Fast Catch-Up, Late Switching
Optimal Batch Size Scheduling via FSL

Jinbo Wang*, Binghui Li *, Zhanpeng Zhou, Mingze Wang, Yuxuan Sun, Jiaqi Zhang †, Xunliang Cai & Lei Wu†

* denotes equal contributions, † denotes corresponding author

1. Summary

Theory insights for practice
We study how to design the best batch size scheduling (BSS) strategy under a **fixed learning rate**. This setting is aligned with recent LLM pre-training practice at *Meituan* and *InternLM*, but differs substantially from the classical **Critical Batch Size (CBS)** regime.

What do we do?
We develop a theoretical study of batch size scheduling under this setting. Building on the **Functional Scaling Laws (FSL)** framework, we:
- characterize the optimal BSS in our theoretical setup,
- reveal a **fast catch-up effect** in training dynamics.
- derive a **late-switch strategy** for efficient training.
- finally validate these insights in LLM pre-training.

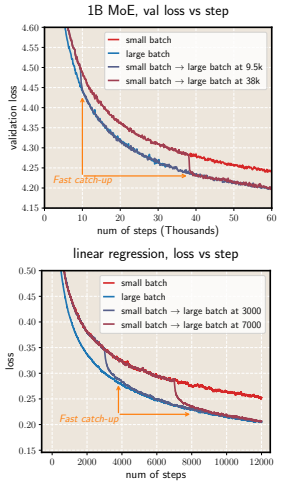


Figure. illustration of fast catch-up.

2. Setting

Feature-space linear regression

Infinity dimension linear regression
with power-law relationships for model and data size.
- s Capacity condition, larger s model can contain more information
- β Source condition, smaller β data can be easier to learn.

Task regime: $s+\frac{1}{\beta} \le 1$ Easy-task regime; $s+\frac{1}{\beta} > 1$ Hard-task regime.

We derive Functional Scaling Law for batch size schedule $b(t)$ under this setup

$$\mathbb{E}[\mathcal{E}(\tilde{\boldsymbol{\theta}}_t)] \eqsim \underbrace{t^{-s}}_{\text{signal learning}} + \underbrace{\eta\sigma^2 \int_0^t \frac{\mathcal{K}(t-\tau)}{b(\tau)}\,\mathrm{d}\tau}_{\text{noise accumulation}},$$

$$\text{where } \mathcal{K}(t) := (t+1)^{-(2-1/\beta)}.$$

Signal learning The main contributor to the loss decrease, decaying with the number of steps t (larger $b(t)$ under fixed D leads to fewer total steps, which slows down the decay of the signal learning term).
Noise accumulation accumulation of gradient noise, decreasing as $b(t)$ increases; due to the convolution kernel, the values of $b(t)$ later in training have a larger impact.

3. Theoretical Analyses via Functional Scaling Laws

Given data budget D, what is optimal batch size schedule (BSS) when loss dynamics follows the FSL in 2.?

Optimal Batch Size Scheduling without Shape Constraints. The optimal batch-size schedule (BSS) differs across task regimes:

- **Easy-task:** The batch size **progressively increases**, and the excess risk rate can achieve the statistically optimal minimax rate $D^{-\frac{s\beta}{1+s\beta}}$.
- **Hard-task:** The optimal BSS exhibits a **stable–growth** structure: initially remains minimum batch size B_{min} and then increases rapidly toward the end, allowing the excess risk rate to reach the optimal rate D^{-s} achievable by SGD.

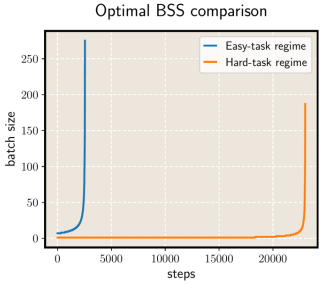


Figure. an illustration of the optimal BSS under two regimes.

Optimal Batch Size Scheduling with Shape Constraints. Batch size is constrained in practical, can only take limited values. We study *minimal setting*: two-stage BS switching.

Theorem 3.2 (Optimal two-stage batch size schedule). *Let $B_1 < B_2$ be constants independent of D, and assume D is sufficiently large. Define $P_D^\star = \arg\min_{P\in[0,D]} \mathcal{E}^D_{B_1\to B_2}(P)$. Then:*

- *If $s > 1-1/\beta$, then $P_D^\star = 0$.*
- *If $s \le 1-1/\beta$, then $\frac{D-P_D^\star}{D} \approx D^{-\frac{1-1/\beta-s}{2-1/\beta}}$.*

Insight

Later Switching Principle

In the hard-task regime, closely aligned with LLM pre-training, BS switches should occur *as late as possible*.

4. Fast Catch-up and Later Switching

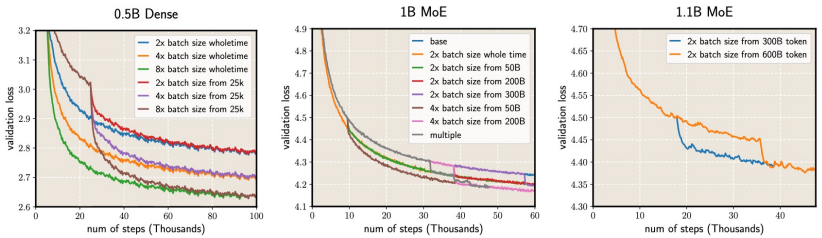


*Figure. **Fast Catch-up** Once the batch size increases, the loss rapidly collapses onto the trajectory of large-batch training.*

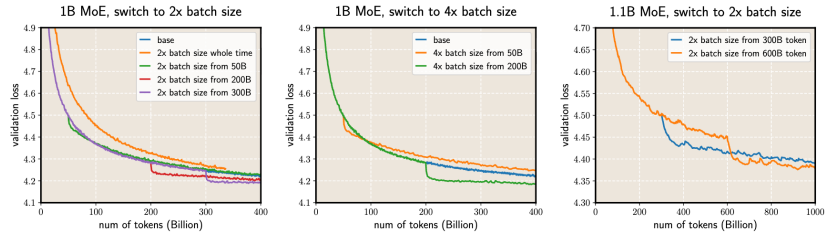


*Figure. **Later Switching** The validation loss of constant large-batch training can be matched by starting with a small batch and switching to large batch late in training.*

Why is later switching better?

Later Switching in Data ← Fast Catch-up in Step

- In terms of steps, both early-switch and late-switch curves quickly catch up to the large-batch trajectory, with the loss roughly equalized.
- However, the **late-switch curve**, spending more steps in small-batch training, effectively uses fewer tokens D, which provides a clear advantage in final performance.

Theoretical understanding of fast catch-up

- **Loss gap at the switching point:** The loss difference at $t_\star$ arises solely from the accumulated noise.

$$G_\star := \mathcal{E}_{B_1}(t_\star) - \mathcal{E}_{B_2}(t_\star) = \eta\sigma^2\left(\frac{1}{B_1}-\frac{1}{B_2}\right)$$

- **Catch-up dynamics:** After switching to a larger batch, the gap decays as the trajectory progressively "forgets" the initial small-batch noise, with a rate determined by the capacity exponent β.

$$\mathcal{E}_{B_1\to B_2}(t_\star+\delta) - \mathcal{E}_{B_2}(t_\star+\delta) \sim G_\star\,\delta^{-(1-1/\beta)}$$

- **When is the catch-up fast?** The catch-up time δ_ϵ is the minimal interval after which the switched trajectory lies within a $(1+\epsilon)$ factor of the large-batch baseline, scaling as $\delta_\epsilon \sim t_\star^{\,1-1/\beta}$.