Scripps Research

Compute-optimal scaling laws for data-constrained antibody language models

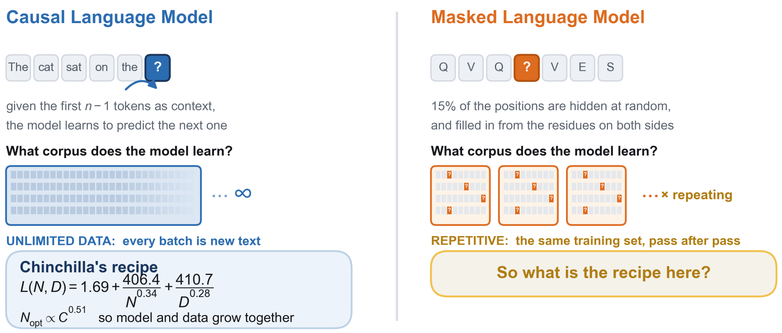
Jinglin Jian · Karenna Ng · Sarah M. Burbach · Shelby Ferrier · Mahdi Shafiei Neyestanak · Praneeth Gangavarapu · Benjamin Nemoz · Jonathan Hurtado · Stefano Forli · Bryan Briney

Scripps Research, La Jolla, CA, USA

Antibody MLMs must repeat their data. What is the recipe?

Scaling laws are the organizing principle for training large models: loss falls as a predictable power law in model size, data and compute[4], and the Chinchilla law[3] turns that surface into a recipe that grows the model and the data together (below left). Masked language models (MLMs) have the same law with a different objective: **Cheng et al.**[1] fitted compute-optimal rules for masked protein language models. But every token those models are fed is fresh.

Antibody masked language models sit in the more serve data-constrained setting. The corpus is small, fixed, and expensive to collect, so more tokens mean more epochs over the same data[8]. Following **Muennighoff et al.**[5] we count effective data, where a repeated epoch counts for less than a fresh one, and we measure how much less.



Hypothesis: repeated data still teaches an MLM something

A masked model hides a random 15% of the residues and predicts them from the rest, re-drawing **which** residues to hide on every pass. So we expect a masked model to tolerate far more passes over one corpus than a causal model can, and to reach the same error with a smaller model.

The fitting law

Following **Muennighoff et al.**[5] we ran the experiments in the next two columns to fit this surface. **L is the eval loss**, cross-entropy on held-out antibodies the model never trained on, so it scores the test task and lower is better.

$$L(N, U, R) = E + \frac{A}{N^{\alpha}} + \frac{B}{D_{\text{eff}}^{\beta}}$$

$$D_{\text{eff}} = U\left[1 + R_d^{\star}\left(1 - \exp\left(\frac{-R-1}{R_d^{\star}}\right)\right)\right]$$

L	= eval loss, lower is better	E	= the floor no model beats
N	= how big the model is	A, α	= how fast error falls with size
U	= unique data we hold, fixed	B, β	= how fast it falls with data
R	= how many passes over it	$R_d^{\star}$	= how fast a repeat loses value

Step 1: muP + complete-P makes the sweep affordable

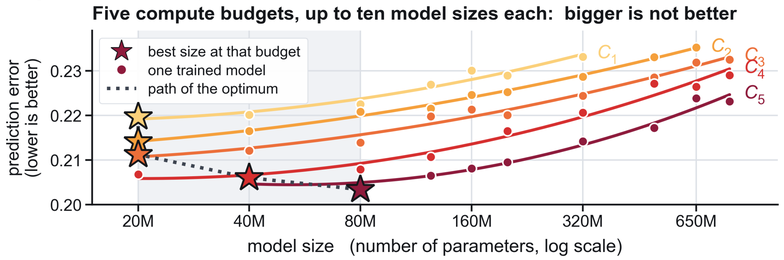
Setting. Tuning a learning rate at every size would make this ladder unaffordable. **muP**[9] **+ complete-P**[2] derives each tensor's scaling from width and depth, so one base rate serves every size: we swept base learning rates once on a 124M model and transferred it across the 20M to 800M ladder.

transfer score (lower is better)	α_d = 1.0	α_d = 0.5	no muP
across width	0.138	**0.091**	**0.55**
across depth	**0.275**	**0.091**	**0.30**
width + depth + heads at once	not run	**0.075-0.120**	**1.024**

Result. Coordinate checks pass at every cell of the width x depth matrix, **0.075 to 0.120** against a 0.20 bar. α_d = 1.0 breaks depth transfer (0.275); **α_d = 0.5** holds it (0.091).

Step 2: the isoFLOP ladder locates the allocation

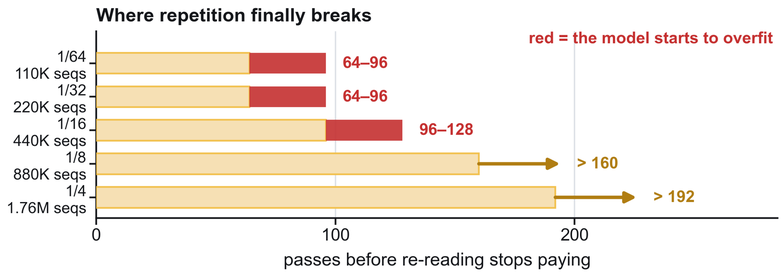
Setting. The corpus is held at U_0 = 7,044,237 sequences ≈ 2.254×10^9 tokens. Each budget C_k fixes the total tokens D, and we sweep N from 20M to 800M inside it. The corpus never grows, so the pass count R = D / U_0 is locked by the budget: the ladder is purely an allocation experiment.



Result. Every budget traces a bowl and the best model stays small, 20M to 80M over a 6.5× compute range. At the top of the ladder that is 80M, against 127M for the masked free-data rule[1] and 533M for the causal repetition rule[5].

Step 3: the repetition arm prices a repeated pass

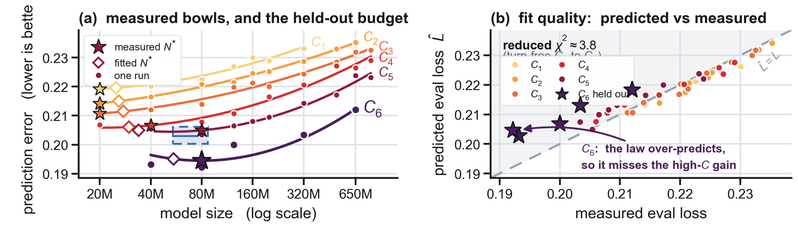
Setting. A **fresh arm** at R = 1 varies the unique subset U at 40M and 124M, which pins B and β. A **repetition grid** then sweeps the pass count R at five fractions of the corpus, pushing R as deep as 256 so that $R_d^{\star}$ separates from B.



Result. All six parameters are fitted jointly by a robust, σ-weighted least squares from a grid of starts, dropping runs whose loss rises with epochs, with 95% intervals from 300 bootstrap resamples.

Held-out test at C_6

Setting. At C_6 = 8.7×10^{19} FLOPs, 1.6× past the fitting range, the fitted surface predicts a compute-optimal size of **54M to 87M** at an eval loss of **0.2029** (interval 0.2003 to 0.2062). We fixed that call, then ran the ladder.

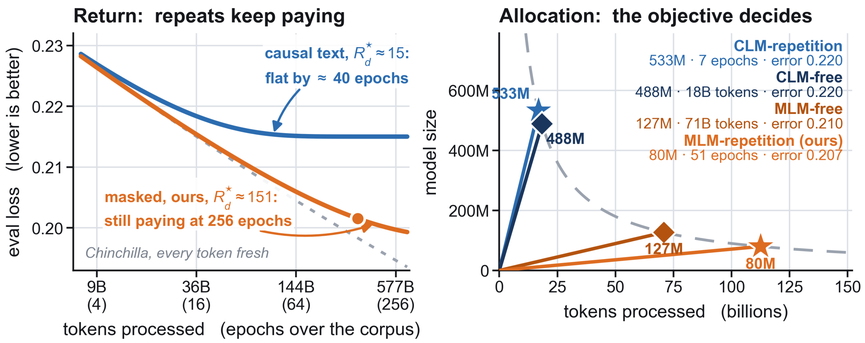


Results. The C_6 bowl bottoms at **80M** (eval loss **0.1922**), inside the predicted [54, 87]M interval and below the predicted loss, so the fit runs conservative at high compute; measured beating predicted is itself a robustness result. The loss lands 0.0107 under the point prediction, 0.0081 under the interval floor.

Conclusion

- **Infrastructure works.** muP + complete-P (α_d = 0.5) transfers one learning rate across width and depth, over the whole 20M to 800M ladder.
- **Compute-optimal models are small.** Smaller models, more epochs. Extra compute buys epochs of a small model until they approach the repetition ceiling, and only then grows the model.
- **Repeated data is nearly free, until it isn't.** Fitted $R_d^{\star} \approx 151$ for masked antibody LMs on our fixed 7.04M-sequence corpus, over budgets C_1 to C_5 inside the turn-free R ≤ 118 domain.
- **Masking is why repeats keep paying.** Re-sampling the mask each epoch gives different supervised positions every pass, so a repeat is only partial: $R_d^{\star} \approx 151$ for masked models against ≈ 15 for causal ones[5].
- **Data-constrained scaling is sharper than Chinchilla.** With the corpus dead at $U_0 \approx 2.25\times10^9$ tokens and repeat value saturating, effective data has a hard ceiling: D_eff^max ≈ $U_0(1 + R_d^{\star})$ ≈ **343 billion tokens**, about 1.07 billion fresh-equivalent sequences.

Main Takeaways



Left: what a repeated pass is worth. **Right:** where one fixed budget of 5.4×10^{19} FLOPs lands under four recipes.

Scripps Research

Setup. 7,044,237 unique paired human antibody sequences (Observed Antibody Space plus in-house healthy-donor repertoires) clustered at 99% identity; the 50,000 evaluation sequences were split off at 80% identity first, so nothing held out is a near-duplicate of anything trained on. Error is held-out masked-LM cross-entropy at 15% masking, length 320. **Limits.** Several low-budget bowls bottom at our smallest trained size, so those optima are upper bounds. On a single corpus the six surface parameters are not individually identifiable, so we report the derived frontier and the measured bowl minima, not parameter point values, and where fit and measurement disagree the measurement is what we report. Because the data term enters as D_eff^(−β) with β = 0.05, any corpus a target implies scales as the twentieth power of the loss gap, so we give the direction, not a number. **Contact.** briney@scripps.edu, jjian@scripps.edu

WORKS CITED

1. Cheng, Xingyi, et al. "Training Compute-Optimal Protein Language Models." *Advances in Neural Information Processing Systems*, 2024.
2. Dey, Nolan, et al. "Don't Be Lazy: CompleteP Enables Compute-Efficient Deep Transformers." *Advances in Neural Information Processing Systems*, 2025.
3. Hoffmann, Jordan, et al. "Training Compute-Optimal Large Language Models." *arXiv*, arXiv:2203.15556, 2022.
4. Kaplan, Jared, et al. "Scaling Laws for Neural Language Models." *arXiv*, arXiv:2001.08361, 2020.
5. Muennighoff, Niklas, et al. "Scaling Data-Constrained Language Models." *arXiv*, arXiv:2305.16264, 2023.
6. Neyestanak, Mahdi Shafiei, et al. "Data-Optimal Scaling of Paired Antibody Language Models." *bioRxiv*, 2025.09.02.673765, 2025.
7. Olsen, Tobias H., et al. "Observed Antibody Space: A Diverse Database of Cleaned, Annotated, and Translated Unpaired and Paired Antibody Sequences." *Protein Science*, vol. 31, no. 1, 2022, pp. 141–46.
8. Qin, Tian, et al. "Bridging Compute- and Data-Optimal Pretraining." *arXiv*, arXiv:2607.25271, 2026.
9. Yang, Greg, et al. "Tensor Programs V: Tuning Large Neural Networks via Zero-Shot Hyperparameter Transfer." *arXiv*, arXiv:2203.03466, 2022.