IEEE Information Theory Workshop 2026

Pipelined Gradient Coding

Xian Su · Florida International University

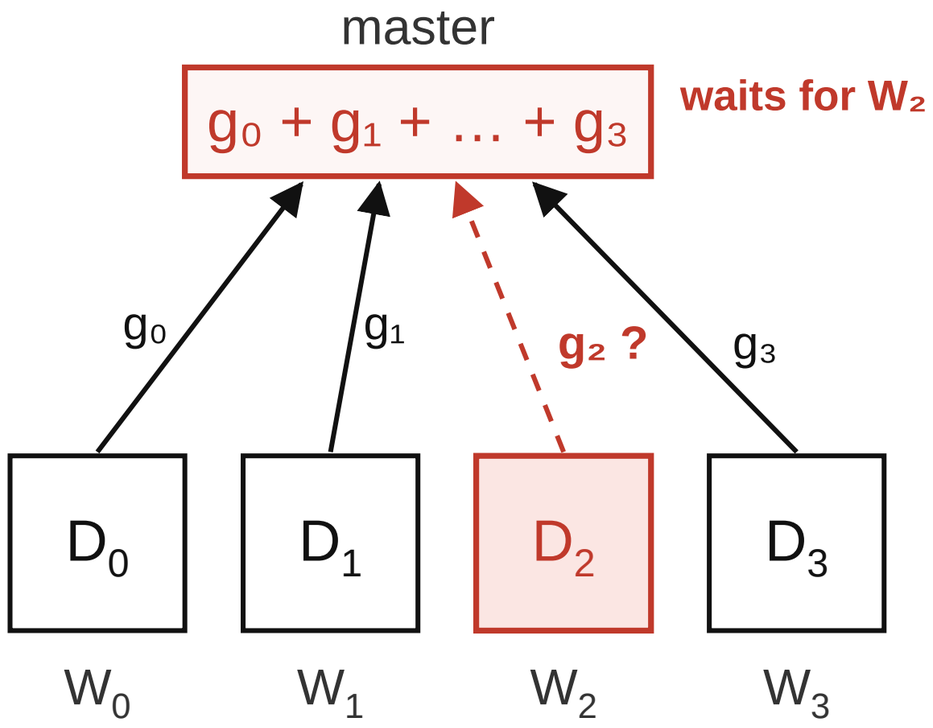
Jun Li · Queens College & The Graduate Center, CUNY **SPEAKER**

Extended version with full proofs

arxiv.org/abs/2607.20739

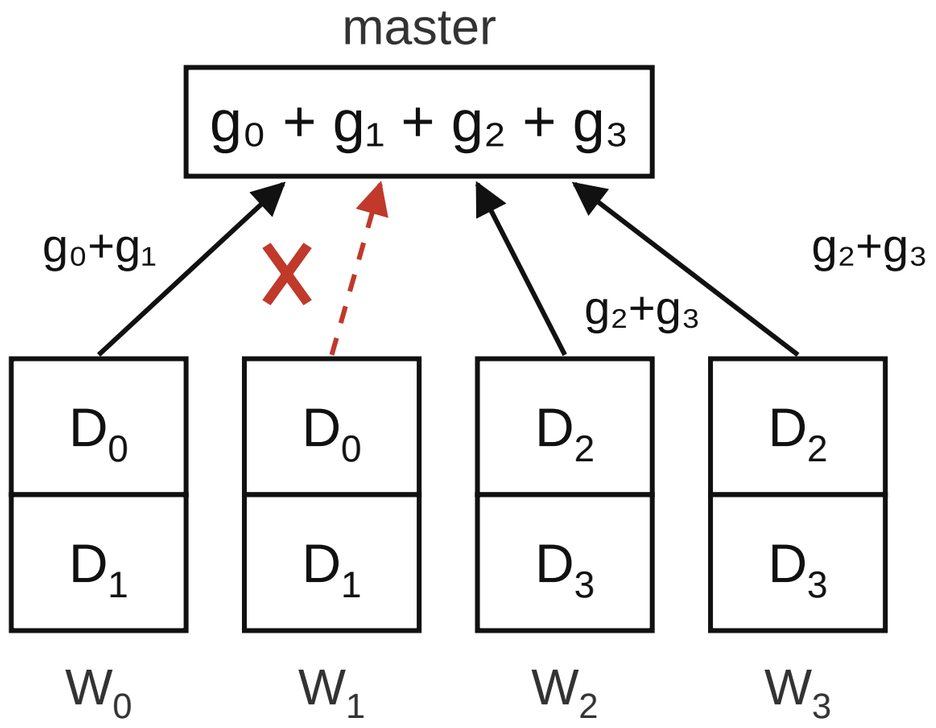
Distributed gradient descent, stragglers, and gradient coding

`gradient coding: Tandon, Lei, Dimakis, Karampatziakis — ICML 2017`



(a) distributed gradient descent

one straggler stalls the whole step,
and next step it is a different worker

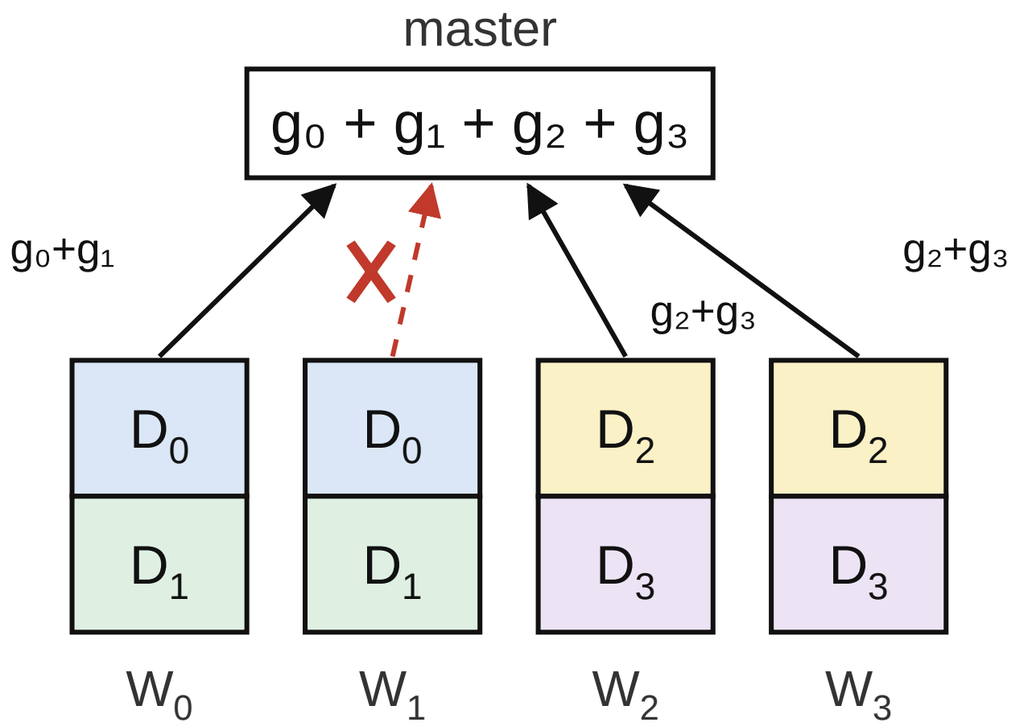


(b) gradient coding

c = s+1 partitions per worker, one coded upload each;
the full sum is recovered from any n−s workers

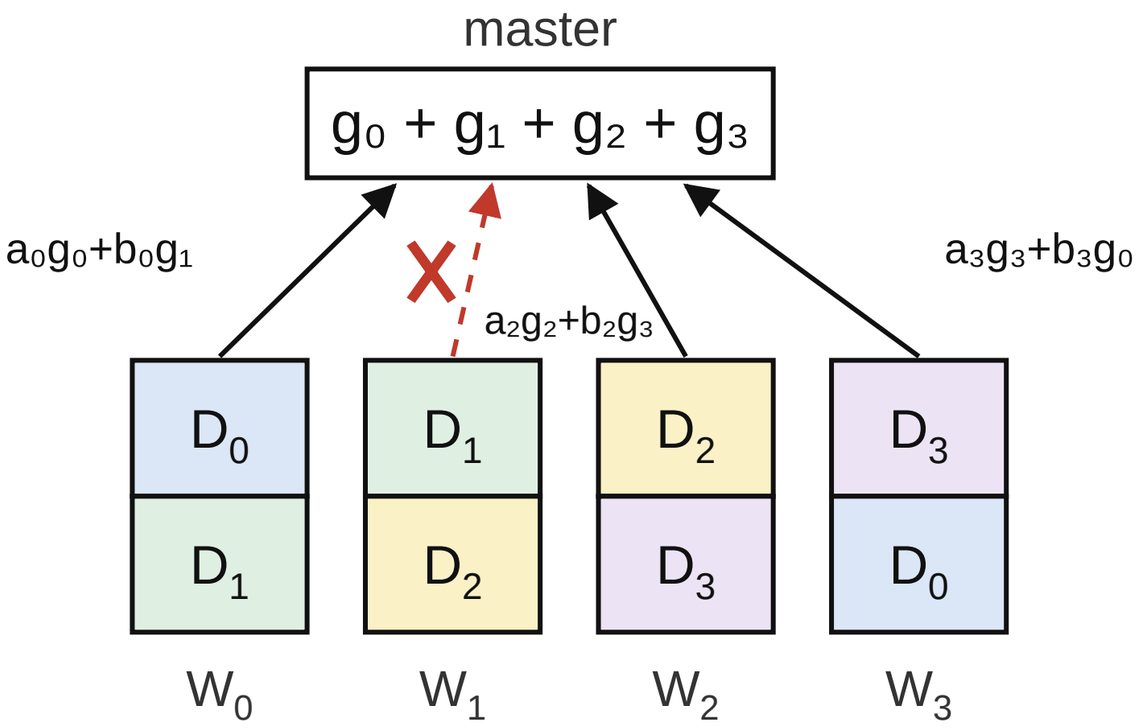
Two placements: fractional repetition and cyclic repetition

`any three of the four coded gradients recover the full sum (n = 4, c = 2, s = 1)`



(a) fractional repetition (FR)

groups of c workers share the same partitions
plain-sum uploads; decode: one survivor per group
every coefficient is exactly 1



(b) cyclic repetition (CR)

W_i holds D_i, …, D_{i+c-1} (mod n); weighted uploads
decode: solve a linear system per straggler pattern
coefficients depend on who straggled

But straggler tolerance is bought with c-fold computation

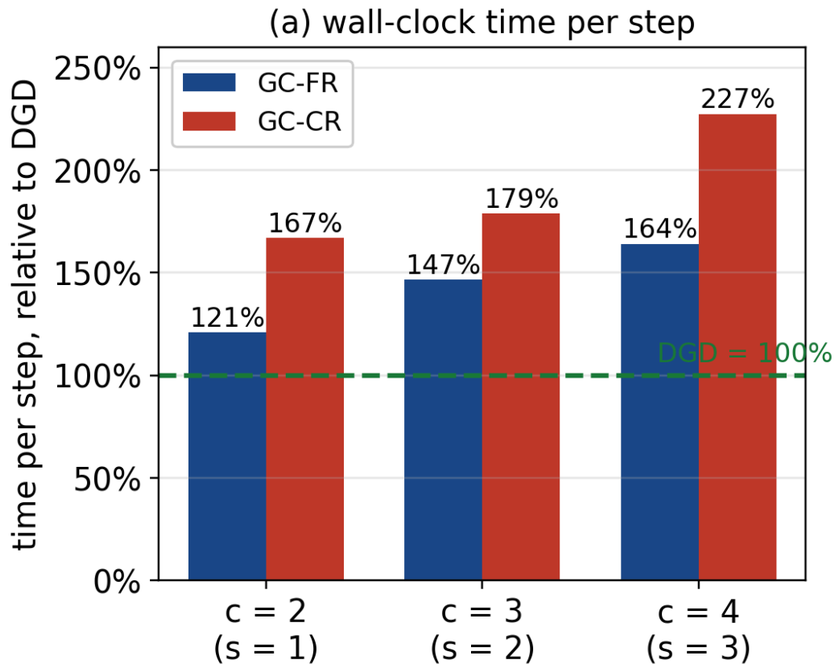
`Google Cloud · n = 12 workers · ResNet-18 / CIFAR-10 · natural stragglers`

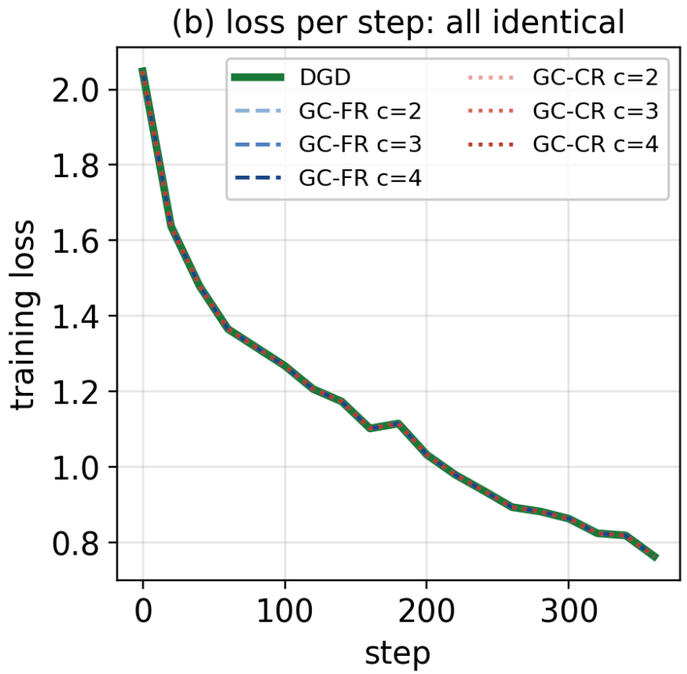
THE COST

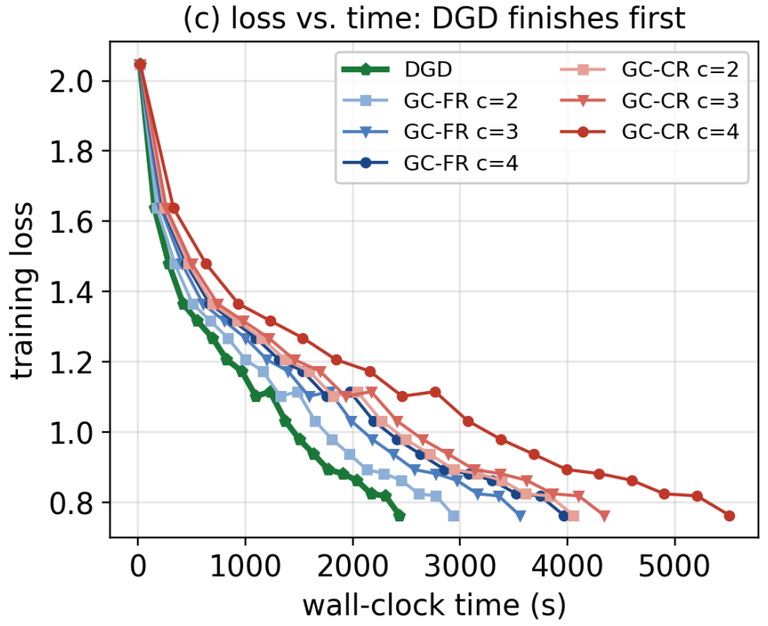
Every worker evaluates $c = s + 1$ partitions *in every step*, so per-step computation is **× c**. Measured: **1.2× to 2.3×** DGD's time per step (a), while the loss per step is **identical** (b). CR is slower than FR: weighted encoding plus a per-pattern decode, on top of the same c gradients.

THE PARADOX

GC only wins if the straggler delay it removes exceeds the $(c-1)\times$ extra work it adds. It does not always: in our cloud runs, every GC configuration is slower than plain DGD to reach the same loss, up to **2.2×** (c).

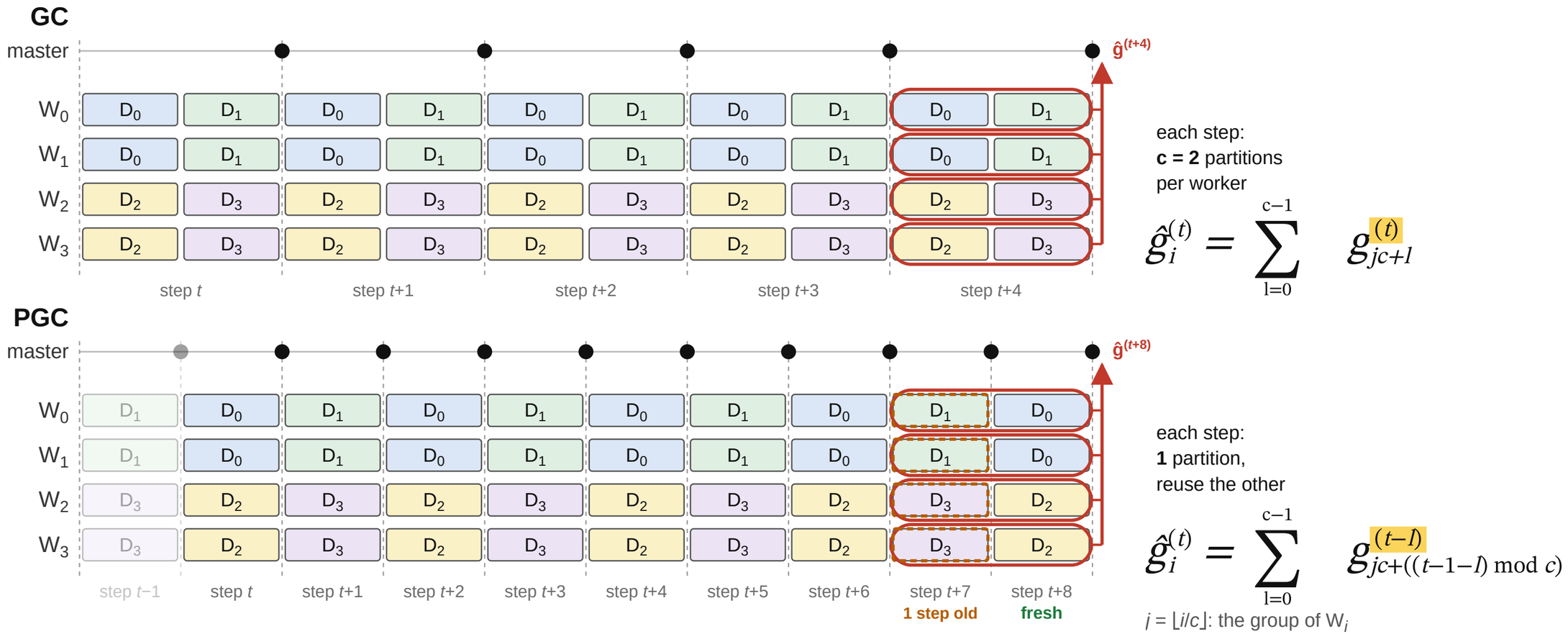






Pipelined gradient coding: one partition per step, reuse the rest

FR placement from before (n = 4, c = 2): W_0, W_1 hold D_0, D_1 and W_2, W_3 hold D_2, D_3



same wall-clock axis, a window in mid-training: PGC delivers a coded gradient every partition-time

COMPUTE

One partition per step — **identical to DGD**.

COMMUNICATE

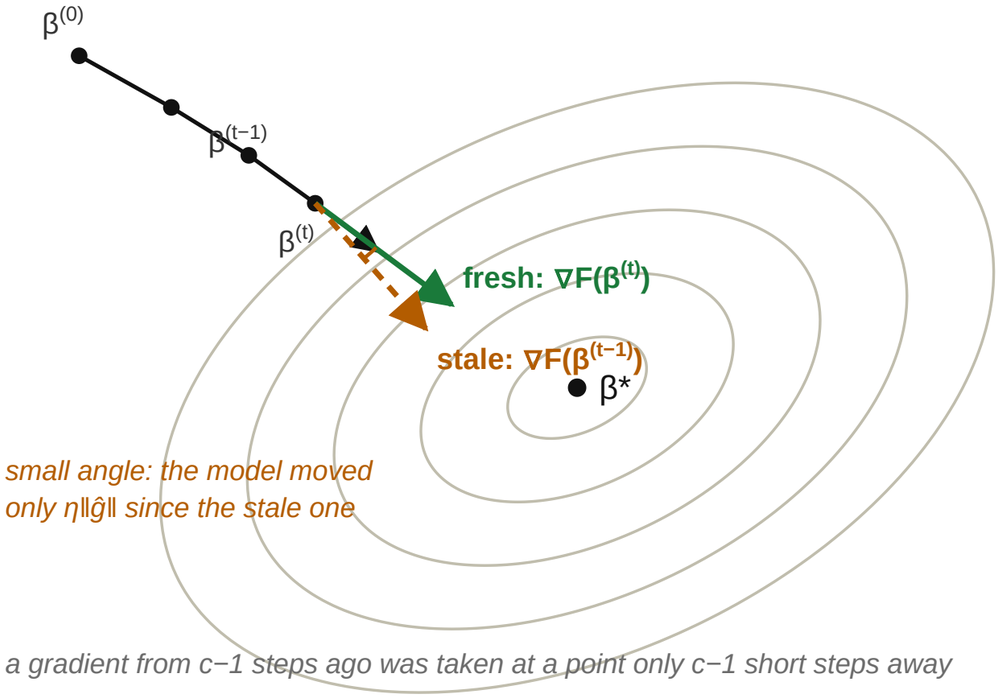
One coded gradient per step; decode from the $n - s$ fastest — **identical to GC**.

PAY

$c - 1$ components are **stale**, by at most $c - 1$ steps — bounded by construction.

Stale gradients still converge, at the same rate up to a factor

Theorem 1 · λ-strongly convex, L-smooth loss, gradient noise ξ, staleness γ



WHAT THE THEOREM SAYS

Staleness $\gamma < 1$, for a stale gradient from any step $s \geq t - c + 1$:

$$\mathbb{E}\|\nabla \mathrm{F}(\beta^{(t)}) - \nabla \mathrm{F}(\beta^{(s)})\|^2 \leq \gamma\, \mathbb{E}\|\nabla \mathrm{F}(\beta^{(t)})\|^2$$

Then, with a small enough learning rate, the gap to the optimum shrinks geometrically:

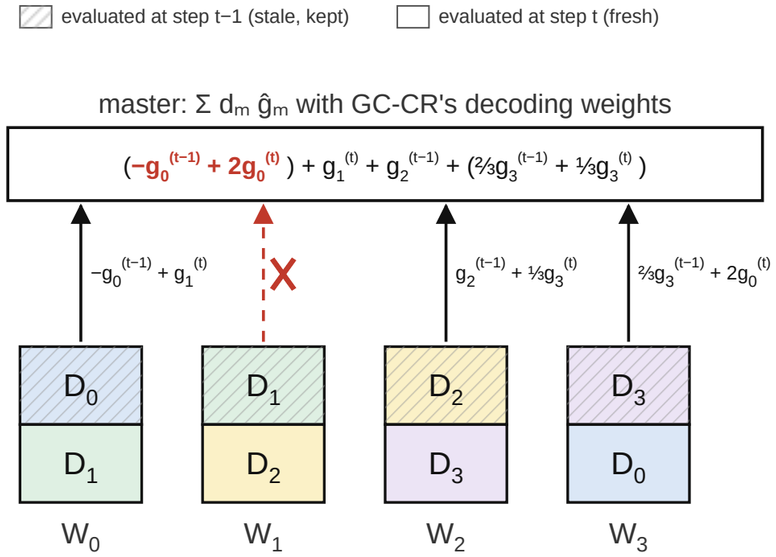
$$\mathbb{E}[\mathrm{F}(\beta^{(t)})] - \mathrm{F}(\beta^{*}) \leq \rho^{t}\,(\mathrm{F}(\beta^{(0)}) - \mathrm{F}(\beta^{*}))$$

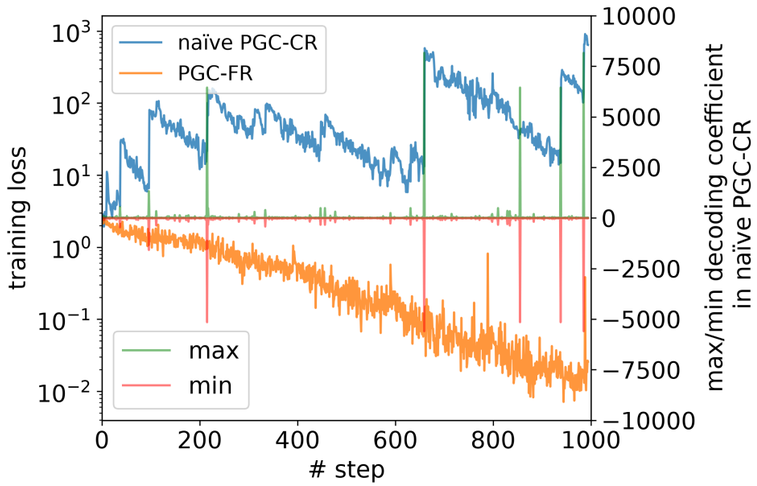
$$\rho = 1 - (1-\gamma) \cdot \frac{\lambda}{nL(n+\xi)}$$

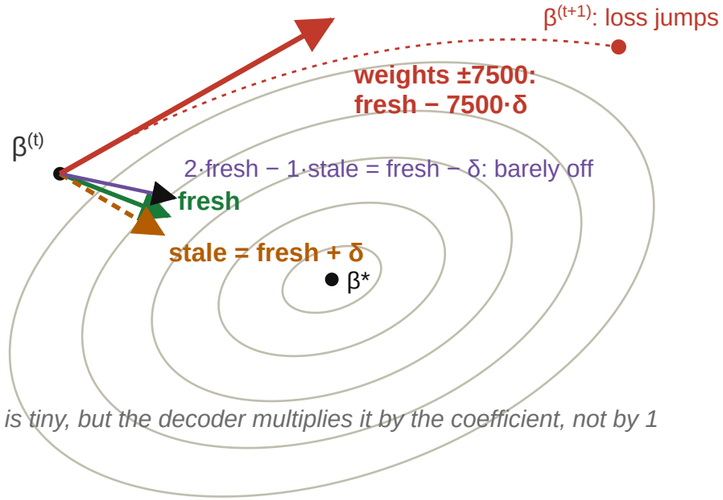
set $\gamma = 0$: the fresh-gradient rate

Do the same for cyclic repetition, and it diverges

naïve PGC-CR: keep GC-CR's coefficients, let the kept components go stale (n = 4, c = 2)







LOOK AT D_0

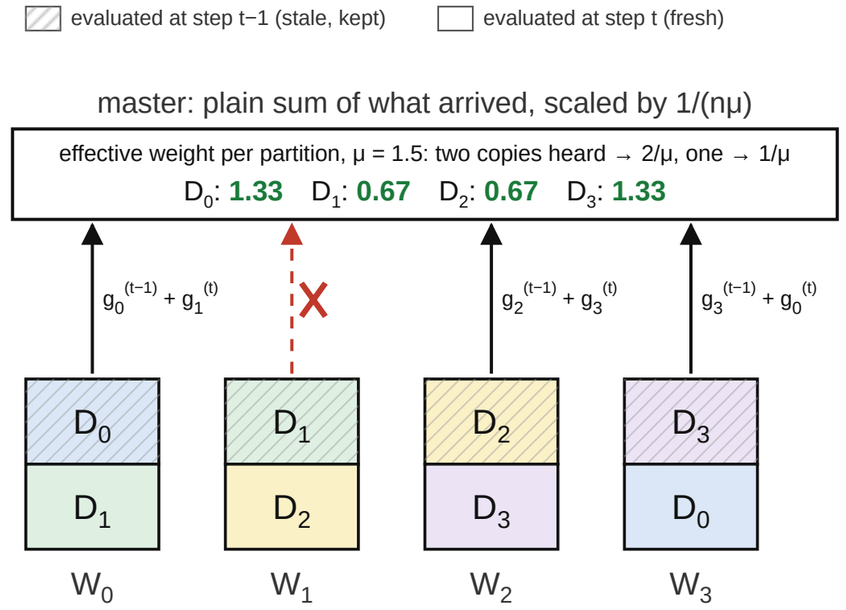
recovered as **−1 × stale + 2 × fresh**. The weights sum to 1, so with fresh copies it is exact. But the stale copy enters with weight **−1**: its error is magnified, not averaged out.

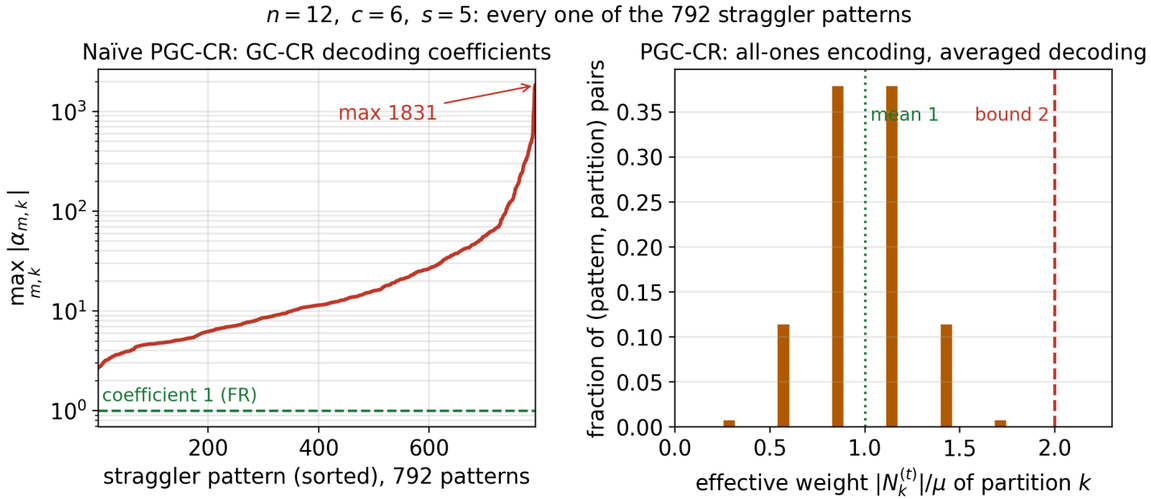
AT SCALE: n = 12, c = 6

Left axis, loss: PGC-FR (orange) converges, naïve PGC-CR (blue) never gets below 10. Right axis, the decoding coefficients: they spike to **±7500**, and every spike coincides with a jump in the loss.

PGC-CR: give up exact recovery, keep every weight small

all-ones encoding · decode by a scaled sum of what arrived · same example (n = 4, c = 2)





NOT EXACT, BUT NEVER WILD

The weight on each partition D_k is no longer exactly 1, only **1 in expectation**:

GC-CR: $\sum_m \alpha_{m,k} = 1$ for *every* straggler pattern (exact)

PGC-CR: $\mathbb{E}[|N_k|/\mu] = 1$ *averaged* over the straggler pattern

$|N_k|$ = times D_k is heard, $\mu = (n-s)c/n$. And **below 2** ($s < n/2$): $|N_k|/\mu \le \min\{c, n-s\}/\mu < 2$, so staleness error is amplified by less than 2, instead of 7500.

EVERY STRAGGLER PATTERN, N = 12, C = 6

Left: GC-CR's decoding coefficients, one random instance of the construction: median 11, up to 1831. Right: PGC-CR's weights stay within **0.29–1.71**, mean 1. Both encoder and decoder got *simpler*.

PGC-CR converges too, at the same geometric rate

Theorem 2 · same λ, L, ξ as Theorem 1 · γ now on the averaged copies

WHAT THE THEOREM SAYS

Staleness $\gamma < 1$, now on the average of all nc copies (c per partition) used at step t, each computed at its *own* earlier step $s(t,i,j) \in [t-c+1,\ t]$:

$$\overline{V}^{(t)} = \frac{1}{nc}\sum_{i,j} \nabla F(\beta^{(s(t,i,j))})$$

$$\mathbb{E}\|\nabla F(\beta^{(t)}) - \overline{V}^{(t)}\|^2 \ \le\ \gamma\, \mathbb{E}\|\nabla F(\beta^{(t)})\|^2$$

Same geometric shape as Theorem 1, staleness again only a (1−γ) factor:

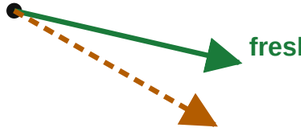
$$FR:\ \rho = 1 - (1-\gamma)\cdot \frac{\lambda}{nL(n+\xi)} \qquad \gamma \text{ per copy, as in Theorem 1}$$

$$CR:\ \rho = 1 - (1-\gamma)\cdot \frac{\lambda}{L(nc+\xi)}$$

INTUITION: MIXING STALENESSES

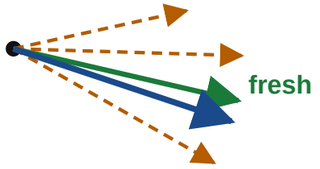
In FR a partition's gradient comes from *one* worker with *one* staleness. In CR it arrives from c workers with *different* delays and is averaged, so we expect the staleness errors to partly cancel.

FR: one copy of D_k, one staleness



the one stale copy: error δ

CR: c copies of D_k, c stalenesses



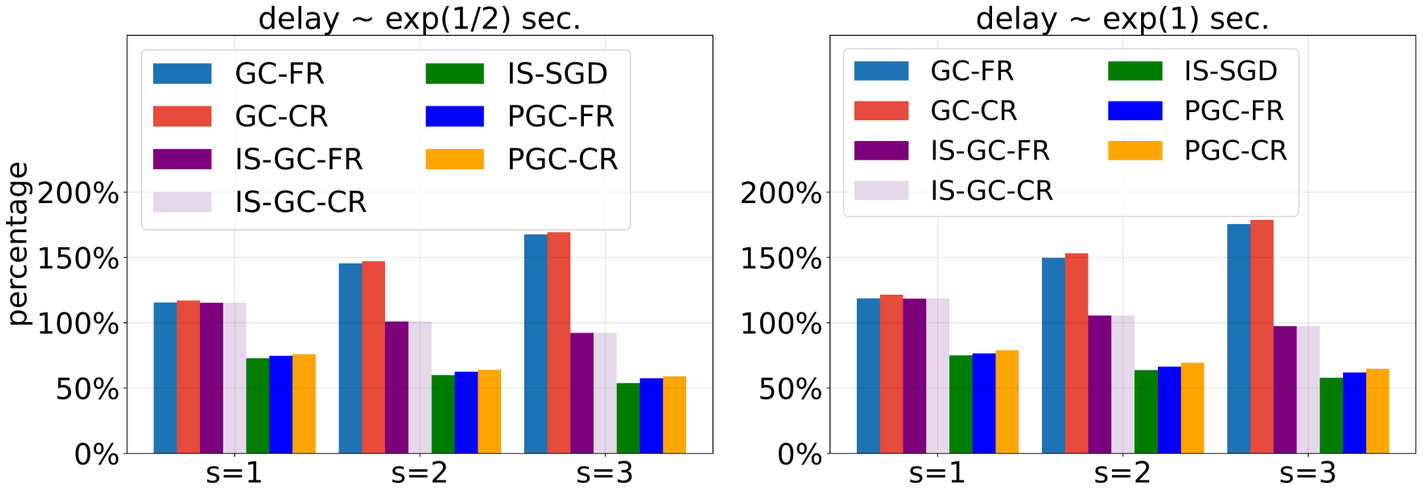
their average: errors partly cancel

same partition, gradients from several workers at several recent iterates: averaging copies with different delays: errors partly cancel (intuition)

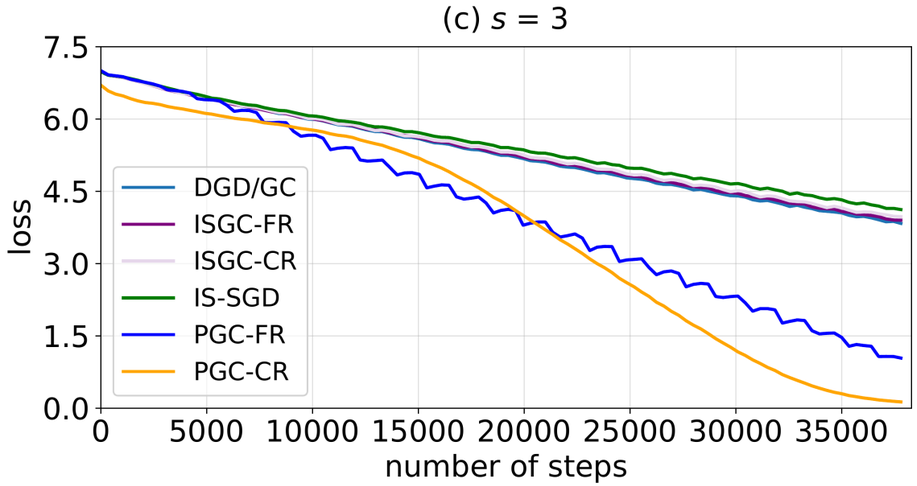
Theorem 2 has an improved statement and proof in the updated arXiv version: arxiv.org/abs/2607.20739

Simulation: DGD's step time, and faster convergence per step

`Ray + PyTorch · n = 12 · ResNet-18 / ImageNet · exp. delays · 25 epochs · 5 runs`



Time per step, DGD = 100%



Loss vs. steps, $s = 3$ (38,150 steps)

TIME PER STEP

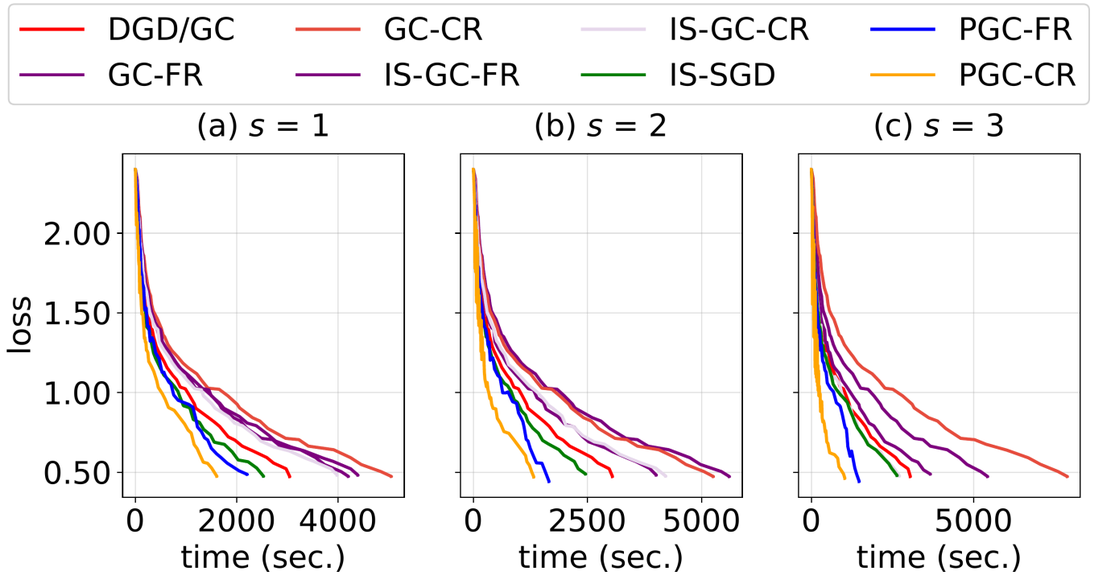
GC always above 100%, up to **180%** at $s = 3$. PGC: **58–79%**, level with IS-SGD which does no redundant work at all.

PER STEP

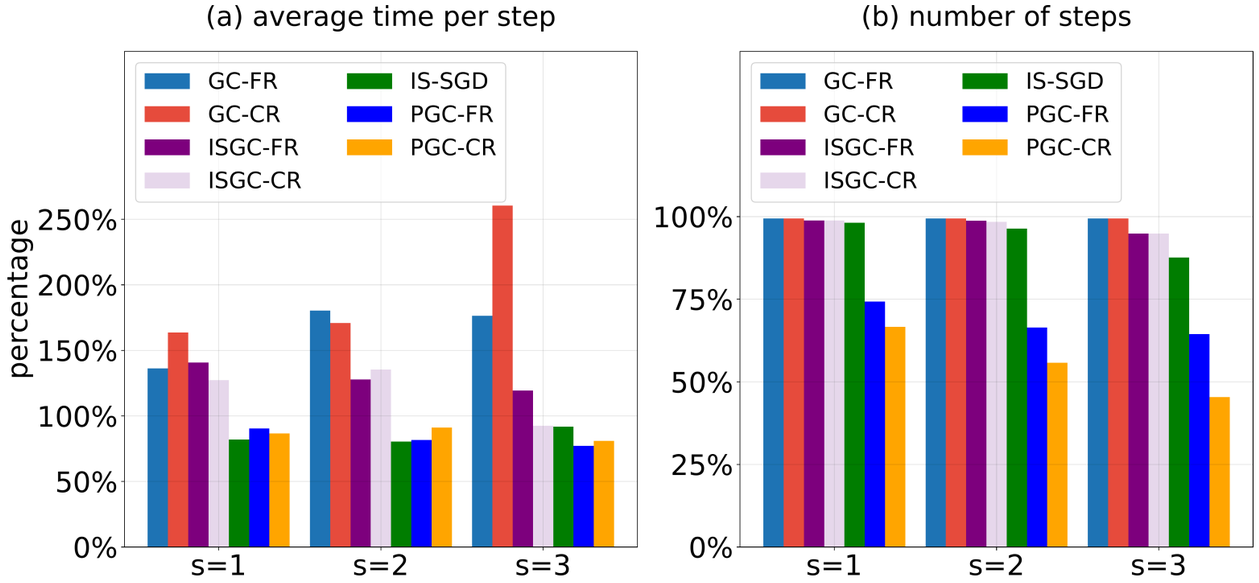
DGD, GC, IS-GC, IS-SGD coincide. PGC pulls ahead, more so for larger s, and **PGC-CR leads PGC-FR**, consistent with the mixed-staleness intuition.

Google Cloud, natural stragglers: both factors at once

`Google Cloud, 12 workers · ResNet-18 / CIFAR-10 · natural stragglers · 10 runs`



Loss vs. wall-clock time



Total time = time per step × number of steps; DGD = 100%

WALL CLOCK

PGC reaches the target first for every s; **PGC-CR fastest in all three panels**; GC slowest.

(A) TIME PER STEP

GC up to **260%** at $s = 3$; PGC **77–91%**, level with IS-SGD.

(B) STEPS

GC = DGD (same gradient). PGC needs **up to about 55% fewer**;
IS-GC ≈ DGD (95–99%).

Takeaways

1 — PIPELINE THE CODING

Spread gradient coding's c-fold computation *across* steps: one partition per step, $c - 1$ retained gradients — DGD's compute, GC's communication and straggler tolerance.

2 — EXACT RECOVERY ≠ ROBUST RECOVERY

GC-CR's decoder is exact for fresh gradients. Reused as-is in the pipeline, its coefficients (up to ± 7500) amplify staleness, and naïve PGC-CR diverges. A weighted sum that is *correct in expectation* keeps every weight below 2.

3 — GUARANTEES AND GAINS

Geometric convergence for both variants. Per-step time at or below DGD's for every s, and up to **about 55% fewer steps**.

Extended version, full proofs:
arxiv.org/abs/2607.20739

Thank you — questions?