第 39 回先端技術大賞応募論文

効率的な通信と収束保証を両立する
大規模 AI 学習における擬似非同期分散学習手法

長沼 大樹
Mila – ケベック人工知能研究所 / モントリオール大学
naganuma.hiroki@mila.quebec

1 緒言

2022 年末の ChatGPT 公開以来、大規模 AI モデルは私たちの生活を急速に変えつつある。医療診断の支援、法律文書の分析、教育における個別指導、さらにはプログラミングの自動化まで、その応用範囲は日々拡大している。しかし、こうした AI の恩恵の裏側には、膨大な計算コストという深刻な課題が潜んでいる。

最先端の AI モデルを 1 つ学習するだけで、数万台の GPU（画像処理装置を転用した並列計算プロセッサ）を数か月稼働させる必要があり、その電力消費量は一般家庭数千世帯分の年間消費電力に匹敵する。国際エネルギー機関（IEA）の報告書 (International Energy Agency, 2025) によれば、世界のデータセンターの電力消費は 2030 年までに約 945TWh に達し、日本全体の消費電力に匹敵する規模になると推計されている。AI 学習はその主要な増加要因である。大規模分散学習では、ワーカー間の通信同期が学習時間の大きな割合を占める。Meta 社の大規模システム分析 (Hsia et al., 2024) によれば、現行の分散学習において全 GPU 時間の 14–32%が計算と重複しない純粋な通信待ちに費やされている。さらに、モデルの大規模化とハードウェアの進化に伴い、この通信オーバーヘッドは将来的に 40–75%に達するとの予測もある (Pati et al., 2023)。例えば Meta 社の LLaMA-3(Dubey et al., 2024) の学習では 1 万 6 千台以上の GPU を同期的に運用しており、通信待ちの間もこれらの GPU は電力を消費し続けて CO_2 の排出へつながる。数十億円規模の計算資源のうち、その 14–32%が通信待ちに浪費されている計算になる。この通信ボトルネックの解消は、AI 開発の効率化のみならず、地球規模のエネルギー・環境問題に直結する喫緊の課題である。

この問題は日本にとって特に切実である。日本政府は 2025 年 12 月に閣議決定した「AI 基本計画」(内閣府, 2025) において、AI 計算基盤の整備を国家戦略の柱に位置づけている。しかし、世界の AI 向け GPU クラスタの性能シェアは米国が約 75%、中国が約 14%を占めるのに対し、日本は約 1%にとどまる (Pilz et al., 2025)。限られた計算資源から最大の成果を引き出す効率的な分散学習技術は、日本の AI 競争力を左右する基盤技術といえる。

本論文では、この課題に対する新たな分散学習手法「擬似非同期局所 SGD（PALSGD: Pseudo-Asynchronous Local SGD）」を提案する[1]。PALSGD の核心は、各計算機が通信なしに自律的にモデルの整合性を維持する「擬似同期」メカニズムにある。画像分類タスクでは従来手法と比較して 18.4%、言語モデリングタスクでは最大 24.4%の学習時間短縮を達成した。本研究は、筆者が主導して開発した分散最適化手法 PALSGD を中心に、その理論解析と実証実験を統合した研究成果である。

本論文の構成は以下の通りである。第 2 節で分散深層学習の仕組みと課題を概説し、第 3 節で提案手法 PALSGD の詳細を述べる。第 4 節で理論的な収束保証を示し、第 5 節で実験結果を報告する。最後に第 6 節で結論と将来の展望を述べる。

[1]本研究の一部は、国際学術誌 Transactions on Machine Learning Research (TMLR) に掲載済みである (Naganuma et al., 2025)。本論文は産経新聞の一般読者を対象に、新たな図表と社会的文脈を加えて日本語で書き下ろしたものである。

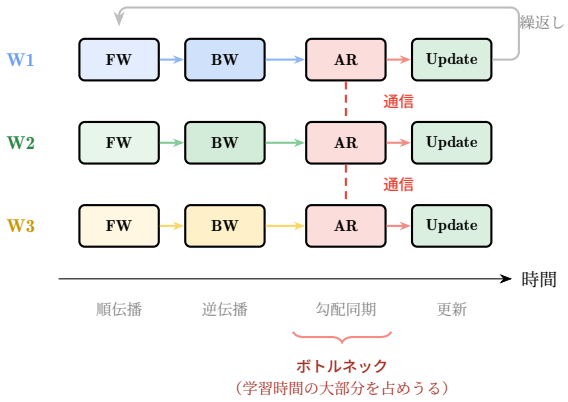


Figure 1: データ並列学習（DDP）の 1 ステップ。各ワーカーが順伝播（FW）・逆伝播（BW）で勾配を計算した後、All-Reduce（AR）により全ワーカー間で勾配を集約・平均する。この通信操作がボトルネックとなり、大規模学習では学習時間の大きな割合を占める。

2　分散深層学習の仕組みと課題

2.1　深層学習における最適化問題

深層学習の目標は、大量のデータからパターンを学習するニューラルネットワーク（人間の脳の神経回路を模した数学的構造）のパラメータを最適に調整することである。モデルは入力（画像やテキスト）を受け取り、出力（分類結果や次の単語の予測）を返す。このモデルが持つ多数のパラメータ（重みと呼ばれる数値）を適切に調整することで、正しい出力が得られるようになる。

数学的には、深層学習は以下の最適化問題として定式化される：

$$\min_{x\in\mathbb{R}^d} F(x), \quad F(x) = \mathbb{E}_{\xi\sim\mathcal{D}} f(x,\xi), \tag{1}$$

ここで x はモデルのパラメータ（最先端モデルでは数千億個の数値の集合）、ξ は学習データの各サンプル、$f(x,\xi)$ はモデルの予測と正解のズレを数値化した「損失関数」を表す。学習の目標は、この損失関数を最小化するパラメータ x を見つけることであり、確率的勾配降下法（SGD）により「勾配」（改善すべき方向と大きさ）を計算し、パラメータを繰り返し更新していく。

2.2　データ並列学習と通信オーバーヘッド

大規模モデルの学習を高速化するため、K 台の計算機（以下、ワーカー）が並列にデータを処理する「データ並列学習」が広く用いられている（図 1）。最も標準的な手法である DDP（Distributed Data Parallel）(Li et al., 2020a) では、各ワーカーがモデルの完全なコピーを保持し、自身に割り当てられたデータ断片で勾配を計算する。その後、All-Reduce と呼ばれる通信操作により全ワーカーの勾配を集約・平均化し、全ワーカーが同一のパラメータ更新を行う。

しかし、DDP では**毎回の学習ステップ**で同期が必要であり、モデルのパラメータ数が数十億に達する場合、1 回の同期で数ギガバイトものデータを全ワーカー間で交換しなければならない。ワーカー数が増えるほど通信路の負荷も増大し、特にワーカーが地理的に離れた場所に分散している場合、通信オーバーヘッドは深刻な問題となる。

2.3　通信頻度削減の既存アプローチとその限界

通信オーバーヘッド削減の代表的手法として、Local SGD(Stich, 2018) がある。各ワーカーが H ステップの間独立して学習し、H ステップ後に一斉同期を行うことで同期回数を $1/H$ に削減する（図 2(b)）。しかし、H を大きくしすぎると各ワーカーのモデルが異なる方向に更新される「モデル発散」が起こり、学習性能が著しく低下する (Lin et al., 2018)。実用上 H は 8 ステップ程度が限界とされ、通信削減効果は限定的である。

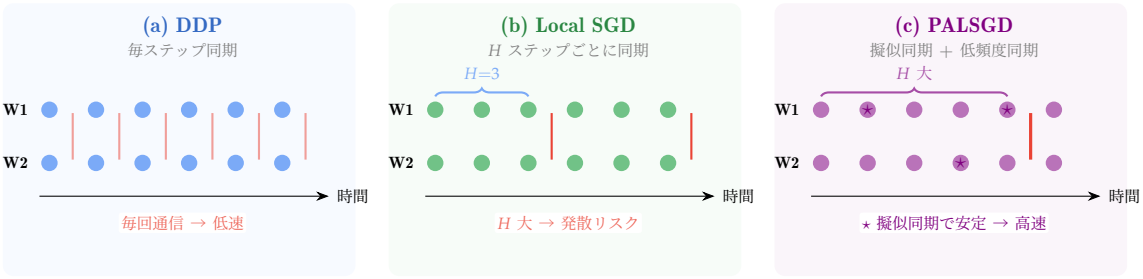


Figure 2: 3 つの分散学習手法の通信パターン比較（2 ワーカーの例）。(a) DDP は毎ステップで全ワーカー間の同期（赤線）を行う。(b) Local SGD は H ステップごとに同期するが、H が大きいとモデルが発散する。(c) PALSGD は確率的な擬似同期（⋆、通信不要）によりモデルの整合性を維持し、同期間隔を大幅に拡大できる。

より新しい手法 DiLoCo(Douillard et al., 2023) は、各ワーカー内部の学習（内部最適化）と全体の集約（外部最適化）を分離するアプローチを採る。内部に AdamW(Kingma & Ba, 2014)、外部に Nesterov モメンタム (Sutskever et al., 2013) を用いることで Local SGD より安定した学習を実現するが、同期間隔が大きくなるとやはり性能低下が見られる。

連合学習分野の FedProx(Li et al., 2020b) は、各ワーカーのモデルが共有モデルから離れすぎないようペナルティ項を追加する手法であり、本研究の提案手法と類似の着想を含む。しかし、追加のハイパーパラメータ調整が必要であり、通信頻度の削減には直接寄与しない。これらの限界を克服するため、我々は PALSGD を提案する。

3 提案手法: 擬似非同期局所 SGD (PALSGD)

3.1 着想の原点

PALSGD の核心的アイデアは、次の問いから生まれた。「通信コストの高い『真の同期』を減らしつつ、モデルの整合性を保つ方法はないか？」

Local SGD でモデルが発散する根本原因は、同期と同期の間にワーカー間のモデルの差が拡大し続けることにある。筆者らの先行研究 (Guille-Escuret et al., 2024) では、ニューラルネットワークの最適化経路が安定した幾何学的構造を持ち、更新方向の一貫性（コサイン類似度の安定性）が学習の収束に寄与することが明らかになった。この知見は、各ワーカーの局所モデルを共有モデルに向けて定期的に修正すれば、最適化経路の一貫性を維持できることを示唆している。すなわち、モデル間の差の拡大を**通信なしに**抑制できれば、同期間隔を大幅に延長しても学習の安定性を保てるはずである。

PALSGD では、各ワーカーが最後の同期時点での共有モデル（全ワーカーの平均モデル）のコピーを自身のメモリ上に保持する。学習の各ステップで、ワーカーは確率的にこの共有モデルのコピーに向かって自身のモデルを「引き寄せる」操作を行う。この操作を「擬似同期」と呼ぶ（図 3）。重要なのは、この擬似同期は各ワーカー内部の計算のみで完結し、**ワーカー間の通信を一切必要としない**点である。

3.2 アルゴリズムの詳細

PALSGD の学習手順を具体的に説明する。K 台のワーカーが同一の初期モデル $x^{(0)}$ から学習を開始し、各ワーカー k は 2 つのモデル状態を保持する。1 つは自身の局所モデル $x_k^{(t)}$（独自に更新するモデル）、もう 1 つは最後の同期で得られた共有モデル $x^{(t)}$（全ワーカー共通の参照点）である。

ステップ 1: 確率的な更新選択　各学習ステップで、ワーカーは確率 p（例えば 5%）で「擬似同期ステップ」を、確率 $1-p$（95%）で「通常の勾配更新ステップ」を選択する。学習の大部分は通常の勾配更新であり、時折（20 回に 1 回程度）擬似同期が挟まれる。

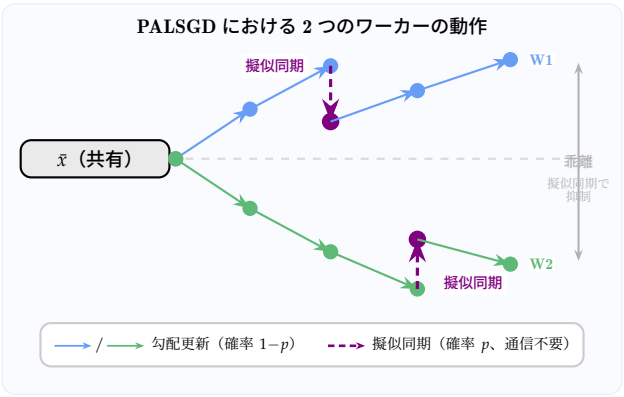


Figure 3: PALSGD における擬似同期メカニズム（2 ワーカーの例）。W1（青）と W2（緑）は共有モデル $\bar{x}$ から出発し、各自の勾配更新により異なる方向に乖離していく。各ワーカーは確率 p で擬似同期（紫矢印）を実行し、局所モデルを $\bar{x}$ に向かって引き戻す。この操作は通信不要であり、ワーカー間の乖離を効果的に抑制する。

ステップ 2-A: 勾配更新（確率 $1-p$ で実行）　自身のデータ断片からサンプルを取得し、損失関数の勾配を計算してモデルを更新する：

$$x_k^{(t+1)} = \textsc{InnerOPT}\left(x_k^{(t)},\ \nabla f(x_k^{(t)}, \xi),\ \frac{\alpha_t}{1-p}\right). \tag{2}$$

ここで α_t は学習率、InnerOPT は内部最適化器（AdamW 等）を表す。

ステップ 2-B: 擬似同期（確率 p で実行）　局所モデルを共有モデルに向かって引き寄せる：

$$x_k^{(t+1)} \leftarrow x_k^{(t)} - \frac{\alpha_t \eta_t}{p}\left(x_k^{(t)} - x^{(t)}\right). \tag{3}$$

η_t は引き寄せの強さを制御する混合率である。括弧内の $(x_k^{(t)} - x^{(t)})$ は局所モデルと共有モデルの「ズレ」を表し、ズレが大きいほど強く引き戻され、小さければほとんど影響を受けない。物理でいうバネの復元力と同様の仕組みであり、モデルの発散を自然に抑制する。

この更新は、二次のペナルティ項 $\frac{\eta_t}{2}\|x_k - x^{(t)}\|^2$ の勾配ステップとしても、指数移動平均（EMA）$(1-\beta)x_k^{(t)} + \beta x^{(t)}$（$\beta = \alpha_t\eta_t/p$）としても解釈でき、更新のばらつきを抑える効果がある。

ステップ 3: 周期的な真の同期（H ステップごと）　H ステップごとに、全ワーカーが All-Reduce 通信を行い差分を集約する：

$$\Delta^{(t)} = \textsc{AllReduce}\left(x^{(t-1)} - x_k^{(t)}\right), \quad x^{(t+1)} \leftarrow \textsc{OuterOPT}(x^{(t)}, \Delta^{(t)}). \tag{4}$$

外部最適化器（Nesterov モメンタム等）で共有モデルに適用し、全ワーカーの学習成果を統合する。擬似同期によりワーカー間の乖離が小さく保たれるため、PALSGD では同期間隔 H を Local SGD よりはるかに大きく設定できる。

3.3　擬似同期の直感的理解

擬似同期の効果は、濃霧の斜面で谷底への進路を探す測量隊にたとえられる。1 人の判断だけでは足元の小さな凹凸に左右されやすいが、K 人が別々の地点で同時に傾きを測り、その結果を平均すれば、より確かな進行方向をより速く決められる。これが、分散学習でワーカー数を増やす利点である。

DDP は、1 歩ごとに全員が無線で結果を共有して進む方法で、確実だが連絡待ちが多い。Local SGD は、しばらく各自で進んでから結果を持ち寄る方法で、速いが隊がばらけやすい。PALSGD は、各隊員が最後に共有した進路メモを手元に持ち、無線しない間も、ときどき自分で向きを修正する方法である。この自己修正は通信なしに行えるため、隊全体のずれを抑えつつ、同期回数を大きく減らせる。

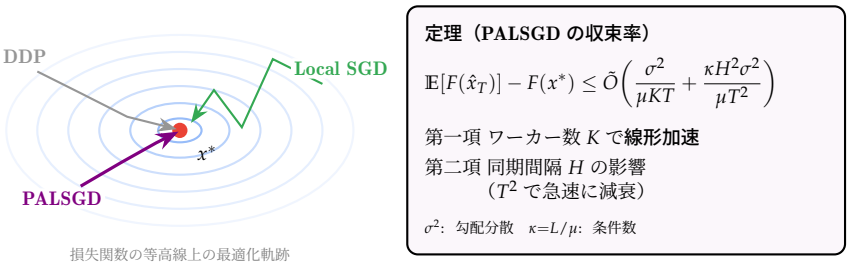


Figure 4: PALSGD の収束性。**左**：損失関数の等高線上の最適化軌跡の模式図。DDP は安定だが低速、Local SGD は振動しやすく、PALSGD は安定かつ高速に最適解 x^* に収束する。**右**：収束率の理論的保証。

3.4 実装上の工夫

実用性を高めるため、2 つの工夫を施している。第一に、Post-Local SGD(Lin et al., 2018) の知見に倣い、学習初期に DDP によるウォームアップを行う。学習初期は勾配が大きく不安定なため、この期間は頻繁な同期で安定性を確保し、モデルがある程度収束した後に PALSGD に切り替える。第二に、DiLoCo(Douillard et al., 2023) で有効性が検証された「分離型最適化」を採用する。内部最適化器として AdamW(Kingma & Ba, 2014) を、外部最適化器として Nesterov モメンタム (Sutskever et al., 2013) を使用し、局所的な学習と全体の集約の両方を効率化する。

メモリ面では共有モデルのコピーと外部最適化器の状態（モデル 1 個分程度）の追加が必要となるが CPU memory などに offload 可能である。通信面では擬似同期が完全に局所計算であるため、通信コストの増加は一切ない。

4 理論的保証

PALSGD の信頼性を裏付けるため、簡略化された設定における収束性の理論解析を行った。内部・外部最適化器を共に SGD とし、目的関数が以下の性質を満たす場合を分析した。すなわち、L-滑らかさ（勾配が急激に変化しないこと）、μ-強凸性（関数が唯一の最小値を持つ「お椀」型の形状）、データ分布の同一性（各ワーカーが同じ統計的性質のデータを扱うこと）、最適解における勾配分散の有界性（勾配のばらつきが有限であること）を仮定する。

定理 1 (PALSGD の収束率、非形式的)**.** *PALSGD* が T ステップ、K 台のワーカー、同期間隔 H で学習した場合、擬似同期確率 $p \le 1/2$ の条件下で、適切なハイパーパラメータの選択により：

$$\mathbb{E}[F(\hat{x}_T)] - F(x^*) \le \tilde{O}\left(\frac{\sigma^2}{\mu KT} + \frac{\kappa H^2\sigma^2}{\mu T^2}\right), \tag{5}$$

ここで $\hat{x}_T$ は *PALSGD* の出力の加重平均、$\kappa = L/\mu$ は条件数、σ^2 は勾配の分散を表す。

この結果の意味は明快である。

第一項 $\frac{\sigma^2}{\mu KT}$ は**線形加速**を示す。ワーカーを 2 倍にすれば誤差がほぼ半分になり、分散学習として理想的なスケーリング特性を持つ。注目すべきは、この項が同期間隔 H にも擬似同期確率 p にも依存しない点であり、PALSGD の擬似同期メカニズムが収束の主要項に悪影響を与えないことを理論的に保証している。十分なステップ数 T のもとでは、この第一項が支配的となり、統計学における Cramer-Rao 下界（パラメータ推定の理論的限界）と一致する最適な速度で収束する。

第二項 $\frac{\kappa H^2\sigma^2}{\mu T^2}$ は同期間隔 H の影響を反映する。H が増えると収束が若干遅くなるが、T の 2 乗に反比例して減衰するため、十分な学習を行えば影響は無視できる。通信が頻繁な場合（H が定数）には強凸関数に対する SGD の理論的下界と一致する (Koloskova et al., 2020)。

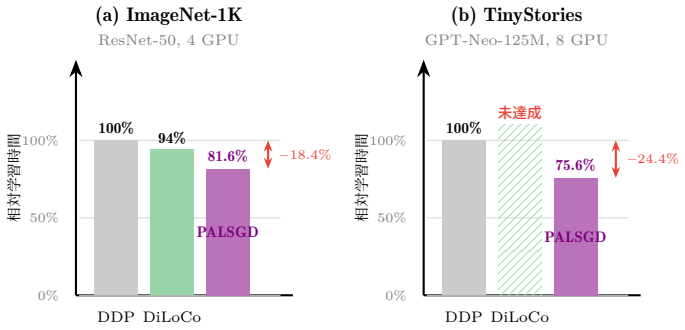
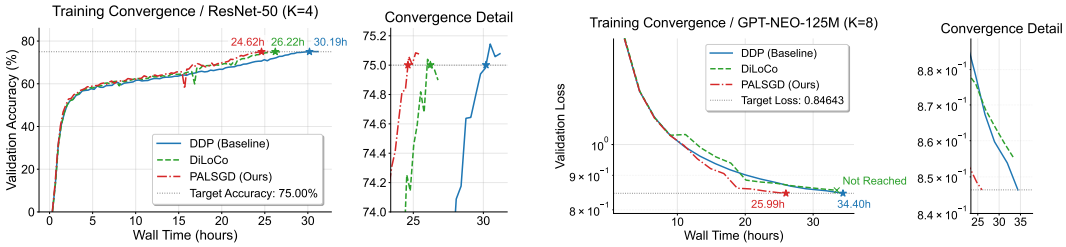


Figure 5: 主要実験結果の比較（DDP の学習時間を 100%とした相対値）。(a) ImageNet-1K では PALSGD が 18.4%の学習時間短縮を達成。(b) TinyStories では 24.4%の短縮を達成し、DiLoCo は目標損失に到達できなかった。



(a) ImageNet-1K（ResNet-50, K=4, H=64）　(b) TinyStories（GPT-Neo-125M, K=8, H=16）

Figure 6: 学習曲線の詳細比較。(a) ImageNet-1K では、PALSGD が目標精度 75%に 24.62 時間で到達し、DDP（30.19 時間）より 18.4%高速。(b) TinyStories では、PALSGD が 25.99 時間で目標損失に到達し、DDP（34.40 時間）より 24.4%高速。DiLoCo は目標に未達成。

5　実験と結果

PALSGD の有効性を実証するため、画像分類と言語モデリングの 2 つのタスクで包括的な実験を行った。比較対象は DDP、DiLoCo(Douillard et al., 2023) の 2 手法である。CIFAR-10 データセットを用いた予備実験により、Local SGD はワーカー数の増加に伴い顕著に性能が劣化するため、以下の大規模実験では DDP と DiLoCo を主要な比較対象とした。

5.1　実験設定

画像分類には ImageNet-1K(Deng et al., 2009)（約 128 万枚、1000 カテゴリ）と ResNet-50(He et al., 2016)（約 2500 万パラメータ）を使用し、4 台の GPU ワーカーで $H = 64$ にて学習した。

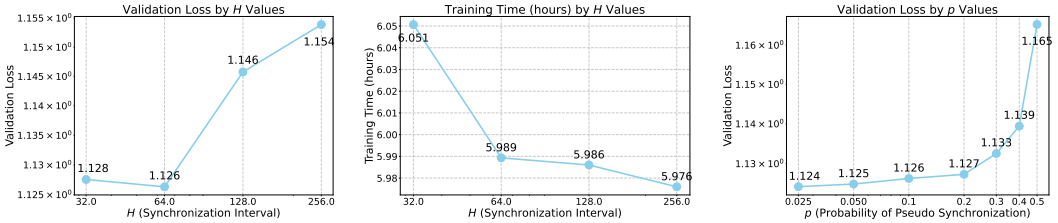
言語モデリングには TinyStories(Eldan & Li, 2023)（子供向け短編物語のデータセット）と GPT-Neo（800 万および 1 億 2500 万パラメータ）を使用し、4〜8 台のワーカーで $H = 16$ にて学習した。

すべてのアルゴリズムの学習率やハイパーパラメータは各設定で独立に最適化し、各手法が最良の条件で比較されることを保証した。

5.2　画像分類: ImageNet-1K

ImageNet-1K の実験（$K = 4, H = 64$）では、3 手法すべてが目標検証精度 75%に到達した。PALSGD は DDP と比較して **18.4%**、DiLoCo と比較して **6%**の学習時間短縮を達成した（図 5(a)、図 6a）。

先行研究 (Ortiz et al., 2021) では Local SGD 系の手法で報告されていた精度低下が、PALSGD では観察されなかった。擬似同期がワーカー間の乖離を効果的に抑制し、Local SGD の弱点を



(a) 同期間隔 H と検証損失　(b) 同期間隔 H と学習時間　(c) 擬似同期確率 p と検証損失

Figure 7: ハイパーパラメータのアブレーション実験（GPT-Neo-8M, K=8）。(a) H を大きくすると検証損失が悪化する（$H \geq 128$ で顕著）。(b) H の増大に伴い学習時間は一貫して短縮される。(c) p は 0.025–0.1 の範囲で安定し、$p \geq 0.3$ で性能が低下する。H=64、$p \approx 0.05$ が最良のバランスを提供する。

克服していることを示唆している。なお、ImageNet-1K は DiLoCo の原論文 (Douillard et al., 2023) では評価されておらず、本実験は DiLoCo の画像分類性能を初めて報告するものでもある。

5.3 言語モデリング: TinyStories

TinyStories の実験では、PALSGD の優位性がさらに顕著に表れた。GPT-Neo-125M（$K = 8, H = 16$）では、DDP と比較して **24.4%**の学習時間短縮を達成した（図 5(b)、図 6b)。DiLoCo は同一の学習時間内に目標損失に到達できなかったのに対し、PALSGD は最速かつ最低の検証損失を達成した。同期間隔が比較的大きい設定において、擬似同期のない DiLoCo ではモデル整合性の維持が不十分であるのに対し、PALSGD は擬似同期により安定した学習を継続できることを示している。

GPT-Neo-8M（800 万パラメータ）でも DDP 比 **21.1%**の短縮を達成し、モデルサイズによらず一貫した改善が得られることを確認した。これらの実験で PALSGD の同期回数は DDP の 6.25%（93.75%の削減）であり、通信コストの観点から劇的な改善を実現している。

5.4 ハイパーパラメータの影響

GPT-Neo-8M（$K = 8$）でアブレーション実験を行い、同期間隔 H と擬似同期確率 p の影響を分析した（図 7)。

H を大きくするほど通信コストが削減される一方、$H \geq 128$ ではモデル発散により検証損失が悪化する。学習時間は H の増大に伴い一貫して短縮されるため、$H = 64$ 付近が効率と品質の最良のバランスを提供する。

p については、$p \approx 0.05$（5%）が多くの設定で良好であった。p が大きすぎると勾配更新の機会が減少し性能が低下する。一方、p が小さすぎると擬似同期の効果が薄れる。少量の擬似同期（5%程度）でもモデル乖離の抑制に十分効果的であるという発見は、PALSGD の実用上の大きな利点である。

6 結論と将来への展望

6.1 本研究のまとめ

本論文では、大規模 AI 学習における通信ボトルネックを解消する新手法 PALSGD を提案した。PALSGD の核心である確率的擬似同期メカニズムは、各ワーカーが通信なしに自律的にモデルの整合性を維持するという、シンプルでありながら効果的なアイデアに基づいている。

理論面では、ワーカー数に対する線形加速を達成し、強凸関数に対する SGD の理論的下界と一致する最適な収束率を証明した。実験面では、画像分類（ImageNet-1K）で 18.4%、言語モデリング（TinyStories）で 24.4%の学習時間短縮を達成し、同期回数を最大 93.75%削減した（図 8)。

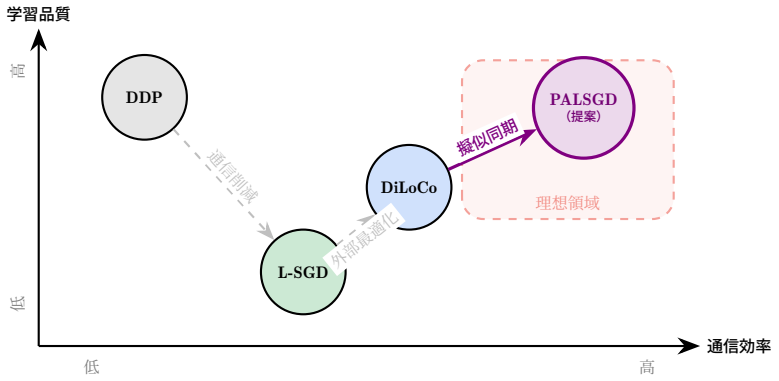


Figure 8: 分散学習手法の通信効率と学習品質における位置づけ。DDP は高品質だが通信効率が低く、Local SGD は通信効率は高いが品質が低下しやすい。PALSGD は擬似同期により両者を高い水準で両立する。

6.2 将来への展望

PALSGD が切り拓く可能性は、単なる学習時間の短縮にとどまらない。本手法がもたらす影響を、CO_2 削減、日本の AI 基盤強化、大規模言語モデル（LLM）のクラウド実装という 3 つの観点から論じる。

AI 学習の CO_2 削減への直接的寄与。 緒言で述べた通り、データセンターの電力消費は急速に拡大しており、AI 学習はその主要因である。PALSGD による通信待ち時間の削減は、その間の GPU 電力消費を直接的に排除する。本実験で実証した 18–24%の学習時間短縮は、同規模の電力消費と CO_2 排出の削減に直結する。例えば、LLaMA-3 の学習では約 27.5GWh の電力が消費されたと報告されている (Dubey et al., 2024)。このような大規模学習に本手法を適用した場合、通信待ち時間の削減だけで数 GWh 規模の電力節約が見込まれ、これは一般家庭約千世帯の年間消費電力に匹敵する。AI 開発の規模が拡大し続ける中で、学習効率の改善は CO_2 削減の最も直接的な手段の一つである。

日本の AI 計算基盤への影響。 日本政府は 2025 年度に AI 関連予算を大幅に拡充し、国内の計算基盤整備を加速させている。しかし、米国の大手テック企業が数十万台規模の GPU クラスタを運用する現状に対し、日本国内の計算資源は依然として限られている。PALSGD が実現する通信コストの大幅削減は、この格差を補う技術的な鍵となる。具体的には、東京・大阪間のように地理的に離れたデータセンターの GPU をネットワーク越しに協調させる「跨拠点分散学習」が現実的となる。PALSGD の擬似同期は通信遅延の大きい環境でも有効に機能するため、単一の巨大クラスタを持たない日本の研究機関や企業が、分散した中規模資源を束ねて大規模モデルの学習に挑む道が開ける。今後は、異種計算環境への適応として、擬似同期確率 p をワーカーの計算能力やネットワーク状況に応じて動的に調整するメカニズムの開発を進め、現実のクラウド環境への展開を目指す。

大規模言語モデルの社会実装に向けて。 ChatGPT に代表される LLM は、今後さらに多くの産業分野に浸透していく。しかし、最先端 LLM の学習コストは数十億円規模に達しており、日本語に特化した LLM の構築には依然として大きな障壁がある。PALSGD の通信削減技術を LLM の事前学習に適用すれば、同一の計算資源でより多くの学習ステップを実行でき、日本語 LLM の品質向上やファインチューニングコストの低減に寄与する。また、筆者らは並列最適化におけるバッチサイズの適応的制御に関する理論的枠組みも構築しており (Naganuma et al., 2026)、PALSGD との統合により通信量と計算量の同時最適化が期待できる。現在の理論は SGD と強凸関数の設定に基づいており、AdamW 等の適応的最適化器や非凸関数への理論的拡張、および勾配圧縮との併用が今後の重要な研究課題である。

研究の展望。 筆者がこの研究を通じて確信したのは、「通信なしの局所的整合性維持」という原理が、分散学習の本質的な制約を打破する可能性を秘めているということである。大規模 AI モデルの学習規模は数兆パラメータに達しつつあり、通信効率の問題はますます深刻化する。この原理をさらに発展させ、限られた計算資源で最大の成果を生み出す技術として確立することで、日本を含む世界中の研究機関が AI 開発に参画できる技術基盤の構築に貢献したい。

謝辞

本研究の共著者である Xinzhi Zhang 氏（ワシントン大学）、Man-Chung Yue 氏（香港大学）、Ioannis Mitliagkas 氏（Mila/モントリオール大学）、Philipp A. Witte 氏（Microsoft）、Russell J. Hewett 氏（NVIDIA）、Yin Tat Lee 氏（ワシントン大学/Microsoft）に深く感謝する。また、本研究は Mila、Digital Research Alliance of Canada、および Hong Kong Research Grants Council の支援を受けて実施された。

References

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Arthur Douillard et al. Diloco: Distributed low-communication training of language models. *arXiv preprint arXiv:2311.08105*, 2023.

Abhimanyu Dubey et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, 2023.

Charles Guille-Escuret, Hiroki Naganuma, Kilian Fatras, and Ioannis Mitliagkas. No wrong turns: The simple geometry of neural networks optimization paths. In *International Conference on Machine Learning (ICML)*, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Samuel Hsia, Alicia Golden, Bilge Acun, Newsha Ardalani, Zachary DeVito, Gu-Yeon Wei, David Brooks, and Carole-Jean Wu. MAD Max beyond single-node: Enabling large machine learning model acceleration on distributed systems. In *Proceedings of the 51st Annual International Symposium on Computer Architecture (ISCA)*, 2024.

International Energy Agency. Energy and ai, 2025. URL https://www.iea.org/reports/energy-and-ai/.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U Stich. A unified theory of decentralized sgd with changing topology and local updates. *arXiv preprint arXiv:2003.10422*, 2020.

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, 2020a.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *MLSys*, 2020b.

Tao Lin, Sebastian U Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't use large mini-batches, use local sgd. *arXiv preprint arXiv:1808.07217*, 2018.

Hiroki Naganuma, Xinzhi Zhang, Man-Chung Yue, Ioannis Mitliagkas, Russell J. Hewett, Philipp Andre Witte, and Yin Tat Lee. Pseudo-asynchronous local SGD: Robust and efficient data-parallel training. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=8VTrvS5vN7.

Hiroki Naganuma, Shagun Gupta, Youssef Briki, Ioannis Mitliagkas, Irina Rish, Parameswaran Raman, and Hao-Jun Michael Shi. Adaptive batch sizes using non-euclidean gradient noise scales for stochastic sign and spectral descent, 2026. URL https://arxiv.org/abs/2602.03001.

Jose Javier Gonzalez Ortiz, Joan Serrà, Takayuki Kishi, et al. Trade-offs of local sgd at scale: An empirical study. *arXiv preprint arXiv:2110.08133*, 2021.

Suchita Pati, Shaizeen Aga, Mahzabeen Islam, Nuwan Jayasena, and Matthew D. Sinclair. Tale of two cs: Computation vs. communication scaling for future transformers on future hardware. In *IEEE International Symposium on Workload Characterization (IISWC)*, pp. 140–153, 2023.

Konstantin F. Pilz, Robi Rahman, James Sanders, and Lennart Heim. Trends in AI supercomputers. *arXiv preprint arXiv:2504.16026*, 2025. Epoch AI Research.

Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

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

内閣府. Ai 基本計画：「信頼できる ai」による「日本再起」. Technical report, 内閣府科学技術・イノベーション推進事務局, 12 2025. URL https://www8.cao.go.jp/cstp/ai/ai_plan/aiplan_20251223.pdf. 閣議決定.