# Mitigating Staleness in Asynchronous Pipeline Parallelism via Basis Rotation

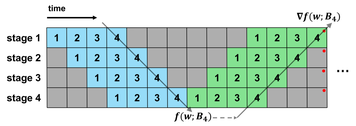
Hyunji Jung*, Sungbin Shin*, Namhoon Lee · POSTECH
*Equal contribution

POSTECH
POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

ICML
International Conference On Machine Learning

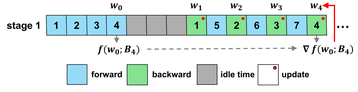
## Motivation

- Large models require deep pipelines across many GPUs.
- **Sync** pipelining makes each stage wait for the others, leaving GPUs idle (*pipeline bubbles*) and **lowering hardware utilization**.
- **Async** pipelining eliminates idle bubbles—but gradients are applied *after* the model has advanced (*staleness*).
- Staleness **scales linearly with pipeline depth**: **5.81**× slowdown at $P=32$ stages.



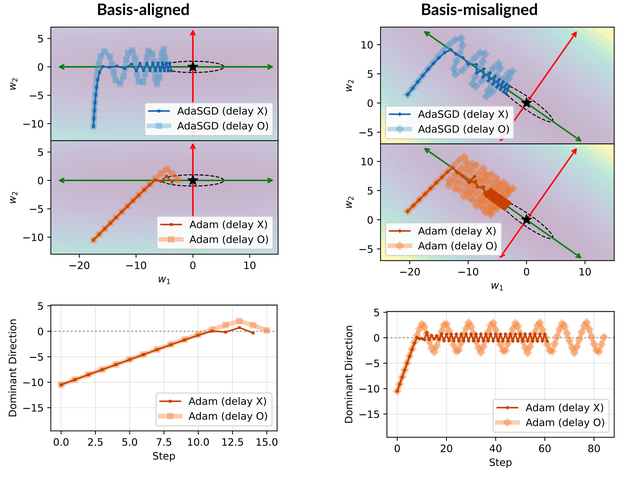

(a) Synchronous PP

(b) Asynchronous PP




(c) Gradient delay in async PP

## Root Cause: Basis Misalignment



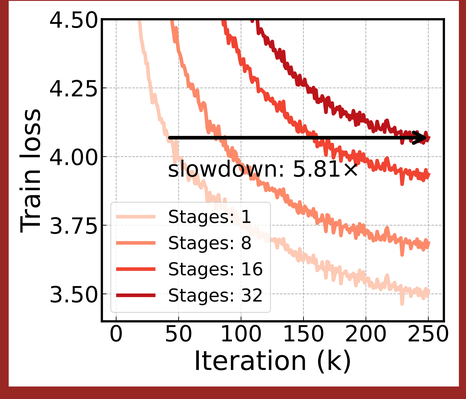

Adam scales correctly → oscillations vanish → trajectory smooth ⇒ **delay harmless**.

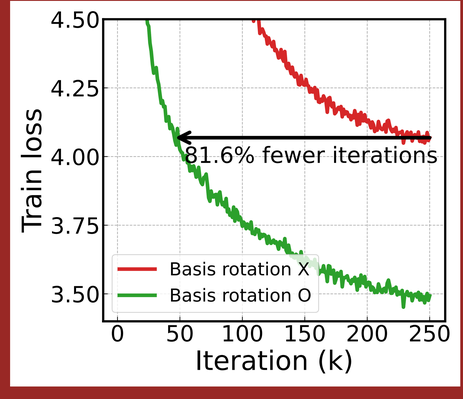
Scaling fails → trajectory oscillates ⇒ delayed gradients **adversarial**.

# Basis misalignment is a fundamental barrier to scaling async pipelines—**Basis Rotation** mitigates it.

arXiv Paper

GitHub Code




### Smooth Trajectory Analysis

When update directions are locally consistent ($u_t \approx u$), delayed iterates $\{\tilde{u}_k\}$ track the originals.

**Moments of the delayed gradient** $\tilde{g}_k$: $\tilde{m}_k \approx m_{k-\tau}$, $\tilde{v}_k \approx v_{k-\tau}$.

Hence $\tilde{u}_k \approx u_{k-\tau} \approx u_k$: a $\tau$-step-stale update still points where the current step needs to go — **delay is harmless**.

An aligned basis keeps the trajectory smooth, so consecutive update directions barely differ.

## Convergence Guarantee

- **A1** (bounded gradient noise): $\mathbb{E}_\xi[\nabla_i f(w;\xi) - \mathbb{E}_\xi \nabla_i f(w;\xi)]^2 \le \sigma_i^2$
- **A2** (coord.-wise $\ell_\infty$-smoothness): $|\nabla_i f(w) - \nabla_i f(w')| \le C_i \|w - w'\|_\infty$

$C \triangleq \sum_i C_i$ locally corresponds to the Hessian $(1,1)$-norm $\|\nabla^2 f\|_{1,1} = \sum_{i,j} |(\nabla^2 f)_{ij}|$, minimized when $\nabla^2 f$ is diagonal ⇒ **proxy for basis misalignment**.

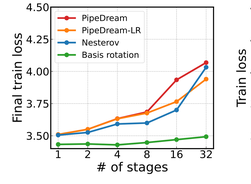
$$\min_{\frac{T}{2}<t\le T} \mathbb{E}\|\nabla f(w_t)\|_1 = \mathcal{O}\left(\sqrt{\frac{(1+d\tau)\Delta_0 C}{T}} + \sqrt{\sum_i \sigma_i}\left(\frac{(1+d\tau)\Delta_0 C}{T}\right)^{1/4} + \sum_i \sigma_i \left(\frac{\log T}{T}\right)^{1/4}\right)$$

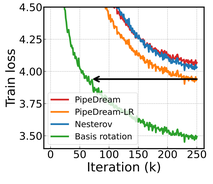
Delay $\tau$ couples **multiplicatively** with $C$: Reduced $C$ ⇒ suppressed impact of delays.

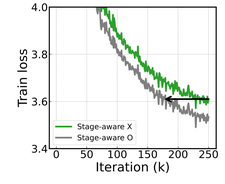
## The Solution: Basis Rotation

- **Approximate Hessian** via Kronecker factorization: $H \approx L \otimes R$, where
  $L = \mathbb{E}[GG^\top], \quad R = \mathbb{E}[G^\top G]$
- **Eigenbases:** $U = \text{eigvecs}(L)$, $V = \text{eigvecs}(R)$ approximately diagonalize $H$.
- **Rotate the moments** each step. From gradient $G_t$ and first moment $M_t = \text{EMA}(G_t)$, form the rotated first and second moments:
  $\tilde{M}_t = U^\top M_t V, \qquad \tilde{V}_t = \text{EMA}\big((U^\top G_t V)^2\big)$
- **Adam in the rotated basis, then rotate back:**
  $$W_{t+1} = W_t - \eta_t U\left(\frac{\tilde{M}_t}{\sqrt{\tilde{V}_t}+\epsilon}\right)V^\top$$
- **Why it works:** in the rotated space the Hessian is *approximately* diagonal—Adam's scaling matches curvature, oscillations vanish, trajectory is smooth.
- **Smooth trajectory** ⇒ **robust to delay:** delayed gradients stay on-direction—staleness stops compounding.

## Experimental Results




- **Setup:** 95M–1B Transformers on OpenWebText, up to $P=32$ stages.
- Basis Rotation stays stable as $P$ grows; baselines degrade severely.
- At $P=32$: **71.6%** fewer iterations than best baseline.
- Advantage grows with model size: **76.8%** at 1B parameters.
- **Stage-aware Basis Rotation:** stage $k$ has delay $\propto (K-k)$—earlier stages suffer most. Budget-neutral redistribution allocates more frequent rotations to earlier stages, suppressing the dominant term in effective delay. **+29.2%** faster convergence vs. uniform frequency.