Datasetv4 Scaling Laws

Date	2026-02-02 17-09
Commit	5e73257
Prompt	Train 10M, 100M, 500M, 1B models from scratch on dataset-v4 for 4 epochs each. Compare FLOP-efficiency and scaling behavior across all sizes.

Methodology

Four ResNet models (10M, 100M, 500M, 1B parameters) were trained from scratch on dataset-v4, a fixed corpus of ~6.6M Go positions. All models used identical hyperparameters: muP-scaled AdamW with LR=3e-3, cosine schedule with 500 warmup steps, weight decay 0.01, mixed-precision (bfloat16), batch size 64. Training targeted 4 epochs (~414k steps) per model. The 1B model completed 3 of 4 epochs (~220k steps) before the spot instance was reclaimed.

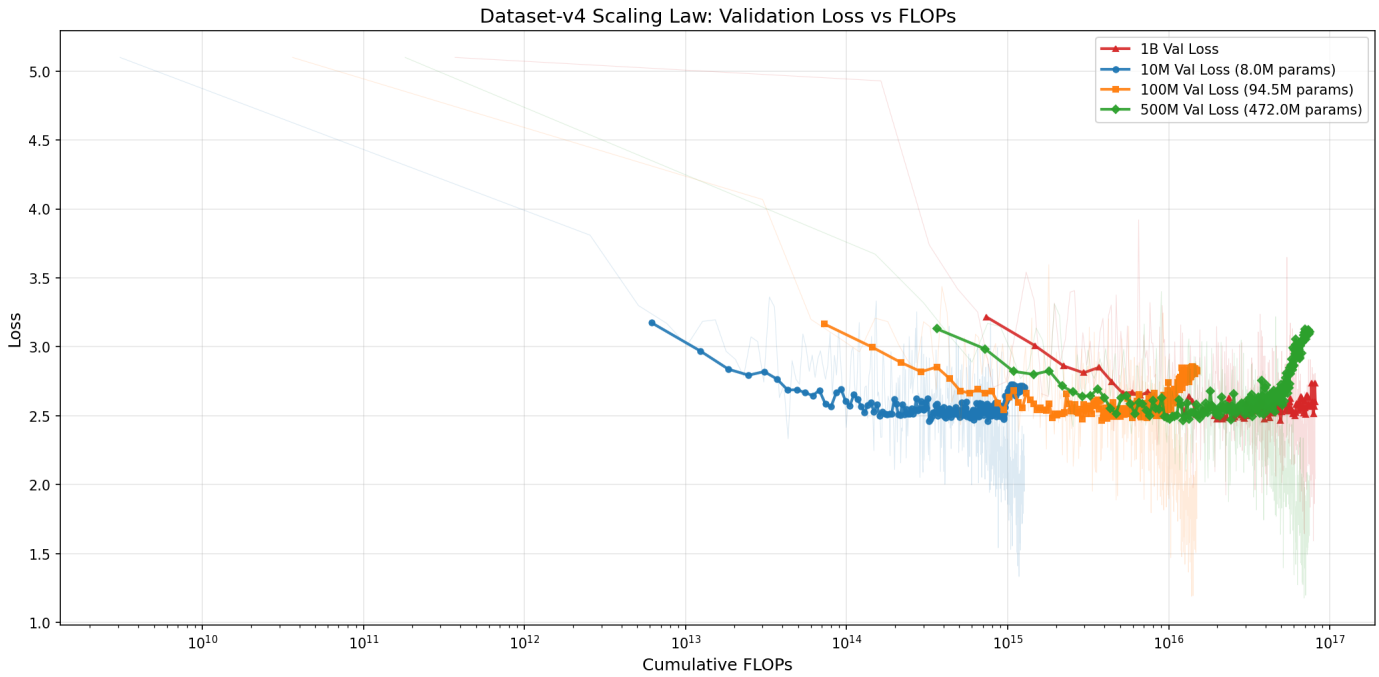
Model configurations:

Model	Parameters	Channels	Blocks	FLOPs/step
10M	8.0M	—	—	3.08e9
100M	94.5M	—	—	3.63e10
500M	472.0M	—	—	1.81e11
1B	963.7M	1408	27	3.70e11

Data regime: With ~6.6M positions, even 4 epochs provides only 3.3 tokens/parameter for the 10M model, 0.3 for 100M, 0.06 for 500M, and 0.03 for 1B. For reference, Chinchilla scaling recommends ~20 tokens/parameter. This experiment is deeply data-limited for all models larger than 10M.

Results

Validation Loss vs FLOPs



All four models converge to nearly identical best validation loss (~2.46) despite 120x parameter difference. The 10M model reaches this minimum with ~7.5e14 FLOPs; the 1B model requires ~5.0e16 FLOPs (67x more compute) for the same result. After reaching their minimum, larger models diverge upward sharply due to overfitting.

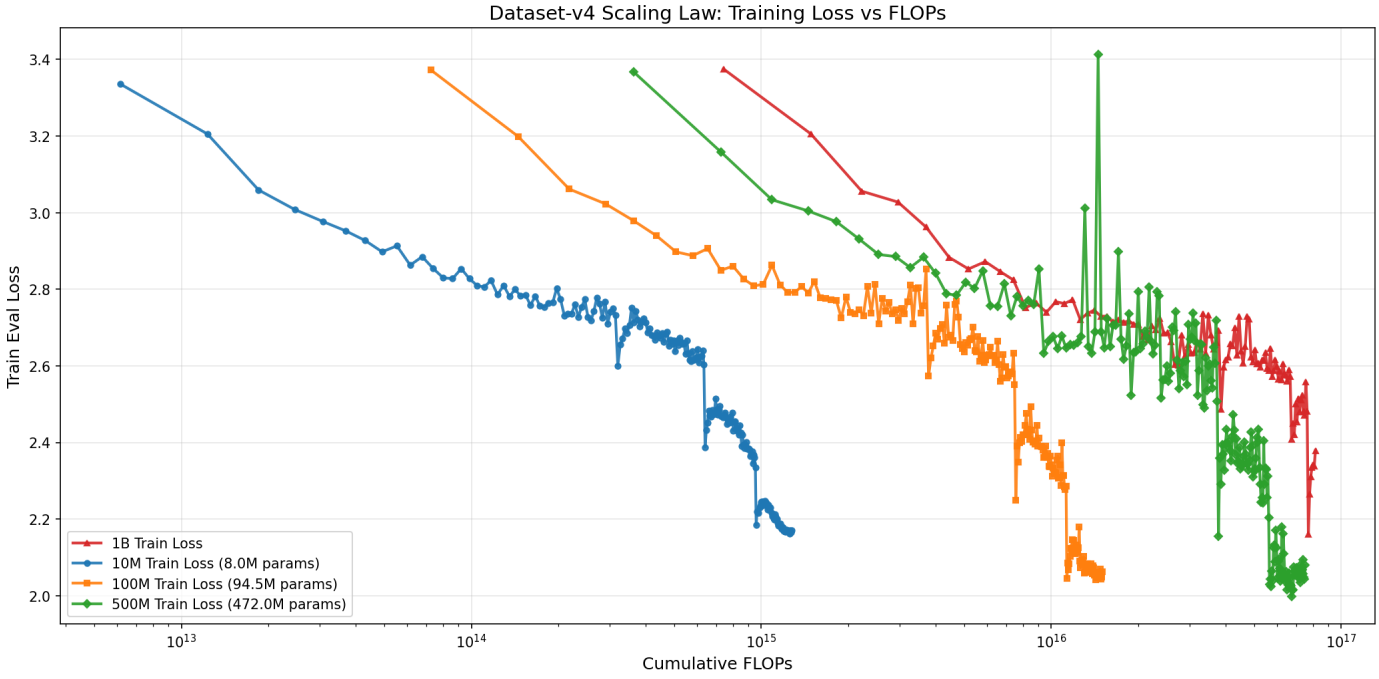
Validation Loss vs Step

Larger models reach their best val loss earlier in training (fewer steps/epochs):

- **500M**: best at step 68k (epoch 1)
- **100M**: best at step 106k (epoch 2)
- **10M**: best at step 244k (epoch 3)
- **1B**: best at step 134k (epoch 2)

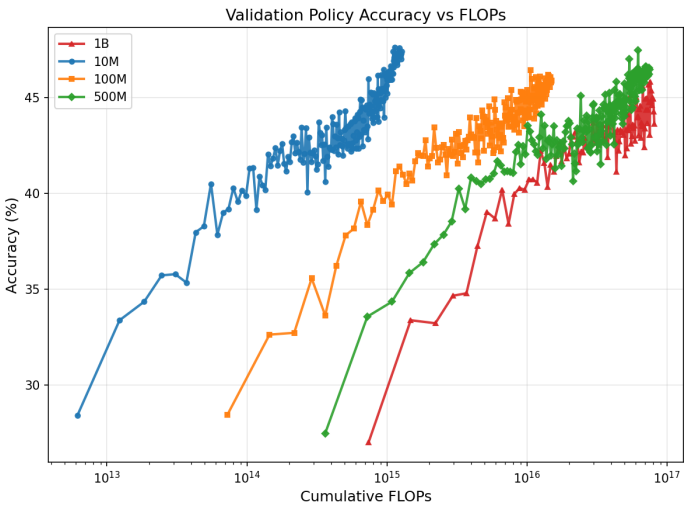
After the optimum, val loss rises monotonically. By epoch 4, the 500M model has degraded from 2.47 to 3.11 (+0.64), while the 10M model goes from 2.46 to 2.70 (+0.24). Overfitting severity is proportional to model capacity.

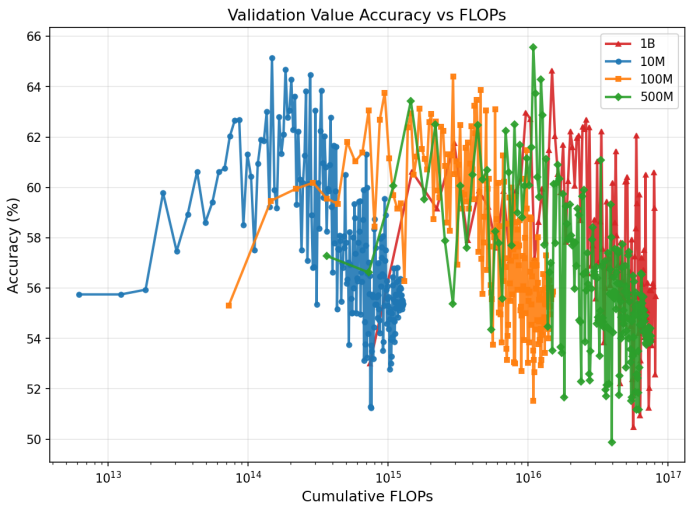
Training Eval Loss vs FLOPs



Training eval loss continues to decrease for larger models (500M train eval loss reaches ~2.0) even as validation loss rises. This confirms the train-val gap is driven by memorization of the finite training set.

Accuracy vs FLOPs

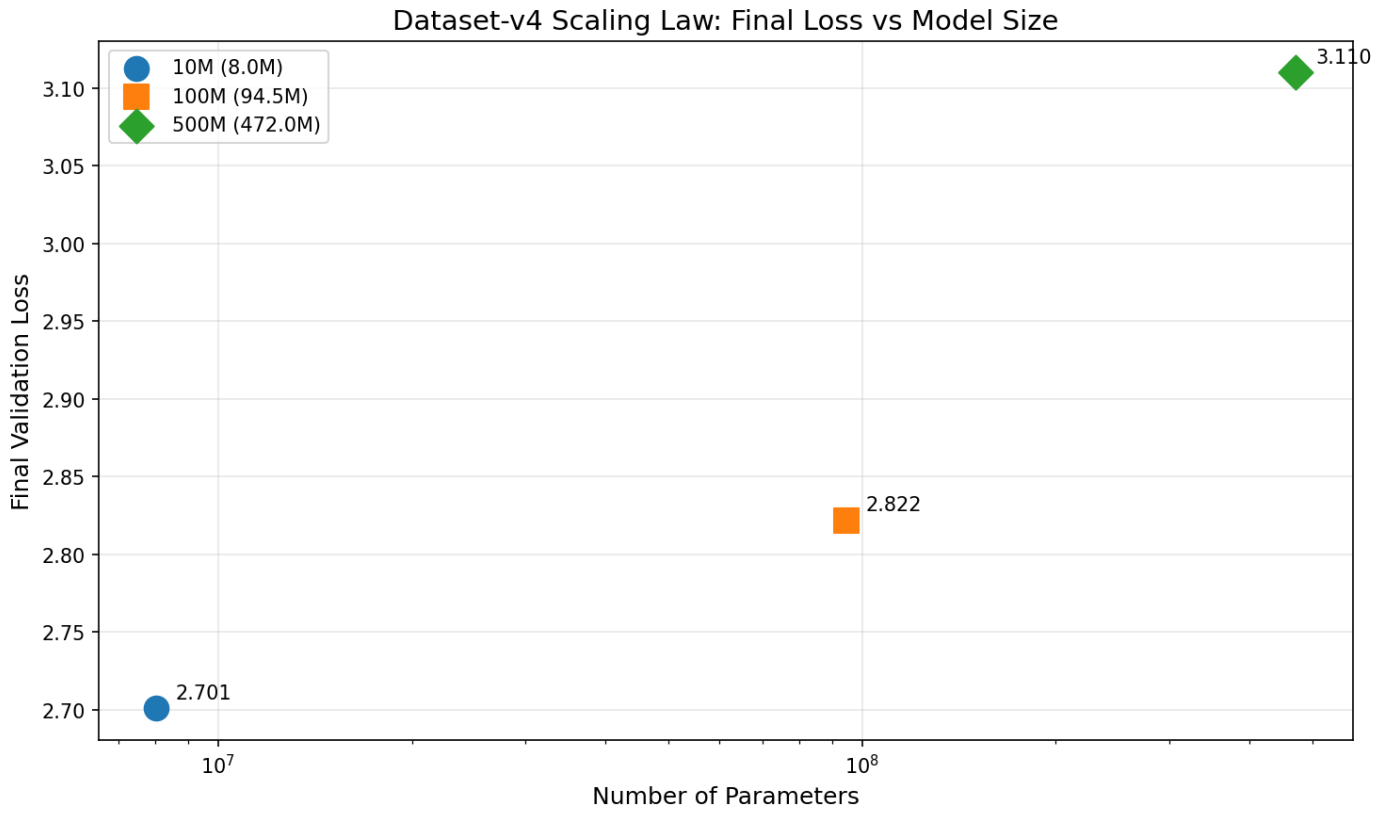




Policy accuracy plateaus around 43-48% for all models. The 10M model peaks highest at 47.6%, while larger models achieve marginally lower peaks (500M: 47.5%, 1B: 45.8%). On a per-FLOP basis, the 10M model is dramatically more efficient at reaching any given accuracy level.

Value accuracy shows more variation: 500M peaks at 64.3%, 100M at 60.9%, 10M at 55.9%. Larger models appear to extract value information more effectively, but this advantage does not persist after overfitting sets in.

Final Loss vs Model Size



Final (end-of-training) validation loss *increases* with model size: 2.70 (10M) → 2.82 (100M) → 3.11 (500M). This is the opposite of the expected scaling law, driven entirely by overfitting in the data-limited regime.

Best Val Loss Per Model

Model	Best Val Loss	At Step	At FLOPs	Policy Acc	Value Acc
10M	2.4618	244,000	7.51e14	44.5%	53.7%
100M	2.4668	106,000	3.85e15	43.0%	60.9%
500M	2.4667	68,000	1.23e16	42.9%	64.3%
1B	2.4667	134,000	4.96e16	44.3%	60.2%

Overfitting by Epoch

Model	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Overfit Gap
10M	2.560	2.511	2.519	2.701	+0.190
100M	2.547	2.543	2.593	2.822	+0.279
500M	2.576	2.576	2.730	3.110	+0.534
1B	2.563	2.570	2.599	—	+0.037

FLOP-Optimal Frontier

The 10M model is FLOP-optimal at every compute budget tested (1e14 through 8e16 FLOPs). No larger model achieves a lower validation loss at any FLOP budget.

Summary

model	train_loss_min	train_loss_max	train_loss_last	train_policy_loss_min	train_policy_loss_max	train_policy_loss_last	train_value_loss_mi
100M	0.852	5.2131	1.7615	0.5998	4.5193	1.5046	0.109
10M	1.0003	5.1115	1.8858	0.692	4.5129	1.5536	0.155
1B	1.3938	5.188	2.2558	0.9379	4.5427	1.8376	0.206
500M	0.7943	5.1535	1.7413	0.5052	4.4889	1.4955	0.094

Key Findings

- **All models converge to the same best val loss (~2.46)** regardless of parameter count (8M to 964M). This ceiling is set by the dataset, not model capacity. The dataset's irreducible loss is approximately 2.46.
- **Larger models are strictly worse on a FLOP-efficiency basis.** The 10M model dominates the FLOP-optimal frontier at every compute budget tested. Increasing model size 120x yields zero improvement in best achievable loss while costing 67x more FLOPs.
- **Overfitting severity scales with model size.** By the end of training, val loss degradation is +0.19 (10M), +0.28 (100M), +0.53 (500M). Larger models memorize the training set faster and more thoroughly.
- **Larger models reach their optimum earlier (in steps).** The 500M model peaks at step 68k (epoch 1) while the 10M peaks at step 244k (epoch 3), consistent with faster memorization of the fixed dataset.
- **The dataset is severely undersized.** At ~6.6M positions, even the 10M model gets only 3.3 tokens/parameter over 4 epochs (vs Chinchilla's recommended ~20). The 1B model gets 0.03 tokens/parameter — 600x below the recommended ratio.
- **Value accuracy shows mild model-size benefit.** Larger models peak higher on value accuracy (500M: 64.3% vs 10M: 53.7%), suggesting the value prediction task has more learnable signal at higher capacity, but this advantage is lost to overfitting.
- **Policy accuracy is not improved by scale.** All models plateau around 43-48% policy accuracy, with the 10M model marginally highest (47.6%).

Conclusions

This experiment demonstrates that dataset-v4 (~6.6M positions) is deeply insufficient to support models larger than 10M parameters. The experiment is operating in a strongly data-limited regime where:

1. **Model scale provides no benefit** — all models hit the same loss floor.
2. **Additional epochs are harmful for large models** — 500M should stop after 1 epoch, 100M after 2.
3. **Compute is wasted** — training the 1B model costs 67x more FLOPs than the 10M model for the same result.

Next steps

- **Scale the dataset.** To benefit from the 1B model, dataset-v4 would need to grow from ~6.6M to ~100-200M+ unique positions (approaching 0.2 tokens/parameter at minimum, ideally much more).
- **Use early stopping.** If training larger models on this dataset, stop after 1-2 epochs and pick the checkpoint with best val loss (especially for 500M/1B).
- **Investigate the loss floor at 2.46.** This ceiling appears to be an intrinsic property of dataset-v4. It could be driven by label noise, position diversity, or limitations in the supervision signal. Understanding this floor is important before scaling data.
- **Train 10M longer or with more data.** The 10M model is the most FLOP-efficient and has the most room to benefit from additional unique data.

Artifacts

- Data: `data/combined_training_log.csv`
- Figure: `figures/loss_vs_flops.png`
- Figure: `figures/loss_vs_step.png`
- Figure: `figures/train_loss_vs_flops.png`
- Figure: `figures/accuracy_vs_flops.png`
- Figure: `figures/perplexity_vs_flops.png`
- Figure: `figures/final_loss_vs_params.png`