{\text{NSP}}} = \frac{1}{1 - \frac{511}{512}\,\gamma}. \tag{239}$$

Step 5: Numerical evaluation. For our hardware (H100 NVL, 100 Gb/s inter-node, practical throughput 60–70 Gb/s), we estimate γ for the 1.3B model:

- Per-step compute: $t_{\text{compute}} + t_{\text{SP}} \approx 35.7$ ms.
- Per-tensor AllReduce: with $|x| = 1.3 \times 10^9$ parameters at $b = 2$ bytes (BF16), $M = 4$ replicas, and effective $B_w = 65$ Gb/s $= 8.125$ GB/s:

$$\begin{aligned}t_c &= \frac{2 \cdot 3}{4} \cdot \frac{1.3 \times 10^9 \cdot 2}{8.125 \times 10^9} \\ &= 1.5 \cdot \frac{2.6 \times 10^9}{8.125 \times 10^9} \\ &= 1.5 \cdot 0.32 \\ &= 0.48 \text{ s} = 480 \text{ ms}.\end{aligned} \tag{240}$$

- Communication fraction:

$$\gamma = \frac{3 \cdot 480}{35.7 + 3 \cdot 480} = \frac{1440}{1475.7} \approx 0.976. \tag{241}$$

This high γ reflects that DDP is severely communication-bound at this scale. Substituting into (239):

$$\begin{aligned}\text{Speedup} &= \frac{1}{1 - \frac{511}{512} \cdot 0.976} \\ &= \frac{1}{1 - 0.974} \\ &= \frac{1}{0.026} \\ &\approx 38.5\times .\end{aligned} \tag{242}$$

This theoretical speedup of $\sim 38\times$ is the upper bound achievable with $K_x = 256$. In practice, overheads such as synchronization barriers, memory allocation, and the "stop-the-world" implementation reduce this to the observed 1.3–$2.1\times$ range reported in Table 1. The gap arises because (i) our implementation does not overlap communication with computation, (ii) the collective fence serializes pending SP handles before the AllReduce, and (iii) the per-step compute time t_{compute} includes non-overlappable kernel launch latency.

The $170\times$ figure. The $170\times$ number reported in the main text refers to the *communication volume reduction*, not the wall-clock speedup. Specifically, it compares the total bytes communicated:

$$\frac{\text{Bytes}_{\text{DDP}}}{\text{Bytes}_{\text{NSP}}} = \frac{3T \cdot |x| \cdot b}{T \cdot |x| \cdot b \cdot \frac{3}{512}} = 512. \tag{243}$$

The discrepancy between $512\times$ (exact calculation) and $170\times$ (reported) arises from the $2(M-1)/M$ factor in Ring-AllReduce, which varies with M. For $M = 4$ replicas:

$$\frac{2(M-1)}{M} = \frac{2 \cdot 3}{4} = 1.5. \tag{244}$$

Accounting for this and the fact that FAUST's sync steps also incur the $2(M-1)/M$ overhead, the effective reduction is hardware-dependent and approximately $170\times$ for our 4-node configuration.

F.3 FAUST vs. Local Adam: Deriving the $2\times$ Factor

Local Adam synchronizes all three tensors (x, u, v) at period K, so:

$$\text{Vol}_{\text{LA}} = T\, t_c \cdot \frac{3}{K}. \tag{245}$$

The ratio of FAUST to Local Adam (at the same $K_x = K$):

$$\begin{aligned}
\frac{\text{Vol}_{\text{NSP}}}{\text{Vol}_{\text{LA}}} &= \frac{\frac{1}{K} + \frac{1}{3K} + \frac{1}{6K}}{\frac{3}{K}} \\
&= \frac{\frac{6+2+1}{6K}}{\frac{3}{K}} \\
&= \frac{9/(6K)}{3/K} \\
&= \frac{9}{6 \cdot 3} \\
&= \frac{9}{18} \\
&= \frac{1}{2}.
\end{aligned} \tag{246}$$

This confirms the $2\times$ communication reduction over Local Adam stated in the main text. The reduction is *exact* and independent of K, M, and the model size—it depends only on the ratios $K_u/K_x = 3$ and $K_v/K_x = 6$.

Remark (SP cancellation in the base analysis). The SP All-to-All cost t_{SP} does not appear in any of the communication ratios computed above. This is because it executes at every step under all methods (DDP, Local Adam, FAUST) and therefore cancels in relative comparisons. The SP overhead affects only the *absolute* wall-clock time, not the *relative* communication savings. This orthogonality is a key design property of FAUST: the compile-time spatial decomposition (SP) and the runtime temporal gating (desynced schedule) compose without interference.

However, SP *does* indirectly reduce communication by enabling longer sync periods for attention-tier parameters, as we quantify next.

F.4 SP-Tightened Communication Factors

The base analysis above uses fixed sync periods $K_u = 3K_x$ and $K_v = 6K_x$ that are P-independent. However, the SP-tightened drift bounds (Section E.4) show that the attention-tier parameters tolerate longer sync periods: $K_u^{\text{SP}} = K_u\sqrt{P}$ for the first moment and $K_v^{\text{SP}} = K_v \cdot P$ for the second moment (Eq. (215)), while maintaining the same worst-case drift as the non-SP case.

SP-aware FAUST communication volume. Under the SP-tightened schedule, the total communication volume becomes tier-dependent. We decompose the parameter space into attention-layer parameters (fraction $\phi = d_{\text{attn}}/d$) and non-attention parameters (fraction $1-\phi$):

$$\text{Vol}_{\text{NSP}}^{\text{SP}} = T\,t_c\left[\frac{1}{K_x} + (1-\phi)\frac{1}{K_u} + \phi\frac{1}{K_u\sqrt{P}} + (1-\phi)\frac{1}{K_v} + \phi\frac{1}{K_v\cdot P}\right]. \tag{247}$$

The first term (parameter sync) is unchanged—all parameters sync at K_x regardless of SP. The momentum terms split: non-attention coordinates use the base periods, while attention coordinates use the extended SP-tightened periods.

SP-aware FAUST vs. DDP. For typical Transformer architectures, attention projection parameters constitute approximately $\phi \approx 1/3$ of total parameters (the Q, K, V, O projections across all layers). Substituting $K_x = 256$, $K_u = 768$, $K_v = 1536$, and $P = 2$:

$$\begin{aligned}\text{Vol}_{\text{NSP}}^{\text{SP}} &= T\,t_c\left[\frac{1}{256} + \frac{2}{3}\cdot\frac{1}{768} + \frac{1}{3}\cdot\frac{1}{768\sqrt{2}} + \frac{2}{3}\cdot\frac{1}{1536} + \frac{1}{3}\cdot\frac{1}{3072}\right]\\ &= T\,t_c\left[3.906\times10^{-3} + 8.681\times10^{-4} + 3.069\times10^{-4} + 4.340\times10^{-4} + 1.085\times10^{-4}\right]\\ &= T\,t_c\cdot 5.624\times10^{-3}.\end{aligned} \tag{248}$$

The base (non-SP-tightened) volume from (227) is:

$$\text{Vol}_{\text{NSP}}^{\text{base}} = T\,t_c\cdot\frac{3}{512} = T\,t_c\cdot 5.859\times10^{-3}. \tag{249}$$

The SP-tightened volume is $5.624/5.859 = 96.0\%$ of the base, yielding an additional 4.0% communication reduction on top of the $512\times$ base factor. At $P = 4$:

$$\begin{aligned}\text{Vol}_{\text{NSP}}^{\text{SP}}\big|_{P=4} &= T\,t_c\left[\frac{1}{256} + \frac{2}{3}\cdot\frac{1}{768} + \frac{1}{3}\cdot\frac{1}{1536} + \frac{2}{3}\cdot\frac{1}{1536} + \frac{1}{3}\cdot\frac{1}{6144}\right]\\ &= T\,t_c\cdot 5.371\times10^{-3},\end{aligned} \tag{250}$$

which is 91.7% of the base—an 8.3% additional reduction, or equivalently a combined $558\times$ communication reduction vs. DDP (up from $512\times$).

SP-aware FAUST vs. Local Adam. For the comparison with Local Adam (which does not benefit from SP-tightened periods since it synchronizes all states uniformly), the SP-tightened FAUST volume ratio becomes:

$$\frac{\text{Vol}_{\text{NSP}}^{\text{SP}}}{\text{Vol}_{\text{LA}}} = \frac{5.624\times10^{-3}}{3/256} = \frac{5.624\times10^{-3}}{1.172\times10^{-2}} = 0.480 \tag{251}$$

at $P = 2$, i.e., a $2.08\times$ reduction (vs. the base $2\times$). At $P = 4$, this becomes $0.458\times$, or a $2.18\times$ reduction.

The incremental improvement from SP-tightened periods is modest (4–8%) because the parameter sync (K_x-tier) dominates the total volume and is not SP-dependent. The improvement is more pronounced for momentum-heavy optimizers like ADOPT, where u and v together constitute $2/3$ of the total communication.

F.5 Throughput Comparison

Table 7 confirms the wall-clock model's predictions:

- At $K_x = 16$ (high-frequency sync), FAUST achieves $1.75\times$ the throughput of DDP for the 1B model (105k vs. 60k tokens/sec/GPU). The gap between FAUST (105k) and Local Adam (89k) reflects the $2\times$ communication reduction: Local Adam synchronizes 3 tensors per round while FAUST synchronizes on average $1 + 1/3 + 1/6 = 3/2$ tensors per round.

Table 7: Throughput (tokens/sec/GPU) at different scales. FAUST matches FAVG+OPT throughput while providing convergence guarantees and fault tolerance. The SP All-to-All overhead ($< 2\%$) is included in all measurements.

	1B Model		7B Model	
K_x	16	256	16	256
DDP	60k	60k	8.5k	8.5k
FAVG+OPT	106k	135k	11.5k	13.7k
Local Adam	89k	132k	10.4k	13.6k
FAUST	105k	134k	11.4k	13.7k

- At $K_x = 256$ (low-frequency sync), all communication-efficient methods approach the compute-bound ceiling (~ 135k), and the throughput difference between FAUST and Local Adam narrows to $< 2\%$. This is expected: when K_x is large, the amortized communication cost per step is small regardless of the method.
- FAUST matches the throughput of FAVG+OPT (which never synchronizes optimizer states, $K_u = K_v = \infty$) to within 1%, confirming that the additional momentum synchronization rounds at $K_u = 3K_x$ and $K_v = 6K_x$ contribute negligible overhead.

G Checkpointing vs. Periodic State Synchronization

A natural question is whether periodic checkpointing—saving the full model and optimizer state to persistent storage at regular intervals—can replace FAUST's explicit cross-replica state synchronization for handling replica failures. We argue it cannot, for four reasons. The first two are inherited from the distributed optimization theory; the third and fourth are specific to FAUST's composed SP + desynced architecture.

G.1 Reason 1: State staleness and unbounded drift

When a new replica joins the training run from a checkpoint written at step t_{ckpt}, its optimizer states (u, v) reflect the *global* state at step t_{ckpt}. If the current step is $t > t_{\text{ckpt}}$, the joining replica's states are stale by $\Delta t = t - t_{\text{ckpt}}$ steps.

Quantifying the staleness. Using the drift bounds derived in Section E, the maximum ℓ_∞ deviation of the first moment between the checkpoint state and the current global state is:

$$\|u_t - u_{t_{\text{ckpt}}}\|_\infty \leq 2\rho\,(1 - \beta_1^{\Delta t}). \tag{252}$$

For the second moment:

$$\|v_t - v_{t_{\text{ckpt}}}\|_\infty \leq 2\rho^2\,(1 - \beta_2^{\Delta t}). \tag{253}$$

Now consider two regimes:

Frequent checkpointing ($\Delta t \ll \tau_{0.5}$). The staleness is small: $1-\beta^{\Delta t} \approx \Delta t\,(1-\beta)$, so the drift grows linearly in Δt. However, frequent checkpointing is expensive: writing the full optimizer state ($3|x|$ parameters in FP32 $= 12|x|$ bytes) to persistent storage at every step negates the communication savings that FAUST provides. For the 1.3B model, $12|x| \approx 15.6$ GB per checkpoint.

Infrequent checkpointing ($\Delta t \gg \tau_{0.5}$). The drift saturates: $1 - \beta^{\Delta t} \to 1$, and the stale states bear no relation to the current optimizer trajectory. A joining replica effectively re-initializes its optimizer states from scratch, which is precisely the "reset" strategy that Section 5.1 shows to be unstable.

Contrast with FAUST's periodic synchronization. Under FAUST, the maximum staleness of any optimizer state is bounded by the corresponding synchronization period:

$$\text{First moment:} \quad \max_t \|u_t - u_t^m\|_\infty \leq 2\rho\,(1 - \beta_1^{K_u}), \tag{254}$$

$$\text{Second moment:} \quad \max_t \|v_t - v_t^m\|_\infty \leq 2\rho^2\,(1 - \beta_2^{K_v}). \tag{255}$$

With $K_u = 3K_x$ and $K_v = 6K_x$, these bounds are *finite and deterministic*—they hold at every step, not just at checkpoint boundaries. A newly joining replica receives the current global state at the next synchronization event (at most K_v steps away), after which its local states are within the bounded-drift envelope.

SP-tightened staleness for attention parameters. For the d_{attn} attention-layer coordinates, the SP-tightened drift bounds (Section E.4) yield even sharper staleness guarantees under FAUST:

$$\text{First moment (attn):} \qquad \max_t \left|[u_t - u_t^m]_j\right| \le \frac{2\rho}{\sqrt{P}}\,(1-\beta_1^{K_u}), \tag{256}$$

$$\text{Second moment (attn):} \qquad \max_t \left|[v_t - v_t^m]_j\right| \le \frac{2\rho^2}{P}\,(1-\beta_2^{K_v}). \tag{257}$$

At $P = 2$, the attention first-moment staleness is reduced by 29% and the second-moment staleness by 50% compared to checkpointing, which cannot exploit this SP structure at all (checkpoint states are stored monolithically without tier or SP awareness). Moreover, the SP-aware schedule extension (Section E.4) allows using longer sync periods $K_u^{\text{SP}} = K_u\sqrt{P}$ for attention parameters while maintaining the same drift envelope—widening the gap between FAUST's bounded staleness and checkpointing's unbounded staleness.

G.2 Reason 2: No convergence guarantees from checkpoint averaging

Ad-hoc checkpoint averaging—loading a checkpoint, running K local steps, then averaging the resulting model with the current global model—provides no theoretical guarantee about the quality of the averaged optimizer states. The difficulty is that the averaging operation

$$u_{\text{avg}} = \frac{1}{2}\big(u_{\text{checkpoint}} + u_{\text{current}}\big) \tag{258}$$

does not correspond to any step of the FAUST algorithm (Algorithm 1). In particular:

1. The convergence proof (Theorem 3.1) relies on the $\mathbb{E}_m[\cdot]$ averaging operator being applied at *probabilistically regular* intervals, so that the drift coefficient ψ remains bounded. A one-time checkpoint average at an arbitrary step does not satisfy this regularity.
2. The step-size restriction $\eta_0 \propto 1/\sqrt{\psi}$ (equation (68)) is calibrated to the *worst-case* drift over the synchronization period. A checkpoint written at an arbitrary interval may produce a drift that exceeds the step-size budget, potentially violating the conditions of Theorem 3.1.

Formal statement. Let Δ_{ckpt} denote the staleness of a checkpoint-based recovery. For the convergence bound (71) to remain valid, we require $\Delta_{\text{ckpt}} \le K_v$—i.e., the checkpoint must be no older than the longest synchronization period. But if checkpoints are written every K_v steps, the checkpointing cost is:

$$\text{Cost}_{\text{ckpt}} = \frac{T}{K_v} \cdot t_{\text{write}}(3|x|), \tag{259}$$

where $t_{\text{write}}(3|x|)$ is the time to write $3|x|$ FP32 parameters to persistent storage. For the 1.3B model with $t_{\text{write}} \approx 2\,\text{s}$ (NVMe SSD), this adds $\frac{T}{1536} \cdot 2 \approx 0.0013\,T$ seconds—comparable to the communication cost that FAUST's desynced schedule was designed to eliminate. Moreover, the checkpoint write must be *synchronous* (all replicas pause to ensure a consistent snapshot), introducing a barrier that defeats the purpose of asynchronous local training.

G.3 Reason 3: SP group atomicity and compiler state

Under FAUST's compile-time sequence-parallel decomposition, each replica consists of P SP ranks that jointly compute the attention layers. The SP All-to-All collectives require *all P ranks* to participate: if any SP rank within a replica fails, the replica's forward pass is incomplete and produces no valid gradient.

This has three consequences for checkpoint-based recovery:

(a) The recovery unit is the full SP group, not a single rank. Checkpointing at the individual-rank level is insufficient. When a rank fails, the entire SP group must be restored from the checkpoint, which requires P coordinated reads from persistent storage. The AutoSP compile-time graph rewrite (commit `5efb24a`) has specialized the computation graph for the specific SP degree P: the SHARDSEQDIM pass rewrites all symbolic sequence-dimension consumers to use S/P; the INSERTA2A pass places $4L$ `torch.ops.autosp.all_to_all` nodes at fixed positions in the FX graph; and the PROPAGATESHAPES pass bakes the sharded activation shapes into every intermediate node's metadata. Restoring a single rank into a running SP group requires re-executing the full 7-pass compilation pipeline or maintaining P copies of the compiled graph artifacts—neither of which is free.

(b) Checkpoint state lacks SP-tier metadata. FAUST's runtime maintains per-parameter tier annotations (`param._desloc_tier`) and per-bucket tier tags that govern communication scheduling. These annotations are set during `init_desloc_torch_extensions` and are not part of the standard PyTorch state dict. A checkpoint-restored replica would need to re-derive the tier map from the model structure, introducing a potential inconsistency if the model architecture has been modified between the checkpoint and recovery point (e.g., layer freezing, adapter insertion).

(c) FAUST's periodic synchronization naturally handles SP group recovery. When a failed SP group is replaced, the new group initializes its parameters and optimizer states from the cross-replica average at the next synchronization boundary (at most K_v steps away). The SP All-to-All within the new group executes correctly because all P ranks start from the same state and process the same compiled graph. No checkpoint read is required—the recovery state is obtained from the *live* replicas via the existing AllReduce infrastructure. The tier annotations are restored via the same `register_param_tiers` call that initializes any new replica, and the SP-tightened sync periods (Section E.4) ensure that attention-tier states converge within K_u^{SP} steps.

This is the key advantage of periodic synchronization over checkpointing in the SP setting: the recovery mechanism uses the same communication primitives that are already part of the training loop, rather than requiring an additional I/O subsystem.

G.4 Reason 4: Interaction with the collective fence

FAUST's collective fence mechanism (Phase 3 of Algorithm 2; implemented as `_desloc_fence_all_pending` with a configurable timeout) ensures that all pending SP All-to-All handles in the set $\mathcal{H}$ are retired before the data-parallel gradient reduction begins. This fence enforces the collective orthogonality constraint: without it, a data-parallel AllReduce could overlap with an in-flight SP All-to-All, causing communicator contention or, on shared-interconnect clusters, data corruption.

Checkpoint-based recovery introduces a subtle interaction with the fence. When a replica restores from a checkpoint, its SP All-to-All handles are in an undefined state: the restored ranks have no knowledge of which collectives were in flight at the time of failure. The fence mechanism assumes that all ranks within an SP group have consistent handle state ($\mathcal{H}$ is identical across ranks); a checkpoint-restored rank violates this assumption because its $\mathcal{H} = \emptyset$ while the surviving ranks may have $|\mathcal{H}| = L$ pending handles.

Consequences for gradient correctness. If a checkpoint-restored rank participates in a DP AllReduce while other ranks still have pending A2A handles, the gradient g_t^m entering the AllReduce may be partially SP-reduced on some ranks and fully reduced on others. This violates the SP gradient identity (16) and the variance invariance (17), invalidating the convergence analysis (specifically, the unbiasedness assumption in the variance decomposition (42)).

Under FAUST's periodic synchronization, this problem does not arise: the synchronization boundary is a *global barrier* at which all pending handles are guaranteed to have completed (the fence $\text{WAITALL}(\mathcal{H})$ fires *before* the AllReduce). A newly joining replica enters the training loop at the post-fence state, where $\mathcal{H} = \emptyset$ for all ranks and the communicator is in a clean state.

G.5 Summary

Property	Checkpointing	FAUST sync
Bounded state drift	Only if $\Delta t \leq K_v$	Always ((254), (255))
SP-tightened attn drift	N/A (monolithic)	$1/\sqrt{P}$ first, $1/P$ second ((256), (257))
Convergence guarantee	None	Theorem 3.1
SP group recovery	P coordinated reads + recompile	Via existing AllReduce
Tier metadata	Not preserved	Restored via `register_param_tiers`
Fence consistency	$\mathcal{H}$ undefined	$\mathcal{H} = \emptyset$ (clean post-fence)
I/O overhead	$T/K_v \cdot t_{\text{write}}$	Zero (uses AllReduce)

Practical recommendation. Checkpointing remains valuable as a *disaster recovery* mechanism (e.g., full cluster power loss), but it should not be used as a *substitute* for FAUST's periodic optimizer-state synchronization. The two mechanisms serve different purposes: checkpointing protects against catastrophic failures that affect all replicas simultaneously, while periodic synchronization protects against individual replica failures and ensures bounded optimizer-state drift for convergence.

The SP-tightened analysis strengthens this recommendation: under SP, FAUST's periodic synchronization gains an *additional* advantage that checkpointing fundamentally cannot match—the structural awareness that attention-layer states drift more slowly (by $1/\sqrt{P}$ for the first moment and $1/P$ for the second). Checkpointing, being a monolithic state-dump mechanism, cannot exploit this per-tier structure and must therefore checkpoint all states at the frequency dictated by the *fastest-drifting* component, negating the communication savings that FAUST's tiered schedule provides.

H LLM Usage Declaration

Large Language Models were used during the preparation of this work for editing and improving the text quality of the manuscript. No LLM was used for generating scientific content, proofs, or experimental results. All technical contributions are the original work of the authors.

Algorithm 2 FAUST-Adam with Explicit SP–DP Collective Orchestration

Require: Model $x_0 \in \mathbb{R}^d$; first/second moment seeds $u_{-1}, v_{-1} \in \mathbb{R}^d$
Require: $\{\eta_t\}_{t=0}^{T-1} \subset \mathbb{R}_{>0}$ {K_x-aligned WSD step-size schedule}
Require: $\beta_1, \beta_2 \in [0, 1)$; $\rho, \lambda \in \mathbb{R}_{>0}$ {Adam decays; clip bound; stability}
Require: $T, M \in \mathbb{N}_+$ {total iterations; DP workers}
Require: $P \in \mathbb{N}_+$ {SP degree ($P = 1$ recovers non-SP)}
Require: $K_x, K_u, K_v \in \mathbb{N}_+$ with $K_x \mid K_u \mid K_v$ {tier sync periods, divisibility chain}
Require: L {number of attention (SDPA) layers}
Ensure: x_T, u_{T-1}, v_{T-1}

$\forall m$: $x_0^m \leftarrow x_0$; $u_{-1}^m \leftarrow v_{-1}^m \leftarrow 0$
$\mathcal{H} \leftarrow \emptyset$ {set of pending A2A async handles}
for $t = 0, \ldots, T-1$ **do**
  for all $m = 0, \ldots, M-1$ **in parallel do**
    {**— Phase 1: Intra-step SP forward (compile-time inserted) —**}
    Partition mini-batch ξ_t^m into P chunks along sequence dim
    for $\ell = 1, \ldots, L$ **do**
      $Q_\ell, K_\ell, V_\ell \leftarrow \mathcal{A}_P\big(Q_\ell^{\text{loc}}, K_\ell^{\text{loc}}, V_\ell^{\text{loc}}\big)$ {$[B, N, \frac{S}{P}, H] \to [B, \frac{N}{P}, S, H]$}
      $O_\ell \leftarrow \text{SDPA}(Q_\ell, K_\ell, V_\ell)$ {full-S attention on N/P heads}
      $O_\ell^{\text{loc}} \leftarrow \mathcal{A}_P^{-1}(O_\ell)$; $\mathcal{H} \leftarrow \mathcal{H} \cup \{h_\ell\}$ {async handle}
    end for
    {**— Phase 2: Backward with FP32 A2A accumulation —**}
    $\tilde{g}_t^m \leftarrow \nabla_x \mathcal{L}\big(x_t^m;\, \xi_t^m,\, \{O_\ell^{\text{loc}}\}\big)$ {backward through compiled graph $\mathcal{G}'$}
    $g_t^m \leftarrow \sum_{p=0}^{P-1} \tilde{g}_t^{m,p}$ {backward A2A reduces partial grads; FP32 accum.}
    $\hat{g}_t^m \leftarrow \text{clip}\big(g_t^m,\, \rho_j\big)$ where $\rho_j = \begin{cases} \rho/\sqrt{P} & j \in \mathcal{I}_{\text{attn}} \\ \rho & \text{otherwise} \end{cases}$ {SP-tightened clip}
    {**— Phase 3: Collective fence —**}
    if $t \bmod K_x = 0$ **then**
      WAITALL($\mathcal{H}$); $\mathcal{H} \leftarrow \emptyset$ {fence: all A2A handles must retire before DP sync}
    end if
    {**— Phase 4: Inter-step DP optimizer with tiered sync —**}
    if $t \bmod K_u = 0$ **then**
      $u_t^m \leftarrow \beta_1 \mathbb{E}_m[u_{t-1}^m] + (1 - \beta_1)\hat{g}_t^m$ {AllReduce first moment}
    else
      $u_t^m \leftarrow \beta_1 u_{t-1}^m + (1 - \beta_1)\hat{g}_t^m$ {local EMA}
    end if
    if $t \bmod K_v = 0$ **then**
      $v_t^m \leftarrow \beta_2 \mathbb{E}_m[v_{t-1}^m] + (1 - \beta_2)(\hat{g}_t^m \odot \hat{g}_t^m)$ {AllReduce second moment}
    else
      $v_t^m \leftarrow \beta_2 v_{t-1}^m + (1 - \beta_2)(\hat{g}_t^m \odot \hat{g}_t^m)$ {local EMA}
    end if
    $d_t^m \leftarrow \dfrac{\eta_t}{\sqrt{v_t^m + \lambda^2}} \odot u_t^m$ {preconditioned step}
    if $t \bmod K_x = 0$ **then**
      $x_{t+1}^m \leftarrow \mathbb{E}_m[x_t^m] - d_t^m$ {AllReduce parameters}
    else
      $x_{t+1}^m \leftarrow x_t^m - d_t^m$ {local step}
    end if
  end for
end for

Algorithm 3 FAUST-ADOPT with Explicit SP–DP Collective Orchestration

Require: Model $x_0 \in \mathbb{R}^d$; moment seeds $m_{-1}, v_{-1} \in \mathbb{R}^d$
Require: $\{\eta_t\}_{t=0}^{T-1} \subset \mathbb{R}_{>0}$ {K_x-aligned WSD learning rate schedule}
Require: $\beta_1, \beta_2 \in [0, 1)$; $\rho, \epsilon \in \mathbb{R}_{>0}$ {decay; clip bound; stability}
Require: $T, M \in \mathbb{N}_+$ {total iterations; DP workers}
Require: $P \in \mathbb{N}_+$; $L \in \mathbb{N}_+$ {SP degree; number of SDPA layers}
Require: $K_x, K_m, K_v \in \mathbb{N}_+$ with $K_x \mid K_m \mid K_v$ {tier sync periods}
Ensure: x_T, m_{T-1}, v_{T-1}

$\forall w$: $x_0^w \leftarrow x_0$; $m_{-1}^w \leftarrow v_{-1}^w \leftarrow 0$
$\mathcal{H} \leftarrow \emptyset$ {pending A2A async handles}
for $t = 0, \ldots, T-1$ **do**
  for all $w = 0, \ldots, M-1$ **in parallel do**
    {**— Phase 1: Intra-step SP forward —**}
    Partition ξ_t^w into P chunks along sequence dim {compiled $\mathcal{G}'$}
    for $\ell = 1, \ldots, L$ **do**
      $Q_\ell, K_\ell, V_\ell \leftarrow \mathcal{A}_P(Q_\ell^{\text{loc}}, K_\ell^{\text{loc}}, V_\ell^{\text{loc}})$ $\{[B, N, \frac{S}{P}, H] \to [B, \frac{N}{P}, S, H]\}$
      $O_\ell \leftarrow \text{SDPA}(Q_\ell, K_\ell, V_\ell)$; $O_\ell^{\text{loc}} \leftarrow \mathcal{A}_P^{-1}(O_\ell)$; $\mathcal{H} \leftarrow \mathcal{H} \cup \{h_\ell\}$
    end for
    {**— Phase 2: Backward with FP32 A2A + SP-tightened clip —**}
    $g_t^w \leftarrow \sum_{p=0}^{P-1} \tilde{g}_t^{w,p}$ {backward A2A: $\nabla \mathcal{A}_P = \mathcal{A}_P^{-1}$; FP32 accum.}
    $\hat{g}_t^w \leftarrow \text{clip}(g_t^w, \rho_j)$ where $\rho_j = \rho/\sqrt{P}$ for $j \in \mathcal{I}_{\text{attn}}$, ρ otherwise
    {**— Phase 3: Collective fence —**}
    if $t \bmod K_x = 0$ **then**
      WAITALL($\mathcal{H}$); $\mathcal{H} \leftarrow \emptyset$ {A2A $\perp$ DP: fence before any AllReduce}
    end if
    {**— Phase 4: ADOPT optimizer with tiered DP sync —**}
    if $t \bmod K_v = 0$ **then**
      $v_t^w \leftarrow \beta_2 \mathbb{E}_w[v_{t-1}^w] + (1-\beta_2)(\hat{g}_t^w \odot \hat{g}_t^w)$ {AllReduce v}
    else
      $v_t^w \leftarrow \beta_2 v_{t-1}^w + (1-\beta_2)(\hat{g}_t^w \odot \hat{g}_t^w)$ {local EMA}
    end if
    if $t \bmod K_m = 0$ **then**
      $m_t^w \leftarrow \beta_1 \mathbb{E}_w[m_{t-1}^w] + (1-\beta_1)\dfrac{\hat{g}_t^w}{\max\{\sqrt{v_{t-1}^w},\, \epsilon\}}$ {AllReduce m}
    else
      $m_t^w \leftarrow \beta_1 m_{t-1}^w + (1-\beta_1)\dfrac{\hat{g}_t^w}{\max\{\sqrt{v_{t-1}^w},\, \epsilon\}}$ {local ADOPT}
    end if
    $d_t^w \leftarrow \eta_t\, m_t^w$ {ADOPT step (no per-coord preconditioning)}
    if $t \bmod K_x = 0$ **then**
      $x_{t+1}^w \leftarrow \mathbb{E}_w[x_t^w] - d_t^w$ {AllReduce parameters}
    else
      $x_{t+1}^w \leftarrow x_t^w - d_t^w$ {local step}
    end if
  end for
end for

Algorithm 4 FAUST-SGDM (probabilistic synchronization formulation)

Require: Model tensors
$x_0 \in \mathbb{R}^d$ initial parameter vector
$u_{-1} \in \mathbb{R}^d$ seed for the momentum, initialised to $\mathbf{0}$
Require: Hyper-parameters
$\{\eta_t\}_{t=0}^{T-1} \subset \mathbb{R}_{>0}$ K_x-aligned WSD step-size schedule
$\beta \in [0, 1)$ momentum decay factor
$T \in \mathbb{N}_+$ total optimisation steps
$M \in \mathbb{N}_+$ number of replicas
$p_x = \frac{1}{K_x},\ p_u = \frac{1}{K_u} \in [0, 1]$ synchronization probabilities for parameters and momentum
Ensure: $x_T,\ u_{T-1}^m$
for each replica m: $x_0^m = x_0,\ u_{-1}^m = 0$ local init
for $t = 0, \dots, T-1$ **do**
  for all replicas $m = 0, \dots, M-1$ **in parallel do**
    $g_t^m \leftarrow \nabla F_m(x_t^m; \xi_t^m)$ stochastic gradient (SP-reduced when $P > 1$)
    $u_t^m \leftarrow \begin{cases} \mathbb{E}_m[\beta u_{t-1}^m + (1-\beta) g_t^m], & \text{with probability } p_u \\ \beta u_{t-1}^m + (1-\beta) g_t^m, & \text{with probability } 1 - p_u \end{cases}$
    $x_{t+1}^m \leftarrow \begin{cases} \mathbb{E}_m[x_t^m - \eta_t u_t^m], & \text{with probability } p_x \\ x_t^m - \eta_t u_t^m, & \text{with probability } 1 - p_x \end{cases}$
  end for
end for

Algorithm 5 FAUST-Adam (probabilistic synchronization formulation)

Require: Model tensors
$x_0 \in \mathbb{R}^d$ initial parameter vector
$u_{-1}, v_{-1} \in \mathbb{R}^d$ seeds for first and second moments, initialised to $\mathbf{0}$
Require: Hyper-parameters
$\{\eta_t\}_{t=0}^{T-1} \subset \mathbb{R}_{>0}$ K_x-aligned WSD step-size schedule
$\beta_1, \beta_2 \in [0, 1)$ Adam decay factors
$\lambda \in \mathbb{R}_{\geq 0}$ ℓ_2 stability term
$T \in \mathbb{N}_+$ total optimisation steps
$M \in \mathbb{N}_+$ number of replicas
$p_x = \frac{1}{K_x},\ p_u = \frac{1}{K_u},\ p_v = \frac{1}{K_v} \in [0, 1]$ synchronization probabilities for parameters and momenta
Ensure: $x_T,\ u_{T-1},\ v_{T-1}$
for each replica m: $x_0^m = x_0,\ u_{-1}^m = v_{-1}^m = 0$ local init
for $t = 0, \dots, T-1$ **do**
  for all replicas $m = 0, \dots, M-1$ **in parallel do**
    $g_t^m \leftarrow \nabla F_m(x_t^m; \xi_t^m)$ stochastic gradient (SP-reduced when $P > 1$)
    $u_t^m \leftarrow \begin{cases} \mathbb{E}_m[\beta_1 u_{t-1}^m + (1-\beta_1) g_t^m], & \text{w.p. } p_u \\ \beta_1 u_{t-1}^m + (1-\beta_1) g_t^m, & \text{w.p. } 1 - p_u \end{cases}$
    $v_t^m \leftarrow \begin{cases} \mathbb{E}_m[\beta_2 v_{t-1}^m + (1-\beta_2)(g_t^m \odot g_t^m)], & \text{w.p. } p_v \\ \beta_2 v_{t-1}^m + (1-\beta_2)(g_t^m \odot g_t^m), & \text{w.p. } 1 - p_v \end{cases}$
    $\tilde{v}_t^m \leftarrow \max(v_t^m,\ \tilde{v}_{t-1}^m)$ AMSGrad normalization ($\tilde{v}_{-1} = v_{-1}$)
    $d_t^m \leftarrow \frac{\eta_t}{\sqrt{\tilde{v}_t^m + \lambda^2}} \odot u_t^m$ preconditioned update
    $x_{t+1}^m \leftarrow \begin{cases} \mathbb{E}_m[x_t^m - d_t^m], & \text{w.p. } p_x \\ x_t^m - d_t^m, & \text{w.p. } 1 - p_x \end{cases}$
  end for
end for

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract states three quantitative claims ($303\times$ communication reduction over DDP, $1.78\times$ over DES-LOC, perplexity within 4.6% of baseline) and a convergence guarantee. All are supported by Theorem 3.1 (Section 3) and the experimental evaluation (Section 5).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The conclusion (Section 7) identifies open directions including layer-wise synchronization, adaptive frequencies, and compressed updates. The theoretical analysis is limited to smooth non-convex objectives under bounded gradient assumptions, which are stated explicitly in the assumptions of Theorem 3.1.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Assumptions are stated explicitly (Section 3). The full proof of Theorem 3.1 is provided in Appendix D, including all supporting lemmas with complete proofs in Appendix D.2. The Adam extension is given in Appendix D.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4 and Appendix A provide model architectures, optimizer hyperparameters (including sweep results), sequence-parallel configuration, batch sizes, learning rate schedules, and random seeds. The training data (FineWeb-Edu) is publicly available.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Code is not released at submission time to preserve anonymity. The training data (FineWeb-Edu) is publicly available. We plan to release the code upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 describes the experimental setup. Appendix A provides full architecture details, hyperparameter sweep procedures, and the final selected hyperparameters for each configuration.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to the high computational cost of pretraining language models at the 135M–1.3B scale, each configuration was run once with a fixed seed. We report final perplexity values rather than confidence intervals. The deterministic seed and full hyperparameter specification enable exact replication.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A specifies the GPU types (NVIDIA H100 SXM, B200), interconnect bandwidths (NVLink, inter-node), and the number of GPUs per experiment. Throughput measurements and wall-clock times are reported throughout Section 5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work proposes a communication-efficient distributed training algorithm. It does not involve human subjects, personal data, or dual-use concerns. The research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: FAUST is a systems-level optimization for distributed training communication. It does not introduce new model capabilities, datasets, or applications. Its primary societal effect is reducing the energy and hardware cost of large-scale training, which is broadly positive. We do not foresee negative societal impacts specific to this contribution beyond those inherent to large language model training in general.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not release pre-trained models, datasets, or other artifacts that pose misuse risks. The contribution is a training algorithm and system.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the original papers for all models (Llama-3.1), datasets (FineWeb-Edu), and baseline methods (Local SGD, FedAvg, Local Adam, DiLoCo, DES-LOC) used in this work. FineWeb-Edu is released under an open license.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: No new datasets or pre-trained models are released in this submission.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This work does not involve crowdsourcing or human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This work does not involve human subjects research.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Section H declares that LLMs were used only for editing and improving text quality. No LLM was used for generating scientific content, proofs, or experimental results.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.