Mitigating Staleness in Asynchronous Pipeline Parallelism via Basis Rotation[1]

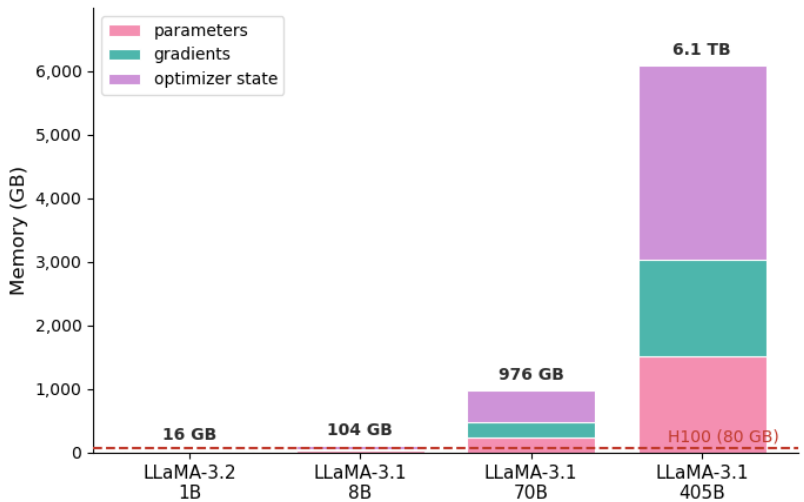
Hyunji Jung*, Sungbin Shin*, Namhoon Lee

POSTECH

25 May 2026

[1] arXiv:2602.03515

Large models must be distributed across many devices

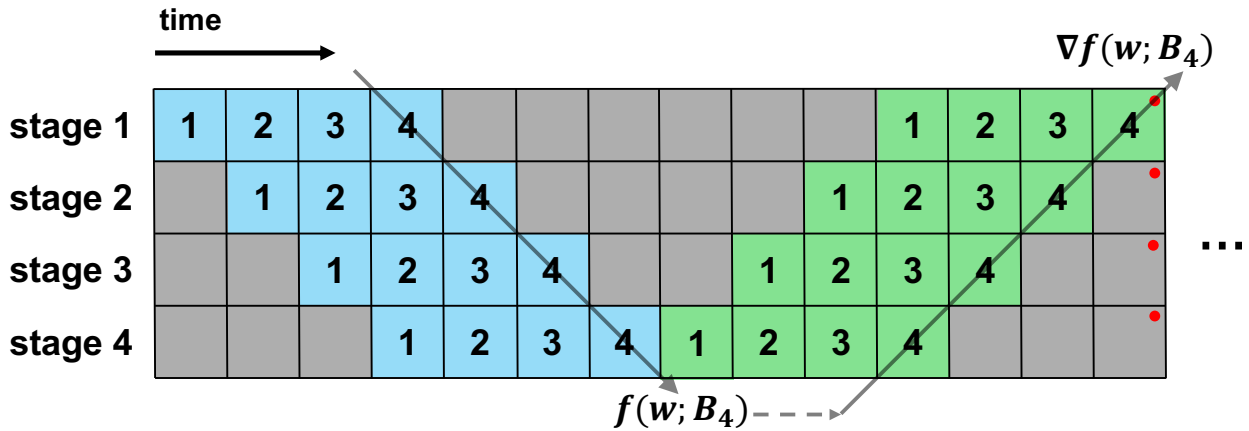


Memory requirement by model size

- Even an 8B model requires 104 GB — already exceeds one H100 (80 GB).
- A 70B model requires 976 GB — $12\times$ a single GPU.

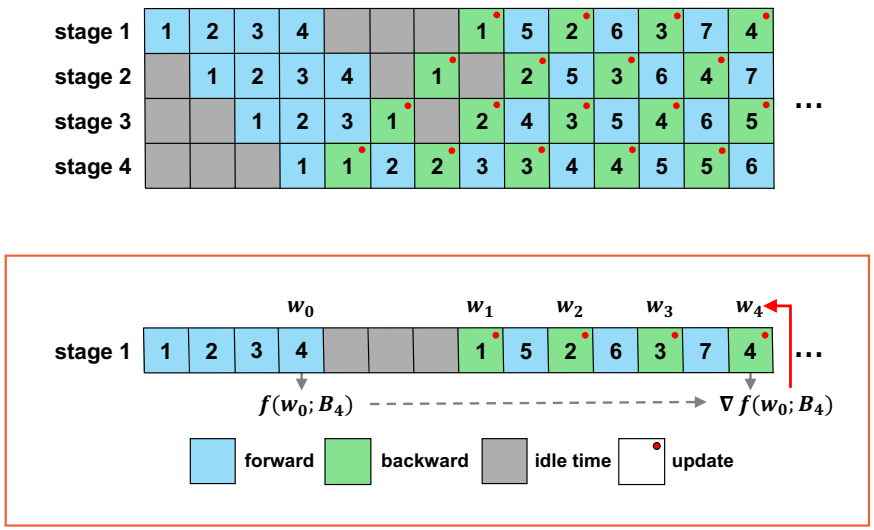
Pipeline parallelism (PP): split model layers into P sequential stages, each on a separate device. Only activations and activation gradients are passed between stages.

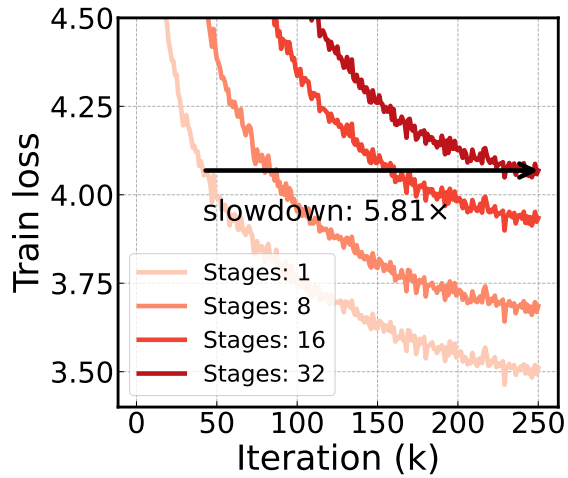
Synchronous PP: pipeline bubbles waste hardware



- Each stage must wait for *all* backward passes to complete before updating weights.
- This creates idle periods — pipeline bubbles — leading to severely suboptimal hardware utilization (Huang et al. 2019; Fan et al. 2021; Li and Hoefler 2021).

Async PP: no bubbles — but gradients become stale





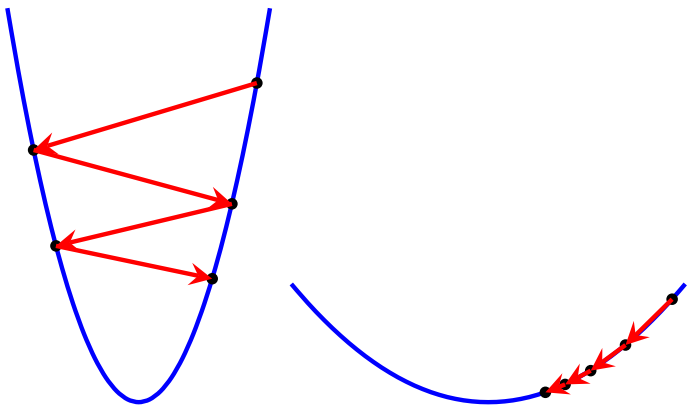
- Stages proceed without waiting ⇒ no bubbles.
- But gradients at $\mathbf{w}_t$ are applied to $\mathbf{w}_{t+\tau}$⇒ delay $\tau \propto P$.

- At P=32: $5.81\times$ slowdown — worsens linearly with depth.

How Adam succeeds: coordinate-wise adaptivity

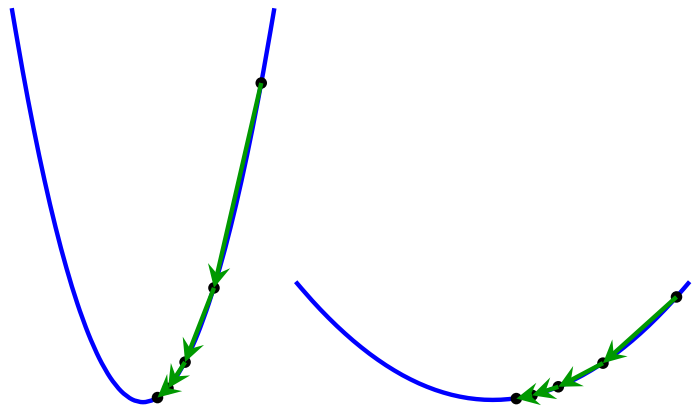
Adam maintains $V_t = \mathrm{EMA}(g_t \odot g_t)$ and rescales each coordinate by $\mathrm{diag}(V_t)^{-1/2}$, adapting the step size to the local curvature per coordinate.

SGD



Oscillates in steep direction

Adam



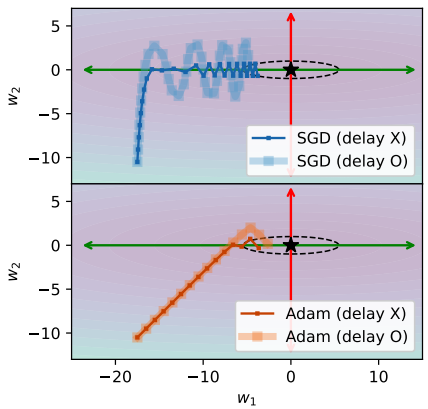
smooth update in every coordinate

Why Adam fails under basis misalignment

Coordinate-wise adaptivity is effective only when the Hessian eigenbasis $\approx$ the coordinate basis (Xie et al. 2025; Zhang et al. 2025).

Consider minimizing $\min_w \frac{1}{2} w^\top H w$ with $H \succ 0$.

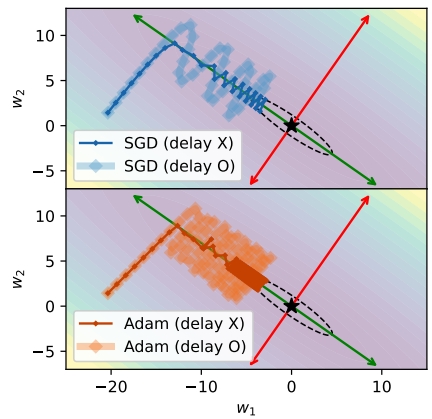
H diagonal (basis aligned)



$\mathrm{diag}(V_t)^{-1/2}$ correctly scales each direction

$\Rightarrow$ straight trajectory

H non-diagonal (basis misaligned)



$\mathrm{diag}(V_t)^{-1/2}$ distorts the geometry

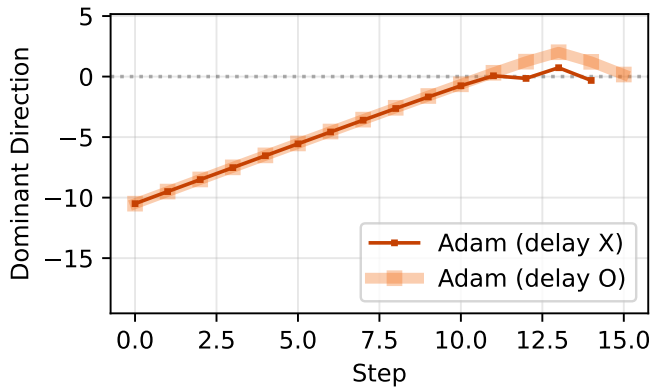
$\Rightarrow$ update direction oscillates

Basis misalignment makes delay hurt training

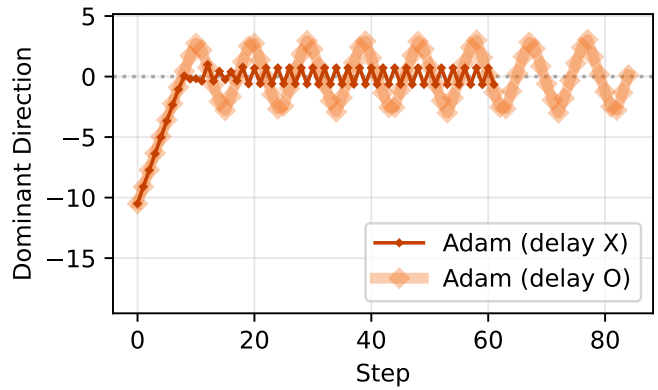
Key insight

When the update direction is stable, delayed update directions stay on-track. ⇒ delay is harmless. When it oscillates due to basis misalignment, delayed update directions may point opposite to the true update direction ⇒ delay hurts severely.

Basis aligned: delay is harmless



Basis misaligned: delay is catastrophic



$$\min_t \mathbb{E}\|\nabla f(w_t)\|_1 = \mathcal{O}\left(\sqrt{\frac{(1+d\tau)\Delta_0 C}{T}} + \sqrt{\sum_{i=1}^{d}\sigma_i}\left(\frac{(1+d\tau)\Delta_0 C}{T}\right)^{1/4} + \sum_{i=1}^{d}\sigma_i\left(\frac{\log T}{T}\right)^{1/4}\right)$$

Solution: Basis Rotation

Apply adam in the **rotated** space $\tilde{w} \triangleq \mathcal{U} w$:

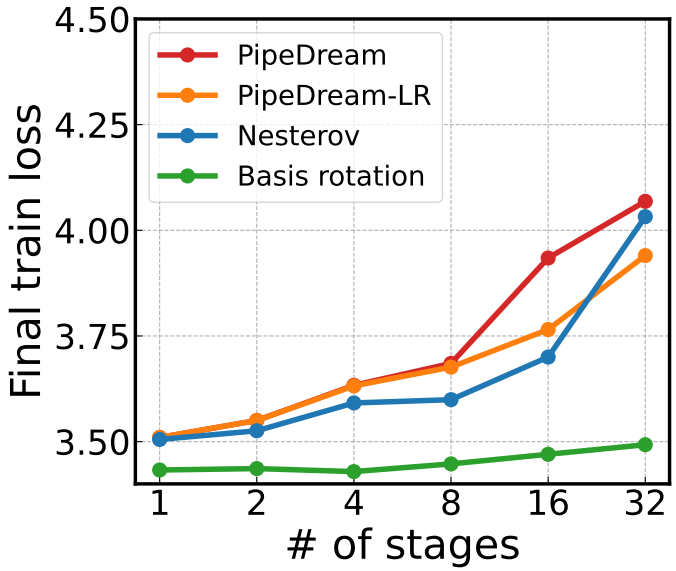
$$w_t = w_{t-1} - \eta_{t-1} \mathcal{U} \frac{\mathsf{EMA}(\mathcal{U}^\top \nabla f(w_{t-1}))}{\sqrt{\mathsf{EMA}((\mathcal{U}^\top \nabla f(w_{t-1}))^2) + \epsilon}}.$$

Adam with Basis Rotation

1. $G_t \leftarrow \nabla f_W(W_t; B_t)$
2. $M_t \leftarrow \beta_1 M_{t-1} + (1-\beta_1) G_t$
3. (every `freq` steps):
   $U \leftarrow \mathrm{Power}(L,\, U),\ V \leftarrow \mathrm{Power}(R,\, V)$
3. $\widetilde{G}_t \leftarrow U^\top G_t V, \quad \widetilde{M}_t \leftarrow U^\top M_t V$
4. $\widetilde{V}_t \leftarrow \beta_2 \widetilde{V}_{t-1} + (1-\beta_2) \widetilde{G}_t \odot \widetilde{G}_t$
5. $W_{t+1} \leftarrow W_t - \eta_t\, U \left(\frac{\widetilde{M}_t}{\sqrt{\widetilde{V}_t + \epsilon}} \right) V^\top$

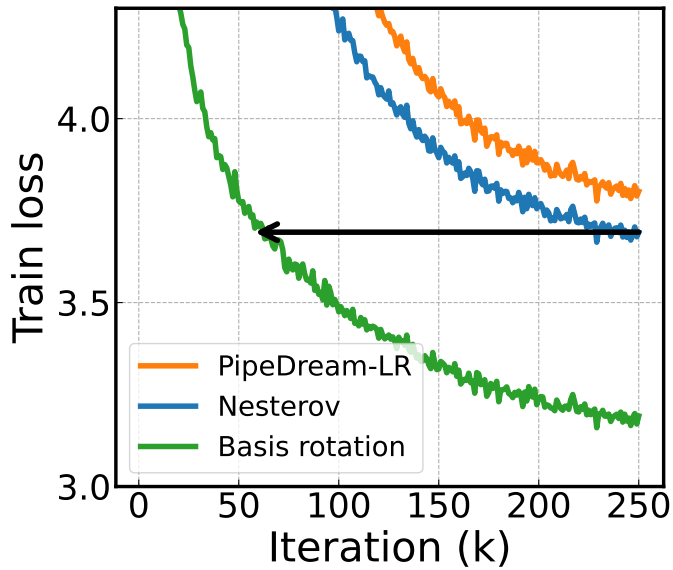
Results

95M model, across # pipeline stages



basis rotation is robust to the # pipeline stages

1B model, P=32



76.8% fewer iterations vs. baselines

Conclusion

- **Problem**. Async PP erases bubbles but injects a gradient delay that grows linearly with depth.
- **Cause**. When the Hessian eigenbasis is rotated off the coordinate axes, Adam oscillates, and a delayed update direction becomes outdated.
- **Fix**. Optimize in a rotated basis.
- **result**. 76.8% fewer iterations vs. baselines

Tue, Jul 7, 2026 • 10:30 AM – 12:15 PM KST, Coex: HALL A