ICML
International Conference On Machine Learning
2026

When Data Is Scarce: Scaling Sparse Language Models with Repeated Training

Boqian Wu*, Qiao Xiao*, Patrik Okanovic, Tomasz Sternal, Maurice van Keulen, Mykola Pechenizkiy, Elena Mocanu, Torsten Hoefler, Decebal Constantin Mocanu

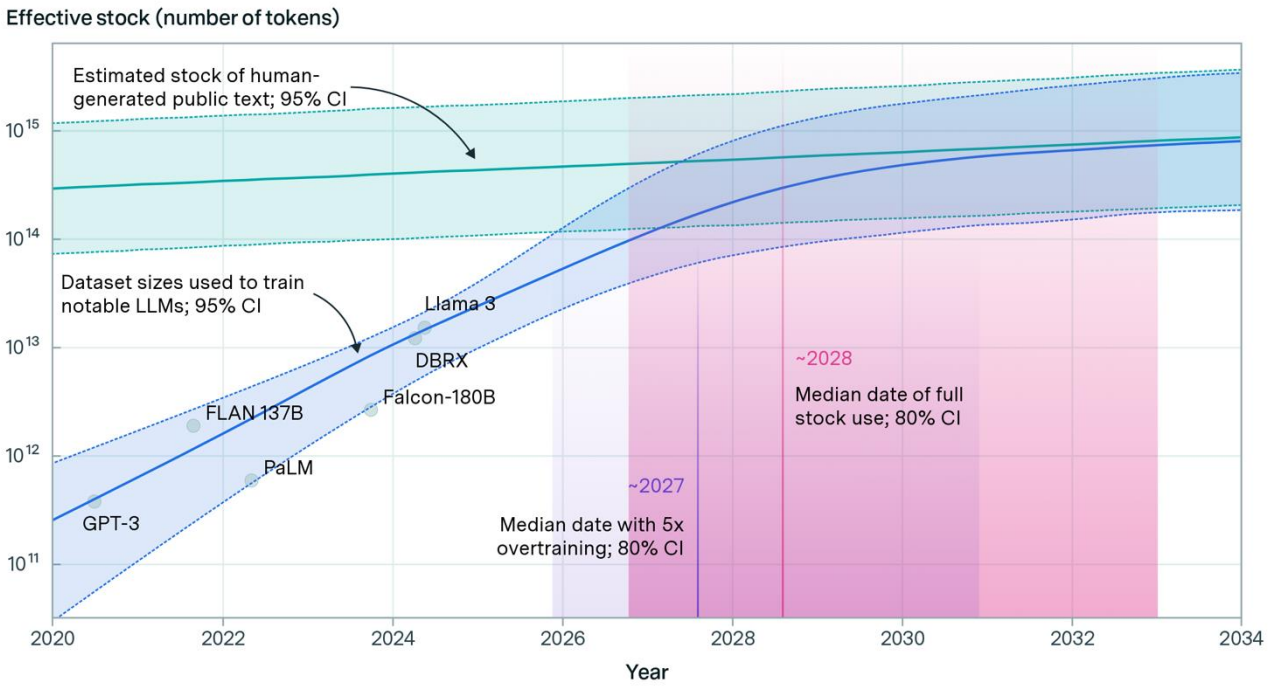
* Equal contributions

Part III: When Data Is Short: What to Do

ICML International Conference On Machine Learning 2026

Projections of the stock of public text and data usage

EPOCH AI



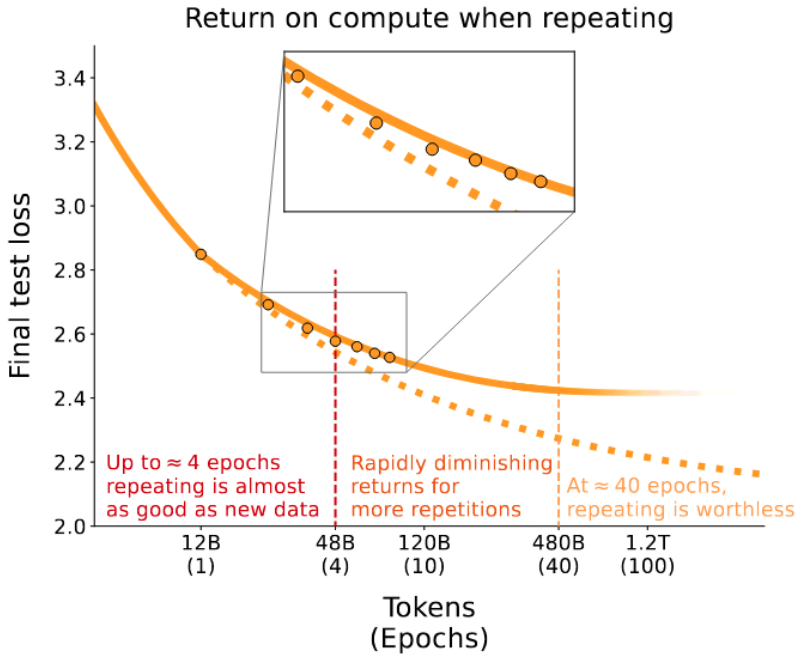
High-quality data grows slowly ≈ 1.03×/year

Pre-training data usage grows quickly ≈ 4×/year

Limited data must be reused multi-epoch

Scaling Data-Constrained Language Models

Niklas Muennighoff [1] Alexander M. Rush [1] Boaz Barak [2] Teven Le Scao [1]
Aleksandra Piktus [1] Nouamane Tazi [1] Sampo Pyysalo [3] Thomas Wolf [1] Colin Raffel [1]
[1] Hugging Face [2] Harvard University [3] University of Turku
n.muennighoff@gmail.com



Marginal returns decline

Limited Data: Typical Scenarios

ICML International Conference On Machine Learning 2026

① Medical Data

Privacy/compliance limits · Expensive expert labels

② Low-resource Languages

Limited corpora · Few labels · Few benchmarks

③ Legal / Financial Data

Private domain data · Strong regulation

④ Enterprise Private Data

Small, specialized internal knowledge bases · Hard to share

Data-constrained regimes arise when high-quality, domain-specific, private, or expert-labeled data cannot scale with compute.

Why might DST interact non-trivially with data repetition?

Dense training

Repeated exposure concentrates on a same set of weights

DST mask evolves

Repeated exposure concentrates on a different set of weights

Core intuition

Dense training repeats updates on the same weights, so it saturates earlier. DST changes the active weights over time, so repeated data is used more broadly and saturates later.

Background & Motivation

ICML International Conference On Machine Learning 2026

① Scaling laws

Chinchilla: data-sufficient

$$L(N, D) = \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}} + E$$

C ≈ 6ND | N: # parameters, D: # training tokens

- **High-quality unique data grows slowly, continued training ⇒ multi-epoch reuse**

Data-constrained: repeated tokens

$$L(N', D') = \frac{A}{(N')^{\alpha}} + \frac{B}{(D')^{\beta}} + E,$$

$$D' = U_d + U_d R_d^* \left(1 - e^{-R_d / R_d^*}\right),$$
$$N' = U_n + U_n R_n^* \left(1 - e^{-R_n / R_n^*}\right),$$

U_d: unique tokens | R_d / R_n: repetition ratio

R_d^* / R_n^*: saturation scale

Dense law is not enough

- Dense: same parameters repeatedly see the same data
- Sparse: spread updates across a larger parameter space

Does sparsity change these saturation scales?

- Repeated data contributes sub-linearly : $D' < U_d * (1 + R_d)$

Research Questions

ICML International Conference On Machine Learning 2026

Q1 · Scaling law

How does pre-training loss change with active parameters, unique data, and repetition across sparsity levels?

Q2 · Allocation

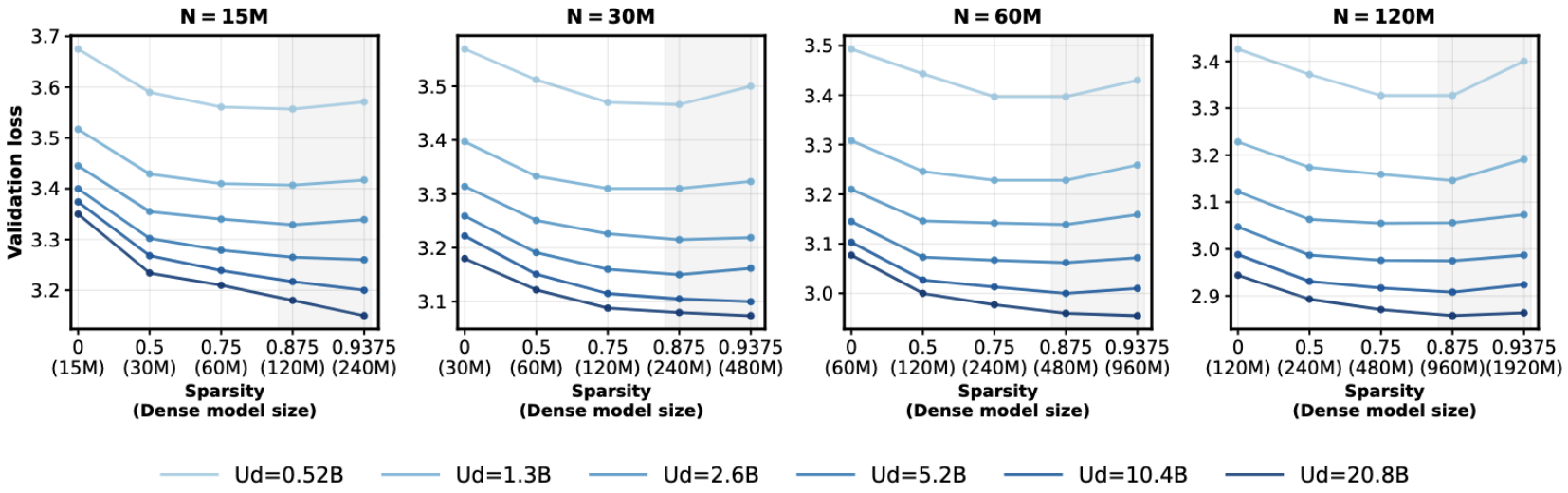
With fixed unique data and compute, should compute go to larger models or more epochs?

Q3 · Optimal sparsity

Given a data budget, what are the loss-optimal and compute-optimal sparsity levels?

Q1 - Observation 1: Sparse Capacity Gains and Saturation

ICML International Conference On Machine Learning 2026



- Moderate sparsity improves effective capacity
- As S increases ⇒ dense-equivalent size N/(1−S) increases
- At extreme sparsity, fixed data cannot fully exploit the extra capacity

Before

$$\frac{A}{N^{\alpha}}$$

Observation 1

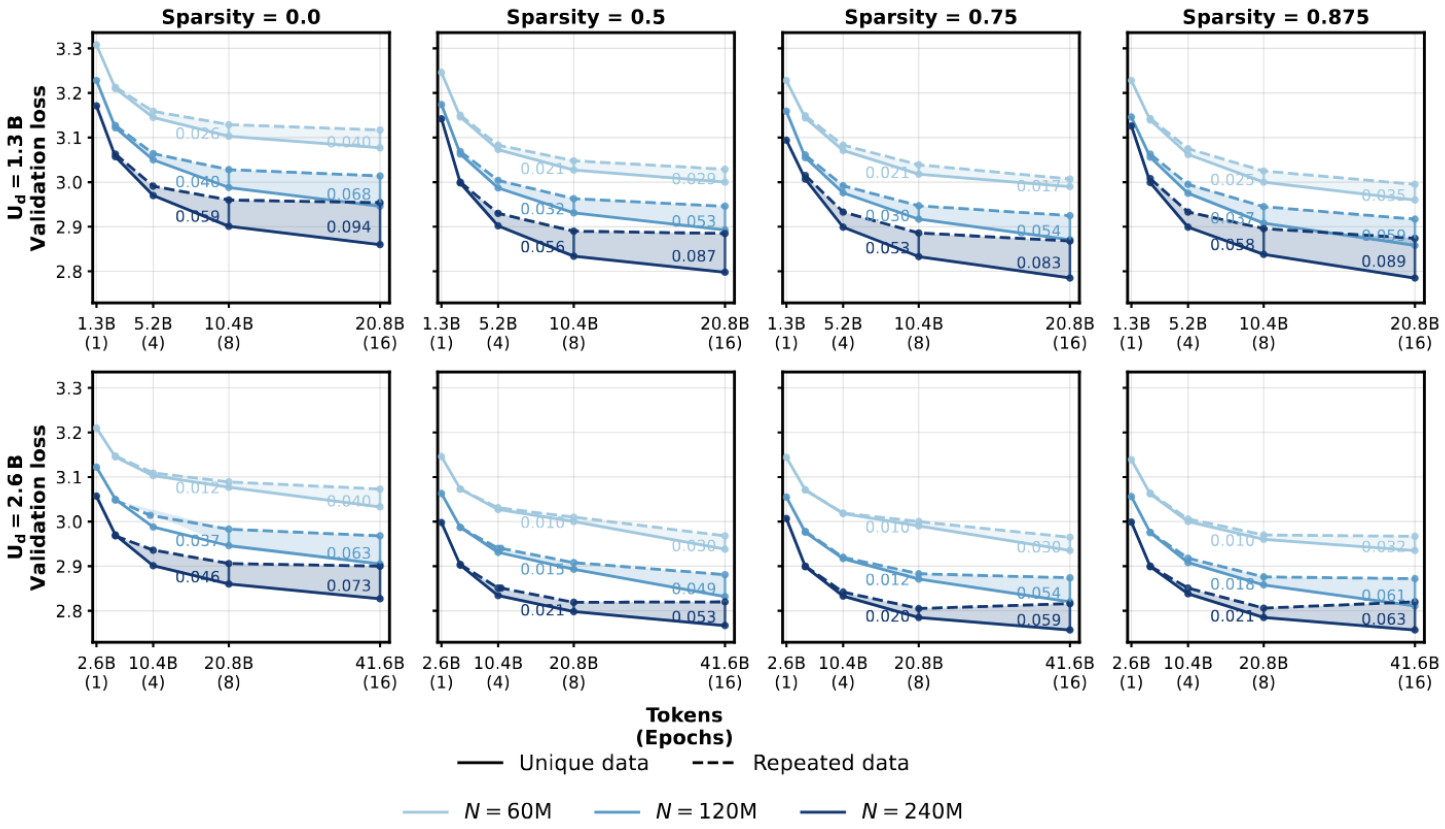
sparsity changes parameter efficiency

After

$$\frac{A\left((1-S)^{\epsilon}+PS^{\mu}\right)}{N^{\alpha}}$$

Q1 - Observation 2: Sparsity Delays Data-Reuse Saturation

ICML International Conference On Machine Learning 2026



- Sparsity delays repeated-data saturation

Scaling law update

The data saturation threshold R_d^* depends on sparsity

$$R_d^*(S) = R_d^*(1 + \lambda_1 S + \sigma_1 S^2)$$

A larger $R_d^*(S)$ means repeated data saturates later

Experimental Design: From Data Points to Fitted Parameters

R_d

2026

Sparsity-aware data-constrained scaling law

$$L(N', D', S) = \frac{A\left((1-S)^{\epsilon} + PS^{\mu}\right)}{N'^{\alpha}} + \frac{B}{D'^{\beta}} + E$$

Effective model size

$$N' = U_n + U_n R_n^*(S)\left(1 - e^{-R_n / R_n^*(S)}\right)$$

Effective data size

$$D' = U_d + U_d R_d^*(S)\left(1 - e^{-R_d / R_d^*(S)}\right)$$

Parameters to fit:
θ = {A, B, E, α, β, ε, P, μ, R_d^*, R_n^*, λ_1, σ_1, λ_2, σ_2}

Fitting flow

① Each training run is one data point

input: (N, U_d, epoch, S) → target: validation loss L

② Compute derived variables in the formula

D = epoch · U_d, R_d= D/ U_d − 1

U_n denotes the optimal parameter count for U_d then obtain R_n

③ Fit θ using all points

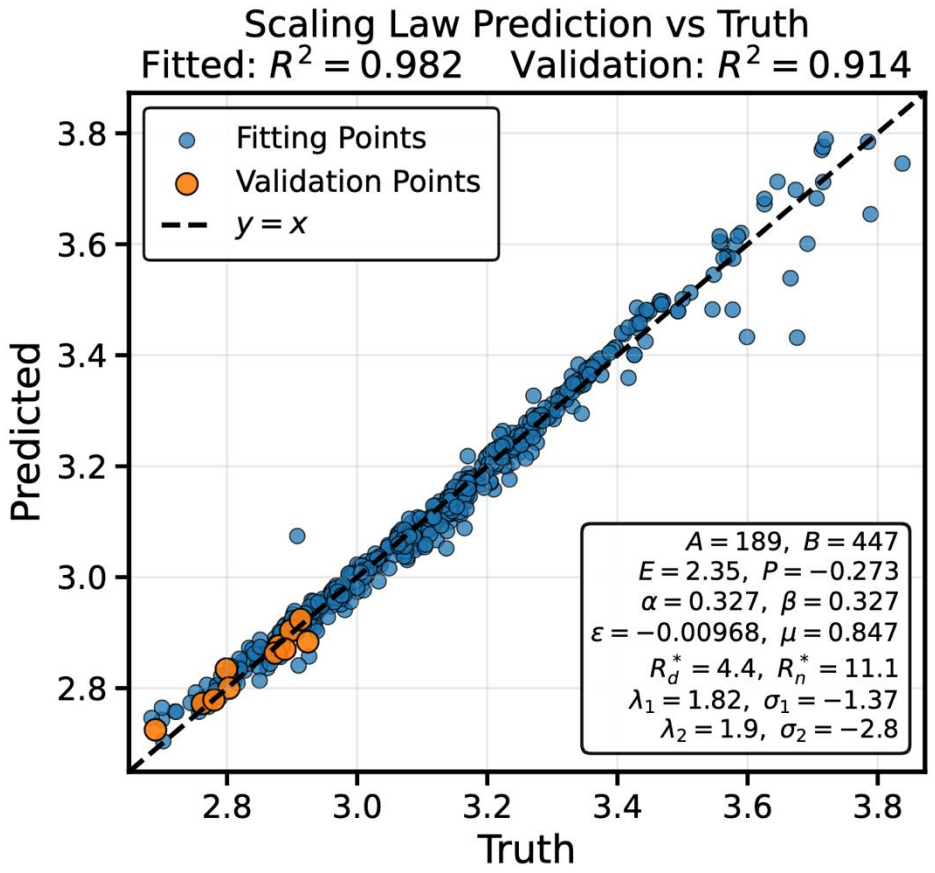
min over θ: Σ (predicted loss − observed loss)²

optimizer: L-BFGS;

500 training runs; N=15M→960M, Ud=0.52B/1.3B/2.6B, epochs=1/2/4/8/16, S=0→93.75%.

Fit Quality: The Formula Predicts Unseen Configurations

ICML International Conference On Machine Learning 2026



Sparse data-constrained scaling law

$$L(N', D', S) = \frac{A\left((1-S)^{\epsilon} + PS^{\mu}\right)}{N'^{\alpha}} + \frac{B}{D'^{\beta}} + E$$

$$N' = U_n + U_n R_n^*(S)\left(1 - e^{-R_n/R_n^*(S)}\right)$$

$$D' = U_d + U_d R_d^*(S)\left(1 - e^{-R_d/R_d^*(S)}\right)$$

Mechanisms read from fitted parameters

$P < 0,\ \mu > 0 \rightarrow$ sparsity reduces the parameter-scaling penalty and improves parameter efficiency

$\epsilon < 0 \rightarrow$ sparsity gains have diminishing returns as S increases

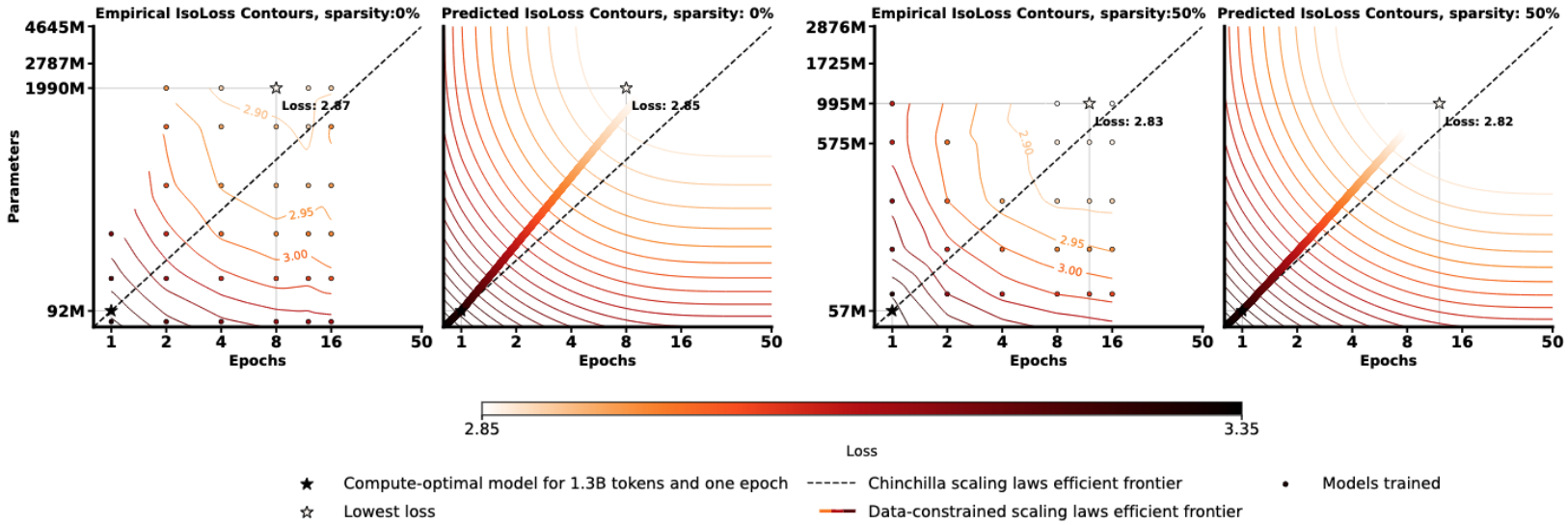
$R_d^*(S) \rightarrow$ repeated data saturates more slowly; around 6.9 at S=0.5

$R_n^*(S)$ peaks early $\rightarrow$ excessive sparsity limits model scaling

Resource Allocation I: Sparsity Changes the Efficient Frontier

ICML International Conference On Machine Learning 2026

Example with Ud = 1.3B

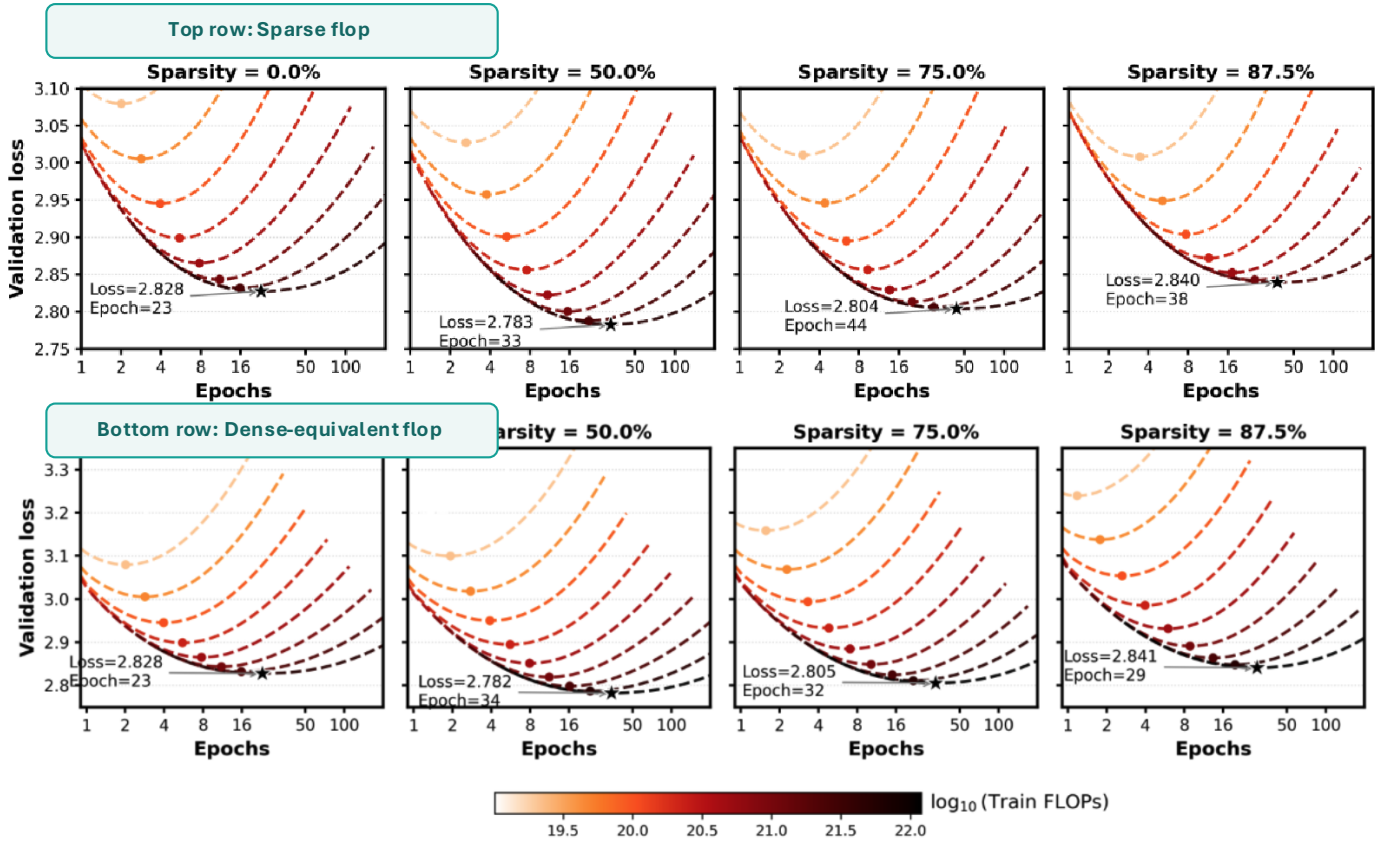


- Dense near-optimal: about 8 epochs + 2B non-zero params
- S=50%: about 12 epochs + 1B non-zero params
- parameter/token ratio is reduced by about 3×, inference is also lighter

In data-constrained settings, sparse models benefit more from training longer, while using fewer active parameters.

Isoflop

ICML International Conference On Machine Learning 2026



Common conclusion

Moderate sparsity (≈50) achieves lower validation loss in the high-compute regime.

Sparse flop

Only non-zero parameters are counted

moderate sparsity shows advantages earlier at lower compute.

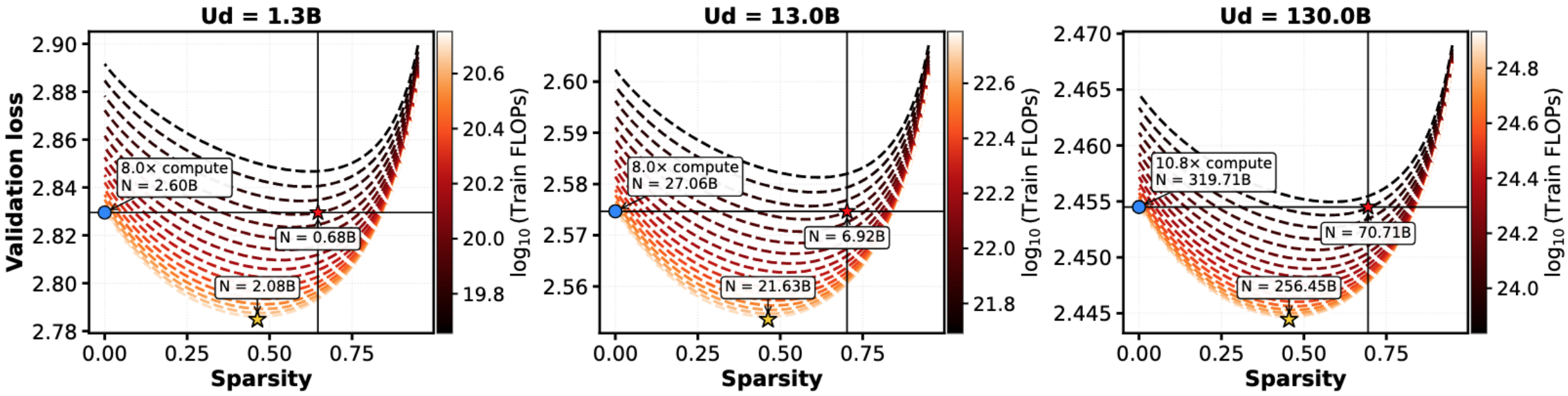
Dense-equivalent flop

Even when counting the dense base model cost, moderate sparsity still maintains lower loss in the high-compute regime.

Moderate sparsity is not useful only under ideal sparse flop; under realistic dense-equivalent flop, it can still achieve lower loss in the high-compute regime.

Optimal Sparsity: Two Different Optimization Goals

ICML International Conference On Machine Learning 2026



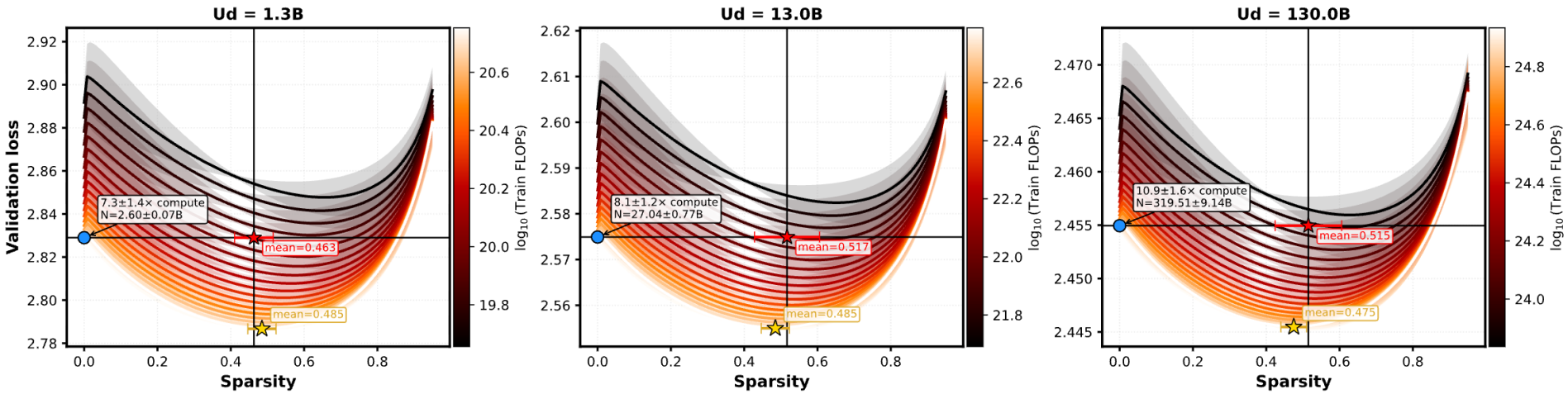
Loss-optimal: “How low can the loss go?”

Compute-optimal: “How much compute can be saved while matching dense optimum?”

Under compute-optimal : fewer non-zero parameters and about 8×–10× fewer training flop

Sensitivity analysis

ICML International Conference On Machine Learning 2026



Leave-one-sparsity refits show that the curve shape is most stable in the moderate-sparsity region; compute savings remain about 7–11× across refits.

Limitation: Results are stable within our studied regime, but larger scales and more diverse domains may introduce additional effects and require further validation.

sparse training changes the marginal value of data reuse

ICML International Conference On Machine Learning 2026

When data is limited and multi-epoch training is necessary, DST spreads updates over a larger parameter space, making repeated data saturate later.

R^*_d: 4.4 → 6.9

At 50% sparsity, the repeated-data saturation threshold increases substantially

≈ 3× smaller

At 1.3B tokens and ~10^{20} flop, the parameter/token ratio is reduced

8–10× fewer

Training flop drop sharply at dense-equivalent performance

Three core findings

① Sparse training improves finite-data utilization;

② with fixed data, sparse models benefit more from training longer than simply becoming larger;

③ Loss-optimal S≈0.45–0.5; compute-optimal S increases with data scale.

Q & A

Thank you !