INTERMITTENT ERROR RECYCLING: NARROWING THE GAP BETWEEN COMPRESSED AND UNCOMPRESSED SGD

Anonymous authors
Paper under double-blind review

ABSTRACT

Gradient compression is a standard remedy for communication overhead in distributed optimization. Yet, existing unbiased compression methods can substantially degrade convergence by inflating the variance of stochastic updates, creating a gap between compressed and uncompressed training. In this work, we show that this degradation is not fundamental and can be mitigated by exploiting a simple distinction: unlike stochastic gradient noise, compression error is directly observable. We propose Intermittent Error Recycling (IER), a framework in which workers accumulate compression residuals locally and periodically communicate an uncompressed correction. This allows compression errors to be amortized over time rather than repeatedly injected into the optimization dynamics. We show that IER recovers the dominant stochastic convergence term of uncompressed distributed SGD without compression-induced variance inflation, while confining compression-dependent effects to faster-decaying terms. This mechanism does not increase the total communication cost, with the correction frequency determined directly from the communication budget, allowing IER to retain the benefits of compression without its associated variance penalty. We empirically corroborate our theoretical guarantees and demonstrate the efficacy of our approach.

1 INTRODUCTION

Distributed stochastic optimization is the engine of modern machine learning, enabling efficient training across data-parallel workers. However, inter-worker communication remains a primary bottleneck, limiting the efficiency of distributed training. This challenge has catalyzed a rich literature on communication-efficient optimization, with *gradient compression*, where workers transmit low-precision or sparsified updates, as a principal solution.

Gradient compression schemes are broadly categorized by their statistical properties into two families. The first consists of **(i) Unbiased compressors**, such as QSGD (Alistarh et al., 2017), Rand-k (Stich et al., 2018), and stochastic rounding (Gupta et al., 2015). These schemes preserve the unbiasedness of stochastic gradients, a fundamental property that facilitates the convergence analysis of stochastic optimization algorithms like SGD. The second category comprises **(ii) Biased compressors**, like Top-k sparsification (Alistarh et al., 2018) and low-rank approximations like PowerSGD (Vogels et al., 2019). These methods generally produce compressed updates whose expectations differ from the original gradients, violating unbiasedness, and introducing bias into the optimization process.

Among these, unbiased compressors occupy a unique theoretical position. By preserving the expectation of the stochastic gradient, they retain the defining property that underpins classical SGD analysis. Consequently, unbiasedness has become a standard assumption in much of the theoretical literature on compressed optimization.

The Variance Inflation Problem. Despite their theoretical appeal, unbiased compressors suffer from a critical drawback: *variance inflation*. Concretely, as unbiased compressors become more aggressive, they inject more noise into the compressed gradients, which in turn inflates their variance relative to the original (uncompressed) stochastic gradients. Consequently, the communication savings achieved through coarse quantization or extreme sparsification in existing approaches can come at the cost of substantially slower convergence.

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

Crucially, this phenomenon is not limited to naive compressed SGD. It persists even in substantially more sophisticated unbiased compression methods, including control-variate and gradient-difference schemes such as DIANA (Mishchenko et al., 2023), MARINA (Gorbunov et al., 2021), and SCALLION (Huang et al., 2024). Despite their mechanisms for mitigating compression error, they still suffer a performance degradation due to variance inflation as their leading stochastic convergence terms retain an explicit dependence on the compression parameter. This raises a fundamental question:

Is the stochastic degradation induced by unbiased compression unavoidable, or is it an artifact of algorithmic design?

In this work, we show that this degradation is not fundamental, and design a new training approach that eliminates compression-induced variance inflation in the leading stochastic term. Our approach exploits the key observation that, unlike the inherent stochastic gradient noise, compression noise is *fully observable and exactly measurable* at every iteration. This asymmetry suggests that compression error should not be treated as just another source of randomness. Instead, it can be *tracked and corrected* in a more deliberate manner.

Motivated by this insight, we propose *Intermittent Error Recycling (IER)*. Rather than repeatedly transmitting compressed correction signals and injecting more compression noise into the optimization process, each worker *locally* accumulates its compression errors and transmits an *uncompressed* correction only intermittently, once every K steps. These periodic corrections cancel the accumulated compression error, while their communication cost is amortized over the intervening compressed iterations, with K determined explicitly by a prescribed communication budget. By correcting residuals exactly but infrequently, our method preserves communication efficiency and confines compression-dependent effects to lower-order terms, *decoupling* them from the leading stochastic convergence term.

We show that this novel mechanism fundamentally changes how unbiased compression affects convergence. Our analysis demonstrates that IER recovers the dominant stochastic term of uncompressed distributed SGD, independent of the compression level, while confining compression-dependent effects to lower-order terms. Thus, the leading variance-inflation penalty is not an inevitable cost of compression: through a different algorithmic treatment of compression noise, IER eliminates this penalty while retaining the communication benefits of compressed training. Empirical results reinforce these findings, demonstrating that IER substantially narrows the gap between compressed and uncompressed optimization while respecting fixed communication budgets.

1.1 Related Work

Communication bottlenecks are a central challenge in distributed learning, motivating extensive work on reducing communication costs in data-parallel optimization. A dominant approach is to reduce communication *algorithmically* through lossy compression of gradients or updates.

Gradient quantization and sparsification. A major family of communication-efficient methods compresses gradients or updates through *quantization* or *sparsification*. Quantized SGD (QSGD) introduced randomized quantization schemes that preserve unbiasedness while trading off communication cost against convergence speed (Alistarh et al., 2017). More aggressive schemes such as 1-bit SGD rely on explicit error compensation to stabilize training (Seide et al., 2014). TernGrad explored stochastic ternary quantization, establishing convergence under a bound on gradients (Wen et al., 2017). Sign-based methods such as signSGD with majority-vote aggregation represent an extreme quantization regime and have been studied both theoretically and empirically (Bernstein et al., 2018). Sparsification methods reduce communication by transmitting only a subset of gradient coordinates. Deep Gradient Compression (DGC) demonstrated very high compression ratios through magnitude-based sparsification combined with residual accumulation and other stabilization techniques (Lin et al., 2018). A related line of work combines compression with variance-reduction mechanisms to address heterogeneity and stochastic noise. DIANA compresses gradient differences using local control variates, reducing variance while maintaining communication efficiency (Mishchenko et al., 2023). MARINA and its variants interleave compressed gradient-difference updates with occasional gradient refreshes, yielding improved communication complexity guarantees in nonconvex distributed optimization (Gorbunov et al., 2021).

Error feedback. Error feedback (EF) is a popular mechanism for stabilizing compressed optimization by reinjecting compression errors into subsequent updates. This mechanism addresses

convergence failures associated with directly applying biased compressors such as sign-based schemes and Top-k by counteracting systematic bias accumulation (Seide et al., 2014; Karimireddy et al., 2019; Stich et al., 2018). More recently, EF21 introduced an error-feedback framework with improved convergence guarantees (Richtárik et al., 2021), and subsequent work showed that incorporating Polyak momentum can further improve communication and sample complexity in stochastic optimization (Fatkhullin et al., 2023). These methods represent the state of the art in principled EF approaches for mitigating compression bias.

Intermittent communication and local-update methods. Communication reduction has also been studied through local-update methods, where workers perform multiple local steps before synchronizing models, reducing communication while preserving convergence (Stich, 2019; Dahan & Levy, 2024). While both approaches introduce temporal structure into communication, local-update methods synchronize models intermittently, whereas IER aggregates worker updates at every iteration and communicates accumulated compression errors only periodically.

Unbiased compressors and the "noise inflation" phenomenon. A common approach to preserving SGD-style guarantees under compression is to employ *unbiased* compressors, such as QSGD and Rand-k, which preserve the expectation of stochastic gradients and facilitate clean convergence analyses (Alistarh et al., 2017). Comparisons of biased and unbiased compressors highlight substantial differences in their compression error and convergence behavior (Beznosikov et al., 2020). Recent work further shows that unbiasedness can be achieved from biased compressors via a multilevel Monte Carlo construction, which converts bias into variance (Zukerman et al., 2025). Yet, when applied naively, unbiased compression inflates the stochastic convergence term through additional compression variance. This effect manifests as slower convergence and greater sensitivity to stochastic gradient noise under aggressive compression, and has been demonstrated both theoretically and empirically (He et al., 2023; Compagnoni et al., 2025). Notably, this limitation extends beyond naive compressed SGD to more sophisticated mechanisms: despite their variance-reduction mechanisms, methods such as DIANA, VR-MARINA, and SCALLION still incur compression-dependent penalties in their stochastic convergence guarantees (Mishchenko et al., 2023; Gorbunov et al., 2021; Huang et al., 2024).

Positioning of our contribution. In contrast to methods whose stochastic convergence guarantees retain compression-dependent penalties, we bridge the gap between uncompressed SGD and communication-efficient training by changing how compression errors are corrected over time. Rather than modifying the compression operator, repeatedly transmitting compressed correction signals, or requiring large gradient batches, we propose an *intermittent error recycling* mechanism that accumulates compression errors locally and transmits *uncompressed* corrections only periodically. This design preserves communication efficiency, with the correction frequency determined explicitly by the communication budget. Our analysis recovers the leading stochastic convergence term of uncompressed distributed SGD while confining compression-dependent effects to lower-order terms, thus eliminating the leading-order variance-inflation penalty.

2 Background

2.1 Problem Setup

We consider a centralized distributed optimization setting with a parameter server and M workers over a time horizon of T iterations. The goal is to minimize a possibly nonconvex objective:

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{M}\sum_{i=1}^{M} f_i(x), \tag{1}$$

where each local objective $f_i(x) = \mathbb{E}_{z\sim\mathcal{D}_i}[f_i(x,z)]$ corresponds to the expected loss over data sampled from a local distribution $\mathcal{D}_i$. We allow for data heterogeneity across workers, i.e., the distributions $\{\mathcal{D}_i\}_{i=1}^M$ may differ.

At each iteration $t \in \{1,\dots,T\}$, the server broadcasts the current model $x_t \in \mathbb{R}^d$ to all workers. Each worker $i \in [M]$ draws a sample $z_{t,i} \sim \mathcal{D}_i$, independently across workers and independently of

the past, and computes the stochastic gradient $v_{t,i} = \nabla f_i(x_t, z_{t,i})$. In standard compressed SGD, each worker applies a possibly randomized compression operator $\mathcal{C}$ and transmits $g_{t,i} = \mathcal{C}(v_{t,i})$ to the server. The server averages the received updates, $g_t = \frac{1}{M}\sum_{i=1}^{M} g_{t,i}$, and updates the model according to $x_{t+1} = x_t - \eta g_t$, where $\eta > 0$ is the stepsize. In Section 4, we extend this framework by modifying the update rule to account for accumulated compression errors.

Throughout the paper, we make the following standard assumptions.

Assumption 2.1 (Smoothness)**.** Each local objective f_i is L-smooth. That is, for all $x, y \in \mathbb{R}^d$ and all $i \in [M]$: $\|\nabla f_i(y) - \nabla f_i(x)\| \leq L \|y - x\|$

Assumption 2.2 (Unbiasedness and Bounded Variance)**.** Unbiased stochastic gradients with bounded variance: $\forall x \in \mathbb{R}^d$, $i \in [M]$: $\mathbb{E}[\nabla f_i(x, z)|x] = \nabla f_i(x)$, and $\mathbb{E}\left[\|\nabla f_i(x, z) - \nabla f_i(x)\|^2 \mid x\right] \leq \sigma^2$

2.2 Unbiased Gradient Compression

Uncompressed data-parallel SGD enjoys strong theoretical guarantees but incurs substantial communication costs, as full-precision gradients must be transmitted at every iteration. A key property underlying the classical convergence analysis of SGD is the unbiasedness of the stochastic gradients.

To reduce communication overhead, a common approach is to compress the stochastic gradients before transmission. Unbiased compressors are designed to preserve this property while reducing communication costs.

Definition 2.3 (Unbiased Compressor)**.** A possibly randomized operator $\mathcal{C} : \mathbb{R}^d \to \mathbb{R}^d$ is an *unbiased compressor* with compression parameter $\omega \geq 0$ if, for all $x \in \mathbb{R}^d$,

$$\mathbb{E}[\mathcal{C}(x)] = x \quad \text{and} \quad \mathbb{E}\left[\|\mathcal{C}(x) - x\|^2\right] \leq \omega \|x\|^2,$$

where the expectations are taken over the internal randomness of the compressor.

The first condition ensures that compression preserves the unbiasedness of the stochastic gradient, while the second bounds the mean-squared compression error relative to the squared norm of the input.

3 Baseline Methods and the Compression-Induced Gap

In this section, we compare the convergence guarantees of standard distributed SGD and its counterpart using unbiased gradient compression. This comparison exposes the compression-induced variance penalty in the dominant stochastic convergence term, motivating our proposed approach.

3.1 Distributed SGD

We begin with standard distributed stochastic gradient descent (SGD) in the centralized parameter-server setting described in Section 2. At each iteration t, each worker computes an unbiased stochastic gradient $v_{t,i} = \nabla f_i(x_t, z_{t,i})$ and transmits it to the server. The server averages the gradients and performs the update $x_{t+1} = x_t - \eta \left(\frac{1}{M}\sum_{i=1}^{M} v_{t,i}\right)$. Under Assumptions 2.1 and 2.2, the standard nonconvex SGD analysis yields the following guarantee for an appropriately chosen stepsize (Ghadimi & Lan, 2013):

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\|\nabla f(x_t)\|^2 \;\leq\; \mathcal{O}\left(\frac{\Delta_1 L}{T} + \frac{\sigma\sqrt{\Delta_1 L}}{\sqrt{MT}}\right), \tag{2}$$

where $\Delta_1 = f(x_1) - f^*$, $f^* = \inf_x f(x) > -\infty$, and σ^2 bounds the variance of each worker's stochastic gradients. Averaging independent stochastic gradients across M workers reduces the variance bound to σ^2/M, yielding the $1/\sqrt{MT}$ scaling of the stochastic term.

3.2 Unbiased Gradient Compression

To reduce communication costs, a natural extension of distributed SGD is to compress the stochastic gradients before transmission. Each worker applies an unbiased compressor $\mathcal{C}$ to its stochastic

gradient, $g_{t,i} = \mathcal{C}(v_{t,i})$, and the server performs the update $x_{t+1} = x_t - \eta \left(\frac{1}{M} \sum_{i=1}^{M} g_{t,i} \right)$. Since $\mathcal{C}$ is unbiased (Definition 2.3), the averaged update remains an unbiased estimator of the true gradient. However, compression introduces additional variance. Specifically, combining Assumption 2.2 with the variance bound of the compressor yields:

$$\mathbb{E}\left[\|g_{t,i} - \nabla f_i(x_t)\|^2\right] \leq \sigma^2 + \omega\, \mathbb{E}\left[\|v_{t,i}\|^2\right]. \tag{3}$$

Thus, compression inflates the stochastic variance bound, and as a result, the convergence bound degrades accordingly (Zukerman et al., 2025):

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}\left[\|\nabla f(x_t)\|^2\right] \;\leq\; \mathcal{O}\left(\frac{\Delta_1 L}{T} + \frac{(\sigma + \sqrt{\omega}(\sigma + \xi))\sqrt{\Delta_1 L}}{\sqrt{MT}} \right). \tag{4}$$

Although compression preserves the unbiasedness of the update direction, it degrades the dominant stochastic term in the convergence bound, making variance inflation a leading-order effect.

3.3 The Compression Gap

Comparing the leading stochastic terms in Eqs.(2) and (4) highlights the variance inflation gap:

$$\underbrace{\mathcal{O}\left(\frac{\sigma}{\sqrt{MT}} \right)}_{\text{Distributed SGD}} \quad \longrightarrow \quad \underbrace{\mathcal{O}\left(\frac{\sigma + \sqrt{\omega}(\sigma + \xi)}{\sqrt{MT}} \right)}_{\text{Unbiased Compression}}$$

For aggressive compression schemes, large ω can substantially degrade the convergence guarantee. This phenomenon has been consistently observed both in theory and in practice (He et al., 2023; Compagnoni et al., 2025). Moreover, such compression-dependent penalties also persist in the stochastic guarantees of more sophisticated methods, including DIANA (Mishchenko et al., 2023) and VR-MARINA (Gorbunov et al., 2021), despite their variance-reduction mechanisms. This raises a fundamental question:

Is the ω-dependent degradation in the stochastic error term an inherent consequence of unbiased gradient compression, or is it an artifact of how compression noise is handled during optimization?

In the following section, we show that this leading-order degradation is *not inherent*. We introduce Intermittent Error Recycling, which recovers the leading stochastic term of uncompressed distributed SGD and confines compression-dependent effects to faster-decaying terms, while, crucially, respecting the prescribed communication budget.

4 Intermittent Error Recycling

In this section, we introduce our framework for mitigating compression-induced variance inflation. We first develop the intuition behind the method, then formalize the algorithm and establish its convergence guarantees. Finally, we extend our framework to SGD with momentum.

4.1 Why Compression Noise Persists

Unbiased gradient compression preserves the expected update, but fundamentally alters the variance structure of the optimization process. In compressed SGD, each update introduces additional *compression noise* that is added to the inherent stochastic gradient noise, inflating the effective noise level and degrading the leading stochastic convergence term.

More sophisticated methods mitigate compression error through control variates or gradient differences. However, these mechanisms still rely on compressed signals, which similarly introduce fresh compression noise. Thus, reducing compression error does not necessarily eliminate its contribution to the leading stochastic term.

The central question is thus not simply how much noise compression introduces, but *how* it is handled temporally. Repeatedly transmitting compressed corrections handles compression noise on the same temporal scale as stochastic noise. This iteration-wise treatment couples their variance contributions in the leading convergence term, incurring the variance-inflation penalty. This motivates a different strategy: accumulating compression errors and correcting them *exactly*, but only intermittently.

4.2 Key Idea: Temporal Amortization of Compression Error

While stochastic noise is not directly observable, *compression noise is fully measurable*. This distinction enables explicit correction, but correcting compression error exactly at every iteration is communication-intensive. Rather than compressing and reinjecting residuals or repeatedly transmitting compressed gradient differences, we propose a different strategy based on a crucial observation:

Compression noise need not be corrected at every iteration to preserve convergence.

Instead of correcting compression error at every iteration, we *accumulate* compression errors locally over multiple steps and transmit a correction only intermittently. Crucially, this correction is sent *uncompressed*, canceling the accumulated residual without introducing further distortion. We show that correcting errors exactly but infrequently eliminates the leading variance-inflation penalty.

This mechanism does not increase the total communication cost relative to the compressed baseline. We use more aggressive compression during regular iterations, making them cheaper and amortizing the cost of each exact correction over these intervening steps. The correction period K is therefore determined explicitly by the prescribed communication budget, as detailed in Section 5.

4.3 Algorithm Description

We now formalize the proposed method. Each worker maintains a local error buffer that accumulates the discrepancy between its stochastic gradients and their compressed versions, initialized as $E_{0,i} = 0$.

At iteration t, each worker i computes a stochastic gradient $v_{t,i} = \nabla f_i(x_t, z_{t,i})$, compresses it as $g_{t,i} = \mathcal{C}(v_{t,i})$, and accumulates the compression error: $E_{t,i} = E_{t-1,i} + (v_{t,i} - g_{t,i})$. On regular iterations, each worker transmits $g_{t,i}$, and the server averages the received gradients and updates the model: $x_{t+1} = x_t - \eta \left(\frac{1}{M} \sum_{i=1}^{M} g_{t,i} \right)$.

Every K iterations, when $t \bmod K = 0$, each worker adds its accumulated correction to the compressed gradient and transmits the combined update $g_{t,i} + E_{t,i}$ *without compression*. The server incorporates these corrected updates:

$$x_{t+1} = x_t - \eta \left(\frac{1}{M} \sum_{i=1}^{M} \left(g_{t,i} + E_{t,i} \right) \right),$$

and the buffers are then reset, $E_{t,i} \leftarrow 0$. These periodic corrections account for the accumulated compression residual without introducing further distortion. This mechanism is key to decoupling the effects of compression and stochastic noise, recovering the $\mathcal{O}(\sigma/\sqrt{MT})$ leading stochastic term of uncompressed SGD, as established in the next subsection. The method is summarized in Algorithm 1.

4.4 Convergence Guarantees

We now present the main guarantee of IER: periodic exact corrections recover the leading stochastic term of uncompressed distributed SGD, with compression-dependent effects confined to faster-decaying terms.

Theorem 4.1. *Suppose Assumptions 2.1 and 2.2 hold, and let $\mathcal{C}$ be an unbiased compressor with compression parameter ω. Consider Algorithm 1 with M workers, correction period $K \geq 1$, and an appropriately chosen stepsize η. Then, the iterates $\{x_t\}_{t=1}^{T}$ satisfy:*

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E} \|\nabla f(x_t)\|^2 \in \mathcal{O}\left(\frac{\Delta_1 L}{T} + \frac{\Delta_1 L \sqrt{\omega} \sqrt{K-1}}{\sqrt{M} T} + \frac{((\sigma + \xi)\sqrt{\omega} L \Delta_1)^{2/3} (K-1)^{1/3}}{M^{1/3} T^{2/3}} + \frac{\sigma \sqrt{2 \Delta_1 L}}{\sqrt{MT}} \right)$$

where ξ quantifies heterogeneity via the standard bound $\frac{1}{M} \sum_{i=1}^{M} \|\nabla f_i(x) - \nabla f(x)\|^2 \leq \xi^2, \forall x$. Theorem 4.1 shows that IER recovers the $\mathcal{O}(\sigma/\sqrt{MT})$ leading stochastic term of uncompressed distributed SGD, *independently of the compression factor* ω. Thus, IER eliminates the compression-induced variance inflation that persists in the leading stochastic guarantees of existing methods. The proof is provided in Appendix A.

Algorithm 1: Distributed SGD with IER

Input: $x_1 \in \mathbb{R}^d$, $E_{0,i} = 0$, stepsize η, $K \geq 1$, compressor $\mathcal{C}$
for $t = 1$ **to** T **do**
  Broadcast x_t to all workers
  for $i = 1$ **to** M **in parallel do**
    Sample $z_{t,i} \sim \mathcal{D}_i$
    $v_{t,i} = \nabla f_i(x_t, z_{t,i})$
    $g_{t,i} = \mathcal{C}(v_{t,i})$
    $\varepsilon_{t,i} = v_{t,i} - g_{t,i}$
    $E_{t,i} = E_{t-1,i} + \varepsilon_{t,i}$
    if $t \bmod K \neq 0$ **then**
      Send $h_{t,i} = g_{t,i}$
    else
      Send $h_{t,i} = g_{t,i} + E_{t,i}$
      $E_{t,i} = 0$
    end if
  end for
  $g_t = \frac{1}{M} \sum_{i=1}^{M} h_{t,i}$
  $x_{t+1} = x_t - \eta g_t$
end for

Algorithm 2: Distributed SGD-M with IER

Input: $x_1 \in \mathbb{R}^d$, $E_{0,i} = 0$, $m_{0,i} = 0$, stepsize η, $\beta \in (0, 1)$, $K \geq 1$, compressor $\mathcal{C}$
for $t = 1$ **to** T **do**
  Broadcast x_t to all workers
  for $i = 1$ **to** M **in parallel do**
    Sample $z_{t,i} \sim \mathcal{D}_i$
    $v_{t,i} = \nabla f_i(x_t, z_{t,i})$
    $m_{t,i} = \beta v_{t,i} + (1 - \beta) m_{t-1,i}$
    $g_{t,i} = \mathcal{C}(m_{t,i})$
    $\varepsilon_{t,i} = m_{t,i} - g_{t,i}$
    $E_{t,i} = E_{t-1,i} + \varepsilon_{t,i}$
    if $t \bmod K \neq 0$ **then**
      Send $h_{t,i} = g_{t,i}$
    else
      Send $h_{t,i} = g_{t,i} + E_{t,i}$
      $E_{t,i} = 0$
    end if
  end for
  $g_t = \frac{1}{M} \sum_{i=1}^{M} h_{t,i}$
  $x_{t+1} = x_t - \eta g_t$
end for

Compression-dependent effects are confined to terms decaying as $1/T$ and $1/T^{2/3}$, faster than the stochastic term. Moreover, this recovery is achieved without increasing the average communication cost relative to the compressed baseline: K is determined explicitly by the communication budget, with cheaper compressed iterations offsetting the cost of exact corrections, as detailed in Section 5.

The central conclusion is that *leading-order variance inflation is not an unavoidable cost of compression*. By correcting accumulated errors *exactly, but only intermittently*, IER changes how compression noise enters the convergence guarantee, retaining the communication savings of compression without degrading the leading stochastic term.

4.5 Extension to SGD-momentum

To facilitate better stability in practical training scenarios, we extend IER to SGD with Polyak's momentum. Each machine maintains a local momentum vector: $m_{t,i} = \beta v_{t,i} + (1 - \beta) m_{t-1,i}$ and applies compression and error recycling to this vector. The resulting method, presented in Alg. 2, admits convergence guarantees analogous to those of Alg. 1. We provide the full analysis in App. B.

5 Communication Complexity

IER preserves the communication budget of a target compressed baseline by using more aggressive compression during regular iterations to offset the cost of exact recycling steps. The savings from these cheaper updates fund the periodic transmission of accumulated compression errors. The recycling period K follows directly from this budget balance.

Specifically, let $\mathcal{C}$ denote the baseline compressor, with per-iteration communication cost $c(\mathcal{C})$, and let $\mathcal{C}'$ denote the more aggressive compressor used by IER, with $c(\mathcal{C}') < c(\mathcal{C})$. Let $c_{\max}$ denote the cost of an uncompressed update. Over a period of K iterations, IER transmits $K - 1$ compressed updates and one uncompressed corrected update for an average cost of $\frac{1}{K}((K-1)c(\mathcal{C}') + c_{\max})$ per update, incurring over T iterations a total communication cost bounded by:

$$\frac{T}{K}((K-1)c(\mathcal{C}') + c_{\max}),$$

while regular compression incurs a total cost of $Tc(\mathcal{C})$. Matching these costs determines the recycling period explicitly:

$$K = \left\lceil \frac{c_{\max} - c(\mathcal{C}')}{c(\mathcal{C}) - c(\mathcal{C}')} \right\rceil. \tag{5}$$

Thus, once the baseline budget and regular-step compressor $\mathcal{C}'$ are specified, the correction period is fixed, and we have $C_{\mathrm{IER}}^{\mathrm{total}} \leq Tc(\mathcal{C})$. Each exact correction is fully funded by the communication savings from the preceding compressed iterations, ensuring that IER does not increase the total communication cost over the training horizon.

The communication-budget guarantee above ensures that IER incurs an average communication cost of at most $c(\mathcal{C})$ per iteration. Combining this cost with the iteration complexity implied by Theorem 4.1 yields the total communication required to reach an ϵ-stationary point. We formalize this in Corollary 5.1.

Corollary 5.1 (Communication complexity of IER)**.** *Suppose the conditions of Theorem 4.1 hold, and let $\mathcal{C}'$ denote the compressor used by IER, with compression parameter ω. Let $\mathcal{C}$ be a target baseline compressor with communication cost $c(\mathcal{C})$, and choose*

$$K = \left\lceil \frac{c_{\max} - c(\mathcal{C}')}{c(\mathcal{C}) - c(\mathcal{C}')} \right\rceil,$$

where $c(\mathcal{C}') < c(\mathcal{C})$ and $c_{\max}$ is the cost of an uncompressed update. Then IER incurs an average communication cost of at most $c(\mathcal{C})$ per iteration. Moreover, to achieve

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\|\nabla f(x_t)\|^2 \leq \epsilon,$$

its total communication cost satisfies

$$C_{\mathrm{IER}}^{\mathrm{total}} \in \mathcal{O}\left(c(\mathcal{C}) \left[\frac{1}{\epsilon} + \frac{\sqrt{\omega(K-1)}}{\sqrt{M}\,\epsilon} + \frac{\sqrt{\omega(K-1)}}{\sqrt{M}\,\epsilon^{3/2}} + \frac{1}{M\epsilon^2} \right] \right),$$

where fixed problem-dependent quantities are suppressed.

The stochastic contribution $1/(M\epsilon^2)$ is free of compression-induced variance inflation and matches that of *uncompressed* distributed SGD, while compression-dependent effects are confined to the lower-order ϵ^{-1} and $\epsilon^{-3/2}$ contributions. Notably, IER achieves this guarantee at an average communication cost of at most $c(\mathcal{C})$ per iteration, retaining the benefits of compression without the leading-order variance penalty.

6 Experiments

We evaluate intermittent error recycling on image classification using ResNet-18 trained on ImageNet-Mini, considering both $M = 4$ and $M = 32$ workers. We compare against uncompressed distributed SGD and established compressed-optimization baselines, EF21-SGDM (Fatkhullin et al., 2023), VR-MARINA (Gorbunov et al., 2021), and QSGD (Alistarh et al., 2017). We consider both sparsification and quantization.

We report accuracy against both training epochs and cumulative communication, separating optimization behavior from communication efficiency. Communication cost is measured as cumulative transmitted bits, including the cost of indices for sparsification and the full-vector transmission on recycling steps.

In the sparsification setting, IER uses Rand-k with $k' = 5{,}000$ and $K = 1000$, matched to the communication budget of $k = 10{,}000$ sparsification; EF21-SGDM baselines use Top-k, the latter following its standard contractive-compressor setting. VR-MARINA, uses rand-k with $k = 10{,}000$. For quantization, QSGD and VR-MARINA use $s = 2$, while IER uses $s' = 1$ and $K = 14$. In both settings, K follows directly from the communication-budget matching rule in Sec. 5. Full implementation details and additional experiments on BERT-base/SST-2 and CIFAR-10, including comparisons with DIANA, are provided in App. D.

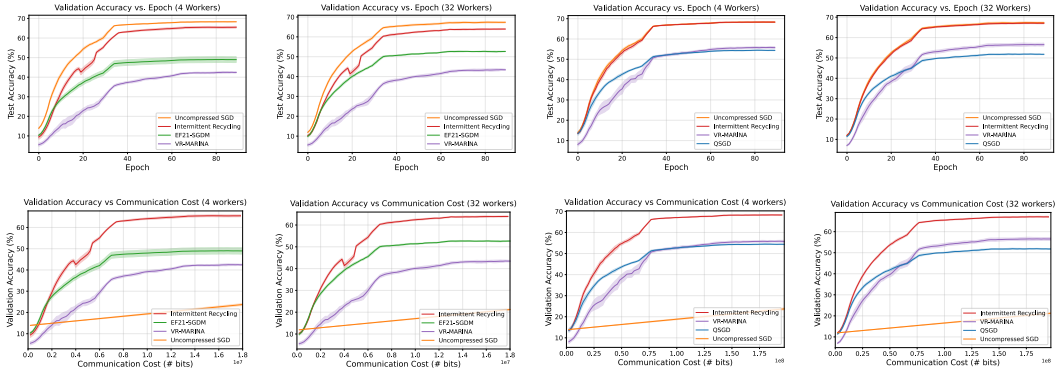


Figure 1.a: ImageNet-Mini with ResNet-18 under sparsification ($k = 10{,}000$). **Top**: validation accuracy vs. epochs. **Bottom**: validation accuracy vs. communication cost. **Left**: 4 workers. **Right**: 32 workers. EF21-SGDM uses Top-k with $k = 10{,}000$, following its standard contractive-compressor setting. IER uses the more aggressive Rand-k compressor with $k' = 5{,}000$ and $K = 1000$, while respecting the same communication budget. VR-Marina uses Rand-k with $k = 10{,}000$.

Figure 1.b: ImageNet-Mini with ResNet-18 under QSGD quantization ($s = 2$). **Top**: validation accuracy vs. epochs. **Bottom**: validation accuracy vs. communication cost. **Left**: 4 workers. **Right**: 32 workers. IER uses $s' = 1$ and $K = 14$ and substantially narrows the gap to uncompressed SGD under the same communication budget. VR-Marina uses QSGD with $s = 2$.

Figures 1.a-b show that IER substantially narrows the gap between aggressively compressed and uncompressed training across both sparsification and quantization, and across worker counts. The improvement is evident both as a function of training epochs and of cumulative communication, showing that the gains persist after accounting for communication. IER also compares favorably with the strong compressed-optimization baselines EF21-SGDM and VR-MARINA.

These results corroborate the central prediction of our analysis: accumulating compression errors and correcting them exactly but intermittently can substantially mitigate the degradation induced by aggressive compression without increasing communication costs.

7 Conclusion

In this work, we showed that the leading-order stochastic degradation commonly associated with unbiased gradient compression is not fundamental, but rather a consequence of how compression noise is handled temporally. By exploiting the fact that compression error is fully observable, we introduced intermittent error recycling, a simple yet principled mechanism that amortizes compression errors over time while preserving the communication budget. Our theoretical analysis demonstrates that IER recovers the leading stochastic convergence term of uncompressed distributed SGD, while confining compression-dependent effects to faster-decaying terms. Empirical results corroborate these findings, showing that intermittent recycling substantially narrows the gap between aggressively compressed and uncompressed training under matched communication budgets. Together, these results demonstrate that temporally decoupling compression noise from stochastic gradient noise can mitigate compression-induced variance inflation without sacrificing the communication benefits of compressed training.

AI use statement

In this work, we have not used generative AI tools for any tasks for which disclosure is required under the ICLR 2027 AI Policy for Authors. Additionally, we used generative AI tools to polish the writing and improve the readability of the manuscript. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

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

Dan Alistarh, Torsten Hoefler, Mikael Johansson, Sarit Khirirat, and Cedric Renggli. The convergence of sparsified gradient methods. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd: compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, 2018.

Aleksandr Beznosikov, Samuel Horvath, Peter Richtárik, and Mher Safaryan. On biased compression for distributed learning. *J. Mach. Learn. Res.*, 24:276:1–276:50, 2020.

Enea Monzio Compagnoni, Rustem Islamov, Frank Norbert Proske, and Aurelien Lucchi. Unbiased and sign compression in distributed learning: Comparing noise resilience via SDEs. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025.

Tehila Dahan and Kfir Yehuda Levy. SLowcalSGD : Slow query points improve local-SGD for stochastic convex optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Tehila Dahan, Roie Reshef, Sharon Goldstein, and Kfir Y. Levy. Bringing order to asynchronous sgd: Towards optimality under data-dependent delays with momentum, 2026. URL https://arxiv.org/abs/2605.02043.

Ron Dorfman, Naseem Yehya, and Kfir Y. Levy. Dynamic byzantine-robust learning: adapting to switching byzantine workers. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, 2024.

Ilyas Fatkhullin, Alexander Tyurin, and Peter Richtárik. Momentum provably improves error feedback! In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, 2023.

Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM J. Optim.*, 23:2341–2368, 2013.

Eduard Gorbunov, Konstantin P. Burlachenko, Zhize Li, and Peter Richtarik. Marina: Faster nonconvex distributed learning with compression. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3788–3798. PMLR, 2021.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *International Conference on Machine Learning (ICML)*, pp. 1737–1746. PMLR, 2015.

Yutong He, Xinmeng Huang, and Kun Yuan. Unbiased compression saves communication in distributed optimization: When and how much? In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Xinmeng Huang, Ping Li, and Xiaoyun Li. Stochastic controlled averaging for federated learning with communication compression. In *The Twelfth International Conference on Learning Representations*, 2024.

Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian U. Stich, and Martin Jaggi. Error feedback fixes signsgd and other gradient compression schemes. *International Conference on Machine Learning*, 2019.

Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *ICLR*, 2018.

Konstantin Mishchenko, Eduard Gorbunov, Martin Takáč, and Peter Richtárik. Distributed learning with compressed gradient differences, 2023. URL https://arxiv.org/abs/1901.09269.

Peter Richtárik, Igor Sokolov, and Ilyas Fatkhullin. Ef21: A new, simpler, theoretically better, and practically faster error feedback. In *Advances in Neural Information Processing Systems*, 2021.

Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *Interspeech*, 2014.

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

Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified sgd with memory. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. Powersgd: Practical low-rank gradient compression for distributed optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Terngrad: Ternary gradients to reduce communication in distributed deep learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Zeev Zukerman, Bassel Hamoud, and Kfir Y. Levy. Beyond communication overhead: A multilevel monte carlo approach for mitigating compression bias in distributed learning. In *International Conference on Machine Learning (ICML)*, 2025.