ADAPAD: Adaptive Parallel Deflation for PEFT with Self-Correcting Rank Discovery

Barbara Su
Stanford University

Fangshuo Liao
New York University

Anastasios Kyrillidis*
Rice University

Abstract

Fine-tuning large language models with LoRA [Hu et al., 2022] requires choosing a rank r before training starts. Existing approaches either extract rank-1 components sequentially [Liao et al., 2024, Vandchali et al., 2025], freezing each component's error permanently into every subsequent residual, or optimize the full low-rank factorization jointly [Zangrando et al., 2025, Shinwari and Usama, 2026] with guarantees that describe only the joint update, not individual rank-1 directions. We present ADAPAD (Adaptive Parallel Deflation), which trains all rank-1 components simultaneously: each worker refines its component against a deflation target built from the latest estimates of all predecessors, and as those estimates improve, the targets improve too. We call this property *self-correction*: deflation errors converge to zero over rounds rather than persisting as fixed residuals. On top of this backbone, ADAPAD adds *advance learning* (private pre-training before activation) and *per-module dynamic rank discovery* (importance-based growth until a shared budget is exhausted), making the rank distribution an output rather than an input. We prove that every component's error decays exponentially after a warm-up period, with a generalization bound that splits into a vanishing algorithmic term and an irreducible statistical floor. Empirically, ADAPAD is competitive with adaptive-rank LoRA baselines on GLUE with DeBERTaV3-base at matched parameter budgets, stays within two points of the best baseline on three math-reasoning benchmarks, and is competitive with fixed-rank LoRA on Qwen3-0.6B SQuAD/SQuAD v2 while deploying an adapter that is on average 30.7% smaller.

1 Introduction

Parameter-efficient fine-tuning (PEFT) is the standard way to adapt large pretrained models to downstream tasks. Among the many approaches [Houlsby et al., 2019, Li and Liang, 2021, Ben Zaken et al., 2022], Low-Rank Adaptation (LoRA) [Hu et al., 2022] stands out for its simplicity: for a pretrained weight $\mathbf{W} \in \mathbb{R}^{m\times d}$, LoRA freezes $\mathbf{W}$ and trains a low-rank update $\Delta\mathbf{W} = \mathbf{BA}$ with $\mathbf{B} \in \mathbb{R}^{m\times r}$, $\mathbf{A} \in \mathbb{R}^{r\times d}$ and rank $r \ll \min(m, d)$ a hyperparameter. LoRA is effective in practice, but leaves a fundamental design choice unresolved: how should r be chosen?

In practice r is set by hand and applied uniformly to every layer [Hu et al., 2022, Hayou et al., 2024, Liu et al., 2024], yet the appropriate rank varies across modules, tasks, and training stages. Adaptive-rank methods address this in different ways: AdaLoRA [Zhang et al., 2023b] prunes from high rank using SVD importance, IncreLoRA [Zhang et al., 2023a] grows from low rank by activating reserve components, DyLoRA [Valipour et al., 2023] trains across a range of ranks, and SoRA [Ding et al., 2023] learns sparse gates. Recent theoretical work analyzes the joint $\mathbf{BA}$ update: dEBORA [Zangrando et al., 2025] uses bilevel Frank-Wolfe with rank-identifiability guarantees, and ARD-LoRA [Shinwari and Usama, 2026] adds learnable per-head scaling with a stationary-point theorem. Both treat the full low-rank matrix as one object; we instead decompose the problem into

*Corresponding author.

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

independent rank-1 subproblems trained in parallel and analyze each separately, yielding a guarantee joint optimization cannot express: *self-correction*, where the deflation mismatch in each rank-1 worker provably vanishes over rounds, so individual rank-1 directions can be certified correct rather than only their joint sum.

Key insight: parallel deflation with self-correction. In sequential deflation, errors in early components are permanently frozen and compound multiplicatively through the spectral gap [Vandchali et al., 2025]. We remove this dependency by training all rank-1 components *simultaneously*: each is assigned to a dedicated worker that, at every communication round, rebuilds its deflation target from the *latest* predecessor estimates, so early mistakes are progressively corrected rather than locked in. We call this property *self-correction* and prove that the deflation mismatch $\|\mathbf{Y}_{k,\ell} - \mathbf{Y}_k^\star\|_F$ vanishes as ℓ grows (Proposition 2), in contrast to sequential deflation [Vandchali et al., 2025, Liao et al., 2024] where the mismatch is fixed and generically nonzero.

The bilinear setting differs from symmetric eigendecomposition [Liao et al., 2025]: the asymmetric structure $\mathbf{b}\mathbf{a}^\top\mathbf{X}$ calls for Wedin perturbation theory [Wedin, 1972] and a data-dependent recovery scale. Two further mechanisms make parallel deflation practical for LoRA: *advance learning* (private warm-up of dormant components against improving targets) and *per-module dynamic rank discovery* (importance-driven growth under a global budget).

Contributions.

- **C1. AdaPaD algorithm** (Algorithms 1 and 2). Parallel rank-1 deflation with *advance learning* (private pre-training before activation) and *per-module dynamic rank discovery* (importance-based growth under a shared budget); the rank distribution is an output, not an input.
- **C2. Theoretical guarantees.** Exponential convergence with vanishing deflation mismatch (Theorem 1 and Proposition 2) and a weight-recovery bound that splits into a vanishing algorithmic term plus an irreducible statistical floor (Theorems 3 and 4); to our knowledge, the first self-correcting guarantee for parallel rank-1 deflation in bilinear regression, structurally distinct from the joint-update guarantees of [Zangrando et al., 2025, Shinwari and Usama, 2026].
- **C3. Empirical evaluation.** At approximately matched budgets (0.32–0.47 M) on GLUE with DeBERTaV3-base, AdaPaD is competitive with IncreLoRA, AdaLoRA, SoRA, and dEBORA across all eight tasks (Section 5.1); on three math-reasoning benchmarks with Qwen3-0.6B it is within two points of the best of LoRA and IncreLoRA on every task (Section 5.4); on Qwen3-0.6B SQuAD/SQuAD v2 it matches fixed-rank LoRA on F1/EM with adapters 30.7% smaller on average and has the best SQuAD v2 F1 against IncreLoRA and AdaLoRA (Section 5.2); component-parallel sharding delivers up to 2.66× speedup on 4×H200 (Section 5.3).

Related work.

Deflation-based methods. Hotelling's deflation [Hotelling, 1933] is the classical approach for extracting principal components; Mackey [2008] extended it to sparse PCA. Error propagation in inexact sequential deflation was analyzed by Liao et al. [2024] and Vandchali et al. [2025]. The model-parallel perspective was introduced by EigenGame [Gemp et al., 2021, 2022], which casts top-K PCA as a multi-player game. Liao et al. [2025] gave the first provable parallel deflation algorithm for PCA; we develop parallel deflation for the bilinear regression setting, where the asymmetric structure $\mathbf{b}\mathbf{a}^\top\mathbf{X}$ requires Wedin-type perturbation analysis and the self-correction guarantee is tied to a data-dependent recovery scale. ReLoRA [Lialin et al., 2024] trains a sequence of low-rank updates that sum to a high-rank result, which is sequential deflation under a different name; our approach differs by running all components in parallel with provable self-correction.

Low-rank optimization. Problem (1) has been attacked via nuclear norm minimization [Candès and Recht, 2009, Recht et al., 2010], projected gradient descent [Jain et al., 2010, Kyrillidis and Cevher, 2014], and factored methods [Burer and Monteiro, 2003, Tu et al., 2016, Kyrillidis et al., 2018, Park et al., 2017, Bhojanapalli et al., 2016, Park et al., 2018, Kim et al., 2023, Geyer et al., 2020, Hsieh et al., 2018]. These optimize the full low-rank matrix jointly but do not decompose into independent rank-1 subproblems. Kim et al. [2025] show that fixed-rank LoRA training converges to a low-rank global minimum or diverges detectably, but this addresses the fixed-rank setting without rank adaptation.

Adaptive-rank methods with convergence theory. dEBORA [Zangrando et al., 2025] formulates rank selection as bilevel optimization: an upper-level Frank-Wolfe on an ℓ_1-constrained simplex selects singular-value scaling factors while a lower-level solver trains the basis matrices. dEBORA proves

$\mathcal{O}(1/\sqrt{T})$ convergence of the upper level and, notably, finite-time rank identification (the algorithm provably discovers which singular values are zero). As a joint optimization method, dEBORA does not decompose into rank-1 subproblems and therefore does not face the deflation error propagation problem; its convergence guarantee applies to the upper-level simplex variable, not to the bilinear $\mathbf{BA}$ factorization itself. ARD-LoRA [Shinwari and Usama, 2026] introduces learnable per-head scaling factors with $\mathcal{O}(1/T)$ convergence to a stationary point via a standard descent lemma; the guarantee is generic smooth optimization and does not exploit the bilinear structure. Neither decomposes into rank-1 subproblems nor exhibits self-correction (Proposition 2).

2 Background and Problem Setup

Low-Rank Regression and LoRA. We study the linear model $\mathbf{Y} = \mathbf{W}^\star\mathbf{X} + \mathcal{E}$, where $\mathbf{W}^\star \in \mathbb{R}^{m\times d}$ has rank $r^\star$, $\mathbf{X} \in \mathbb{R}^{d\times n}$ collects input features, $\mathbf{Y} \in \mathbb{R}^{m\times n}$ is the response, and $\mathcal{E}$ is noise ($\mathcal{E} = \mathbf{0}$ in the noiseless case; $\mathcal{E}_{ij} \sim \mathcal{N}(0, \varepsilon^2)$ in the noisy case). The goal is to solve

$$\min_{\mathbf{A}\in\mathbb{R}^{r\times d},\mathbf{B}\in\mathbb{R}^{m\times r}} \frac{1}{2}\left\|\mathbf{Y} - \mathbf{BAX}\right\|_F^2. \tag{1}$$

This directly models LoRA: for a pretrained layer with weight $\mathbf{W}_0$, taking $\mathbf{X}$ as the input activations and $\mathbf{Y} = \Delta\mathbf{W}\cdot\mathbf{X}$ reduces LoRA training to (1) with $\mathbf{W}^\star = \Delta\mathbf{W} = \mathbf{BA}$, where $\mathbf{B} \in \mathbb{R}^{m\times r}$ and $\mathbf{A} \in \mathbb{R}^{r\times d}$.

Sequential vs. Parallel Deflation. *Sequential deflation* extracts components one at a time: set $\mathbf{Y}_1 = \mathbf{Y}$, compute $(\mathbf{a}_k, \mathbf{b}_k) = \texttt{Rank1}(\mathbf{Y}_k, \mathbf{X}, T_k)$ with budget T_k, then $\mathbf{Y}_{k+1} = \mathbf{Y}_k - \mathbf{b}_k\mathbf{a}_k^\top\mathbf{X}$. Two defects: wall-clock time $\sum_k T_k$, and frozen errors that compound multiplicatively [Vandchali et al., 2025]. *Parallel deflation* assigns each component to its own worker and indexes rounds by $\ell = 1, \ldots, L$. At round ℓ, worker k holds $(\mathbf{a}_{k,\ell}, \mathbf{b}_{k,\ell})$ and uses the latest predecessor estimates to recompute $\mathbf{Y}_{k,\ell} := \mathbf{Y} - \sum_{k'<k} \mathbf{b}_{k',\ell-1}\mathbf{a}_{k',\ell-1}^\top\mathbf{X}$. The starred quantities $(\mathbf{a}_k^\star, \mathbf{b}_k^\star)$ denote the *ideal* components of the clean target (Section 2). Table 1 contrasts the three regimes: the parallel target $\mathbf{Y}_{k,\ell}$ improves over rounds rather than being fixed.

Table 1: Deflation targets for component k under three deflation regimes. Exact sequential [Hotelling, 1933, Mackey, 2008] uses ideal predecessor outputs (unavailable in practice, since they require ground-truth singular vectors); inexact sequential [Liao et al., 2024, Vandchali et al., 2025] freezes approximate predecessors; parallel deflation refreshes the target every round with the latest predecessor estimates, so the target itself converges as $\ell \to \infty$.

Method	**Deflation target for component k**	**Property**
Exact Sequential	$\mathbf{Y}_k^\star = \mathbf{Y} - \sum_{k'<k} \mathbf{b}_{k'}^\star\mathbf{a}_{k'}^{\star\top}\mathbf{X}$	Fixed, ideal
Inexact Sequential	$\mathbf{Y}_k = \mathbf{Y} - \sum_{k'<k} \mathbf{b}_{k'}\mathbf{a}_{k'}^\top\mathbf{X}$	Fixed, approximate
Parallel (Ours)	$\mathbf{Y}_{k,\ell} = \mathbf{Y} - \sum_{k'<k} \mathbf{b}_{k',\ell-1}\mathbf{a}_{k',\ell-1}^\top\mathbf{X}$	**Improves over rounds**

Ideal Rank-1 Fit and Error Decomposition. Fix $k \in [r]$ and round ℓ. The *ideal rank-1 fit* of $\mathbf{Y}_{k,\ell}$ is its best rank-1 Frobenius approximation,

$$(\bar{\mathbf{a}}_{k,\ell}, \bar{\mathbf{b}}_{k,\ell}) \in \operatorname*{arg\,min}_{\mathbf{a}\in\mathbb{R}^d,\mathbf{b}\in\mathbb{R}^m} \left\|\mathbf{Y}_{k,\ell} - \mathbf{ba}^\top\mathbf{X}\right\|_F, \tag{2}$$

satisfying $\bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top\mathbf{X} = \sigma_{1_k,\ell}\mathbf{u}_{1_k,\ell}\mathbf{v}_{1_k,\ell}^\top$ when the top triplet is unique, where $\sigma_{1_k,\ell}$ is the largest singular value of the current target $\mathbf{Y}_{k,\ell}$ and $\mathbf{u}_{1_k,\ell}$, $\mathbf{v}_{1_k,\ell}$ are its top left and right singular vectors. Define the *clean* deflation target $\mathbf{Y}_k^\star = \mathbf{Y} - \sum_{k'<k} \mathbf{b}_{k'}^\star\mathbf{a}_{k'}^{\star\top}\mathbf{X}$ (exact sequential, top row of Table 1); its best rank-1 fit defines the *ideal k-th component* $(\mathbf{a}_k^\star, \mathbf{b}_k^\star)$ with $\mathbf{b}_k^\star\mathbf{a}_k^{\star\top}\mathbf{X} = \sigma_k^\star\mathbf{u}_k^\star\mathbf{v}_k^{\star\top}$ via Eckart–Young–Mirsky [Eckart and Young, 1936, Mirsky, 1960], where $\sigma_k^\star$, the top singular value of $\mathbf{Y}_k^\star$, equals the k-th singular value of the clean target $\mathbf{W}^\star\mathbf{X}$, and $\mathbf{u}_k^\star$, $\mathbf{v}_k^\star$ are the corresponding singular vectors. Each component thus has three versions at round ℓ, from least to most idealized: the plain iterate $(\mathbf{a}_{k,\ell}, \mathbf{b}_{k,\ell})$ is the subroutine's actual, inexact estimate; the barred pair $(\bar{\mathbf{a}}_{k,\ell}, \bar{\mathbf{b}}_{k,\ell})$ is the exact best rank-1 fit of the current target $\mathbf{Y}_{k,\ell}$; and the starred pair $(\mathbf{a}_k^\star, \mathbf{b}_k^\star)$ is the exact best rank-1 fit of the clean target $\mathbf{Y}_k^\star$, a fixed optimum that does not depend on ℓ. Following the numerical-vs.-mismatch

split of [Liao et al., 2024, Vandchali et al., 2025, Liao et al., 2025], we decompose worker k's error at round ℓ into the *numerical error*, *deflation mismatch*, and *total error*:

$$D_{k,\ell} \triangleq \left\|\mathbf{b}_{k,\ell}\mathbf{a}_{k,\ell}^\top\mathbf{X} - \bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top\mathbf{X}\right\|_F, \qquad \text{(Numerical error)} \tag{3}$$

$$B_{k,\ell} \triangleq \left\|\bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top\mathbf{X} - \mathbf{b}_k^\star\mathbf{a}_k^{\star\top}\mathbf{X}\right\|_F, \qquad \text{(Deflation mismatch)} \tag{4}$$

$$G_{k,\ell} \triangleq \left\|\mathbf{b}_{k,\ell}\mathbf{a}_{k,\ell}^\top\mathbf{X} - \mathbf{b}_k^\star\mathbf{a}_k^{\star\top}\mathbf{X}\right\|_F. \qquad \text{(Total error)} \tag{5}$$

The triangle inequality gives $G_{k,\ell} \le D_{k,\ell} + B_{k,\ell}$: $D_{k,\ell}$ vanishes as the subroutine iterates, $B_{k,\ell}$ shrinks as predecessors improve, and our proof shows both vanish as $\ell \to \infty$.

2.1 Assumptions

Assumption 1 (Rank-1 Subroutine Contraction)**.** For each component k, there exists a *contraction factor* $\mathcal{F}_k \in (0,1)$ such that the rank-1 subroutine, warm-started from $(\mathbf{a}_{k,\ell-1}, \mathbf{b}_{k,\ell-1})$, satisfies for all $\ell \ge k$: $\left\|\mathbf{b}_{k,\ell}\mathbf{a}_{k,\ell}^\top\mathbf{X} - \bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top\mathbf{X}\right\|_F \le \mathcal{F}_k \left\|\mathbf{b}_{k,\ell-1}\mathbf{a}_{k,\ell-1}^\top\mathbf{X} - \bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top\mathbf{X}\right\|_F$, where $(\bar{\mathbf{a}}_{k,\ell}, \bar{\mathbf{b}}_{k,\ell})$ is the ideal rank-1 fit defined in (2).

Assumption 2 (Spectral Gap)**.** $\mathbf{Y} = \mathbf{W}^\star\mathbf{X}$ has distinct singular values $\sigma_1^\star > \sigma_2^\star > \cdots > \sigma_r^\star > 0$ with gaps $\mathcal{T}_k^\star := \min\{\min_{j>k} |\sigma_k^\star - \sigma_j^\star|, \sigma_k^\star\} > 0$. We normalize $\sigma_1^\star = 1$.

Assumption 3 (Bounded Rank-1 Recovery via Projection)**.** Parameter Q in Algorithm 1 satisfies $Q \ge 2\sigma_1^\star$.

The projection in Algorithm 1 together with the assumption above implies the per-component magnitude bound $G_{k,\ell} \le \mathcal{R}_k := Q + \sigma_k^\star$ used throughout the proofs (Lemma 7 in Section E).

All three assumptions are standard: Assumption 1 is the rank-1 subroutine regularity condition [Liao et al., 2025], holding for ALS ($\mathcal{F}_k$ controlled by the local spectral ratio) and factored GD at small step size [Jain et al., 2013, Zheng and Lafferty, 2015, Kyrillidis et al., 2018, Park et al., 2017, Bhojanapalli et al., 2016, Park et al., 2018, Kim et al., 2023, Geyer et al., 2020, Hsieh et al., 2018] (pseudocode in Section B); Assumption 2 is the spectral-gap condition for deflation [Liao et al., 2024, Vandchali et al., 2025, Liao et al., 2025] ($\sigma_1^\star = 1$ is a rescaling); Assumption 3 is automatic given the projection step in Algorithm 1, and the resulting magnitude bound (Lemma 7) is in the same spirit as bounded-iterate constructions in nonconvex factor recovery [Tu et al., 2016, Bhojanapalli et al., 2016, Park et al., 2018, Ge et al., 2017]. In practice, Assumption 1 only needs to hold for $\ell \ge s_k$ (the warm-up round in Theorem 1 via Equation (32)), by which point predecessor errors preserve the top-triplet structure of $\mathbf{Y}_{k,\ell}$. Background perturbation results (Weyl, Wedin) are in Section C.

3 ADAPAD: The Algorithm

Base Algorithm: Parallel Rank-1 Deflation. Algorithm 1 parallelizes inexact sequential deflation [Liao et al., 2024, Vandchali et al., 2025, Liao et al., 2025] for the bilinear model; we assume one worker per component for exposition (here P denotes the number of parallel workers, i.e. physical GPUs; with one worker per component $P = r$, and when $P < r$ each worker owns several components, sharded round-robin via a single `all_gather` per round). Three mechanisms drive convergence: *staggered activation* (worker k begins at $\ell = k$), *warm-starting* from $(\mathbf{a}_{k,\ell-1}, \mathbf{b}_{k,\ell-1})$, and *deflation with current estimates*. The projection on Algorithm 1 caps each rank-1 estimate's Frobenius norm at $Q \ge \sigma_1^\star$, making the bounded-recovery bound Lemma 7 automatic without affecting fixed points (the ideal estimate has norm $\sigma_k^\star \le Q$). The projection in Algorithm 1 prevents the recovered component $\mathbf{b}_{k,\ell}\mathbf{a}_{k,\ell}^\top\mathbf{X}$ from blowing up while the deflated target $\mathbf{Y}_{k,\ell}$ has not yet tightened around $\mathbf{Y}_k^\star$ enough for the rank-1 subroutine to begin contracting. In practice we find that taking $Q = \infty$ (i.e., skipping the projection) suffices. Two extensions, *advance learning* and *dynamic rank discovery*, follow.

Advance Learning. In Algorithm 1, worker k is idle for rounds $1, \ldots, k-1$. Inspired by IncreLoRA's reserve components [Zhang et al., 2023a], we use this idle time: before activation, worker k privately runs the rank-1 subroutine against its current deflation target without broadcasting, accumulating a good initialization. Unlike IncreLoRA's reserves, the warming component trains against the *correct* target $\mathbf{Y}_{k,\ell}$, which itself improves over rounds via self-correction.

Algorithm 1 Parallel Rank-1 Deflation for Low-Rank Regression

Require: $\mathbf{X} \in \mathbb{R}^{d\times n}$, $\mathbf{Y} \in \mathbb{R}^{m\times n}$, rank r, local iterations T, rounds $L \geq r$, projection radius $Q \geq \sigma_1^\star$
Ensure: Components $\{(\mathbf{a}_{k,L}, \mathbf{b}_{k,L})\}_{k=1}^r$

for $k = 1, \ldots, r$ **do**
  Randomly initialize $(\mathbf{a}_{k,0}, \mathbf{b}_{k,0})$
end for
for $\ell = 1, \ldots, L$ **do**
  parfor $k = 1, \ldots, r$ **do** ▷ All r workers in parallel
    if $k \leq \ell$ **then** ▷ Staggered activation
      Receive $\{(\mathbf{a}_{k',\ell-1}, \mathbf{b}_{k',\ell-1})\}_{k'<k}$
      $\mathbf{Y}_{k,\ell} \leftarrow \mathbf{Y} - \sum_{k'<k} \mathbf{b}_{k',\ell-1}\mathbf{a}_{k',\ell-1}^\top \mathbf{X}$ ▷ Deflate with latest estimates
      $(\mathbf{a}_{k,\ell}, \mathbf{b}_{k,\ell}) \leftarrow \texttt{Rank1}(\mathbf{Y}_{k,\ell}, \mathbf{X}, T, \mathbf{a}_{k,\ell-1}, \mathbf{b}_{k,\ell-1})$ ▷ Warm-started
      $\mathbf{b}_{k,\ell}\mathbf{a}_{k,\ell}^\top \mathbf{X} \leftarrow \Pi_Q\big(\mathbf{b}_{k,\ell}\mathbf{a}_{k,\ell}^\top \mathbf{X}\big)$ ▷ Project onto Frobenius ball of radius Q
      Broadcast $(\mathbf{a}_{k,\ell}, \mathbf{b}_{k,\ell})$
    else
      $(\mathbf{a}_{k,\ell}, \mathbf{b}_{k,\ell}) \leftarrow (\mathbf{a}_{k,\ell-1}, \mathbf{b}_{k,\ell-1})$
    end if
  end parfor
end for

Theoretically, advance learning tightens the convergence constant: the worst-case leading factor $3\mathcal{R}_k$ in Theorem 1 is replaced by $\mathcal{R}_k(\mathcal{F}_k^{s_k-1} + 2)$ when pre-activation contraction yields $D_{k,s_k-1} \ll \mathcal{R}_k$.

Dynamic Rank Discovery. Attention and FFN sub-layers need different capacity, and the right allocation varies with depth and task [Zhang et al., 2023b,a, Ding et al., 2023], motivating per-module rank assignment. AdaPaD (Algorithm 2) maintains a per-module active rank r_i (initially 1, capped at $r_{\max}$), a shared budget B, and one private *reserve* per module via advance learning. Every Δ mini-batches (one *global round*, after a one-round warm-up), the top-h eligible modules ($r_i < r_{\max}$) promote their reserve to a committed component and start advance learning on a new reserve; growth stops when $\sum_i r_i \geq B$.

Importance scoring. We adopt the importance-uncertainty product from IncreLoRA [Zhang et al., 2023a]: a module deserves capacity if its parameters are both large in magnitude and still receiving large gradient updates. With $\boldsymbol{\theta}_i$ the committed $(\mathbf{a}, \mathbf{b})$ entries of module i, the raw importance at step t is

$$S_i^{(t)} = \frac{1}{\sqrt{\|\boldsymbol{\theta}_i\|_2}} \sum_{p\in\boldsymbol{\theta}_i} |p| \cdot |\nabla_p \mathcal{L}|, \tag{6}$$

normalized for comparability across module sizes, where $\mathcal{L}$ is the training loss, p ranges over the scalar entries of $\boldsymbol{\theta}_i$, and t indexes optimizer steps. Two EMAs (β_1=β_2=0.85) track signal and volatility, and the allocation score is their product $\bar{S}_i^{(t)} U_i^{(t)}$:

$$\bar{S}_i^{(t)} = \beta_1 \bar{S}_i^{(t-1)} + (1-\beta_1) S_i^{(t)}, \qquad U_i^{(t)} = \beta_2 U_i^{(t-1)} + (1-\beta_2)|S_i^{(t)} - \bar{S}_i^{(t)}|. \tag{7}$$

This per-module allocation is analogous to IncreLoRA's incremental growth but combined with parallel deflation and self-correction: committed components are snapshotted every Δ mini-batches, and new components train against improving deflation targets.

4 Theoretical Guarantees

Scope. Our analysis is formulated for the bilinear regression model $\mathbf{Y} = \mathbf{W}^\star\mathbf{X} + \mathcal{E}$ (Equation (1)), the exact structure of a single LoRA adapter layer; assumptions hold for ALS and factored GD [Jain et al., 2013, Zheng and Lafferty, 2015]. We treat each layer's subproblem in isolation and do not model the end-to-end nonlinear loss, as in dEBORA [Zangrando et al., 2025] and ARD-LoRA [Shinwari and Usama, 2026]; the gap to nonlinear practice is bridged empirically in Section 5.1, with NTK [Jacot et al., 2018, Arora et al., 2019] or local-linearization [Lee et al., 2019, Malladi et al., 2023] extensions deferred (Section 6). Proofs are in Section E; recall $G_{k,\ell} \leq D_{k,\ell} + B_{k,\ell}$ from Section 2.

Algorithm 2 ADAPAD: Parallel Deflation with Per-Module Rank Discovery

Require: Model with M adapter modules, max rank $r_{\max}$, budget B, growth interval Δ, top-h, projection radius $Q \geq \sigma_1^\star$ (inherited by per-component updates as in Algorithm 1)
Ensure: Per-module components $\{(\mathbf{a}_{i,k}, \mathbf{b}_{i,k})\}_{k=1}^{r_i}$ for $i = 1, \ldots, M$

$r_i \leftarrow 1$ for all i; $R \leftarrow M$ ▷ Total active rank
for round $= 1, \ldots, \lceil T_{\text{total}}/\Delta \rceil$ **do**
  Cache Δ mini-batches $\mathcal{B} = \{B_1, \ldots, B_\Delta\}$
  Snapshot committed components for all modules ▷ Self-correction
  for each batch $B_j \in \mathcal{B}$ **do**
    parfor component k owned by this worker **do**
      Forward with snapshot prefix $[0, k)$ + current component k
      Backward; update $(\mathbf{a}_{i,k}, \mathbf{b}_{i,k})$
    end parfor
  end for
  Update importance EMAs $\bar{S}_i, U_i$ via (7)
  if $R < B$ **then**
    Select top-h modules by $\bar{S}_i \cdot U_i$ (among those with $r_i < r_{\max}$)
    for each selected module i **do**
      Activate reserve → committed; $r_i \leftarrow r_i + 1$; $R \leftarrow R + 1$
      Spawn new reserve (advance learning)
    end for
  end if
end for

Main Convergence Result. After a component-dependent warm-up, every component's total error decays nearly-linearly (exponential decay modulated by a mild linear prefactor).

Theorem 1 (Convergence of Parallel Rank-1 Deflation)**.** *Under Assumptions 1 to 3, where $\mathcal{F}_k \in (0, 1)$ is the per-component contraction factor of Assumption 1 and $\mathcal{R}_k > 0$ is the magnitude bound of Lemma 7, define effective rates $m_1 = \mathcal{F}_1$ and*

$$m_k = \max\Big\{\mathcal{F}_k, \frac{1}{k} + \frac{k-1}{k} m_{k-1}\Big\} \quad \text{for } k \geq 2, \tag{8}$$

and starting rounds $s_1 = 1$ with s_{k+1} chosen per-predecessor (i.e., as a maximum over $k' \leq k$) so that

$$s_{k+1} \;=\; \max_{k' \leq k} s_{k'} \;+\; \widetilde{\mathcal{O}}\left(\frac{\log\big(k\, C_{k+1} \mathcal{R}_{k'} / \mathcal{T}^\star_{k+1}\big)}{1 - m_k}\right) \;+\; \frac{(k+1)\, m_k}{1 - m_k} \tag{9}$$

(full Lambert-W form and proof in Section E.2, Corollary 11). Then for all $k \in [r]$ and $\ell \geq s_k - 1$:

$$G_{k,\ell} \leq 3\mathcal{R}_k(\ell - s_k + 2) m_k^{\ell - s_k + 1}. \tag{10}$$

Remark 1 (Interpretation of (10)). Each $m_k \in (0, 1)$ (Section E.7), so the geometric factor dominates the linear prefactor and the bound vanishes exponentially. Proof by strong induction on k (Section E.2); empirical validation in Figures 2, 5 and 6.

Self-Correction. The defining advantage of parallel over sequential deflation: errors are transient, not permanent.

Proposition 2 (Self-Correction)**.** *Under the conditions of Theorem 1, for all $\ell \geq s_k$:*

$$\|\mathbf{Y}_{k,\ell} - \mathbf{Y}^\star_k\|_F \leq 3 \sum_{k'=1}^{k-1} \mathcal{R}_{k'}(\ell - s_{k'} + 1) m_{k'}^{\ell - s_{k'}} \xrightarrow{\ell \to \infty} 0. \tag{11}$$

In sequential deflation, $\|\mathbf{Y}_k - \mathbf{Y}^\star_k\|_F$ is fixed after step k and generically nonzero.

The proof bounds $\|\mathbf{Y}_{k,\ell} - \mathbf{Y}^\star_k\|_F \leq \sum_{k' < k} G_{k',\ell-1}$ via Lemma 9 and applies Theorem 1 to force each $G_{k',\ell-1} \to 0$ (Section E). Table 2 summarizes the differences; see Figure 3.

Estimation Bounds. The self-correction advantage propagates to weight estimation.

Table 2: Sequential vs. parallel rank-1 deflation. "Front-loaded compute" means a disproportionate share of iterations T_k must be spent on the first few components, since their errors propagate to all later ones; under parallel deflation a uniform per-component budget suffices because predecessor errors are corrected over rounds.

Property	**Sequential** [Vandchali et al., 2025]	**Parallel (ours)**
Wall-clock time	$\sum_{k=1}^{r} T_k$	LT (up to $r\times$ faster)
Deflation mismatch	Fixed, generally nonzero	Vanishes as $\ell \to \infty$
Error propagation	Accumulates and persists	Self-corrects over rounds
Compute allocation	Front-loaded compute essential	Uniform per-component budget suffices
Estimation error	Fixed algorithmic term	Vanishes as $L \to \infty$

Theorem 3 (Noiseless estimation)**.** *If* $\mathbf{Y} = \mathbf{W}^\star\mathbf{X}$ *with* $\operatorname{rank}(\mathbf{W}^\star) = r^\star$ *and* $\sigma_{\min}(\mathbf{X}) > 0$*, then after* L *rounds:*

$$\left\|\mathbf{W}^\star - \textstyle\sum_{k=1}^{r} \mathbf{b}_{k,L}\mathbf{a}_{k,L}^\top\right\|_F \le \frac{1}{\sigma_{\min}(\mathbf{X})}\Big(\sum_{k=r+1}^{r^\star} \sigma_k^\star + \sum_{k=1}^{r} G_{k,L}\Big), \tag{12}$$

where $\sigma_k^\star$ *denotes the* k*-th singular value of* $\mathbf{Y} = \mathbf{W}^\star\mathbf{X}$*.*

The second term vanishes as $L \to \infty$ by Theorem 1; in the sequential baseline [Vandchali et al., 2025] the corresponding term is permanently positive.

Theorem 4 (Noisy estimation)**.** *Under* $\mathcal{E}_{ij} \sim \mathcal{N}(0, \varepsilon^2)$ *with* $\varepsilon \le \mathcal{O}(\mathcal{T}^\star_{\min}/(\sqrt{n} + \sqrt{\log(1/\gamma)}))$*, with probability* $\ge 1 - \gamma$*:*

$$\begin{aligned}\left\|\mathbf{W}^\star - \textstyle\sum_k \mathbf{b}_{k,L}\mathbf{a}_{k,L}^\top\right\|_F \le\ & \underbrace{\kappa(\mathbf{X})\Big(\textstyle\sum_{k>r}\sigma_k(\mathbf{W}^\star) + \sigma_{\max}(\mathbf{X})^{-1}\sum_k G_{k,L}\Big)}_{\textit{algorithmic (vanishes)}} \\ & + \underbrace{\mathcal{O}\Big(\frac{\varepsilon\sqrt{n\log(1/\gamma)}}{\sigma_{\min}(\mathbf{X})}\Big(r + \sqrt{\frac{r}{\mathcal{T}^\star_{\min}}}\Big)\Big)}_{\textit{statistical (irreducible)}}.\end{aligned} \tag{13}$$

The algorithmic term vanishes via $\sum_k G_{k,L} \to 0$ (Theorem 1); the statistical floor matches the Gaussian-perturbation bound for deflation-based recovery [Vandchali et al., 2025, Thm. 3], so parallelization adds no new noise penalty (the sequential analogue's algorithmic term stays positive; here it decays in L; floor matches Figure 4). Advance learning tightens the leading constant $3\mathcal{R}_k$ in (10) via $s_k - 1$ rounds of private refinement.

5 Experiments

We evaluate AdaPaD along four axes. Section 5.1 compares it against four adaptive-rank LoRA baselines on GLUE classification with DeBERTaV3-base. Section 5.2 measures its parameter efficiency against fixed-rank LoRA on Qwen3-0.6B SQuAD/SQuAD v2 reading comprehension and compares it with IncreLoRA and AdaLoRA at a matched budget. Section 5.4 evaluates it on three math-reasoning benchmarks with the same backbone. Section 5.3 measures multi-GPU wall-clock scaling of *component-parallel sharding* (assigning disjoint subsets of AdaPaD's rank-1 components to different workers, each holding a private copy of the backbone) on the same Qwen3-0.6B + SQuAD workload. All experiments run on 4×H200 NVLink; settings in Section H. All reported experiments use the gradient-descent rank-1 update (Algorithm 4).

5.1 GLUE Classification

Setup. We fine-tune DeBERTaV3-base on the eight GLUE tasks [Wang et al., 2019] and compare AdaPaD against fixed-rank LoRA and three adaptive-rank baselines: (i) **LoRA** (r=2); (ii) **AdaPaD** ($r_{\max}$=4, dynamic) at total rank budget B=144, matching IncreLoRA's parameter budget; (iii) **IncreLoRA** [Zhang et al., 2023a], reproduced with the authors' code and published hyperparameters;

Table 3: GLUE development-set results with DeBERTaV3-base at matched parameter budgets. AdaPaD discovers rank dynamically ($r_{\max}$=4). AdaPaD and IncreLoRA* (our reproduction with the authors' code) are mean ± std over five seeds; AdaLoRA, SoRA, and dEBORA are cited from their papers. **Bold**: best in column. *Avg.* is the unweighted mean of the eight per-task primary metrics (MCC for CoLA, Pearson for STS-B, accuracy otherwise), following the standard GLUE reporting convention [Wang et al., 2019]; every method in the table is averaged identically, so the comparison is like-for-like.

Method	#Params	CoLA (MCC)	RTE (Acc)	MRPC (Acc)	STS-B (Pear.)	SST-2 (Acc)	QNLI (Acc)	QQP (Acc)	MNLI (Acc)	Avg.
LoRA (r=2)	0.34M	63.56	79.06	78.86	90.47	94.95	94.03	91.61	90.30	85.36
AdaPaD	0.34M	70.98±0.75	**87.73**±0.69	**90.78**±0.33	**91.83**±0.14	**96.08**±0.27	94.24±0.09	91.34±0.11	90.18±0.13	**89.14**
IncreLoRA*	0.34M	70.54±1.20	87.44±0.70	**90.78**±0.70	91.80±0.21	95.99±0.32	94.25±0.17	91.66±0.15	90.20±0.19	89.08
AdaLoRA	0.32M	70.04	87.36	90.44	91.63	95.80	**94.49**	91.78	**90.66**	89.03
SoRA ($r_{\max}$=4)	0.47M	**71.05**	86.04	90.20	91.76	95.57	93.92	**92.06**	90.38	88.87
dEBORA	0.40M	68.72	83.75	90.16	90.84	95.29	93.42	91.88	90.01	88.01

(iv) **AdaLoRA** [Zhang et al., 2023b], **SoRA** [Ding et al., 2023], and **dEBORA** [Zangrando et al., 2025] from their published numbers. Parameter budgets are approximately matched (0.32–0.47 M trainable parameters across methods); rank-1 adapters attach to all 72 encoder linear layers (six per layer × 12 layers: W_q, W_k, W_v, W_o, W_{f1}, W_{f2}), matching IncreLoRA. In practice we use $Q = \infty$ (no projection) as the simplest design choice and find this is sufficient for empirical convergence in all our experiments.

Main results. AdaPaD has the highest GLUE average in Table 3 (89.14), above our IncreLoRA reproduction (89.08) and the published AdaLoRA (89.03). On the five small- and medium-scale tasks (CoLA, RTE, MRPC, STS-B, SST-2; $\leq$ 67k training examples) AdaPaD is best on RTE, STS-B, and SST-2, ties IncreLoRA on MRPC, and trails SoRA on CoLA by 0.07, within one standard deviation. On the three large-scale tasks (QNLI, QQP, MNLI; $\geq$ 105k training examples) it is within 0.72 points of the best baseline (-0.25, -0.72, -0.48).

Discovered rank distribution. Figure 1 shows per-module rank on CoLA at convergence: of 72 modules, 16/44/8/4 sit at rank $1/2/3/4$. FFN intermediate W_{f1} averages 2.67 and attention output W_o averages 2.17, while W_{f2} stays lowest (1.58); layer 11 averages 3.67, consistent with last-layer task-specificity [Zhang et al., 2023b]. Forcing fixed rank costs 8.70 MCC (Table 4).

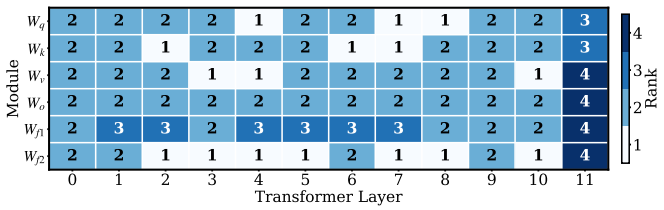


Figure 1: Per-module rank discovered by AdaPaD on CoLA (avg = 2.0); 72 modules (6 types × 12 layers). Higher rank concentrates in the last layer and FFN intermediate projections.

Ablation study. Table 4 isolates AdaPaD's three ingredients on CoLA and MRPC: dynamic rank discovery is the largest contributor (-8.70 MCC, -12.81 Acc when replaced by fixed r=2), then importance scoring (-2.69, -1.72 vs. uniform), then advance learning (-1.21, -1.23).

5.2 Parameter Efficiency on Reading Comprehension

We test AdaPaD on Qwen3-0.6B fine-tuned for QA on SQuAD v1.1 and SQuAD v2 [Rajpurkar et al., 2016, 2018] (generative decoder, adapters on seven projections $W_q, W_k, W_v, W_o, W_{\text{gate}}, W_{\text{up}}, W_{\text{down}}$; 196 modules). LoRA uses uniform rank $r \in \{4, 6\}$; AdaPaD uses average-rank budget $\bar{r} \in \{4, 6\}$ with $r_{\max}$=8 (full settings in Section H.2). For each cell we report the smallest AdaPaD adapter that matches or beats same-budget LoRA on F1, with parameter reduction ΔPar.

Table 4: Ablation on CoLA and MRPC. Each row removes one component from the full AdaPaD algorithm. Δ: change from the full model. The *w/o dynamic rank (fixed r=2)* row pins every module at uniform rank 2, which reduces AdaPaD exactly to the LoRA (r=2) baseline of Table 3; its CoLA and MRPC entries are therefore the *same run* by construction, reproduced here for reference rather than a coincidental duplicate.

	CoLA (MCC)		MRPC (Acc)	
Variant	Score	Δ	Score	Δ
AdaPaD	**72.26**	—	**91.67**	—
w/o advance learning	71.05	−1.21	90.44	−1.23
w/o dynamic rank (fixed r=2)	63.56	−8.70	78.86	−12.81
w/o importance scoring (uniform)	69.57	−2.69	89.95	−1.72

Table 5: Matched-budget head-to-head on Qwen3-0.6B SQuAD/SQuAD v2 (seed=43; bf16; identical recipe across all cells; full settings in Section H.2). **Bold**: AdaPaD matches or beats LoRA at the cell.

		LoRA (fixed-rank r)			AdaPaD ($\bar{r}$)			
Task	b	Params	F1	EM	Params	F1	EM	**ΔPar**
SQuAD v1.1	4	2.52M	88.48	80.65	2.12M	**88.83**	**80.77**	−15.9%
	6	3.79M	88.93	81.42	2.11M	**89.02**	81.09	−44.3%
SQuAD v2	4	2.52M	73.45	70.33	1.93M	**73.63**	**70.70**	−23.4%
	6	3.79M	73.91	70.66	2.30M	**73.99**	**70.91**	−39.3%

Table 6: Adaptive baselines at the average-rank-4 budget (Qwen3-0.6B). Best F1 over training, with EM at that step. The AdaPaD run is the one in Table 5, which reports its earliest checkpoint that beats LoRA with fewer parameters.

	SQuAD v1.1		SQuAD v2	
Method	F1	EM	F1	EM
LoRA (r=4)	88.48	80.65	73.45	70.33
IncreLoRA	88.52	80.49	73.25	70.14
AdaLoRA	**89.45**	**81.69**	71.84	69.46
AdaPaD	89.05	80.75	**74.04**	**70.91**

Table 7: Math reasoning with Qwen3-0.6B at the average-rank-2 budget. Exact-match accuracy.

Method	SVAMP	MAWPS	GSM8K
LoRA (r=2)	52.67	83.33	**49.50**
IncreLoRA	**56.33**	79.00	49.00
AdaPaD	56.00	**85.33**	47.50

Across the four cells (Table 5), AdaPaD yields adapters 15.9–44.3% smaller than fixed-rank LoRA (mean 30.7%) while remaining competitive on F1/EM; a complementary view to GLUE, which fixes the budget and reads off quality.

Table 6 adds the adaptive baselines IncreLoRA and AdaLoRA at the average-rank-4 budget (784 rank units). AdaPaD has the best SQuAD v2 F1 and EM. On SQuAD v1.1 AdaLoRA leads by 0.40 F1 and AdaPaD is second.

5.3 Multi-GPU Scaling on Qwen3-0.6B

AdaPaD admits component-parallel sharding: each worker holds a disjoint subset of the rank-1 components and trains them independently between syncs, predicting near-linear speedup. We measure this on the SQuAD v1.1 workload of Section 5.2, disabling per-module growth and pinning every module at $r_{\max}$ to fix the total at $196 \times r_{\max}$ across $P \in \{1, 2, 3, 4\}$ at each $r_{\max} \in \{4, 8\}$. We report mean $\pm$ std over five global rounds (hyperparameters in Section H.2).

The 1-3 GPU cells are highly stable (rel. std $< 1.5\%$): three workers deliver $2.03\times$ at $r_{\max}$=4 and $2.53\times$ at $r_{\max}$=8. At P=4 we see $2.45\times$ and $2.66\times$ with higher variance (rel. std $\approx 10\%$), as the 4-way NVLink all-reduce makes NCCL latency dominate when per-worker compute shrinks; overlap-friendly schedules [Li et al., 2020, Sergeev and Balso, 2018] and bandwidth-aware placement [Rajbhandari et al., 2020, Shoeybi et al., 2020] are future work. Each worker holds one copy

Table 8: Component-parallel scaling on Qwen3-0.6B + SQuAD v1.1. Per-batch wall-clock cost (mean $\pm$ std over five global rounds) and speedup vs. $P{=}1$ at the same rank. **Bold**: best speedup in each rank column.

	$r_{\max}{=}4$		$r_{\max}{=}8$	
#GPUs	per-batch (s)	speedup	per-batch (s)	speedup
1	0.986 ± 0.011	$1.00\times$	2.039 ± 0.015	$1.00\times$
2	0.532 ± 0.006	$1.85 \pm 0.03\times$	1.062 ± 0.008	$1.92 \pm 0.02\times$
3	0.487 ± 0.006	$2.03 \pm 0.03\times$	0.806 ± 0.009	$2.53 \pm 0.04\times$
4	0.403 ± 0.047	$\mathbf{2.45} \pm 0.29\times$	0.768 ± 0.074	$\mathbf{2.66} \pm 0.26\times$

of the frozen backbone, as in data-parallel LoRA, and exchanges only rank-1 factors. On one GPU, per-step time grows with rank while peak memory stays flat (Section H.5).

5.4 Math Reasoning on Qwen3-0.6B

We fine-tune Qwen3-0.6B on SVAMP, MAWPS, and GSM8K with the 196 adapted modules of Section 5.2, at the average-rank-2 budget of LoRA with $r{=}2$ (1.26M parameters). AdaPaD is best on MAWPS, 2.00 points above LoRA, and trails IncreLoRA on SVAMP by 0.33 and LoRA on GSM8K by 2.00 (Table 7).

6 Discussion and Conclusion

AdaPaD addresses two LoRA design questions: parallel training keeps early rank-1 components from corrupting later ones, and per-module dynamic rank discovery makes the rank distribution an output of importance scores rather than a fixed input. We prove nearly-linear convergence with a self-correcting deflation mismatch and an estimation bound that splits into a vanishing algorithmic term plus an irreducible statistical floor. Empirically, AdaPaD achieves the highest GLUE average against four adaptive-rank baselines (Section 5.1), stays within two points of the best baseline on three math-reasoning benchmarks (Section 5.4), produces Qwen3-0.6B SQuAD/SQuAD v2 adapters 30.7% smaller than fixed-rank LoRA at competitive F1/EM (Section 5.2), and component-parallel sharding delivers up to $2.66\times$ speedup on $4\times$H200 (Section 5.3).

Limitations and future work. The main practical limitation is training cost on a single GPU. Each step runs one backbone pass per active component, so per-step time grows with rank, and the rank warm-up delays the drop in loss (Section H.5). Component-parallel sharding offsets part of this cost when several GPUs are available (Section 5.3). The theory covers the bilinear model (1); extension to nonlinear LoRA losses is empirical, bounds degrade as $\mathcal{T}_k^\star \to 0$, and synchronous $\mathcal{O}(r^2(d+m))$ communication can offset the speedup at very large r. Validation at 7B+ scale, NTK [Jacot et al., 2018, Arora et al., 2019] or local-linearization [Lee et al., 2019, Malladi et al., 2023] extensions, asynchronous communication, and richer importance scores are future work, as are a fused rank-1 kernel that sends all components through one backbone pass and coding and instruction-tuning benchmarks.

Acknowledgments and Disclosure of Funding

This research was funded in part by Rice University, the U.S. National Science Foundation (CAREER award no. 2145629), and a Microsoft Research Award. AK also thanks the Ken Kennedy Institute at Rice University for its support through the research cluster QuanTAS. We thank the anonymous reviewers and the area chair for comments that improved this paper.

References

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing*

Systems, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/dbc4d84bfcfe2284ba11beffb853a8c4-Paper.pdf.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.1. URL https://aclanthology.org/2022.acl-short.1/.

Srinadh Bhojanapalli, Anastasios Kyrillidis, and Sujay Sanghavi. Dropping convexity for faster semi-definite optimization. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 530–582, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR. URL https://proceedings.mlr.press/v49/bhojanapalli16.html.

Samuel Burer and Renato D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, February 2003. ISSN 1436-4646. doi: 10.1007/s10107-002-0352-8. URL http://dx.doi.org/10.1007/s10107-002-0352-8.

Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, April 2009. ISSN 1615-3383. doi: 10.1007/s10208-009-9045-5. URL http://dx.doi.org/10.1007/s10208-009-9045-5.

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. Sparse low-rank adaptation of pre-trained language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4133–4145, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.252. URL https://aclanthology.org/2023.emnlp-main.252/.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, September 1936. ISSN 1860-0980. doi: 10.1007/bf02288367. URL http://dx.doi.org/10.1007/bf02288367.

Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1233–1242. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/ge17a.html.

Ian Gemp, Brian McWilliams, Claire Vernade, and Thore Graepel. EigenGame: PCA as a Nash equilibrium. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=NzTU59SYbNq.

Ian Gemp, Brian McWilliams, Claire Vernade, and Thore Graepel. EigenGame unloaded: When playing games is better than optimizing. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=So6YAqnqgMj.

Kelly Geyer, Anastasios Kyrillidis, and Amir Kalev. Low-rank regularization and solution uniqueness in over-parameterized matrix sensing. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 930–940. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/geyer20a.html.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low rank adaptation of large models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 17783–17806. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/hayou24a.html.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=sE7-XhLxHA.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, September 1933. ISSN 0022-0663. doi: 10.1037/h0071325. URL http://dx.doi.org/10.1037/h0071325.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

Ya-Ping Hsieh, Yu-Chun Kao, Rabeeh Karimi Mahabadi, Alp Yurtsever, Anastasios Kyrillidis, and Volkan Cevher. A non-Euclidean gradient descent framework for non-convex matrix factorization. *IEEE Transactions on Signal Processing*, 66(22):5917–5926, November 2018. ISSN 1941-0476. doi: 10.1109/tsp.2018.2870353. URL http://dx.doi.org/10.1109/tsp.2018.2870353.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/5a4be1fa34e62bb8a6ec6b91d2462f5a-Paper.pdf.

Prateek Jain, Raghu Meka, and Inderjit Dhillon. Guaranteed rank minimization via singular value projection. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010. URL https://proceedings.neurips.cc/paper_files/paper/2010/file/08d98638c6fcd194a4b1e6992063e944-Paper.pdf.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of Computing*, STOC′13, pages 665–674. ACM, June 2013. doi: 10.1145/2488608.2488693. URL http://dx.doi.org/10.1145/2488608.2488693.

Junhyung Lyle Kim, George Kollias, Amir Kalev, Ken X. Wei, and Anastasios Kyrillidis. Fast quantum state reconstruction via accelerated non-convex programming. *Photonics*, 10(2):116, January 2023. ISSN 2304-6732. doi: 10.3390/photonics10020116. URL http://dx.doi.org/10.3390/photonics10020116.

Junsu Kim, Jaeyeon Kim, and Ernest K. Ryu. LoRA training provably converges to a low-rank global minimum or it fails loudly (But it probably won't fail). In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 30224–30247. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/kim25n.html.

Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M Asano. VeRA: Vector-based random matrix adaptation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NjNfLdxr3A.

Anastasios Kyrillidis and Volkan Cevher. Matrix recipes for hard thresholding methods. *Journal of Mathematical Imaging and Vision*, 48(2):235–265, February 2014. ISSN 1573-7683. doi: 10.1007/s10851-013-0434-7. URL http://dx.doi.org/10.1007/s10851-013-0434-7.

Anastasios Kyrillidis, Amir Kalev, Dohyung Park, Srinadh Bhojanapalli, Constantine Caramanis, and Sujay Sanghavi. Provable compressed sensing quantum state tomography via non-convex methods. *npj Quantum Information*, 4(1), August 2018. ISSN 2056-6387. doi: 10.1038/s41534-018-0080-4. URL http://dx.doi.org/10.1038/s41534-018-0080-4.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/0d1a9651497a38d8b1c3871c84528bd4-Paper.pdf.

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. *Proceedings of the VLDB Endowment*, 13(12):3005–3018, August 2020. ISSN 2150-8097. doi: 10.14778/3415478.3415530. URL http://dx.doi.org/10.14778/3415478.3415530.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353/.

Yang Li, Shaobo Han, and Shihao Ji. VB-LoRA: Extreme parameter efficient fine-tuning with vector banks. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 16724–16751. Curran Associates, Inc., 2024. doi: 10.52202/079017-0533. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/1e0d38c676d5855bcfab7f6d29d20ad9-Paper-Conference.pdf.

Vladislav Lialin, Sherin Muckatira, Namrata Shivagunde, and Anna Rumshisky. ReLoRA: High-rank training through low-rank updates. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=DLJznSp6X3.

Fangshuo Liao, Junhyung Lyle Kim, Cruz Barnum, and Anastasios Kyrillidis. On the error-propagation of inexact Hotelling's deflation for principal component analysis. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 29720–29747. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/liao24a.html.

Fangshuo Liao, Wenyi Su, and Anastasios Kyrillidis. Provable model-parallel distributed principal component analysis with parallel deflation. In Beidi Chen, Shijia Liu, Mert Pilanci, Weijie Su, Jeremias Sulam, Yuxiang Wang, and Zhihui Zhu, editors, *Conference on Parsimony and Learning*, volume 280 of *Proceedings of Machine Learning Research*, pages 392–416. PMLR, 24–27 Mar 2025. URL https://proceedings.mlr.press/v280/liao25a.html.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 32100–32121. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/liu24bn.html.

Lester Mackey. Deflation methods for sparse PCA. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper_files/paper/2008/file/85d8ce590ad8981ca2c8286f79f59954-Paper.pdf.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A kernel-based view of language model fine-tuning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 23610–23641. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/malladi23a.html.

L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 11(1):50–59, 1960. ISSN 1464-3847. doi: 10.1093/qmath/11.1.50. URL http://dx.doi.org/10.1093/qmath/11.1.50.

Dohyung Park, Anastasios Kyrillidis, Constantine Caramanis, and Sujay Sanghavi. Non-square matrix sensing without spurious local minima via the Burer-Monteiro approach. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 65–74. PMLR, 20–22 Apr 2017. URL https://proceedings.mlr.press/v54/park17a.html.

Dohyung Park, Anastasios Kyrillidis, Constantine Caramanis, and Sujay Sanghavi. Finding low-rank solutions via nonconvex matrix factorization, efficiently and provably. *SIAM Journal on Imaging Sciences*, 11(4):2165–2204, January 2018. ISSN 1936-4954. doi: 10.1137/17m1150189. URL http://dx.doi.org/10.1137/17m1150189.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, November 2020. doi: 10.1109/sc41405.2020.00024. URL http://dx.doi.org/10.1109/sc41405.2020.00024.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras, editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2124. URL https://aclanthology.org/P18-2124/.

Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, January 2010. ISSN 1095-7200. doi: 10.1137/070697835. URL http://dx.doi.org/10.1137/070697835.

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

Haseeb Ullah Khan Shinwari and Muhammad Usama. ARD-LoRA: Dynamic rank allocation for parameter-efficient fine-tuning of foundation models with heterogeneous adaptation needs. *IEEE Transactions on Artificial Intelligence*, 7(3):1666–1676, March 2026. ISSN 2691-4581. doi: 10.1109/tai.2025.3605569. URL http://dx.doi.org/10.1109/tai.2025.3605569.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-LM: Training multi-billion parameter language models using model parallelism, 2020. URL https://arxiv.org/abs/1909.08053.

Stephen Tu, Ross Boczar, Max Simchowitz, Mahdi Soltanolkotabi, and Ben Recht. Low-rank solutions of linear matrix equations via Procrustes flow. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 964–973, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/tu16.html.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. DyLoRA: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3274–3287, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.239. URL https://aclanthology.org/2023.eacl-main.239/.

Mahtab Alizadeh Vandchali, Fangshuo Liao, and Anastasios Kyrillidis. One rank at a time: Cascading error dynamics in sequential learning, 2025. URL https://arxiv.org/abs/2505.22602.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

Per-Åke Wedin. Perturbation bounds in connection with singular value decomposition. *BIT*, 12(1): 99–111, March 1972. ISSN 1572-9125. doi: 10.1007/bf01932678. URL http://dx.doi.org/10.1007/bf01932678.

Hermann Weyl. Das asymptotische Verteilungsgesetz der Eigenwerte linearer partieller Differentialgleichungen (mit einer Anwendung auf die Theorie der Hohlraumstrahlung). *Mathematische Annalen*, 71(4):441–479, December 1912. ISSN 1432-1807. doi: 10.1007/bf01456804. URL http://dx.doi.org/10.1007/bf01456804.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Emanuele Zangrando, Sara Venturini, Francesco Rinaldi, and Francesco Tudisco. dEBORA: Efficient bilevel optimization-based low-rank adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=5M0ic2RxQZ.

Feiyu Zhang, Liangzhi Li, Junhao Chen, Zhouqiang Jiang, Bowen Wang, and Yiming Qian. IncreLoRA: Incremental parameter allocation method for parameter-efficient fine-tuning, 2023a. URL https://arxiv.org/abs/2308.12043.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=lq62uWRJjiY.

Qinqing Zheng and John Lafferty. A convergent gradient descent algorithm for rank minimization and semidefinite programming from random linear measurements. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/32bb90e8976aab5298d5da10fe66f21d-Paper.pdf.

A Broader Impact

ADAPAD is a parameter-efficient fine-tuning method, and its primary societal benefit is efficiency: by allocating rank only where it is needed and matching fixed-rank LoRA quality with adapters that are roughly 30% smaller, it lowers the storage, memory, and energy cost of adapting large models, broadening access to fine-tuning for practitioners with limited compute. As a general-purpose adaptation technique, ADAPAD inherits the risks of the models it fine-tunes: like any PEFT method it can specialize a model for harmful or biased downstream behavior, and it does not by itself remove biases present in the pretrained backbone or the fine-tuning data. We view it as complementary to, not a substitute for, dataset curation, safety alignment, and evaluation.

B Rank-1 Subroutines

We provide pseudocode for two rank-1 subroutines that satisfy Assumption 1.

Algorithm 3 `Rank1` via Alternating Least Squares

Require: $\mathbf{Y}_k \in \mathbb{R}^{m\times n}$, $\mathbf{X} \in \mathbb{R}^{d\times n}$, iterations T, warm-start $(\mathbf{a}^{(0)}, \mathbf{b}^{(0)})$

$\mathbf{M} \leftarrow \mathbf{X}\mathbf{X}^\top$; $\mathbf{N} \leftarrow \mathbf{Y}_k\mathbf{X}^\top$
for $t = 0, \dots, T-1$ **do**
  $\mathbf{b}^{(t+1)} \leftarrow \mathbf{N}\mathbf{a}^{(t)} / (\mathbf{a}^{(t)\top}\mathbf{M}\mathbf{a}^{(t)})$
  $\mathbf{a}^{(t+1)} \leftarrow (\|\mathbf{b}^{(t+1)}\|^2\mathbf{M})^{-1}\mathbf{N}^\top\mathbf{b}^{(t+1)}$
end for
return $(\mathbf{a}^{(T)}, \mathbf{b}^{(T)})$

Algorithm 4 `Rank1` via Factored Gradient Descent

Require: $\mathbf{Y}_k$, $\mathbf{X}$, T, warm-start $(\mathbf{a}^{(0)}, \mathbf{b}^{(0)})$, learning rates $\eta_{\mathbf{a}}, \eta_{\mathbf{b}}$

$\mathbf{M} \leftarrow \mathbf{X}\mathbf{X}^\top$; $\mathbf{N} \leftarrow \mathbf{Y}_k\mathbf{X}^\top$
for $t = 0, \dots, T-1$ **do**
  $\nabla_{\mathbf{a}} \leftarrow \mathbf{M}\mathbf{a}^{(t)}\|\mathbf{b}^{(t)}\|^2 - \mathbf{N}^\top\mathbf{b}^{(t)}$
  $\nabla_{\mathbf{b}} \leftarrow \mathbf{b}^{(t)}(\mathbf{a}^{(t)\top}\mathbf{M}\mathbf{a}^{(t)}) - \mathbf{N}\mathbf{a}^{(t)}$
  $\mathbf{a}^{(t+1)} \leftarrow \mathbf{a}^{(t)} - \eta_{\mathbf{a}}\nabla_{\mathbf{a}}$; $\mathbf{b}^{(t+1)} \leftarrow \mathbf{b}^{(t)} - \eta_{\mathbf{b}}\nabla_{\mathbf{b}}$
end for
return $(\mathbf{a}^{(T)}, \mathbf{b}^{(T)})$

C Background Perturbation Results

Theorem 5 (Weyl's Inequality [Weyl, 1912])**.** *For* $\mathbf{M}, \mathbf{\Delta} \in \mathbb{R}^{m\times n}$*, define* $\tilde{\mathbf{M}} = \mathbf{M} + \mathbf{\Delta}$*. Then* $|\tilde{\sigma}_i - \sigma_i| \le \|\mathbf{\Delta}\|_2$ *for all* i*.*

Theorem 6 (Wedin's Theorem [Wedin, 1972])**.** *Let* $\mathbf{M}$ *and* $\tilde{\mathbf{M}} = \mathbf{M} + \mathbf{\Delta}$ *have top singular triplets* $(\sigma_1, \mathbf{u}_1, \mathbf{v}_1)$ *and* $(\tilde{\sigma}_1, \tilde{\mathbf{u}}_1, \tilde{\mathbf{v}}_1)$*. If* $\delta := \min\{\min_{j\neq 1}|\sigma_1 - \tilde{\sigma}_j|, \sigma_1\} > 0$*, then* $\sin^2\angle(\mathbf{u}_1, \tilde{\mathbf{u}}_1) + \sin^2\angle(\mathbf{v}_1, \tilde{\mathbf{v}}_1) \le (\|\mathbf{\Delta}^\top\mathbf{u}_1\|_2^2 + \|\mathbf{\Delta}\mathbf{v}_1\|_2^2)/\delta^2$.

D Notation Index

E Complete Proofs

Recall the following definitions:

$$D_{k,\ell} \triangleq \left\|\mathbf{b}_{k,\ell}\mathbf{a}_{k,\ell}^\top\mathbf{X} - \bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top\mathbf{X}\right\|_F, \qquad \text{(Numerical error)} \tag{14}$$

$$B_{k,\ell} \triangleq \left\|\bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top\mathbf{X} - \mathbf{b}_k^\star\mathbf{a}_k^{\star\top}\mathbf{X}\right\|_F, \qquad \text{(Deflation mismatch)} \tag{15}$$

$$G_{k,\ell} \triangleq \left\|\mathbf{b}_{k,\ell}\mathbf{a}_{k,\ell}^\top\mathbf{X} - \mathbf{b}_k^\star\mathbf{a}_k^{\star\top}\mathbf{X}\right\|_F. \qquad \text{(Total error)} \tag{16}$$

Table 9: Summary of notation.

Symbol	Type	Defined in	Meaning
$\mathbf{a}_{k,\ell}$	$\mathbb{R}^d$	Algorithm 1	Worker k's right factor at round ℓ
$\mathbf{a}_k^\star$	$\mathbb{R}^d$	Section 2	Ideal right factor for component k
$\bar{\mathbf{a}}_{k,\ell}, \bar{\mathbf{b}}_{k,\ell}$	$\mathbb{R}^d, \mathbb{R}^m$	(2)	Exact best rank-1 fit of the current target $\mathbf{Y}_{k,\ell}$
$B_{k,\ell}$	$\mathbb{R}_{\geq 0}$	Section 2, (4)	Deflation mismatch $\left\|\bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top \mathbf{X} - \mathbf{b}_k^\star \mathbf{a}_k^{\star\top}\mathbf{X}\right\|_F$
$\mathbf{b}_{k,\ell}$	$\mathbb{R}^m$	Algorithm 1	Worker k's left factor at round ℓ
C_k	$\mathbb{R}_{>0}$	Lemma 10	$3\sigma_k^\star/\mathcal{T}_k^\star + 1$
$D_{k,\ell}$	$\mathbb{R}_{\geq 0}$	Section 2, (3)	Numerical error $\left\|\mathbf{b}_{k,\ell}\mathbf{a}_{k,\ell}^\top \mathbf{X} - \bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top \mathbf{X}\right\|_F$
$G_{k,\ell}$	$\mathbb{R}_{\geq 0}$	Section 2, (5)	Total error $\left\|\mathbf{b}_{k,\ell}\mathbf{a}_{k,\ell}^\top \mathbf{X} - \mathbf{b}_k^\star \mathbf{a}_k^{\star\top}\mathbf{X}\right\|_F$
$\mathcal{F}_k$	$(0,1)$	Assumption 1	Subroutine contraction factor
m_k	$(0,1)$	Theorem 1	Effective convergence rate
P	$\mathbb{N}$	Section 3	Number of parallel workers (GPUs); $P = r$ with one worker per component
Q	$\mathbb{R}_{>0}$	Algorithm 1	Projection radius
$\mathcal{R}_k$	$\mathbb{R}_{>0}$	Lemma 7	Magnitude bound on the rank-1 recovery
s_k	$\mathbb{N}$	Theorem 1	Warm-up round
$\hat{s}_k$	$\mathbb{N}$	Lemma 14	Earliest round at which the Wedin gap activates
$\mathcal{T}_k^\star$	$\mathbb{R}_{>0}$	Assumption 2	Singular-value gap
$\sigma_{1_k,\ell}, \mathbf{u}_{1_k,\ell}, \mathbf{v}_{1_k,\ell}$	$\mathbb{R}_{>0}, \mathbb{R}^m, \mathbb{R}^n$	Section 2	Top singular triplet of the current target $\mathbf{Y}_{k,\ell}$
$\sigma_k^\star, \mathbf{u}_k^\star, \mathbf{v}_k^\star$	$\mathbb{R}_{>0}, \mathbb{R}^m, \mathbb{R}^n$	Section 2	Top singular triplet of the clean target $\mathbf{Y}_k^\star$
$\mathbf{Y}_{k,\ell}$	$\mathbb{R}^{m\times n}$	Algorithm 1	Current deflation target of worker k at round ℓ
$\mathbf{Y}_k^\star$	$\mathbb{R}^{m\times n}$	Section 2	Clean (exact sequential) deflation target
$r_i, r_{\max}, B$	$\mathbb{N}$	Section 3	Active rank of module i, per-module cap, total rank budget
$S_i^{(t)}, \bar{S}_i^{(t)}, U_i^{(t)}$	$\mathbb{R}_{\geq 0}$	(6), (7)	Raw importance, its EMA, and its uncertainty
$\hat{W}$	–	Theorem 1	$\max\{1, -W_{-1}(-\cdot)\}$

where the ideal fit $(\bar{\mathbf{a}}_{k,\ell}, \bar{\mathbf{b}}_{k,\ell})$ is defined in (2); and $(\mathbf{a}_k^\star, \mathbf{b}_k^\star)$ denotes the ideal k-th component of the clean target $\mathbf{Y}_k^\star$.

E.1 Supporting Lemmas

The following lemmas control how these three quantities interact across rounds.

Lemma 7 (Bounded Rank-1 Recovery)**.** *Under Assumption 3, the per-component recovery error satisfies*

$$G_{k,\ell} \;\leq\; \mathcal{R}_k \;:=\; Q + \sigma_k^\star \qquad \text{for every round } \ell \geq 0, \tag{17}$$

and the same bound transfers to $D_{k,\ell}$ and $B_{k,\ell}$.

Proof. By the projection in Algorithm 1, $\left\|\mathbf{b}_{k,\ell}\mathbf{a}_{k,\ell}^\top \mathbf{X}\right\|_F \leq Q$, and the ideal k-th component satisfies $\left\|\mathbf{b}_k^\star \mathbf{a}_k^{\star\top}\mathbf{X}\right\|_F = \sigma_k^\star$. The triangle inequality gives

$$G_{k,\ell} \;\leq\; \left\|\mathbf{b}_{k,\ell}\mathbf{a}_{k,\ell}^\top \mathbf{X}\right\|_F + \left\|\mathbf{b}_k^\star \mathbf{a}_k^{\star\top}\mathbf{X}\right\|_F \;\leq\; Q + \sigma_k^\star \;=\; \mathcal{R}_k.$$

The corresponding ceilings for $D_{k,\ell}$ and $B_{k,\ell}$ follow by the same triangle-inequality argument using the ideal-fit norm $\left\|\bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top \mathbf{X}\right\|_F = \sigma_{1_k,\ell} \leq \|\mathbf{Y}_{k,\ell}\|_2$, which is bounded by Q whenever $\|\mathbf{Y}_{k,\ell} - \mathbf{Y}_k^\star\|_2 \leq \sigma_1^\star$ (using $Q \geq 2\sigma_1^\star$); this is the regime in which the proofs invoke these ceilings. □

Lemma 8 (Numerical Error Recurrence)**.** *Under Assumption 1 and Assumption 3, for all $\ell \geq k$ if $\|\mathbf{Y}_{k,\ell} - \mathbf{Y}_k^\star\| \leq \sigma_k^\star$, then we have that $D_{k,\ell} \leq \mathcal{F}_k(D_{k,\ell-1} + B_{k,\ell} + B_{k,\ell-1})$.*

Proof. Let the output of Line 9 be denoted by $\tilde{\mathbf{a}}_{k,\ell}, \tilde{\mathbf{b}}_{k,\ell}$. If $\left\|\tilde{\mathbf{b}}_{k,\ell}\tilde{\mathbf{a}}_{k,\ell}^\top \mathbf{X}\right\| \le Q$, then $\tilde{\mathbf{a}}_{k,\ell} = \mathbf{a}_{k,\ell}, \tilde{\mathbf{b}}_{k,\ell} = \mathbf{b}_{k,\ell}$. Therefore, by Assumption 1:

$$\begin{aligned} D_{k,\ell} &\le \mathcal{F}_k \left\|\mathbf{b}_{k,\ell-1}\mathbf{a}_{k,\ell-1}^\top \mathbf{X} - \bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top \mathbf{X}\right\|_F \\ &\le \mathcal{F}_k(\left\|\mathbf{b}_{k,\ell-1}\mathbf{a}_{k,\ell-1}^\top \mathbf{X} - \mathbf{b}_k^\star \mathbf{a}_k^{\star\top}\mathbf{X}\right\|_F + \left\|\mathbf{b}_k^\star \mathbf{a}_k^{\star\top}\mathbf{X} - \bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top \mathbf{X}\right\|_F) \\ &= \mathcal{F}_k(G_{k,\ell-1} + B_{k,\ell}) \le \mathcal{F}_k(D_{k,\ell-1} + B_{k,\ell-1} + B_{k,\ell}). \end{aligned}$$

On the other hand, if $\left\|\tilde{\mathbf{b}}_{k,\ell}\tilde{\mathbf{a}}_{k,\ell}^\top \mathbf{X}\right\| \ge Q$, we then notice that

$$\|\mathbf{Y}_{k,\ell}\|_2 \le \|\mathbf{Y}_k^\star\|_2 + \|\mathbf{Y}_{k,\ell} - \mathbf{Y}_k^\star\|_2 \le \sigma_k^\star + \sigma_1^\star \le Q$$

Therefore, $\left\|\bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top \mathbf{X}\right\|_F \le Q$. By the property of projection onto convex sets, we have that

$$\left\|\mathbf{b}_{k,\ell}\mathbf{a}_{k,\ell}^\top \mathbf{X} - \bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top \mathbf{X}\right\| \le \left\|\tilde{\mathbf{b}}_{k,\ell}\tilde{\mathbf{a}}_{k,\ell}^\top \mathbf{X} - \bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top \mathbf{X}\right\|$$

Therefore, by Assumption 1 we have that

$$\begin{aligned} D_{k,\ell} &\le \left\|\tilde{\mathbf{b}}_{k,\ell}\tilde{\mathbf{a}}_{k,\ell}^\top \mathbf{X} - \bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top \mathbf{X}\right\| \\ &\le \mathcal{F}_k \left\|\mathbf{b}_{k,\ell-1}\mathbf{a}_{k,\ell-1}^\top \mathbf{X} - \bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top \mathbf{X}\right\|_F \\ &\le \mathcal{F}_k(\left\|\mathbf{b}_{k,\ell-1}\mathbf{a}_{k,\ell-1}^\top \mathbf{X} - \mathbf{b}_k^\star \mathbf{a}_k^{\star\top}\mathbf{X}\right\|_F + \left\|\mathbf{b}_k^\star \mathbf{a}_k^{\star\top}\mathbf{X} - \bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top \mathbf{X}\right\|_F) \\ &= \mathcal{F}_k(G_{k,\ell-1} + B_{k,\ell}) \le \mathcal{F}_k(D_{k,\ell-1} + B_{k,\ell-1} + B_{k,\ell}). \end{aligned}$$

□

Lemma 9 (Deflated Target Mismatch)**.** *For Algorithm 1:* $\|\mathbf{Y}_{k,\ell} - \mathbf{Y}_k^\star\|_F \le \sum_{k'=1}^{k-1} G_{k',\ell-1}$.

Proof. $\mathbf{Y}_{k,\ell} - \mathbf{Y}_k^\star = \sum_{k'<k}(\mathbf{b}_{k'}^\star \mathbf{a}_{k'}^{\star\top} - \mathbf{b}_{k',\ell-1}\mathbf{a}_{k',\ell-1}^\top)\mathbf{X}$. Taking Frobenius norms: $\|\mathbf{Y}_{k,\ell} - \mathbf{Y}_k^\star\|_F \le \sum_{k'<k} G_{k',\ell-1}$. □

Lemma 10 (Deflation Mismatch Bound)**.** *Under Assumption 2, define* $C_k := 3\sigma_k^\star/\mathcal{T}_k^\star + 1$. *If* $\|\mathbf{Y}_{k,\ell} - \mathbf{Y}_k^\star\|_2 < \frac{1}{2}\min_{j>k}|\sigma_k^\star - \sigma_j^\star|$, *then* $B_{k,\ell} \le C_k \|\mathbf{Y}_{k,\ell} - \mathbf{Y}_k^\star\|_F$.

Proof. The deflation mismatch $B_{k,\ell}$ measures the gap between the best rank-1 fit of the *actual* target $\mathbf{Y}_{k,\ell}$ and the ideal k-th component of the *clean* target $\mathbf{Y}_k^\star$. We bound this gap by decomposing it into a singular-vector error and a singular-value error, then control each via the two perturbation results in Section C: Wedin's theorem for the singular vectors and Weyl's inequality for the singular values.

Step 1 (Rank-1 SVD expressions). By the Eckart-Young-Mirsky theorem, ideal sequential deflation removes exactly the first $k-1$ singular components of $\mathbf{Y}$, so $\mathbf{Y}_k^\star = \sum_{j\ge k}\sigma_j^\star \mathbf{u}_j^\star \mathbf{v}_j^{\star\top}$ has leading singular triplet $(\sigma_k^\star, \mathbf{u}_k^\star, \mathbf{v}_k^\star)$ with gap $\mathcal{T}_k^\star$ [Vandchali et al., 2025, Lemma 1]. The hypothesis $\|\mathbf{Y}_{k,\ell} - \mathbf{Y}_k^\star\|_2 < \frac{1}{2}\mathcal{T}_k^\star$ together with Assumption 2 (distinct singular values of $\mathbf{Y}_k^\star$) ensures that $\mathbf{Y}_{k,\ell}$ has a unique top singular triplet $(\sigma_{1_k,\ell}, \mathbf{u}_{1_k,\ell}, \mathbf{v}_{1_k,\ell})$, so the ideal fit (2) is well-defined. By definition of the ideal fit and the ideal component,

$$\bar{\mathbf{b}}_{k,\ell}\bar{\mathbf{a}}_{k,\ell}^\top \mathbf{X} = \sigma_{1_k,\ell}\mathbf{u}_{1_k,\ell}\mathbf{v}_{1_k,\ell}^\top, \qquad \mathbf{b}_k^\star \mathbf{a}_k^{\star\top}\mathbf{X} = \sigma_k^\star \mathbf{u}_k^\star \mathbf{v}_k^{\star\top}. \tag{18}$$

Step 2 (Decomposition into singular-vector and singular-value errors). Using (18) and the definition of $B_{k,\ell}$ in (4), we add and subtract $\sigma_k^\star \mathbf{u}_{1_k,\ell}\mathbf{v}_{1_k,\ell}^\top$ and apply the triangle inequality:

$$\begin{aligned} B_{k,\ell} &= \left\|\sigma_{1_k,\ell}\mathbf{u}_{1_k,\ell}\mathbf{v}_{1_k,\ell}^\top - \sigma_k^\star \mathbf{u}_k^\star \mathbf{v}_k^{\star\top}\right\|_F \\ &= \left\|(\sigma_{1_k,\ell} - \sigma_k^\star)\mathbf{u}_{1_k,\ell}\mathbf{v}_{1_k,\ell}^\top + \sigma_k^\star(\mathbf{u}_{1_k,\ell}\mathbf{v}_{1_k,\ell}^\top - \mathbf{u}_k^\star \mathbf{v}_k^{\star\top})\right\|_F \\ &\le \underbrace{|\sigma_k^\star - \sigma_{1_k,\ell}|}_{\text{singular-value error}} + \sigma_k^\star \underbrace{\left\|\mathbf{u}_k^\star \mathbf{v}_k^{\star\top} - \mathbf{u}_{1_k,\ell}\mathbf{v}_{1_k,\ell}^\top\right\|_F}_{\text{singular-vector error}}, \end{aligned} \tag{19}$$

using $\left\|\mathbf{u}_{1_k,\ell}\mathbf{v}_{1_k,\ell}^\top\right\|_F = 1$. The decomposition is natural because the actual and ideal objects in (18) are each a scalar times a unit-Frobenius rank-1 term, so the discrepancy decomposes cleanly into a scale mismatch and a direction mismatch.

Step 3 (Singular-value error via Weyl). By Theorem 5, $|\sigma_k^\star - \sigma_{1_k,\ell}| \le \|\mathbf{Y}_k^\star - \mathbf{Y}_{k,\ell}\|_2 \le \|\mathbf{Y}_k^\star - \mathbf{Y}_{k,\ell}\|_F$.

Step 4 (Singular-vector error via Wedin). Insert $\pm\,\mathbf{u}_k^\star\mathbf{v}_{1_k,\ell}^\top$ and apply the triangle inequality:

$$\left\|\mathbf{u}_k^\star\mathbf{v}_k^{\star\top} - \mathbf{u}_{1_k,\ell}\mathbf{v}_{1_k,\ell}^\top\right\|_F \le \|\mathbf{v}_k^\star - \mathbf{v}_{1_k,\ell}\|_2 + \|\mathbf{u}_k^\star - \mathbf{u}_{1_k,\ell}\|_2 \le \sqrt{2}\sqrt{\sin^2\alpha + \sin^2\beta}, \tag{20}$$

where $\alpha := \angle(\mathbf{u}_k^\star, \mathbf{u}_{1_k,\ell})$, $\beta := \angle(\mathbf{v}_k^\star, \mathbf{v}_{1_k,\ell})$, and we used $\|\mathbf{u} - \tilde{\mathbf{u}}\|_2 \le \sqrt{2}\sin\angle(\mathbf{u}, \tilde{\mathbf{u}})$ for unit vectors. To apply Wedin's theorem (Theorem 6) we first verify the gap condition: by Weyl's inequality, $|\sigma_j(\mathbf{Y}_{k,\ell}) - \sigma_j(\mathbf{Y}_k^\star)| \le \|\mathbf{Y}_{k,\ell} - \mathbf{Y}_k^\star\|_2$ for every j, which combined with the hypothesis $\|\mathbf{Y}_{k,\ell} - \mathbf{Y}_k^\star\|_2 < \frac{1}{2}\min_{j>k}|\sigma_k^\star - \sigma_j^\star|$ yields $\delta_k := \min\{\min_{j\ge 2}|\sigma_1(\mathbf{Y}_{k,\ell}) - \sigma_j(\mathbf{Y}_{k,\ell})|, \sigma_1(\mathbf{Y}_{k,\ell})\} \ge \mathcal{T}_k^\star/2 > 0$. Theorem 6 then gives $\sin^2\alpha + \sin^2\beta \le 2\|\mathbf{Y}_k^\star - \mathbf{Y}_{k,\ell}\|_F^2/(\mathcal{T}_k^\star)^2$. Plugging back into (20) and using $2\sqrt{2} < 3$,

$$\left\|\mathbf{u}_k^\star\mathbf{v}_k^{\star\top} - \mathbf{u}_{1_k,\ell}\mathbf{v}_{1_k,\ell}^\top\right\|_F \le \frac{2\sqrt{2}}{\mathcal{T}_k^\star}\|\mathbf{Y}_k^\star - \mathbf{Y}_{k,\ell}\|_F \le \frac{3}{\mathcal{T}_k^\star}\|\mathbf{Y}_k^\star - \mathbf{Y}_{k,\ell}\|_F. \tag{21}$$

Step 5 (Combine). Substituting Steps 3 and (21) into (19):

$$B_{k,\ell} \le \frac{3\sigma_k^\star}{\mathcal{T}_k^\star}\|\mathbf{Y}_k^\star - \mathbf{Y}_{k,\ell}\|_F + \|\mathbf{Y}_k^\star - \mathbf{Y}_{k,\ell}\|_F = C_k\|\mathbf{Y}_k^\star - \mathbf{Y}_{k,\ell}\|_F, \quad C_k := \tfrac{3\sigma_k^\star}{\mathcal{T}_k^\star} + 1. \tag{22}$$

□

E.2 Proof of Theorem 1

For the reader's convenience, we restate the theorem before giving the proof.

Theorem 1 (Restated). *Under Assumptions 1 to 3, define $m_1 = \mathcal{F}_1$ and $m_k = \max\{\mathcal{F}_k, \frac{1}{k} + \frac{k-1}{k}m_{k-1}\}$ for $k \ge 2$, and let $s_1 = 1$ with s_{k+1} chosen via (32). Then for all $k \in [r]$ and all $\ell \ge s_k - 1$, $G_{k,\ell} \le 3\mathcal{R}_k(\ell - s_k + 2)\, m_k^{\ell - s_k + 1}$.*

Proof. We proceed by strong induction on the component index k. We first establish a coupled *triple recursion* on the numerical error, deflation mismatch, and total error; then construct upper-bounding *surrogate sequences* that admit closed-form decay; finally, verify the *five warm-up conditions* on s_k that make the surrogates valid. To start, let s_k satisfy the property that $\|\mathbf{Y}_{k,\ell} - \mathbf{Y}_k^\star\|_F \le \sigma_1^\star$.

Base case ($k = 1$). $\mathbf{Y}_{1,\ell} = \mathbf{Y}$ for all ℓ, so the first deflation target is exact and $B_{1,\ell} = 0$. Then $G_{1,\ell} = D_{1,\ell} \le \mathcal{F}_1^\ell D_{1,0} \le \mathcal{R}_1\mathcal{F}_1^\ell$ by Assumption 1 and Lemma 7. Since $m_1 = \mathcal{F}_1$ and $s_1 = 1$, $G_{1,\ell} \le \mathcal{R}_1 m_1^\ell \le 3\mathcal{R}_1(\ell+1)m_1^\ell$, which satisfies (10). The tighter factor 1 for the base case is loosened to 3 once deflation enters at $k \ge 2$, because the surrogate initialization $\hat{G}_{k,s_k-1} = D_{k,s_k-1} + \hat{B}_{k,s_k-1} + \hat{B}_{k,s_k-2} \le 3\mathcal{R}_k$ via Lemma 7.

Inductive step: triple recursion. Fix $k \ge 2$ and assume (10) holds for every $k' < k$. We collect the three error recursions used below.

(i) Numerical error. Lemma 8 gives, for $\ell \ge s_k$,

$$D_{k,\ell} \le \mathcal{F}_k\big(D_{k,\ell-1} + B_{k,\ell} + B_{k,\ell-1}\big). \tag{23}$$

(ii) Deflation mismatch. Combining Lemma 9 with Lemma 10,

$$B_{k,\ell} \le C_k\|\mathbf{Y}_{k,\ell} - \mathbf{Y}_k^\star\|_F \le C_k\sum_{k'<k} G_{k',\ell-1}. \tag{24}$$

The key structural observation is that $B_{k,\ell}$ depends only on the *earlier* components' total errors $G_{k',\ell-1}$ with $k' < k$: predecessor convergence automatically forces $B_{k,\ell} \to 0$.

(iii) Total error. Unrolling (23) from round $s_k - 1$ up to round ℓ and using $G_{k,\ell} \le D_{k,\ell} + B_{k,\ell}$,

$$G_{k,\ell} \le \mathcal{F}_k^{\ell - s_k + 1} D_{k,s_k-1} + \sum_{\ell'=s_k-1}^{\ell-1} \mathcal{F}_k^{\ell-\ell'} \big(B_{k,\ell'} + B_{k,\ell'-1}\big) + B_{k,\ell}. \tag{25}$$

The three recursions (23)–(25) are the natural triple-recursion structure for parallel deflation, with the mismatch bound supplied by our Wedin-based Lemma 10.

Surrogate sequences. Because the right-hand sides of (24)–(25) are coupled through $\{G_{k',\cdot}\}_{k'<k}$, we replace the actual sequences with explicit upper-bounding surrogates. Let $\hat{s}_k$ denote the earliest round at which the Wedin gap condition in Lemma 10 holds; $\hat{s}_k \le s_k$. Following Lemma 14, define the *surrogate mismatch* and *surrogate total error* piecewise:

$$\hat{B}_{k,\ell} = \begin{cases} \mathcal{R}_k & \text{if } \ell < \hat{s}_k, \\ \min\{\mathcal{R}_k,\ m_{k-1}^{\ell-\hat{s}_k}(\ell - \hat{s}_k + 1)\,\hat{B}_{k,\hat{s}_k}\} & \text{if } \ell \ge \hat{s}_k, \end{cases} \tag{26}$$

with boundary value $\hat{B}_{k,\hat{s}_k} := C_k \sum_{k'<k} \hat{G}_{k',\hat{s}_k-1}$, and

$$\hat{G}_{k,\ell} = \begin{cases} \hat{G}_{k,s_k-1} & \text{if } \ell < s_k, \\ m_k^{\ell-s_k+1}(\ell - s_k + 2)\,\hat{G}_{k,s_k-1} & \text{if } \ell \ge s_k, \end{cases} \tag{27}$$

with initialization $\hat{G}_{k,s_k-1} = D_{k,s_k-1} + \hat{B}_{k,s_k-1} + \hat{B}_{k,s_k-2}$. Note $\hat{G}_{k,s_k-1} \le 3\mathcal{R}_k$ by Lemma 7 and the $\hat{B}_{k,\cdot} \le \mathcal{R}_k$ ceiling in (26).

By Lemma 14, applying the surrogate domination directly,

$$G_{k,\ell} \le \hat{G}_{k,\ell} \le m_k^{\ell-s_k+1}(\ell - s_k + 2)\,\hat{G}_{k,s_k-1} \le 3\mathcal{R}_k(\ell - s_k + 2)\,m_k^{\ell-s_k+1} \quad \text{for all } \ell \ge s_k.$$

It remains to exhibit the conditions on s_{k+1} under which Lemma 14 applies.

Conditions arising in the proof of Lemma 14. The proof of Lemma 14 (given below) requires five inequalities; we collect them here so that the present theorem can secure them via the choice of s_{k+1}. For each condition we identify the proof step at which it is invoked.

(C1) Predecessor envelope folding (per-predecessor form). For every $k' \le k$,

$$m_{k'}^{\hat{s}_{k+1}-s_{k'}}(\hat{s}_{k+1} - s_{k'} + 1) \ \le\ \tfrac{1}{k}.$$

Origin: used in the bound on $B_{k+1,\ell}$ in Lemma 14 (case $\ell \ge \hat{s}_{k+1}$) to fold the polynomial factor $(\hat{s}_{k+1} - s_{k'} + 1)$ inherited from each predecessor surrogate $\hat{G}_{k',\cdot}$ into the target geometric envelope and then average the resulting sum of k terms back to a single $\hat{B}_{k+1,\hat{s}_{k+1}}$ value. The per-predecessor form (with $m_{k'}$ rather than the global m_k) is no looser than the corresponding global statement, since $m_k \ge m_{k'}$ for $k' \le k$ (proved below).

(C2) Wedin-gap activation: $\big\|\mathbf{Y}_{k+1,\ell} - \mathbf{Y}^\star_{k+1}\big\|_2 < \frac{1}{2}\mathcal{T}^\star_{k+1}$ for all $\ell \ge \hat{s}_{k+1}$, equivalently $\hat{s}_{k+1} \ge s^{\mathrm{gap}}_{k+1}$ where s^{gap}_{k+1} is the smallest round such that $\sum_{k' \le k} G_{k',\ell-1} \le \mathcal{T}^\star_{k+1}/(2C_{k+1})$. *Origin:* the hypothesis of Lemma 10; without it, $B_{k+1,\ell}$ has no usable upper bound.

(C3) Coupling compatibility: $\gamma_k := \frac{1}{k+1} + \frac{k}{k+1} m_k \le m_{k+1}$. *Origin:* structural; required when invoking Lemma 13 to re-anchor $\hat{B}_{k+1,\cdot}$ to a pure-geometric envelope at rate γ_k and then absorb that rate into m_{k+1}. This holds automatically because $m_{k+1} = \max\{\mathcal{F}_{k+1}, \gamma_k\}$ by (8).

(C4) Contraction compatibility: $\mathcal{F}_{k+1} \le m_{k+1}$. *Origin:* used to absorb the factor $\mathcal{F}_{k+1}$ from Lemma 8 into the target rate m_{k+1} when bounding $D_{k+1,\ell}$. This holds automatically because $m_{k+1} = \max\{\mathcal{F}_{k+1}, \gamma_k\}$ by (8).

(C5) Re-anchor warm-up offset: the offset between the Wedin-activation index $\hat{s}_{k+1}$ and the warm-up index s_{k+1} satisfies both

$$s_{k+1} - \hat{s}_{k+1} - 2 \ \ge\ \frac{(k+1)m_k}{1 - m_k} \qquad \text{and} \qquad m_k^{s_{k+1}-\hat{s}_{k+1}-2}(s_{k+1} - \hat{s}_{k+1} - 1) \ \le\ 1. \tag{28}$$

Origin: the two warm-up requirements of Lemma 13 (with the second instance $s = s_{k+1} - 2$ giving the tightest case): together they convert the linear-times-geometric form $m_k^{\ell-\hat{s}_{k+1}}(\ell - \hat{s}_{k+1} + 1)\hat{B}_{k+1,\hat{s}_{k+1}}$ into the pure-geometric form $\gamma_k^{\ell-s_{k+1}+1}\hat{B}_{k+1,s_{k+1}-1}$ used to majorize the forcing sum in the unrolled D-recurrence at rate m_{k+1} without incurring an additional polynomial factor.

Conditions **(C3)** and **(C4)** are automatic structural facts (no constraint on s_{k+1}); **(C1)**, **(C2)**, and **(C5)** translate, via Lemma 12(ii) (for the first two) or directly (28) (for the third), into explicit per-predecessor lower bounds on s_{k+1}. Before proceeding, record once that m_k is non-decreasing in k: by (8), $\frac{1}{k} + \frac{k-1}{k} m_{k-1}$ is a convex combination of 1 and m_{k-1}, hence $\geq m_{k-1}$, so $m_k = \max\{\mathcal{F}_k, \gamma_{k-1}\} \geq m_{k-1}$.

Deriving (32) from (C1) and (C2). We compute the per-predecessor lower bound on s_{k+1} that simultaneously secures the two non-trivial conditions.

*From **(C1)**.* For each $k' \leq k$, we seek the smallest $\hat{s}_{k+1}$ such that $m_{k'}^{\hat{s}_{k+1}-s_{k'}}(\hat{s}_{k+1} - s_{k'} + 1) \leq 1/k$. Applying Lemma 12(ii) to $g(x) = m_{k'}^x(x+1)$ with $x = \hat{s}_{k+1} - s_{k'}$ and $\epsilon = 1/k$ yields $\hat{s}_{k+1} - s_{k'} \geq \frac{1}{\log m_{k'}} W_{-1}(\frac{m_{k'} \log m_{k'}}{k}) - 1$. Using the wrapper $\hat{W}(a) := \max\{1, -W_{-1}(-a)\}$, the trailing -1 is absorbed (since $\hat{W}(a) \geq 1$ for $a \in (0, 1/e)$ and the wrapper bounds $-W_{-1}(-a) - 1 \leq \hat{W}(a) - 1 + 1 = \hat{W}(a)$ once we pass to the inverse-sign convention $\epsilon \mapsto -\epsilon m \log m$):

$$\hat{s}_{k+1} - s_{k'} \;\geq\; \frac{\hat{W}\Big(\frac{m_{k'}|\log m_{k'}|}{k}\Big)}{\log(1/m_{k'})}, \qquad \forall k' \leq k. \tag{29}$$

*From **(C2)**.* By Lemma 9 and the inductive hypothesis $G_{k',\ell-1} \leq 3\mathcal{R}_{k'}(\ell - s_{k'} + 1)m_{k'}^{\ell-s_{k'}}$,

$$\big\|\mathbf{Y}_{k+1,\ell} - \mathbf{Y}^\star_{k+1}\big\|_F \leq \sum_{k' \leq k} G_{k',\ell-1} \leq 3 \sum_{k' \leq k} \mathcal{R}_{k'}(\ell - s_{k'} + 1) m_{k'}^{\ell-s_{k'}}. \tag{30}$$

A sufficient condition for the Wedin gap is that each summand is at most $\mathcal{T}^\star_{k+1}/(6kC_{k+1})$, i.e., for each $k' \leq k$, $(\ell - s_{k'} + 1)m_{k'}^{\ell-s_{k'}} \leq \frac{\mathcal{T}^\star_{k+1}}{6k\mathcal{R}_{k'}C_{k+1}}$ (the constant $6 = 3 \times 2$ comes from $\hat{G}_{k',\cdot} \leq 3\mathcal{R}_{k'} \cdot m_{k'}^{\cdot}(\cdot+1)$ post-warm-up and the factor-of-two slack in Lemma 10's gap hypothesis). Applying Lemma 12(ii) with $\epsilon_{k'} := \frac{\mathcal{T}^\star_{k+1}}{6k\mathcal{R}_{k'}C_{k+1}}$ and the same $\hat{W}$-absorption,

$$\ell - s_{k'} \;\geq\; \frac{\hat{W}(\epsilon_{k'}\, m_{k'}|\log m_{k'}|)}{\log(1/m_{k'})}, \qquad \forall k' \leq k. \tag{31}$$

*From **(C5)**.* The first inequality in (28) is an offset between s_{k+1} and $\hat{s}_{k+1}$ that does not depend on k'; it contributes the additive term $(k+1)m_k/(1-m_k)+2$. The second inequality $m_k^{s_{k+1}-\hat{s}_{k+1}-2}(s_{k+1} - \hat{s}_{k+1} - 1) \leq 1$ is a Lambert-W condition of the same form as (29) (applied to $g(x) = m_k^x(x+1)$ with $\epsilon = 1$), so it contributes an additional offset $\hat{W}(m_k|\log m_k|)/\log(1/m_k)$ between $\hat{s}_{k+1}$ and $s_{k+1} - 2$.

Combining. The bounds (29) and (31) are per-predecessor lower bounds on the gap from anchor $s_{k'}$, while (28) adds a uniform offset between $\hat{s}_{k+1}$ and s_{k+1}, so a sufficient choice of s_{k+1} is

$$\begin{aligned} s_{k+1} \;\geq\; \max_{k' \leq k} \Bigg\{ s_{k'} \;+\; \underbrace{\frac{\hat{W}\Big(\frac{m_{k'}|\log m_{k'}|}{k}\Big)}{\log(1/m_{k'})}}_{\text{from (C1), (29)}} \;+\; \underbrace{\frac{\hat{W}\Big(\frac{\mathcal{T}^\star_{k+1}\, m_{k'}|\log m_{k'}|}{6k\mathcal{R}_{k'}C_{k+1}}\Big)}{\log(1/m_{k'})}}_{\text{from (C2), (31)}} \Bigg\} \\ +\; \underbrace{\frac{(k+1)m_k}{1-m_k} + 2 \;+\; \frac{\hat{W}\big(m_k|\log m_k|\big)}{\log(1/m_k)}}_{\text{from (C5), (28)}}. \end{aligned} \tag{32}$$

By Lemma 12 the wrapper $\hat{W}(\cdot)$ is well-defined and $\hat{W}(a) = \mathcal{O}(\max\{1, \log(1/a)\})$, and the re-anchor offset is bounded for any fixed $m_k < 1$, so (32) produces a finite, polynomial-in-$\log$-and-$1/(1-m_k)$ requirement.

Simplified bound on s_{k+1}. The three nested $\hat{W}$ terms in (32) can be folded into a single logarithmic expression by applying the standard Lambert-W upper bound (sharpening the asymptotic of Lemma 12)

$$\hat{W}(a) \;\leq\; \log(1/a) + \log\log(1/a) + 1 \qquad \text{for } a \in (0, 1/e), \tag{33}$$

together with the elementary bound $\log(1/m) \geq 1 - m$ for $m \in (0,1)$ (so that $1/\log(1/m_{k'}) \leq 1/(1-m_{k'})$). For each predecessor $k' \leq k$, applying (33) to $a = m_{k'}|\log m_{k'}|/k$ gives $\hat{W}\big(\frac{m_{k'}|\log m_{k'}|}{k}\big) \leq \log\big(\frac{k}{m_{k'}|\log m_{k'}|}\big) + \log\log\big(\frac{k}{m_{k'}|\log m_{k'}|}\big) + 1$, and the same expansion applied to the **(C2)**-term $\hat{W}$ replaces k by $6kC_{k+1}\mathcal{R}_{k'}/\mathcal{T}^\star_{k+1}$. Absorbing the $\log\log$ and $+1$ remainders into a generic $\widetilde{\mathcal{O}}(\cdot)$ symbol, (32) simplifies to:

Corollary 11 (Simplified warm-up bound)**.** *There exist absolute constants $C_1, C_2, C_3 > 0$ such that the per-predecessor lower bound (32) on s_{k+1} is implied by*

$$s_{k+1} \;\geq\; \max_{k' \leq k} \left\{ s_{k'} + \frac{C_1 \,\log\!\big(k\, C_{k+1}\mathcal{R}_{k'}/\mathcal{T}^\star_{k+1}\big) + C_2}{1 - m_{k'}} \right\} \;+\; \frac{(k+1)m_k}{1-m_k} \;+\; \frac{C_3}{1-m_k}. \tag{34}$$

Equivalently, in $\widetilde{\mathcal{O}}$-form (suppressing constants and $\log\log$ factors),

$$s_{k+1} \;=\; \max_{k' \leq k} s_{k'} \;+\; \widetilde{\mathcal{O}}\left(\frac{\log\!\big(k\, C_{k+1}\mathcal{R}_{k'}/\mathcal{T}^\star_{k+1}\big)}{1-m_k} \right) \;+\; \frac{(k+1)m_k}{1-m_k}. \tag{35}$$

Proof. Each $\hat{W}$ term in (32) is bounded via (33). The denominator $\log(1/m_{k'})$ is replaced by its lower bound $1 - m_{k'}$, which is uniformly bounded above by $1 - m_k$ since $m_{k'} \leq m_k$ (Equation (8)). The two $\hat{W}$ contributions inside the max each yield a $\log(k/(m_{k'}|\log m_{k'}|))$ piece and a $\log(C_{k+1}\mathcal{R}_{k'}/\mathcal{T}^\star_{k+1})$ piece (the latter only from the **(C2)**-term); both are absorbed into $C_1 \log(k\, C_{k+1}\mathcal{R}_{k'}/\mathcal{T}^\star_{k+1})$. The remaining $\log\log$ and $+1$ terms are bounded by a constant C_2, while the third (re-anchor) $\hat{W}$ contributes a constant C_3 once $1/\log(1/m_k) \leq 1/(1-m_k)$ is applied. Combining yields (34); (35) is the same statement with constants and $\log\log$ factors hidden. □

The simplified form (34) is the version referenced in the main-text statement of Theorem 1; it makes explicit the three drivers of the warm-up cost: the predecessor-anchor offset $s_{k'}$, a logarithmic dependence on the spectral-gap-to-magnitude ratio $C_{k+1}\mathcal{R}_{k'}/\mathcal{T}^\star_{k+1}$, and the geometric $1/(1-m_k)$ slowdown as the effective rate approaches one.

Conclusion. With s_{k+1} chosen per (32), conditions **(C1)**, **(C2)**, and **(C5)** are secured; **(C3)** and **(C4)** are automatic from (8). Lemma 14 then yields $G_{k,\ell} \leq \hat{G}_{k,\ell}$ for all $\ell \geq s_k - 1$, and substituting (27) together with $\hat{G}_{k,s_k-1} \leq 3\mathcal{R}_k$ gives

$$G_{k,\ell} \leq m_k^{\ell - s_k + 1}(\ell - s_k + 2)\,\hat{G}_{k,s_k-1} \leq 3\mathcal{R}_k(\ell - s_k + 2)\, m_k^{\ell - s_k + 1},$$

which is the bound (10). This the proof. □

E.3 Auxiliary Lemmas

Lemma 12 (Lambert-W Bound)**.** *Let $m \in (0,1)$ and $g(x) = m^x(x+1)$.*

(i) If $\epsilon \geq -(em\log m)^{-1}$, then $g(x) \leq \epsilon$ for all $x \geq 0$.
(ii) If $\epsilon < -(em\log m)^{-1}$, then $g(x) \leq \epsilon$ whenever $x \geq \frac{1}{\log m} W_{-1}(\epsilon m \log m) - 1$.

Moreover, the wrapper $\hat{W}(a) := \max\{1, -W_{-1}(-a)\}$ satisfies $\hat{W}(a) = \mathcal{O}(\max\{1, \log(1/a)\})$, and admits the explicit upper bound $\hat{W}(a) \leq \log(1/a) + \log\log(1/a) + 1$ for $a \in (0, 1/e)$.

Proof. Differentiating $g(x) = m^x(x+1)$ gives $g'(x) = m^x\big(1 + (x+1)\log m\big)$, so g attains its maximum at $x^\star = (\log(1/m))^{-1} - 1$ with value $g(x^\star) = -(em\log m)^{-1}$.

Case (i). If $\epsilon \geq g(x^\star)$, then $g(x) \leq g(x^\star) \leq \epsilon$ for every $x \geq 0$, and any non-negative x satisfies the bound.

Case (ii). Suppose $\epsilon < g(x^\star)$. On $(x^\star, \infty)$, g is strictly decreasing, so the equation $g(x) = \epsilon$ has a unique solution $x_\epsilon > x^\star$. Taking logarithms and substituting $y = (x+1)\log m$ transforms $g(x) = \epsilon$ into $ye^y = \epsilon m \log m$, which is solved by $y = W_{-1}(\epsilon m \log m)$ on the decreasing branch

(since $\epsilon m \log m \in (-1/e, 0)$). Hence $x_\epsilon = \frac{1}{\log m} W_{-1}(\epsilon m \log m) - 1$, and every $x \geq x_\epsilon$ satisfies $g(x) \leq \epsilon$.

Asymptotic statement. By the standard expansion $-W_{-1}(-a) = \log(1/a) + \log\log(1/a) + o(1)$ as $a \to 0^+$, we get $\hat{W}(a) = \mathcal{O}(\max\{1, \log(1/a)\})$. The explicit non-asymptotic bound $\hat{W}(a) \leq \log(1/a) + \log\log(1/a) + 1$ for $a \in (0, 1/e)$ follows from the standard Lambert-W inequality $-W_{-1}(-e^{-u-1}) \leq u + \log u + 1$ for $u \geq 1$ (substitute $u = \log(1/a) - 1$). □

Lemma 13 (Geometric Re-anchoring of $\hat{B}$)**.** *Fix $k \geq 1$. Suppose the surrogate sequence $\hat{B}_{k+1,\ell}$ is given by (38) with $\hat{B}_{k+1,\hat{s}_{k+1}} \leq \mathcal{R}_{k+1}$ and rate $m_k \in (0,1)$. If an integer s satisfies $s \geq \hat{s}_{k+1}$, $m_k^{s-\hat{s}_{k+1}}(s - \hat{s}_{k+1} + 1) \leq 1$, and*

$$s - \hat{s}_{k+1} \;\geq\; \frac{(k+1)m_k}{1 - m_k}, \tag{36}$$

then for every $\ell \geq s$,

$$\hat{B}_{k+1,\ell} \;\leq\; \gamma_k^{\ell - s}\, \hat{B}_{k+1,s}, \qquad \gamma_k = \tfrac{1}{k+1} + \tfrac{k}{k+1} m_k. \tag{37}$$

Proof. At index s, the conditions $\hat{B}_{k+1,\hat{s}_{k+1}} \leq \mathcal{R}_{k+1}$ and $m_k^{s-\hat{s}_{k+1}}(s - \hat{s}_{k+1} + 1) \leq 1$ together imply $m_k^{s-\hat{s}_{k+1}}(s - \hat{s}_{k+1} + 1)\hat{B}_{k+1,\hat{s}_{k+1}} \leq \mathcal{R}_{k+1}$, so the $\min\{\cdot, \mathcal{R}_{k+1}\}$ ceiling is inactive at s and

$$\hat{B}_{k+1,s} = m_k^{s-\hat{s}_{k+1}}(s - \hat{s}_{k+1} + 1)\hat{B}_{k+1,\hat{s}_{k+1}}, \quad \text{so} \quad \hat{B}_{k+1,\hat{s}_{k+1}} = \left[m_k^{s-\hat{s}_{k+1}}(s - \hat{s}_{k+1} + 1)\right]^{-1} \hat{B}_{k+1,s}.$$

For any $\ell \geq s$, the post-warm-up clause of (38) gives

$$\begin{aligned}
\hat{B}_{k+1,\ell} &\leq m_k^{\ell - \hat{s}_{k+1}}(\ell - \hat{s}_{k+1} + 1)\hat{B}_{k+1,\hat{s}_{k+1}} \\
&= m_k^{\ell - s} \cdot \frac{\ell - \hat{s}_{k+1} + 1}{s - \hat{s}_{k+1} + 1}\, \hat{B}_{k+1,s} \\
&= m_k^{\ell - s}\, \hat{B}_{k+1,s} \prod_{\ell'=s+1}^{\ell} \frac{\ell' - \hat{s}_{k+1} + 1}{\ell' - \hat{s}_{k+1}} \;\leq\; m_k^{\ell - s} \left(\frac{s - \hat{s}_{k+1} + 1}{s - \hat{s}_{k+1}}\right)^{\ell - s} \hat{B}_{k+1,s},
\end{aligned}$$

where the last inequality bounds each of the $\ell - s$ factors by its largest value at $\ell' = s + 1$ (the ratio $(\ell' - \hat{s}_{k+1} + 1)/(\ell' - \hat{s}_{k+1})$ is decreasing in ℓ'). The warm-up condition (36), equivalent to $\frac{s - \hat{s}_{k+1} + 1}{s - \hat{s}_{k+1}} \leq 1 + \frac{1}{(k+1)m_k/(1-m_k)} = \frac{(k+1)m_k + 1 - m_k}{(k+1)m_k} = \frac{km_k + 1}{(k+1)m_k}$, then yields

$$m_k \cdot \frac{s - \hat{s}_{k+1} + 1}{s - \hat{s}_{k+1}} \;\leq\; \frac{km_k + 1}{k+1} \;=\; \tfrac{1}{k+1} + \tfrac{k}{k+1} m_k \;=\; \gamma_k,$$

which combined with the displayed bound gives (37). □

Lemma 14 (Surrogate Sequence Domination)**.** *Under Assumptions 1 to 3, let $\hat{s}_k$ denote the earliest round at which the Wedin gap condition of Lemma 10 holds for component k, and let s_k be the warm-up round chosen via (32). The choice (32) ensures $\hat{s}_k \leq s_k$ for every $k \geq 2$ via the additive offset $(k+1)m_k/(1-m_k) + 2 + \hat{W}(m_k |\log m_k|)/\log(1/m_k)$ contributed by **(C5)**; for indices $j < \hat{s}_k$ we adopt the convention $\hat{B}_{k,j} := \mathcal{R}_k$. Define the surrogate sequences piecewise:*

$$\hat{B}_{k,\ell} = \begin{cases} \mathcal{R}_k & \text{if } \ell < \hat{s}_k, \\ \min\left\{\mathcal{R}_k,\; m_{k-1}^{\ell - \hat{s}_k}(\ell - \hat{s}_k + 1)\, \hat{B}_{k,\hat{s}_k}\right\} & \text{if } \ell \geq \hat{s}_k, \end{cases} \tag{38}$$

$$\hat{G}_{k,\ell} = \begin{cases} \hat{G}_{k,s_k - 1} & \text{if } \ell < s_k, \\ m_k^{\ell - s_k + 1}(\ell - s_k + 2)\, \hat{G}_{k,s_k - 1} & \text{if } \ell \geq s_k, \end{cases} \tag{39}$$

with boundary values $\hat{B}_{k,\hat{s}_k} := C_k \sum_{k' < k} \hat{G}_{k',\hat{s}_k - 1}$ and $\hat{G}_{k,s_k - 1} := D_{k,s_k - 1} + \hat{B}_{k,s_k - 1} + \hat{B}_{k,s_k - 2}$. If s_{k+1} is chosen via (32), then $\hat{B}_{k,\ell} \geq B_{k,\ell}$ for every $k \in [r]$ and every $\ell \geq 0$, and $\hat{G}_{k,\ell} \geq G_{k,\ell}$ for every $\ell \geq s_k - 1$.

Proof. We proceed by strong induction on k. At each inequality below we identify which of the conditions **(C1)**–**(C4)** (collected in the proof of Theorem 1) is being invoked.

Base case ($k = 1$). Since $\mathbf{Y}_{1,\ell} = \mathbf{Y}$, the first target is exact, so $B_{1,\ell} = 0$ by (4); combined with $B_{1,\ell} \le \mathcal{R}_1$ from Lemma 7, the surrogate $\hat{B}_{1,\ell} = \mathcal{R}_1$ dominates trivially. For the total error, Lemma 8 unrolls to $G_{1,\ell} = D_{1,\ell} \le \mathcal{F}_1^{\ell-s_1+1} D_{1,s_1-1}$, which is upper-bounded by $m_1^{\ell-s_1+1}(\ell - s_1 + 2)\hat{G}_{1,s_1-1} = \hat{G}_{1,\ell}$ since $m_1 = \mathcal{F}_1$ and $(\ell - s_1 + 2) \ge 1$.

Inductive step. Fix $k \ge 1$ and assume the inductive hypothesis: for every $k' \le k$, $\hat{B}_{k',\ell} \ge B_{k',\ell}$ for every $\ell \ge 0$ and $\hat{G}_{k',\ell} \ge G_{k',\ell}$ for every $\ell \ge s_{k'} - 1$. We show $\hat{B}_{k+1,\ell} \ge B_{k+1,\ell}$ for every $\ell \ge 0$, and then $\hat{G}_{k+1,\ell} \ge G_{k+1,\ell}$ for every $\ell \ge s_{k+1} - 1$.

We bound $B_{k+1,\ell}$ first. If $\ell < \hat{s}_{k+1}$, then by Lemma 7, $B_{k+1,\ell} \le \mathcal{R}_{k+1} = \hat{B}_{k+1,\ell}$. If $\ell \ge \hat{s}_{k+1}$, then the Wedin-gap hypothesis of Lemma 10 is active by condition **(C2)**, and by definition of $\hat{s}_{k+1}$, $\hat{s}_{k+1} - 1 \ge s_{k'}$ for every $k' \le k$, so each predecessor surrogate lies on the post-warm-up branch of (39). Combining Lemma 9 and Lemma 10 with the inductive hypothesis $G_{k',\ell-1} \le \hat{G}_{k',\ell-1}$,

$$\begin{aligned}
B_{k+1,\ell} &\le C_{k+1} \sum_{k' \le k} \hat{G}_{k',\ell-1} \\
&= C_{k+1} \sum_{k' \le k} m_{k'}^{\ell - s_{k'}} (\ell - s_{k'} + 1)\, \hat{G}_{k',s_{k'}-1} \\
&\le C_{k+1} \sum_{k' \le k} m_{k'}^{\ell - \hat{s}_{k+1}}\, m_{k'}^{\hat{s}_{k+1} - s_{k'}} (\ell - \hat{s}_{k+1} + 1)(\hat{s}_{k+1} - s_{k'} + 1)\, \hat{G}_{k',s_{k'}-1} \\
&\le C_{k+1}\, m_k^{\ell - \hat{s}_{k+1}} (\ell - \hat{s}_{k+1} + 1) \cdot \tfrac{1}{k} \sum_{k' \le k} \hat{G}_{k',s_{k'}-1},
\end{aligned}$$

where the third inequality applies $a + b - 1 \le ab$ for $a, b \ge 1$ (with $a = \ell - \hat{s}_{k+1} + 1$ and $b = \hat{s}_{k+1} - s_{k'} + 1$) to split the polynomial factor along $\hat{s}_{k+1}$, and the fourth uses condition **(C1)** ($m_{k'}^{\hat{s}_{k+1} - s_{k'}} (\hat{s}_{k+1} - s_{k'} + 1) \le 1/k$) together with the monotonicity $m_{k'} \le m_k$ for $k' \le k$ (from (8)). By **(C1)** again, $\hat{G}_{k',\hat{s}_{k+1}-1} = m_{k'}^{\hat{s}_{k+1} - s_{k'}} (\hat{s}_{k+1} - s_{k'} + 1) \hat{G}_{k',s_{k'}-1} \le (1/k) \hat{G}_{k',s_{k'}-1}$, so $\hat{B}_{k+1,\hat{s}_{k+1}} := C_{k+1} \sum_{k' \le k} \hat{G}_{k',\hat{s}_{k+1}-1} \le (C_{k+1}/k) \sum_{k' \le k} \hat{G}_{k',s_{k'}-1}$, and combining with $B_{k+1,\ell} \le \mathcal{R}_{k+1}$ from Lemma 7,

$$B_{k+1,\ell} \;\le\; \min\!\big\{\mathcal{R}_{k+1},\; m_k^{\ell - \hat{s}_{k+1}} (\ell - \hat{s}_{k+1} + 1)\, \hat{B}_{k+1,\hat{s}_{k+1}}\big\} \;=\; \hat{B}_{k+1,\ell}, \tag{40}$$

which is the post-warm-up branch of (38). The min-ceiling enforces $\hat{B}_{k+1,\hat{s}_{k+1}} \le \mathcal{R}_{k+1}$, which we use when invoking Lemma 13. This establishes $\hat{B}_{k+1,\ell} \ge B_{k+1,\ell}$ for every $\ell \ge 0$.

We now bound $G_{k+1,\ell}$, starting with $\ell \ge s_{k+1}$. Iterating Lemma 8 from round $s_{k+1} - 1$ up to round ℓ and using (40),

$$D_{k+1,\ell} \;\le\; \mathcal{F}_{k+1}^{\ell - s_{k+1}+1} D_{k+1,s_{k+1}-1} + \sum_{\ell'=s_{k+1}-1}^{\ell} \mathcal{F}_{k+1}^{\ell-\ell'} \big(\hat{B}_{k+1,\ell'} + \hat{B}_{k+1,\ell'-1}\big), \tag{41}$$

where we re-indexed the sum to start at $s_{k+1} - 1$ (the dropped earlier summand $B_{k+1,s_{k+1}-2}$ is non-negative). To collapse this sum into a clean envelope at rate m_{k+1}, we re-anchor $\hat{B}_{k+1,\cdot}$ from $\hat{s}_{k+1}$ to $s_{k+1} - 1$ via Lemma 13. Apply that lemma twice, at $s = s_{k+1} - 1$ and at $s = s_{k+1} - 2$:

$$\hat{B}_{k+1,\ell'} \;\le\; \gamma_k^{\ell' - (s_{k+1}-1)}\, \hat{B}_{k+1,s_{k+1}-1}, \qquad \forall\, \ell' \ge s_{k+1} - 1, \tag{42}$$

$$\hat{B}_{k+1,\ell'-1} \;\le\; \gamma_k^{\ell' - (s_{k+1}-1)}\, \hat{B}_{k+1,s_{k+1}-2}, \qquad \forall\, \ell' \ge s_{k+1} - 1, \tag{43}$$

where the second display has matching exponent $(\ell' - 1) - (s_{k+1} - 2) = \ell' - s_{k+1} + 1$. The two preconditions of Lemma 13 at $s = s_{k+1} - 2$, namely $m_k^{s_{k+1} - \hat{s}_{k+1} - 2}(s_{k+1} - \hat{s}_{k+1} - 1) \le 1$ and $s_{k+1} - \hat{s}_{k+1} - 2 \ge (k+1)m_k/(1 - m_k)$, are exactly the two clauses of **(C5)** secured by (32); at $s = s_{k+1} - 1$ the offset is one larger and so no looser, and $g(x) = m_k^x(x + 1)$ is decreasing past its peak at $x = 1/\log(1/m_k) - 1$, which lies below **(C5)**'s offset (using $1/\log(1/m_k) \le 1/(1 - m_k)$).

Substituting (42) and (43) into (41),

$$\begin{aligned} D_{k+1,\ell} &\le m_{k+1}^{\ell-s_{k+1}+1} D_{k+1,s_{k+1}-1} \;+\; \sum_{\ell'=s_{k+1}-1}^{\ell} m_{k+1}^{\ell-\ell'}\, \gamma_k^{\ell'-s_{k+1}+1} \big(\hat{B}_{k+1,s_{k+1}-1} + \hat{B}_{k+1,s_{k+1}-2}\big) \\ &\le m_{k+1}^{\ell-s_{k+1}+1} D_{k+1,s_{k+1}-1} \;+\; m_{k+1}^{\ell-s_{k+1}+1} (\ell - s_{k+1} + 2) \big(\hat{B}_{k+1,s_{k+1}-1} + \hat{B}_{k+1,s_{k+1}-2}\big) \\ &\le m_{k+1}^{\ell-s_{k+1}+1} (\ell - s_{k+1} + 2) \big(D_{k+1,s_{k+1}-1} + \hat{B}_{k+1,s_{k+1}-1} + \hat{B}_{k+1,s_{k+1}-2}\big), \end{aligned}$$

where the first inequality uses **(C4)** ($\mathcal{F}_{k+1} \le m_{k+1}$), and the second bounds the inner sum by $(\ell - s_{k+1} + 2)$ copies of $m_{k+1}^{\ell-s_{k+1}+1}$ via **(C3)** ($\gamma_k \le m_{k+1}$) over $\ell - s_{k+1} + 2$ consecutive indices. By (40), (42), and $\gamma_k \le m_{k+1}$, we also have $B_{k+1,\ell} \le \hat{B}_{k+1,\ell} \le m_{k+1}^{\ell-s_{k+1}+1} (\ell - s_{k+1} + 2)\, \hat{B}_{k+1,s_{k+1}-1}$. Combining via $G_{k+1,\ell} \le D_{k+1,\ell} + B_{k+1,\ell}$,

$$\begin{aligned} G_{k+1,\ell} &\le m_{k+1}^{\ell-s_{k+1}+1} (\ell - s_{k+1} + 2) \big(D_{k+1,s_{k+1}-1} + \hat{B}_{k+1,s_{k+1}-1} + \hat{B}_{k+1,s_{k+1}-2}\big) \\ &= m_{k+1}^{\ell-s_{k+1}+1} (\ell - s_{k+1} + 2)\, \hat{G}_{k+1,s_{k+1}-1} \;=\; \hat{G}_{k+1,\ell}. \end{aligned}$$

For the boundary case $\ell = s_{k+1} - 1$, the surrogate is held constant: $\hat{G}_{k+1,s_{k+1}-1} = D_{k+1,s_{k+1}-1} + \hat{B}_{k+1,s_{k+1}-1} + \hat{B}_{k+1,s_{k+1}-2}$, so by the just-established $\hat{B}_{k+1,j} \ge B_{k+1,j}$ and $\hat{B}_{k+1,s_{k+1}-2} \ge 0$,

$$\hat{G}_{k+1,s_{k+1}-1} \;\ge\; D_{k+1,s_{k+1}-1} + B_{k+1,s_{k+1}-1} \;\ge\; G_{k+1,s_{k+1}-1},$$

where the last step is $G_{k+1,\cdot} \le D_{k+1,\cdot} + B_{k+1,\cdot}$. The size bound $\hat{G}_{k+1,s_{k+1}-1} \le 3\mathcal{R}_{k+1}$ used in (10) follows from Lemma 7 applied to each of the three summands ($D_{k+1,s_{k+1}-1} \le \mathcal{R}_{k+1}$ and $\hat{B}_{k+1,j} \le \mathcal{R}_{k+1}$ by the min-ceiling in (38)). This completes the inductive step. □

E.4 Proof of Proposition 2

Proof. By Lemma 9: $\|\mathbf{Y}_{k,\ell} - \mathbf{Y}_k^\star\|_F \le \sum_{k'<k} G_{k',\ell-1}$. Substituting Theorem 1: each term is $3\mathcal{R}_{k'}(\ell - s_{k'} + 1)\, m_{k'}^{\ell-s_{k'}} \to 0$ since $m_{k'} < 1$. □

E.5 Proof of Theorem 3

Proof. Since $\sigma_{\min}(\mathbf{X}) > 0$, for any $\mathbf{W}$ we have $\|\mathbf{W}\|_F \le \sigma_{\min}(\mathbf{X})^{-1} \|\mathbf{W}\mathbf{X}\|_F$. Applying this with $\mathbf{W} = \mathbf{W}^\star - \sum_k \mathbf{b}_{k,L}\mathbf{a}_{k,L}^\top$ and using the triangle inequality: $\left\|\mathbf{W}^\star\mathbf{X} - \sum_k \mathbf{b}_{k,L}\mathbf{a}_{k,L}^\top\mathbf{X}\right\|_F \le \left\|\mathbf{Y} - \sum_{k=1}^r \mathbf{b}_k^\star\mathbf{a}_k^{\star\top}\mathbf{X}\right\|_F + \sum_{k=1}^r \left\|\mathbf{b}_k^\star\mathbf{a}_k^{\star\top}\mathbf{X} - \mathbf{b}_{k,L}\mathbf{a}_{k,L}^\top\mathbf{X}\right\|_F = \sum_{k=r+1}^{r^\star} \sigma_k^\star + \sum_{k=1}^r G_{k,L}$, where the first equality uses Eckart-Young-Mirsky: the best rank-r approximation of $\mathbf{Y}$ in Frobenius norm leaves residual $\sum_{k>r} \sigma_k^\star$. Dividing by $\sigma_{\min}(\mathbf{X})$ yields the stated bound. □

E.6 Proof of Theorem 4

Proof. Decompose into T_1 (truncation, bounded by Eckart-Young-Mirsky), T_2 (noise perturbation, bounded by Weyl/Wedin and Gaussian concentration following [Vandchali et al., 2025, Lemma C.1]), and T_3 (algorithmic error $\sum_k G_{k,L}$). The algorithmic term inherits self-correction; the statistical term matches the sequential baseline. Dividing by $\sigma_{\min}(\mathbf{X})$ yields the stated bound.

The key observation is that parallelization does not affect the noise term T_2. The noise $\mathcal{E}$ enters through the data model $\mathbf{Y} = \mathbf{W}^\star\mathbf{X} + \mathcal{E}$, which is identical for both sequential and parallel algorithms. The T_2 bound depends only on the spectral perturbation $\|\mathbf{Y} - \mathbf{W}^\star\mathbf{X}\|_2 = \|\mathcal{E}\|_2$, controlled by standard Gaussian concentration (e.g., $\|\mathcal{E}\|_2 \le \varepsilon(\sqrt{m} + \sqrt{n})$ with high probability). This perturbation is a property of the data, not the algorithm, so the parallel and sequential statistical terms are the same. The only difference is in T_3: for the parallel algorithm, $\sum_k G_{k,L} \to 0$ by Theorem 1, while for the sequential baseline, the corresponding sum involves permanently fixed numerical errors. □

E.7 Verification that $m_k < 1$

We show by induction on k that the effective rate m_k defined in (8) satisfies $m_k \in (0, 1)$ for every $k \in [r]$.

Base case ($k = 1$). By definition, $m_1 = \mathcal{F}_1$, and Assumption 1 requires $\mathcal{F}_k \in (0, 1)$ for every k. Hence $m_1 \in (0, 1)$.

Inductive step ($k \geq 2$). Assume $m_{k-1} \in (0, 1)$. The auxiliary quantity $\gamma_{k-1} = \frac{1}{k} + \frac{k-1}{k} m_{k-1}$ is a convex combination of 1 and m_{k-1}, so $\gamma_{k-1} \in (m_{k-1},\, 1) \subset (0, 1)$ whenever $m_{k-1} < 1$. By Assumption 1, $\mathcal{F}_k \in (0, 1)$. Therefore $m_k = \max\{\mathcal{F}_k,\, \gamma_{k-1}\} \in (0, 1)$, completing the induction.

F Game-Theoretic Interpretation

Proposition 15 (Nash equilibrium)**.** *Under Assumption 2, define player k's utility as $\mathcal{V}_k(\mathbf{a}, \mathbf{b} \mid \{(\mathbf{a}_{k'}, \mathbf{b}_{k'})\}_{k'<k}) = -\frac{1}{2}\|\mathbf{Y} - \sum_{k'<k} \mathbf{b}_{k'}\mathbf{a}_{k'}^\top \mathbf{X} - \mathbf{b}\mathbf{a}^\top \mathbf{X}\|_F^2$. Then $\{(\mathbf{a}_k^\star, \mathbf{b}_k^\star)\}_{k=1}^r$ is the unique strict Nash equilibrium up to sign.*

Proof. By induction: given optimal predecessors, player k maximizes utility by finding the best rank-1 approximation of $\mathbf{Y}_k^\star$, which is unique under the strict spectral gap (Assumption 2). □

G Theory-Validation Experiments

We evaluate the predictions of Theorems 1 and 4 and Proposition 2 on the bilinear regression model $\mathbf{Y} = \mathbf{W}^\star\mathbf{X} + \mathcal{E}$ from (1). The synthetic instances use a ground-truth weight $\mathbf{W}^\star \in \mathbb{R}^{m\times d}$ of rank $r^\star$, an input matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$ with i.i.d. standard Gaussian entries, and Gaussian noise $\mathcal{E}_{ij} \sim \mathcal{N}(0, \varepsilon^2)$ entrywise, with $\varepsilon = 0$ in the noiseless experiments. Three singular-value profiles for $\mathbf{W}^\star$ are considered, each normalised so that $\sigma_1^\star = 1$: *exponential* $\sigma_k^\star = 2^{r^\star - k}$ (geometric decay, large gaps), *power-law* $\sigma_k^\star = k^{-1.5}$ (algebraic decay), and *uniform* $\sigma_k^\star = 1$ for all k (no spectral gap, so Assumption 2 is degenerate). Each setting is repeated over five random seeds; figures plot the per-round median across seeds with shaded bands giving the seed-to-seed minimum-to-maximum envelope.

G.1 Synthetic Validation

We first verify the qualitative predictions of Theorems 1 and 4 and Proposition 2: the per-round convergence of total reconstruction error, the self-correction of deflation targets, and the matching of the parallel and sequential noise floors.

For the convergence comparison of Figure 2 we set $m = 100$, $d = 200$, $n = 500$, $r^\star = 10$, and run each method with $T = 10$ inner iterations per round and $L = 10$ communication rounds across the three spectral profiles. Parallel ALS matches sequential ALS on the exponential spectrum, with both reaching a relative weight error of about 2.1×10^{-3}. On the power-law spectrum parallel reaches 3.6×10^{-3} against 2.1×10^{-3} for sequential, so the two stay within a small constant factor; on the uniform spectrum both methods degrade together because the spectral gap vanishes, consistent with Assumption 2. For reference we also include two factored gradient-descent variants (parallel and sequential, using the rank-1 solver of Section B in place of ALS) and a joint gradient-descent baseline that updates all $r^\star$ rank-1 factor pairs simultaneously without deflation.

To isolate self-correction (Figure 3) we shrink the problem to $m = 50$, $d = 80$, $n = 200$, $r^\star = 5$ on the exponential spectrum, with $T = 10$ and $L = 30$, so that workers $k = 2, \ldots, 5$ activate round by round under the staggered schedule $s_k = k$ of Theorem 1. The left panel shows the normalised deflation mismatch $\|\mathbf{Y}_{k,\ell} - \mathbf{Y}_k^\star\|_F \,/\, \|\mathbf{Y}_k^\star\|_F$ for each worker, where $\mathbf{Y}_{k,\ell}$ is the worker's current target and $\mathbf{Y}_k^\star = \mathbf{Y} - \sum_{k'<k} \mathbf{b}_{k'}^\star \mathbf{a}_{k'}^{\star\top} \mathbf{X}$ is the clean target built from the ideal rank-1 components $(\mathbf{a}_{k'}^\star, \mathbf{b}_{k'}^\star)$ of Section 2. The mismatch decreases by roughly $5\times$ for workers 2 to 4 between activation and convergence. The right panel decomposes the error of worker 3 into the three quantities $G_{3,\ell}$, $D_{3,\ell}$, and $B_{3,\ell}$ defined in (3), (4), and (5). Both $D_{3,\ell}$ and $B_{3,\ell}$ decay to a small common floor, exactly the behaviour predicted by Proposition 2.

The noise-robustness experiment of Figure 4 keeps the same problem dimensions and exponential profile, sets $T = 10$ and $L = 15$, and adds entrywise Gaussian noise $\mathcal{E}_{ij} \sim \mathcal{N}(0, \varepsilon^2)$ at $\varepsilon \in \{0.01, 0.1, 0.5, 1.0\}$. Across all four noise levels parallel and sequential deflation converge to the same noise floor, and the per-seed final-round parallel-to-sequential ratio stays within $[1.000, 1.014]$ across all four noise settings, so parallelization incurs no statistical penalty relative to sequential

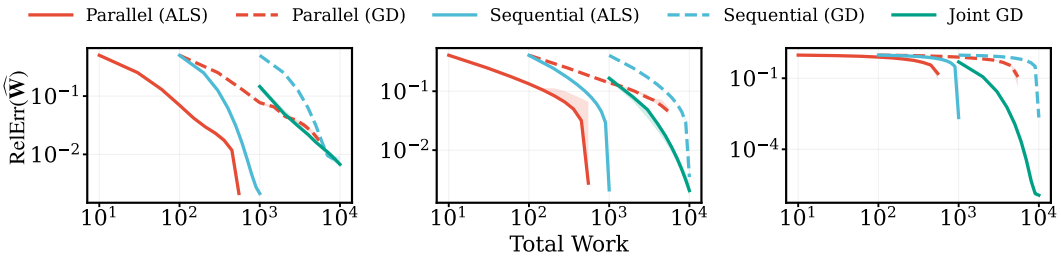


Figure 2: **Convergence under equal total work.** Relative weight error vs. total work for exponential, power-law, and uniform spectra. Parallel ALS matches sequential on the well-separated exponential spectrum and degrades gracefully as spectral gaps shrink.

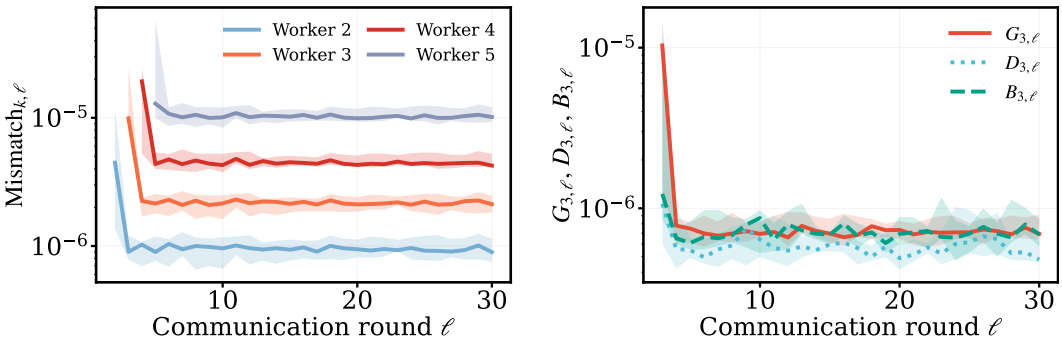


Figure 3: **Self-correction.** Left: normalized deflation mismatch per worker, decreasing over rounds. Right: error decomposition for worker 3 (G, D, B all decay).

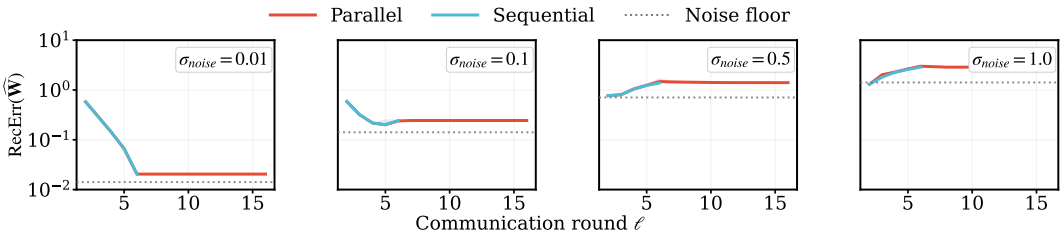


Figure 4: **Noise robustness.** Reconstruction error vs. rounds at four noise levels ε. Parallel and sequential converge to the same noise floor.

deflation, as predicted by Theorem 4. The reference floor in each panel is the theoretical scale $\varepsilon\sqrt{r^\star d/n}$ from the same theorem.

G.2 Theoretical-Bound Validation

This experiment tests the quantitative form of Theorem 1: each worker's total error $G_{k,\ell}$ (defined in (5)) decays geometrically once the warm-up round s_k from Theorem 1 is reached, and the empirical per-worker contraction rate matches the recurrence (8) for m_k. We fix $m = 50$, $d = 80$, $n = 200$, $r^\star = 5$ on the exponential spectrum and run Algorithm 1 with $T = 10$ ALS iterations per round and $L = 40$ communication rounds. For every seed and every worker k we record the empirical trajectory $G_{k,\ell}$ and fit $G_{k,\ell} \approx \widehat{C}_k\, \widehat{m}_k^{\,\ell}$ by least-squares regression on $\log G_{k,\ell}$, restricted to rounds at which $G_{k,\ell}$ has fallen below half of its value at worker activation. Here m_k denotes the per-worker contraction factor of the recurrence (8), and $\mathcal{F}_k$ denotes the per-component contraction factor of the rank-1 subroutine from Assumption 1. The empirical warm-up $\widehat{s}_k$ is taken as the first round after which $G_{k,\ell}$ decays monotonically over the next five rounds, and $\widehat{m}_k$ is the fitted slope; we use $\widehat{s}_k$ and $\widehat{m}_k$ as the empirical estimates of the theoretical s_k and m_k.

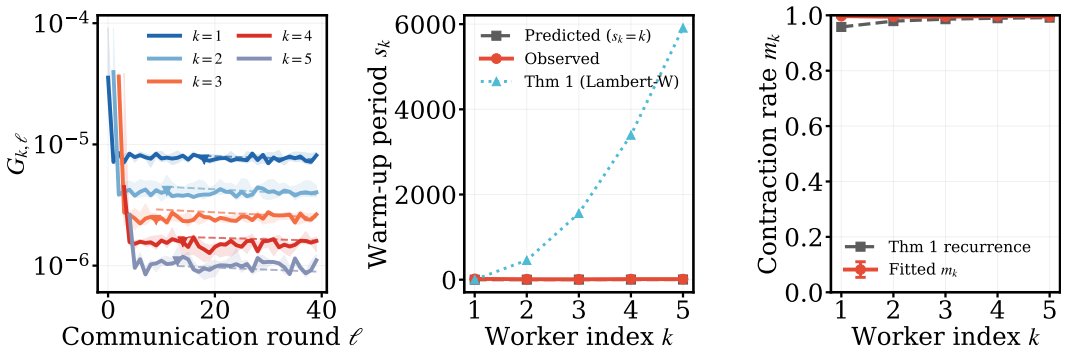


Figure 5: **Theoretical-bound validation for Theorem 1.** (A) Empirical $G_{k,\ell}$ per worker (solid) with fitted exponential decay (dashed) and observed warm-up marker. (B) Predicted ($s_k = k$) vs. observed warm-up period, with the Lambert-W bound from (32) shown for reference. (C) Fitted contraction rate $\widehat{m}_k$ vs. the recurrence (8).

Figure 5 compares these estimates to the theoretical predictions. Panel (A) overlays the fitted exponentials on the empirical $G_{k,\ell}$ curves and shows that every worker enters geometric decay after a brief warm-up. Panel (B) plots $\widehat{s}_k$ together with the staggered-activation prediction $s_k = k$ and the Lambert-W upper bound from (32): the observed warm-up tracks $s_k = k$, while the Lambert-W bound is conservative on this well-separated spectrum, as expected of a worst-case bound (for visual clarity, the displayed curve uses the looser global-m_k form of (32), while the proof states a tighter per-predecessor maximum over $k' \le k$; both are valid upper bounds on s_k). Panel (C) places the fitted $\widehat{m}_k$ next to the recurrence $m_k = \max\{\mathcal{F}_k, 1/k + (k-1)m_{k-1}/k\}$ from (8), and the two agree within seed-to-seed noise. Together the three panels confirm that the convergence rate predicted by Theorem 1 is attained empirically and that the fitted contraction rate $\widehat{m}_k$ matches the recurrence (8) for m_k within seed-to-seed variability.

G.3 Spectral-Gap Sensitivity

The third experiment probes Assumption 2 directly by sweeping the spectral gap and tracking how convergence degrades as it closes. We use a linear profile $\sigma_k^\star = 1 - (k-1)g$ for $k = 1, \dots, r^\star$, clamped at a floor of 0.01 so the problem stays well-posed when g is small. Here g is the spectral-gap step parameter for this experiment, distinct from the growth interval Δ used in the main text (Table 10). The remaining settings match Section G.2: $m = 50$, $d = 80$, $n = 200$, $r^\star = 5$, $T = 10$, $L = 30$. On this profile the per-worker gap from Assumption 2, $\mathcal{T}_k^\star = \min\big(\min_{j>k} |\sigma_k^\star - \sigma_j^\star|, \sigma_k^\star\big)$, equals $\min(g, \sigma_k^\star)$ for $k < r^\star$, so over the swept range the ratio $g/\sigma_1^\star$ directly controls $\min_k \mathcal{T}_k^\star/\sigma_1^\star$. We sweep $g/\sigma_1^\star \in \{0.01, 0.05, 0.1, 0.25, 0.5\}$.

Figure 6 reports the outcome. Panel (a) shows that all five gap settings converge and that the per-round rate degrades smoothly as the gap shrinks: at $g/\sigma_1^\star = 0.5$ the reconstruction error reaches its floor within roughly five communication rounds ℓ, while at $g/\sigma_1^\star = 0.01$ it takes about twenty-five rounds. Panel (b) plots the across-seed median of the final-round error against the gap ratio on a log scale; even at $g/\sigma_1^\star = 0.01$ the error drops below 10^{-4}, so the algorithm degrades gracefully rather than failing as the gap closes. A positive spectral gap is therefore sufficient for convergence on this profile, and shrinking the gap inflates the warm-up period s_k (Theorem 1) smoothly rather than triggering a sharp phase transition, consistent with the role $\mathcal{T}_k^\star$ plays in Assumption 2.

H Additional Experimental Details

H.1 GLUE Training Details

Model and tokenizer. All GLUE experiments use DeBERTaV3-base [He et al., 2023] (`microsoft/deberta-v3-base`), a 12-layer, 768-hidden, 12-head transformer with ∼184M parameters. We freeze all pretrained weights and train only the rank-1 adapter parameters. Inputs are tokenized with the DeBERTaV3 SentencePiece tokenizer at maximum sequence length 128.

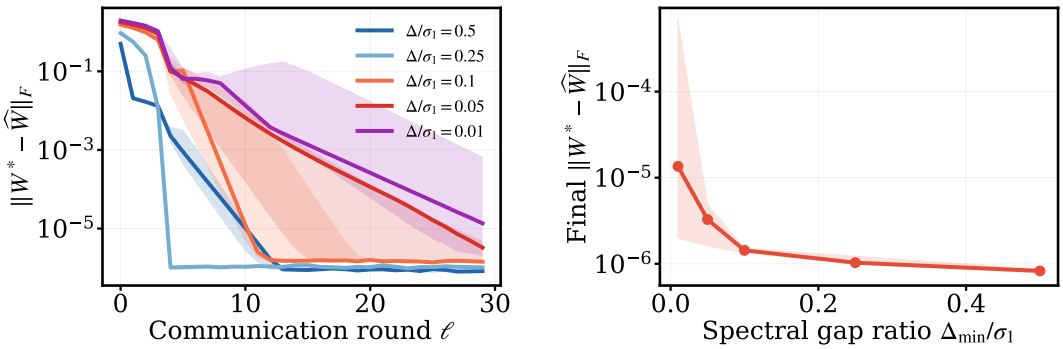


Figure 6: **Spectral-gap sensitivity (validates Assumption 2).** (a) Convergence curves for five gap ratios $g/\sigma_1^\star$; smaller gaps require more rounds but still converge. (b) Final reconstruction error vs. gap ratio decays to a floor below 10^{-4} even at $g/\sigma_1^\star = 0.01$.

Adapter configuration. We attach parallel rank-1 adapters to all 72 encoder linear layers (six per layer $\times$ 12 layers): W_q, W_k, W_v, W_o (attention) and W_{f1}, W_{f2} (FFN). Each adapter has $r_{\max} = 4$ components and both $\mathbf{a}_k$ and $\mathbf{b}_k$ vectors are initialized from $\mathcal{N}(0, 0.02)$. We use the standard LoRA scaling α [Hu et al., 2022], with the per-task values reported in Table 10.

Per-task hyperparameters. Table 10 lists the hyperparameters used for each task on the GLUE benchmark [Wang et al., 2019]; LoRA's scaling convention follows Hu et al. [2022].

Table 10: Per-task hyperparameters for AdaPaD on GLUE (seed=43).

	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI
Epochs	10	50	30	25	24	5	7	9
Batch size	32	32	32	32	64	64	128	256
Learning rate	8×10^{-4}	1.2×10^{-3}	1×10^{-3}	1.97×10^{-3}	1×10^{-3}	1×10^{-3}	7×10^{-4}	1.5×10^{-3}
α	32	24	24	32	8	32	48	24
Weight decay	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Warmup steps	100	100	100	100	1000	500	500	500
Grad clip	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Top-h	3	3	6	1	5	3	5	2
Δ (growth interval)	100	200	100	100	100	50	300	100
Orth reg coef λ_{orth}	0.3	0.3	0	0.5	0.2	0.2	0.15	0.1
Adapter dropout	0.0	0.0	0.3	0.0	0.1	0.0	0.1	0.2
Cls dropout	0.0	0.0	0.0	0.2	0.0	0.0	0.15	0.15
Adapter weight decay	0.0	0.01	0.01	0.0	0.0	0.0	0.0	0.0
LR schedule	linear	linear	linear	linear	linear	linear	linear	cosine
Rank budget B	144	144	144	144	144	144	144	144

Orthogonal regularization. We optionally include the orthogonal regularization term from IncreLoRA [Zhang et al., 2023a], $\lambda_{\text{orth}}\big(\|\mathbf{A}^\top\mathbf{A} - \mathbf{I}\|_F + \|\mathbf{B}\mathbf{B}^\top - \mathbf{I}\|_F\big)/(2n_{\text{layers}})$, which pushes each module's component vectors toward an orthonormal frame. We adopt it on GLUE with the per-task values in Table 10 and disable it on SQuAD.

Baseline reproduction. We reproduce IncreLoRA as the primary comparison baseline because it is the closest method to AdaPaD: both grow rank incrementally from rank-1 components via importance-based allocation with EMA scoring. We use the authors' official loralib (SVDLinear + RankAllocator) and their published hyperparameters, ported to our software environment (Python 3.11, PyTorch 2.6, Transformers 5.2, CUDA 12.9). For the remaining baselines (AdaLoRA, SoRA, dEBORA) we cite published results from their respective papers [Zhang et al., 2023b, Ding et al., 2023, Zangrando et al., 2025].

H.2 SQuAD/SQuAD v2 Training Details

Model and tokenizer. We fine-tune Qwen/Qwen3-0.6B [Yang et al., 2025] with the standard `AutoModelForQuestionAnswering` span-prediction head. The backbone has 28 transformer blocks with hidden size 1024, FFN dim 3072, and grouped-query attention with 16 query heads and 8 KV heads. Inputs are tokenized at sequence length 384 with doc-stride 128, following the HuggingFace canonical preprocessing for both SQuAD v1.1 [Rajpurkar et al., 2016] and SQuAD v2 [Rajpurkar et al., 2018].

Adapter configuration. Both AdaPaD and the fixed-rank LoRA [Hu et al., 2022] baseline adapt all seven projection matrices in every transformer block: W_q, W_k, W_v, W_o, W_{gate}, W_{up}, W_{down}, for a total of $7 \times 28{=}196$ adapted modules. AdaPaD's rank-1 components initialise both $\mathbf{a}_k$ and $\mathbf{b}_k$ vectors from $\mathcal{N}(0, 0.02)$. We set $r_{\max}{=}8$ and allocate the remaining budget by EMA importance scoring with the top-10 modules per growth round.

Per-task hyperparameters. Table 11 lists the single AdaPaD recipe used across all four (task, $\bar{r}$) cells. Only the rank budget changes between cells, driven by the average-rank target; all other hyperparameters are shared.

Table 11: AdaPaD hyperparameters on SQuAD/SQuAD v2 (seed=43). A single recipe is used for all four cells; only the rank budget B changes between $\bar{r}{=}4$ and $\bar{r}{=}6$.

	SQuAD v1.1	SQuAD v2
Epochs	3	3
Batch size	16	16
Max sequence length	384	384
Doc-stride	128	128
Total optimiser steps	16,650	24,777
Learning rate	1×10^{-3}	1×10^{-3}
α	16	16
Weight decay	0.0	0.0
Warmup steps	1,000	1,000
Grad clip	1.0	1.0
Top-h	10	10
Δ (batches per growth round)	400	400
Orth reg coef	0.0	0.0
Adapter dropout	0.0	0.0
LR schedule	linear	linear
$r_{\max}$	8	8
Rank budget B ($\bar{r}{=}4$ / $\bar{r}{=}6$)	784 / 1,176	784 / 1,176
Optimiser	AdamW (bf16)	AdamW (bf16)

H.3 Seeds and Reporting

GLUE results in Table 3 and Table 12 are the mean and population standard deviation over seeds 43, 44, 45, 46, and 47. AdaPaD, IncreLoRA, VeRA, and VB-LoRA use the same five seeds. The LoRA ($r{=}2$) row of Table 3 is a single run with seed 43. AdaLoRA, SoRA, and dEBORA are cited from their papers. All other experiments use seed 43. These are the ablation (Table 4), the rank map (Figure 1), SQuAD/SQuAD v2 (Tables 5 and 6), math reasoning (Table 7), multi-GPU scaling (Table 8), and the cost measurements in Section H.5.

H.4 Parameter-Sharing Baselines on GLUE

Table 12 compares AdaPaD with VeRA [Kopiczko et al., 2024] and VB-LoRA [Li et al., 2024] on the five small- and medium-scale GLUE tasks. Both baselines follow the hyperparameters and parameter budgets of their papers. AdaPaD is at or above both methods on four tasks and within 0.02 of VB-LoRA on STS-B.

Table 12: ADAPAD against parameter-sharing methods on GLUE with DeBERTaV3-base (mean $\pm$ std over seeds 43–47). Metrics as in Table 3.

Method	#Params	CoLA	MRPC	STS-B	RTE	SST-2
VeRA	0.10M	69.95±0.65	89.56±0.45	90.27±0.21	83.61±0.84	95.34±0.21
VB-LoRA	0.29M	70.02±0.87	90.05±0.70	**91.85**±0.12	87.58±0.93	95.73±0.47
ADAPAD	0.34M	**70.98**±0.75	**90.78**±0.33	91.83±0.14	**87.73**±0.69	**96.08**±0.27

H.5 Training Cost

We measure training cost on MAWPS with Qwen3-0.6B on one GPU (seed 43). Table 13 pins every module at rank r and times one training step. A LoRA step is one forward and backward pass of the frozen backbone. An ADAPAD step runs one such pass per active component, so its time grows about linearly with r while the time per component stays near $1.3\times$ a LoRA step. Peak memory is flat in r for both methods.

Table 13: Per-step time and peak memory on one GPU (MAWPS, Qwen3-0.6B, rank pinned at r in every module).

r	ADAPAD ms/step	LoRA ms/step	ratio	ADAPAD ms/component	ADAPAD peak MB	LoRA peak MB
1	214	160	1.3×	214	12095	10746
2	476	161	3.0×	238	12111	10755
3	750	163	4.6×	250	12127	10764
4	1040	167	6.2×	260	12143	10774
8	2170	193	11.3×	271	12207	10810

Table 14 tracks training loss against step and wall-clock time for LoRA (r=2) and ADAPAD at average rank 2 with $r_{\max}$=4. Both reach the same training-loss floor. ADAPAD stays near the frozen-backbone loss during its warm-up and rank growth, then drops to LoRA's level, and the full run takes $3.3\times$ the wall-clock time (1091 s against 331 s for 800 steps). Inference cost is the same for both, since the components merge into one low-rank update.

Table 14: Training loss and elapsed time on MAWPS (one GPU, matched average rank 2).

	LoRA (r=2)		ADAPAD (avg. rank 2)	
Step	loss	elapsed (s)	loss	elapsed (s)
40	0.028	33	3.66	53
80	0.003	61	3.46	86
120	0.0006	77	0.020	122
200	0.0001	109	0.035	188
400	0.0001	203	0.0007	470
800	0.0001	331	0.0001	1091

H.6 Hardware and Software

Unless otherwise stated, all experiments run on $4\times$ NVIDIA H200 80 GB GPUs connected by NVLink on a single node. Software: Python 3.11, PyTorch 2.6, Transformers 5.2, CUDA 12.9.

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 and Section 1 state our three contributions: the AdaPaD algorithm with self-correction, theoretical guarantees for the bilinear regression case, and an empirical evaluation on GLUE, SQuAD, and math reasoning. Each claim is matched to a specific theorem (Theorems 1, 3 and 4 and Proposition 2) or to a results table in 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: Section 6 has a dedicated Limitations paragraph that calls out the bilinear-regression scope of the convergence theory, the spectral-gap dependence, the synchronous-communication requirement, the per-round communication cost at very large rank, and the single-backbone-scale empirical setup.

   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: The three assumptions (Assumptions 1 to 3) are stated in Section 2.1 with the normalization $\sigma_1^\star = 1$ noted explicitly. Full proofs of Theorems 1, 3 and 4 and Proposition 2 together with the supporting lemmas (target mismatch, Wedin-based B-bound, surrogate domination, $\hat{B}$ re-anchoring, and the Lambert-W bound) live in Section E.

   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 H lists per-task hyperparameters (learning rate, batch size, epochs, warmup, dropout, growth interval, rank budget), the seeds, the rank-1 subroutine choice, the IncreLoRA reproduction protocol, and the software stack (Python 3.11, PyTorch 2.6, Transformers 5.2, CUDA 12.9). Section H.2 does the same for the SQuAD experiments.

   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: [Yes]

   Justification: An anonymized code repository with the AdaPaD implementation and run scripts for the GLUE and SQuAD experiments is available at https://anonymous.4open.science/r/ParallelLinearRegression-C604/README.md (also linked from the abstract). The benchmarks (GLUE, SQuAD v1.1, SQuAD v2) and pretrained backbones (DeBERTaV3-base, Qwen3-0.6B) are publicly available; the per-task recipes in Sections H and H.2 match the configurations in the repository.

   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 5 states the matched parameter budgets, module selection, baseline list, and evaluation protocol. The full per-task hyperparameter table for GLUE is in Section H and the SQuAD recipe is in Section H.2.

   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: [Yes]

Justification: GLUE results (Tables 3 and 12) are mean $\pm$ std over five seeds (43–47). The multi-GPU scaling experiment (Table 8) and the synthetic theory-validation experiments (Section G) report mean $\pm$ std over independent trials. SQuAD, math reasoning, the ablation, and the cost measurements use seed 43 (Section H.3).

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: All experiments run on a single node with $4\times$ NVIDIA H200 80 GB GPUs over NVLink, stated in Section 5 and again in Section H; Table 8 reports per-batch wall-clock times (mean $\pm$ std over five global rounds) for the multi-GPU scaling experiment, and the per-task training schedules (epochs, batch size, total optimiser steps) are listed in Tables 10 and 11.

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: We have read the NeurIPS Code of Ethics. The work uses publicly available benchmark datasets and pretrained checkpoints, does not involve human subjects, and does not raise privacy or fairness concerns specific to this paper.

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: [Yes]

Justification: AdaPaD reduces the parameter count and wall-clock cost of LoRA fine-tuning, which lowers the bar for adapting large pretrained models. On the positive side this helps smaller research groups and on-device deployment; on the negative side cheaper fine-tuning lowers the cost of adversarial or misuse-oriented adaptation. We do not introduce a failure mode beyond those already present in standard PEFT methods.

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: We release no models, datasets, or scraped content with this submission, only the algorithmic recipe. The contribution does not introduce a new failure mode requiring safeguards beyond those associated with standard fine-tuning of publicly available checkpoints.

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: [No]

    Justification: We cite GLUE, SQuAD v1.1, SQuAD v2, the DeBERTaV3 paper, and the Qwen3 technical report at the points where each asset is used, and we credit baseline authors (IncreLoRA, AdaLoRA, SoRA, dEBORA) where their numbers or code are used. We do not enumerate the individual licenses for these assets in the submission text.

    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, models, or other assets are released as part of 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: The work uses pre-existing benchmark datasets only. There is no crowdsourcing, no new data collection, and no human-subjects component.

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: No human-subjects research is conducted in this work, so no IRB review was required.

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: [N/A]

Justification: Pretrained LLMs (DeBERTaV3-base, Qwen3-0.6B) appear in this paper as fine-tuning targets in the empirical evaluation, not as components of the algorithmic contribution. The methodology itself (parallel rank-1 deflation with self-correction, advance learning, dynamic rank discovery) does not use LLMs as an important, original, or non-standard component.

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.