Language model scaling laws and zero-sum learning

Andrei Mircea[1,2] * **Nima Chitsazan**[2] **Supriyo Chakraborty**[2] **Renkun Ni**[2]

Austin Zhang[2] **Ekaterina Lobacheva**[1] **Irina Rish**[1]

[1]**University of Montreal / Mila,** [2]**Capital One**

Abstract

This work aims to understand how, in terms of training dynamics, scaling up language model size yields predictable loss improvements. We find that these improvements can be tied back to *loss deceleration*, an abrupt transition in the rate of loss improvement, characterized by piece-wise linear behavior in log-log space. Notably, improvements from increased model size appear to be a result of (1) improving the loss at which this transition occurs; and (2) improving the rate of loss improvement after this transition. As an explanation for the mechanism underlying this transition (and the effect of model size on loss it mediates), we propose the zero-sum learning (ZSL) hypothesis. In ZSL, per-token gradients become systematically opposed, leading to degenerate training dynamics where the model can't improve loss on one token without harming it on another; bottlenecking the overall rate at which loss can improve. We find compelling evidence of ZSL, as well as unexpected results which shed light on other factors contributing to ZSL.

1 Explaining scaling laws in terms of training dynamics

Increasing language model size empirically improves cross-entropy loss with power-law scaling, accurately extrapolated across several orders of magnitude with scaling laws [KMH+20]. Despite their predictive capabilities, scaling laws offer limited explanatory power as to the underlying mechanism [SP12]; i.e. they do not explain *how* scaling improves loss. This question is of particular interest because, by identifying and understanding such a mechanism, we may be able to target it directly to improve models independent of scale, or achieve better improvements from scaling.

While several recent works have sought to explain scaling laws (e.g. in terms of asymptotic behavior [BDK+24] or data distribution properties [MLGT23]), these are typically based on some notion of intrinsic model capacity. In contrast, our work attempts to explain scaling in terms of its effect on training dynamics, which could in turn be targeted directly and independent of scale. First, we find that the effect of scaling on loss can be quantified in terms of its effect on *loss deceleration*, a measurable transition early in training where loss improvements abruptly slow down (Section 2). This leads us to propose the *zero-sum learning* (ZSL) hypothesis (Section 3) as an explanation for loss deceleration and how it mediates, in terms of training dynamics, the effect of scaling on loss. We test and validate ZSL against alternate hypotheses, based on the original setup of [KMH+20], shedding light on and finding promising preliminary evidence of ZSL. In the interest of conserving space, we use the Appendix for background (A), methods (B), and additional results (C) sections.

*Contribution during an internship at Capital One. ⟨mirceara@mila.quebec⟩

Workshop on Scientific Methods for Understanding Deep Learning, NeurIPS 2024.

2 Loss deceleration in language models

In Fig. 1, we observe that LM loss curves exhibit an abrupt slow down in the rate of loss improvement early during training. This transition, which we refer to as *loss deceleration*, is characterized by piecewise linear behavior in log-log space, and can be parametrically described with smoothly broken power laws such as BNSL [CGRK23]:

$$L(t) - a = \left(bt^{-c_0}\right)\left(1 + (t/d_1)^{1/f_1}\right)^{-c_1 f_1} \quad (1)$$

Crucially, by fitting a one-break BNSL (Eqn. 1), we can measure and characterize deceleration with quantities from which an estimate $\hat{L}_T$ of the final loss can be recovered:

t_d : d_1, the step at which deceleration occurs.

L_d : $bd_1^{-c_0}$, the loss at which deceleration occurs.

r_d : $c_0 + c_1$, the log-log loss slope after deceleration.

$\hat{L}_T$: $\log(L_T) \approx \log(\hat{L}_T) = \log(L_d) - r_d \log(T/t_d)$

In Appendix C.1, we show improvements in L_T from scaling can be largely attributable to improvements in L_d and r_d, suggesting that scaling improves performance by mitigating deceleration, i.e. reaching a better loss L_d before deceleration, and converging to a better log-log rate of loss improvement r_d after deceleration. Therefore, to understand *how* scaling improves performance, we instead aim to understand the mechanism that underlies loss deceleration and the mitigating effect of scale on deceleration.

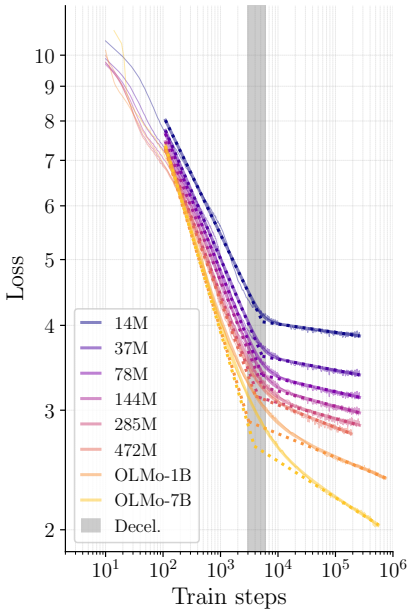


Fig. 1: Loss across model sizes and train steps, with BNSL fit of loss deceleration. We also include the OLMo 1B and 7B models from [GBW+24] to confirm loss deceleration occurs measurably at larger scales.

3 The zero-sum learning hypothesis

In this section, we present the zero-sum learning (ZSL) hypothesis as a mechanism for (1) how loss deceleration arises and (2) how scaling improves loss by mitigating deceleration. We formalize this hypothesis with three claims that we evaluate against alternative explanations: loss deceleration is caused by ZSL (3.1); ZSL is caused by opposing gradients (3.2); and scale mitigates gradient opposition and ZSL (3.3). If true, these claims imply that improvements from scaling model size are at least in part due to a mitigating effect on gradient opposition and ZSL. Crucially, gradient opposition and ZSL can potentially be targeted directly and independent of scale to improve language models. As a first step in this direction, we conduct a series of experiments that corroborate the ZSL hypothesis and shed light on the interplay of scaling and learning dynamics in language models.

> **Zero-sum learning:** degenerate training dynamics where loss improvements in one set of examples are offset by a similar deterioration of loss in another set of examples, bottlenecking the rate at which overall loss can improve with additional training steps.

3.1 Claim #1: Loss deceleration is caused by ZSL

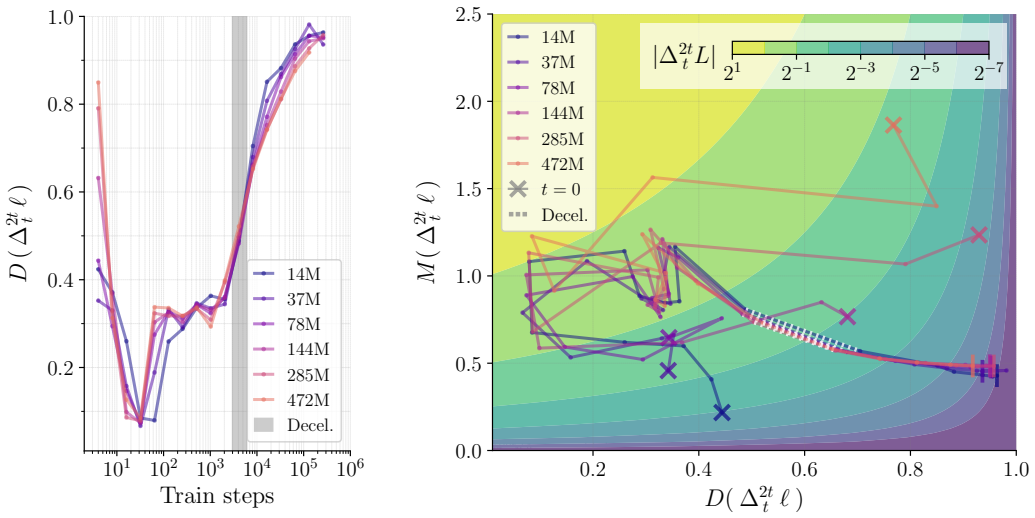
During loss deceleration, the change in loss $\Delta_{t_1}^{t_2} L$ between steps t_1, t_2 abruptly decreases in magnitude. For a dataset $\mathcal{D}$ with per-example losses ℓ_i, we have $\Delta_{t_1}^{t_2} L = \sum_i \Delta_{t_1}^{t_2} \ell_i / |\mathcal{D}|$ such that deceleration can occur for two possible reasons: (1) per-example changes in loss $\Delta_{t_1}^{t_2} \ell_i$ can increasingly cancel one another out (i.e. ZSL); or (2) $\Delta_{t_1}^{t_2} \ell_i$ can shrink in magnitude across examples. To quantify these, we define general metrics that, given a set of measurements x over samples i, $x = [x_1, \ldots, x_N]$, measure destructive interference $D(x)$ between samples, and average magnitude $M(x)$ across samples (Eqns.2 and 3). Importantly, we can express the absolute change in loss $|\Delta L|$ in terms of these two quantities with Eqn. 4, allowing us to disentangle the effects of destructive interference $D(\Delta\ell)$ on loss deceleration, relative to magnitude $M(\Delta\ell)$.

$$D(x) = 1 - |\textstyle\sum_i x_i| \,/\, \sum_i |x_i| \quad (2)$$

$$M(x) = \textstyle\sum_i^N |x_i| \,/\, N \quad (3)$$

$$|\Delta L| = M(\Delta\ell)(1 - D(\Delta\ell)) \quad (4)$$

We first measure[2] $D(\Delta_t^{2t}\ell)$ throughout training and find it rises rapidly during deceleration (Fig. 2a), consistent with our hypothesis. Furthermore, in Fig. 2b we plot model training trajectories with respect to $D(\Delta_t^{2t}\ell)$, $M(\Delta_t^{2t}\ell)$ and Eqn. 4, to quantify the relative contribution of ZSL to deceleration. Notably, we see that during and after deceleration, reductions in $|\Delta_t^{2t}L|$ are largely attributable to increases in D rather than M. Concretely, we know from Eqn. 4 that the observed reduction in M during deceleration, from 0.75 to 0.5, corresponds to a 1.5x reduction in $|\Delta_t^{2t}L|$. In contrast, the increase in D observed in that same period, from 0.5 to 0.95, corresponds to a 10x reduction in loss improvements. More generally, we see that as D increases and approaches 1.0, the required increase in M to maintain $|\Delta_t^{2t}L|$ explodes such that ZSL effectively bottlenecks loss improvements. These results corroborate that ZSL (rather than the alternate hypothesis of magnitude) explain loss deceleration, and in the next section we aim to understand and explain the underlying cause of ZSL.



(a) ZSL arises with loss deceleration **(b)** Reductions in loss improvements are primarily driven by ZSL

Fig. 2: ZSL results in loss deceleration by bottlenecking the rate at which loss can improve

3.2 Claim #2: ZSL is caused by opposing gradients

There exist several potential explanations for ZSL, i.e. for why loss improves for one set of examples but degrades for another. Given a small enough weight update $\Delta\theta$, changes in overall and per-token losses are approximable by first-order Taylor expansions (Eqn. 5) such that destructive interference and ZSL can arise if $\Delta\theta \cdot \nabla_\theta \ell_i$ is positive for some examples and negative for others (Eqn. 6).

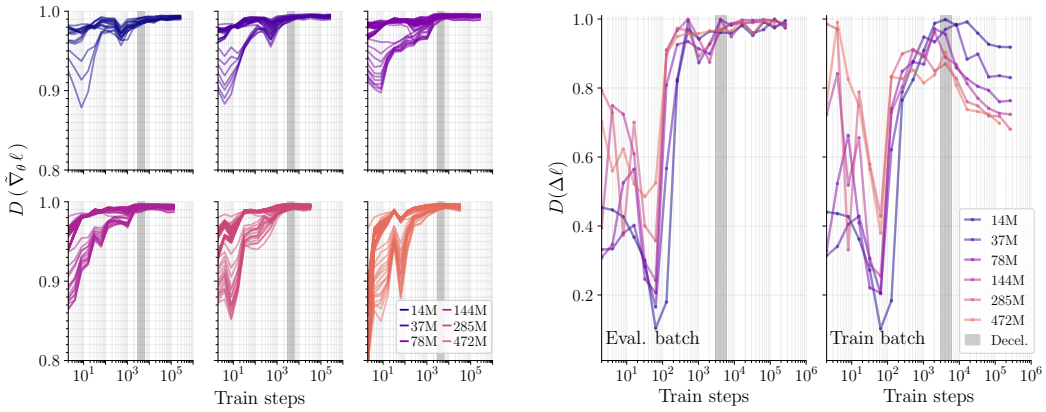
$$\Delta L \approx \tilde{\Delta} L \coloneqq \Delta\theta \cdot \nabla_\theta L = \textstyle\sum_i \tilde{\Delta}\ell_i\,, \qquad \Delta\ell_i \approx \tilde{\Delta}\ell_i \coloneqq \Delta\theta \cdot \nabla_\theta \ell_i \tag{5}$$

$$D(\tilde{\Delta}\ell) = D(\Delta\theta \cdot \nabla_\theta \ell) = 1 - |\textstyle\sum_i \Delta\theta \cdot \nabla_\theta \ell_i| \,/\, \sum_i |\Delta\theta \cdot \nabla_\theta \ell_i| \tag{6}$$

Alternatively, if this approximation does not hold, ZSL might be caused by progressive sharpening [CKL$^+$22, RR23] where $\Delta\theta$ overshoots local minima for some examples but not others. While these explanations are not mutually exclusive, we hypothesize that the principal cause of ZSL during loss deceleration is *systematic* gradient opposition. Concretely, we take this to mean near-complete destructive interference in gradients where $D(\nabla_\theta \ell) \approx 1$ across parameters.

First, we verify our hypothesis under the assumption that Eqn. 5 is a valid approximation. We show in Appendix C.4 that $D(\nabla_\theta \ell) \approx 1$ demonstrably results in $D(\tilde{\Delta}\ell) \approx 1$. In other words, for an optimizer step $\Delta\theta$, systematic gradient opposition fundamentally results in ZSL for $\tilde{\Delta}\ell$. Consistent with our hypothesis, we find in Fig. 3a that systematic gradient opposition occurs with loss deceleration across model sizes, with $D(\tilde{\nabla}_\theta \ell) \approx 1$ across non-embedding parameter tensors (where $D(\tilde{\nabla}_\theta \ell)$ is a tractable proxy approximating $D(\nabla_\theta \ell)$ as described in Appendix B.4).

[2]We checkpoint models every 2^x steps and use Δ_t^{2t} in line with deceleration being observed in log-space.



(a) Systematic gradient opposition occurs with decel. **(b)** Single-step ZSL occurs with gradient opposition

Fig. 3: Increases in gradient opposition correlate increases in single-step ZSL and loss deceleration

In Fig. 3b, we measure actual destructive interference $D(\Delta\ell)$ after an optimizer step, for both `Train` batches used to compute the optimizer step (right), and separate `Eval` batches (left)[3]. Consistent with our hypothesis, we observe a close correspondence between increases in gradient opposition in Fig. 3a and increases in single-step ZSL in Fig. 3b. We also find important qualitative differences between `Train` batches (for which single-step ZSL converges with deceleration before decreasing), and `Eval` batches (for which single-step ZSL converges well before deceleration and does not decrease). The underlying cause of this difference is not clear, but likely relates to increased alignment between $\Delta\theta$ and gradients of the samples used in computing $\Delta\theta$. Furthermore, the discrepancies with multi-step behavior observed in Fig. 2a suggest that interactions across multiple batches and updates play an important role in loss deceleration, not accounted for by local first-order gradient information.

Second, we assess the validity of the first-order Taylor approximation underlying our hypothesis. Our hypothesis relies on the assumption that destructive interference in $\tilde{\Delta}\ell$ and Eqn. 5 is reflective of destructive interference in $\Delta\ell$ and Fig. 3b; causally linking gradient opposition to ZSL from first principles. To verify this assumption, in Fig. 4, we measure and plot the Pearson correlation coefficient between actual and approximate changes in loss $\Delta\ell, \tilde{\Delta}\ell$ throughout training. However, computing $\nabla_\theta \ell_i$ and the corresponding $\tilde{\Delta}\ell_i = \Delta\theta \cdot \nabla_\theta \ell_i$ for each token is intractable, so we empirically measure $\tilde{\Delta}\ell_i$ using a linearization of loss landscape cross-sections as shown in Fig. 7 and described in Appendix B.3. Consistent with our hypothesis, we find strong correlation between $\Delta\ell$ and $\tilde{\Delta}\ell$, particularly after deceleration. In Appendix C.5, we plot a sample of 1000 per-token loss landscapes across train steps and model sizes, finding that they are generally linear in the vicinity of weight updates. Taken together, these results support our hypothesis that gradient opposition and first-order training dynamics account for single-step ZSL.

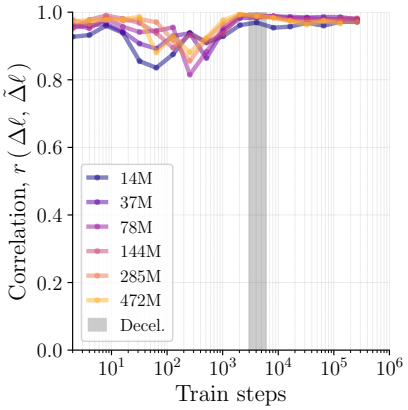


Fig. 4: Correlation between actual and approximated changes in loss during training corroborate validity of Eqn. 5 approximation.

Third, we rule out the alternate hypothesis that ZSL occurs because of progressive sharpening. In addition to Appendix C.5 which suggests that first order rather than second order gradient information dominates per-sample loss landscapes and is responsible for ZSL, we rule out the occurrence of progressive sharpening in the overall loss as a contributing factor. In Appendix C.6 we find that progressive sharpening does not occur with deceleration. Surprisingly, and perhaps counter to conventional wisdom, we find that loss landscapes instead become significantly flatter with deceleration; following an initial phase of high sharpness before deceleration.

[3]Note that unless otherwise stated, results throughout the paper are based on `Eval` batches.

3.3 Claim #3: Scale mitigates gradient opposition and ZSL

In 3.1 and 3.2, we presented evidence across model scales that loss deceleration is a result of ZSL, and that ZSL is a result of systematic gradient opposition. In this section, we re-examine our results to characterize the effects of model size and present speculative explanations for how scaling improves performance by mitigating ZSL and gradient opposition. At a high level, we hypothesize that increasing the number of parameters increases the degrees of freedom in which per-token gradients can co-exist without systematic opposition, resulting in (1) greater improvements before deceleration (and corresponding improvements in deceleration loss L_d); and (2) reduced ZSL after deceleration (and corresponding improvements in post-deceleration log-log rates of loss improvement r_d).

Scaling reduces gradient opposition before loss deceleration In Fig. 3a, we observe that, leading up to deceleration, increasing model size results in reduced gradient opposition across more layers. However, during this period, destructive interference is typically similar or worse with increased model size (Figs. 2a, 3b). Improvements in loss before deceleration seem instead to be primarily attributable to the magnitude of per-token loss improvements $M(\Delta\ell)$ in the first 100 steps (Fig. 5). In Appendix C.5 and C.6, we can see this is likely a result of steeper loss landscapes along weight updates, leading to greater loss improvements per step despite smaller stepsizes in larger models. $\tilde{\Delta}\ell_i = \|\Delta\theta\|\|\nabla_\theta\ell_i\|\cos(\Delta\theta, \nabla_\theta\ell_i)$ such that $\tilde{\Delta}\ell_i$ is a function of update-gradient norms $\|\Delta\theta\|\|\nabla_\theta\ell_i\|$, and update-gradient alignments $\cos(\Delta\theta, \nabla_\theta\ell_i)$. In Fig. 6, we find tentative evidence that reduced gradient opposition increases early-stage loss improvements by increasing update-gradient alignments.

Scaling reduces ZSL and mitigates gradient opposition after loss deceleration In Fig. 2 we observe that post-deceleration loss improvements are bottlenecked by ZSL (as measured by $D(\Delta_t^{2t}\ell)$), and that increased model sizes consistently reduce $D(\Delta_t^{2t}\ell)$ after deceleration. The exact reason for this reduction remains unclear, however our findings suggest an underlying relation to mitigated gradient opposition. Even after deceleration, where all models converge to $D(\tilde{\Delta}\ell) \approx 1$, larger models with more parameters intrinsically have more degrees of freedom for a similar level of destructive interference, in effect mitigating systematic gradient opposition. 99.9% gradient destructive interference in a 1B parameter model is not equivalent to the same value in a 10M model. This may explain why, in Fig. 3, `Train` batch samples exhibit reduced destructive interference in loss improvements when increasing model size. Furthermore, in Appendix C.5 and Appendix C.6, we see that increasing model size results in flatter loss landscapes after loss deceleration, which may also mitigate the effect of gradient opposition by e.g. reducing oscillations between training steps.

While we find evidence that increasing model size reduces gradient opposition as well as single and multi-step ZSL, the connection between these cannot be fully explained by our hypothesis and results. In particular, our findings suggest that interactions across multiple updates and batches play an important role that must be clarified in order to obtain a more precise understanding of loss deceleration and how it is mitigated by scaling model size.

4 Conclusion and outlook

Our results corroborate that ZSL and gradient opposition occur with and *can* explain loss deceleration. Notably, we show the causal relationship between these is mechanistically plausible based on first principles and empirical measurements. Furthermore, we find evidence that the effect of model size on loss is indeed mediated by these factors, although our results suggest several non-trivial interactions with training dynamics beyond gradient opposition that need to be understood to more comprehensively explain loss deceleration. We believe our hypothesis remains to be more rigorously tested with targeted interventions, especially considering our original motivation of improving loss by directly mitigating gradient opposition and ZSL. More generally, the extent to which loss deceleration can be a function of training dynamics independent of model scale, while clearly upper-bounded, also remains to be more rigorously characterized. Beyond ZSL and loss deceleration, we believe the interplay between alignment, orthogonality and opposition of per-token gradients has until now been underexplored and can provide important insights into learning dynamics, scaling and generalization.

References

[AZL23] Zeyuan Allen-Zhu and Yuanzhi Li. Towards Understanding Ensemble, Knowledge Distillation and Self-Distillation in Deep Learning, February 2023. arXiv:2012.09816 [cs, math, stat].

[AZVP24] Alexander Atanasov, Jacob A. Zavatone-Veth, and Cengiz Pehlevan. Scaling and renormalization in high-dimensional regression, June 2024. arXiv:2405.00592 [cond-mat, stat].

[BDK+24] Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining Neural Scaling Laws. *Proceedings of the National Academy of Sciences*, 121(27):e2311878121, July 2024. arXiv:2102.06701 [cond-mat, stat].

[CGRK23] Ethan Caballero, Kshitij Gupta, Irina Rish, and David Krueger. Broken Neural Scaling Laws, July 2023. arXiv:2210.14891 [cs].

[CKL+22] Jeremy M. Cohen, Simran Kaur, Yuanzhi Li, J. Zico Kolter, and Ameet Talwalkar. Gradient Descent on Neural Networks Typically Occurs at the Edge of Stability, November 2022. arXiv:2103.00065 [cs, stat].

[EXW+24] Katie Everett, Lechao Xiao, Mitchell Wortsman, Alexander A. Alemi, Roman Novak, Peter J. Liu, Izzeddin Gur, Jascha Sohl-Dickstein, Leslie Pack Kaelbling, Jaehoon Lee, and Jeffrey Pennington. Scaling Exponents Across Parameterizations and Optimizers, July 2024. arXiv:2407.05872 [cs] version: 1.

[GBW+24] Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. OLMo: Accelerating the Science of Language Models, June 2024. arXiv:2402.00838.

[HBK+24] Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. Scaling Laws and Compute-Optimal Training Beyond Fixed Training Durations. July 2024.

[HBM+22] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training Compute-Optimal Large Language Models, March 2022. arXiv:2203.15556 [cs].

[Hut21] Marcus Hutter. Learning Curve Theory, February 2021. arXiv:2102.04074 [cs, stat].

[KMH+20] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling Laws for Neural Language Models, January 2020. arXiv:2001.08361 [cs, stat].

[LLJ+21] Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-Averse Gradient Descent for Multi-task learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 18878–18890. Curran Associates, Inc., 2021.

[MBH+23] Ian Magnusson, Akshita Bhagia, Valentin Hofmann, Luca Soldaini, Ananya Harsh Jha, Oyvind Tafjord, Dustin Schwenk, Evan Pete Walsh, Yanai Elazar, Kyle Lo, Dirk Groeneveld, Iz Beltagy, Hannaneh Hajishirzi, Noah A. Smith, Kyle Richardson, and Jesse Dodge. Paloma: A Benchmark for Evaluating Language Model Fit, December 2023. arXiv:2312.10523.

[MLGT23] Eric J. Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The Quantization Model of Neural Scaling, March 2023. arXiv:2303.13506 [cond-mat].

[MLR24] Andrei Mircea, Ekaterina Lobacheva, and Irina Rish. Gradient Dissent in Language Model Training and Saturation. June 2024.

[MUP+23] Max Marion, Ahmet Ustun, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. When Less is More: Investigating Data Pruning for Pretraining LLMs at Scale, September 2023. arXiv:2309.04564 [cs].

[PNO+20] Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. Learning explanations that are hard to vary. October 2020.

[RR23] Elan Rosenfeld and Andrej Risteski. Outliers with Opposing Signals Have an Outsized Effect on Neural Network Optimization, November 2023. arXiv:2311.04163 [cs, stat].

[RRR+24] David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-Depths: Dynamically allocating compute in transformer-based language models, April 2024. arXiv:2404.02258 [cs].

[SGS+22] Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 19523–19536. Curran Associates, Inc., 2022.

[SK20] Utkarsh Sharma and Jared Kaplan. A Neural Scaling Law from the Dimension of the Data Manifold, April 2020. arXiv:2004.10802 [cs, stat].

[SP12] Michael P. H. Stumpf and Mason A. Porter. Critical Truths About Power Laws. *Science*, 335(6069):665–666, February 2012. Publisher: American Association for the Advancement of Science.

[TRP+24] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy

Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving Open Language Models at a Practical Size, August 2024. arXiv:2408.00118 [cs].

[TWW24] Howe Tissue, Venus Wang, and Lu Wang. Scaling Law with Learning Rate Annealing, August 2024. arXiv:2408.11029 [cs].

[VL22] Tom Viering and Marco Loog. The Shape of Learning Curves: a Review, November 2022. arXiv:2103.10948 [cs].

[YKG+20] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient Surgery for Multi-Task Learning, December 2020. arXiv:2001.06782 [cs, stat].

[YO20] Yuki Yoshida and Masato Okada. Data-Dependence of Plateau Phenomenon in Learning with Neural Network — Statistical Mechanical Analysis. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124013, December 2020. arXiv:2001.03371 [cs, stat].

[YSS+21] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. Opacus: User-Friendly Differential Privacy Library in PyTorch, September 2021.

A Related Works

Explaining scaling laws Several works have proposed different explanations for neural scaling laws such as [KMH+20, HBM+22, CGRK23, HBK+24, TWW24, EXW+24]. Notably, [BDK+24] explain scaling laws in terms of asymptotic behavior, identifying variance-limited regimes based on concentration around infinite limits, and resolution-limited regimes based on distances between train and test data points on their manifold (see also [SK20]). [AZVP24] analytically explain power-law scaling in high-dimensional ridge regression with tools from random matrix theory. [MLGT23] propose a "quantization model of neural scaling", whereby power law scaling is a result of (1) language models improving loss by learning discrete capabilities from their demonstration in data, (2) larger models being able to learn more capabilities, and (3) rarer capabilities improve loss by smaller and smaller amounts due to their vanishing frequency. Similarly, [Hut21] show how power law scaling with data can arise from long-tail feature distributions.

Improving language models independently of scaling Recent work on e.g. data pruning [MUP+23, SGS+22] model distillation [AZL23, TRP+24] and model pruning [RRR+24] show that improvements predicted from scaling can (up to a point) be realized without scaling. This suggests that scaling may indirectly improve loss by its effect on training dynamics, and that similar effects/improvements can be obtained without necessarily scaling.

Gradient opposition From the perspective of training dynamics, [RR23] discuss the effect of outlier samples with opposing gradients. In the context of multi-task learning, several works have proposed approaches to mitigate gradient opposition between tasks, e.g. [PNO+20, YKG+20, LLJ+21]. Gradient opposition between tokens in language modeling has, to the best of our knowledge, not been characterized. Related but distinct, is the work of [MLR24] characterizes opposition within token gradients rather than between.

Loss deceleration and learning curves To the best of our knowledge, the loss deceleration transition we identify and characterize in this work has not been previously established or explained. We refer the reader to [VL22] for a comprehensive review of learning curve shapes, as well as [Hut21] and [YO20] as examples of attempting to explain features in a learning curve.

B Methodology

B.1 Language model pretraining

We train variants of OLMo with the same training data as OLMo-7B-0724 [GBW+24]. Model dimensions and learning rates are based on [KMH+20] and shown in Table 1, labeled with (rounded) total parameter counts. For pretraining, we again adapt the experimental setup of [KMH+20], training with a batch size of 0.5M tokens for 2^{18} steps. However, instead of a cosine learning rate decay, we adopt the trapezoidal learning rate from [HBK+24] with a learning rate warmup to the values in Table 1 in the first $2{,}000$ steps and no cooldown in the 2^{18} steps considered. Note that the OLMo-1B and OLMo-7B models are those trained by [GBW+24] and could not included in our analysis of ZSL because of insufficient checkpointing frequency before deceleration.

Table 1: Model and Optimizer Parameters for Different Runs

Model size	**14M**	**37M**	**78M**	**144M**	**285M**	**472M**	**OLMo-1B**	**OLMo-7B**
`d_model`	256	512	768	1024	1536	2048	2048	4096
`mlp_dim`	256	512	768	1024	1536	2048	16384	22016
`n_heads`	4	8	12	16	16	16	16	32
`n_layers`	4	8	12	16	16	16	16	32
`peak_lr`	1.3E-3	9.7E-4	8.0E-4	6.8E-4	5.7E-4	4.9E-4	4.0E-4	3.0E-4

B.2 Analysis of ZSL and gradient opposition

During training, we checkpoint the model and optimizer every 2^i steps with $i \in [0, 18]$. Our analyses of ZSL and gradient opposition are done on these checkpoints after pretraining. Methodological details regarding e.g. precision or batch size are kept consistent with pretraining to obtain representative results. All of our evaluations are conducted on the C4 validation set from [MBH+23], using the `allenai_eleuther-ai-gpt-neox-20b-pii-special` tokenizer from [GBW+24], consistent with pretraining.

B.3 Measuring first-order Taylor approximation of per-token changes in loss

To empirically measure first-order Taylor approximations of loss changes, we compute 1D cross-sections of per-token loss landscapes (Fig. 7) by evaluating models along increments of a given weight update $\Delta\theta$, with $\theta(\alpha) = \theta + \alpha\Delta\theta/\|\Delta\theta\|$, $\alpha \in [-10, 10]$. This allows us to tractably measure $\tilde{\Delta}\ell$ as a linearization around $\alpha = 0$ where $\tilde{\Delta}\ell(\alpha) = \alpha\,(\ell_{\theta+\epsilon} - \ell_\theta/\|\epsilon\|)$.

B.4 Computing destructive interference in token-level gradients

Computing per-prediction gradients in transformer language models is intractable due to the combinatorial nature of self-attention. However, we can tractably compute per-token gradients along the hidden state dimension (similar to [YSS+21]) where for any module $\mathcal{M}(x) = y,\ \mathcal{M} : S \times D_1 \mapsto S \times D_2$ with sequence length S and hidden dimensions D_1, D_2; we define $\nabla_\theta \tilde{\ell}_i = \sum_{\mathcal{M}} (\delta L / \delta y_i)(\delta y_i / \delta \theta_{\mathcal{M}})$ for the i^{th} token in a sequence. In the PyTorch code block below, we illustrate how the backward pass of a linear layer with weights W is modified to compute gradient destructive interference across samples: $\frac{\delta L}{\delta W} = \sum_i \frac{\delta L}{\delta y_i} \frac{\delta y_i}{\delta W}$

```
import torch
from torch import nn, autograd, functional as F

def compute_gdi(W: nn.Parameter):
  gdi = 1 - W.sum_grads.abs()/W.sum_abs_grads
  return gdi.mean()


class GDILinearFunction(autograd.Function):
  @staticmethod
  def forward(ctx, x, W):
    ctx.save_for_backward(x, W)
    y = F.linear(x,W)
    return y

  @staticmethod
  def backward(ctx, dLdy):
    x, W = ctx.saved_tensors
    if ctx.needs_input_grad[1]:

      # instantiate metrics if not present
      if not hasattr(W, 'sum_grads'):
        W.sum_grads = torch.zeros_like(W)
        W.sum_abs_grads = torch.zeros_like(W)

      # accumulate sum of gradients
      W.sum_grads.add_(
        torch.einsum(
          'B...d,B...p->pd', x, dLdy
        )
      )
      # accumulate sum of absolute gradients
      W.sum_abs_grads.add_(
        torch.einsum(
          'B...d,B...p->pd', x.abs(), dLdy.abs()
        )
      )
    # compute and return input gradient for backprop
    if ctx.needs_input_grad[0]:
      dLdx = torch.einsum(
        'B...p,pd->B...d', dLdy, W
      )
    else:
      dLdx = None

    return dLdx, None
```

Code 1: Illustrative example computing gradient destructive interference in PyTorch

C Additional results

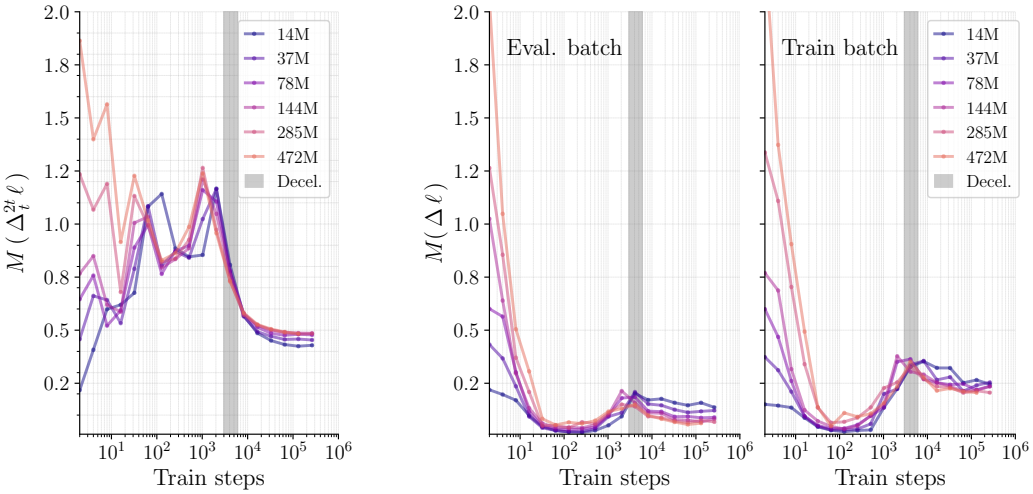
C.1 BNSL fit of loss deceleration

To fit a two-segment BNSL in Section 2, we adapt the methodology and code from [CGRK23]. Similar to [KMH+20], we do not fit data from the initial transient phase of training (i.e. the first 100 steps). Furthermore, we force a in the BNSL to be 0, which we found helpful for preventing instability and explosions in parameters. To validate the quality of the BNSL fit, we use root standard log error as in [CGRK23] and report it along with the fitted parameters in Table 2.

Table 2: Summary of loss deceleration quantities and root standard log error (RSLE) for BNSL fit.

Model	L_d	r_d	d_t	$\hat{L}_T/L_T$	RSLE
14M	4.05	0.012	5400	3.86/3.90	0.012
37M	3.60	0.016	5400	3.38/3.42	0.015
78M	3.38	0.019	5300	3.13/3.17	0.015
144M	3.25	0.023	5400	2.97/3.01	0.014
285M	3.13	0.023	4600	2.85/2.88	0.014
472M	3.15	0.033	4100	2.77/2.78	0.014
OLMo-1B	2.89	0.034	3100	2.39/2.38	0.008
OLMo-7B	2.66	0.054	3800	2.03/2.02	0.008

C.2 Magnitudes of loss improvements



(a) Multi-step magnitudes

(b) Single-step magnitudes

Fig. 5: Magnitude of loss improvements across training and model sizes, for multiple and single steps In both single and multi-step settings, magnitude is typically greater in larger models in the beginning of training. Deceleration also correlates with a decrease in magnitude in both settings, although this is more pronounced for multi-step. Magnitude is also systematically larger for samples in `Train` batches compared to `Eval` batches, but generally exhibit similar behavior in contrast to Fig. 3b.

C.3 Contributions of update-gradient norms and alignment to early loss improvements

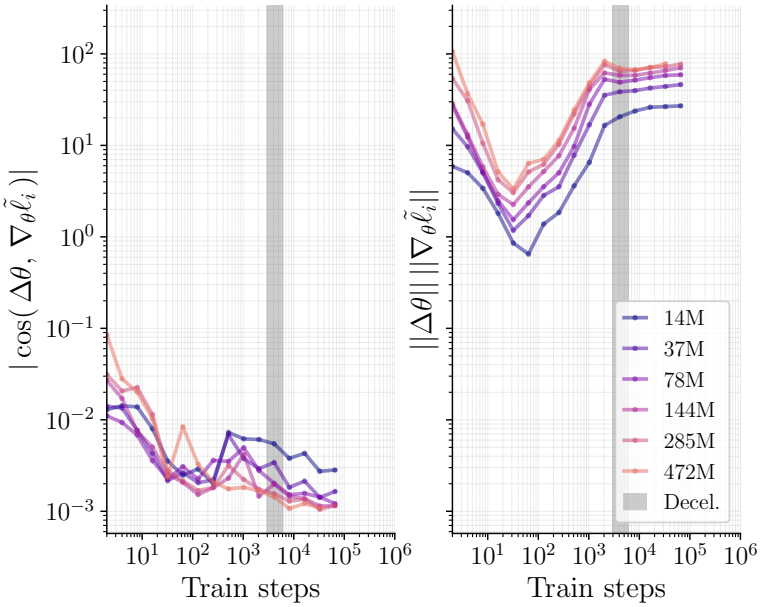


Fig. 6: Relative contributions of update-gradient alignment (left) and norm (right) to loss improvements. Per-token loss improvements are approximated by $\tilde{\Delta}\ell_i = \|\Delta\theta\|\|\nabla_\theta \ell_i\| \cos(\Delta\theta, \nabla_\theta \ell_i)$. Early-stage loss improvements with increasing model size appear attributable to both increased alignment and norms. Notably, relative improvements in norm remain constant throughout training, while improvements in alignment occur only in the early-stage phase associated with pre-deceleration loss improvements.

C.4 Decomposition of destructive interference

For a weight update vector $\mathbf{u}$ and for token-level gradient vectors $\mathbf{g}_i$ and overall gradient $\mathbf{G} = \sum_i \mathbf{g}_i$, the resulting change in loss can be approximated by a first-order Taylor expansion:

$$\tilde{\Delta}L = \mathbf{u} \cdot \mathbf{G} = \textstyle\sum_i \tilde{\Delta}\ell_i = \sum_i \mathbf{u} \cdot \mathbf{g}_i \tag{7}$$

Based on Eqn. 2, we can express destructive interference in this sum as follows:

$$D(\tilde{\Delta}\ell) = D(\mathbf{u} \cdot \mathbf{g}) = 1 - \frac{|\sum_i \mathbf{u} \cdot \mathbf{g}_i|}{\sum_i |\mathbf{u} \cdot \mathbf{g}_i|} = 1 - \frac{|\mathbf{u} \cdot \mathbf{G}|}{\sum_i |\mathbf{u} \cdot \mathbf{g}_i|} \tag{8}$$

Our goal is to isolate the effect of gradient destructive interference in Eqn. 8, as measured by $D(\mathbf{g}_j)$ across parameters θ_j, or $\vec{D}(\mathbf{g})$ in vector form. However, this is non-trivial because directions of high opposition in $\mathbf{g}$ may be orthogonal to the weight update vector $\mathbf{u}$ such that gradient opposition does not cause ZSL. Conversely, two gradients $\mathbf{g}_i$, $\mathbf{g}_k$ with no coordinate-wise destructive interference may result in ZSL if e.g. $\mathbf{u}$ is aligned with $\mathbf{g}_i - \mathbf{g}_k$. In other words, we need to disentangle ZSL due to *systematic* gradient opposition and destructive interference in the canonical basis; as opposed to potentially incidental gradient opposition along $\mathbf{u}$ resulting from suboptimal updates.

We will quantify these contributions as C_g for ZSL due to gradient opposition, and C_u for ZSL due to update directions. To do this we need to isolate $\vec{D}(\mathbf{g})$ and account for its alignment with respect to $\mathbf{u}$. First, note that the denominator $\sum_i |\mathbf{u} \cdot \mathbf{g}_i| = \sum_i \|\mathbf{u}\| \|\mathbf{g}_i\| \, |\cos(\mathbf{u}, \mathbf{g}_i)|$ in Eqn. 8 is independent of coordinate-level gradient opposition, while the contribution of destructive interference in gradients is captured by $\mathbf{G} = \pm\vec{M}(\mathbf{g})(1 - \vec{D}(\mathbf{g}))$, where for compactness we use $\pm\vec{M}(\mathbf{g})$ to denote $\mathrm{sign}(\mathbf{G})\vec{M}(\mathbf{g})$, allowing us to rewrite Eqn. 8 as:

$$\begin{aligned}
D(\tilde{\Delta}\ell) &= 1 - C_u + C_g && (9)\\
C_u &= \frac{\|\pm\vec{M}(\mathbf{g})\| \, |\cos(\mathbf{u}, \pm\vec{M}(\mathbf{g}))|}{\sum_i \|\mathbf{g}_i\| \, |\cos(\mathbf{u}, \mathbf{g}_i)|} && \in [0, 1]\\
C_g &= \frac{\|\pm\vec{M}(\mathbf{g})\vec{D}(\mathbf{g})\| \, |\cos(\mathbf{u}, \pm\vec{M}(\mathbf{g})\vec{D}(\mathbf{g}))|}{\sum_i \|\mathbf{g}_i\| \, |\cos(\mathbf{u}, \mathbf{g}_i)|} && \in [0, C_u]
\end{aligned}$$

Intuitively, C_u captures destructive interference from summing along the canonical basis coordinates when projecting gradients onto $\mathbf{u}$, assuming no coordinate-level gradient opposition (i.e. $\mathbf{G} = \pm\vec{M}(\mathbf{g})$). Note that if $\mathbf{g}_i$ all lie on the same line such that $\forall i$, $|\cos(\mathbf{u}, \mathbf{g}_i)| = |\cos(\mathbf{u}, \mathbf{G})| = |\cos(\mathbf{u}, \pm\vec{M}(\mathbf{g}))|$ and $\|\vec{M}(\mathbf{g})\| = \|\sum_i |\mathbf{g}_i|\| = \sum_i \|\mathbf{g}_i\|$, then we can see that C_u becomes 1, indicating no destructive interference. In contrast, C_g captures destructive interference from $\vec{D}(\mathbf{g})$, i.e. from summing along examples, while taking into account its alignment with $\mathbf{u}$. As a result, C_g is at most C_u, approaching this upper limit as $\vec{D}(\mathbf{g})$ approaches complete destructive interference along $\mathbf{u}$. However, because the values of $\vec{D}(\mathbf{g})$ are bounded between 0 and 1, in the case of complete destructive interference across all coordinates, as observed in Fig. 3a, the alignment with $\mathbf{u}$ becomes irrelevant as expected. In other words, $D(\nabla_\theta \ell) \approx 1$ will result in $D(\tilde{\Delta}\ell) \approx 1$ as described in 3.2.

C.5 Per-token loss landscape cross-sections

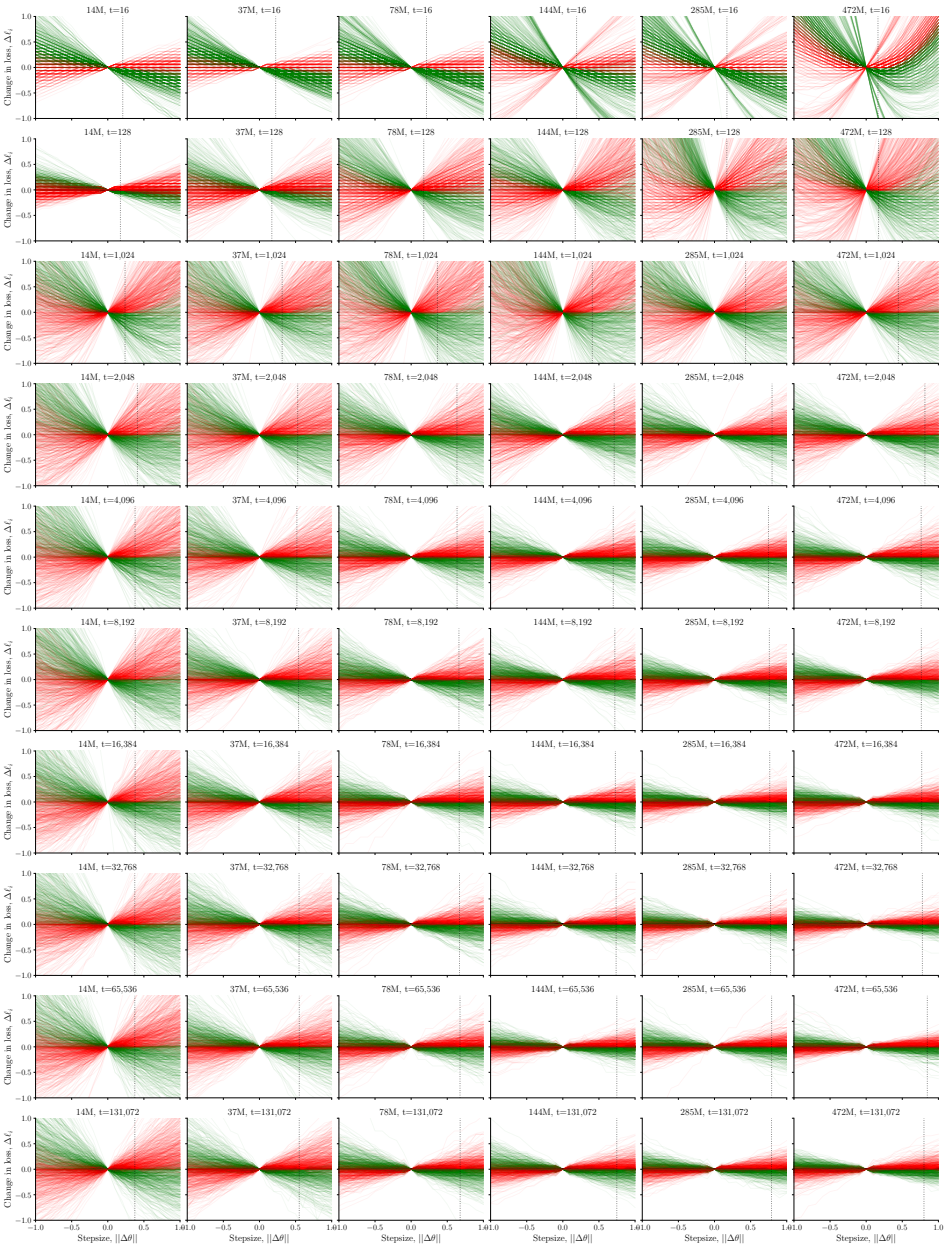


Fig. 7: Sampled per-token loss landscape cross-sections across model sizes and train steps
Across model sizes (columns) and train steps (rows), we randomly sample 1000 tokens and plot a cross-section of their loss landscape along the weight update $\Delta\theta$ for step t, at increments of 0.1. For consistent axes across model sizes and train steps, we plot ΔL rather than L, which has the same geometry but allows more easily distinguishing loss improvements from degradations. The point corresponding to the actual stepsize is indicated with a dotted vertical line. Lines are colored in green or red depending on whether the loss (respectively) improved or deteriorated at the actual stepsize. Visually, it appears that per-token losses are (mostly) well approximated by first-order gradient information in the vicinity of optimizer updates.

Fig. 8: Sampled per-token loss landscape cross-sections across model sizes at the start of training We plot the same data as in Fig. 7, but focused on the beginning of training (before deceleration).

C.6 Overall loss landscape cross-sections throughout training

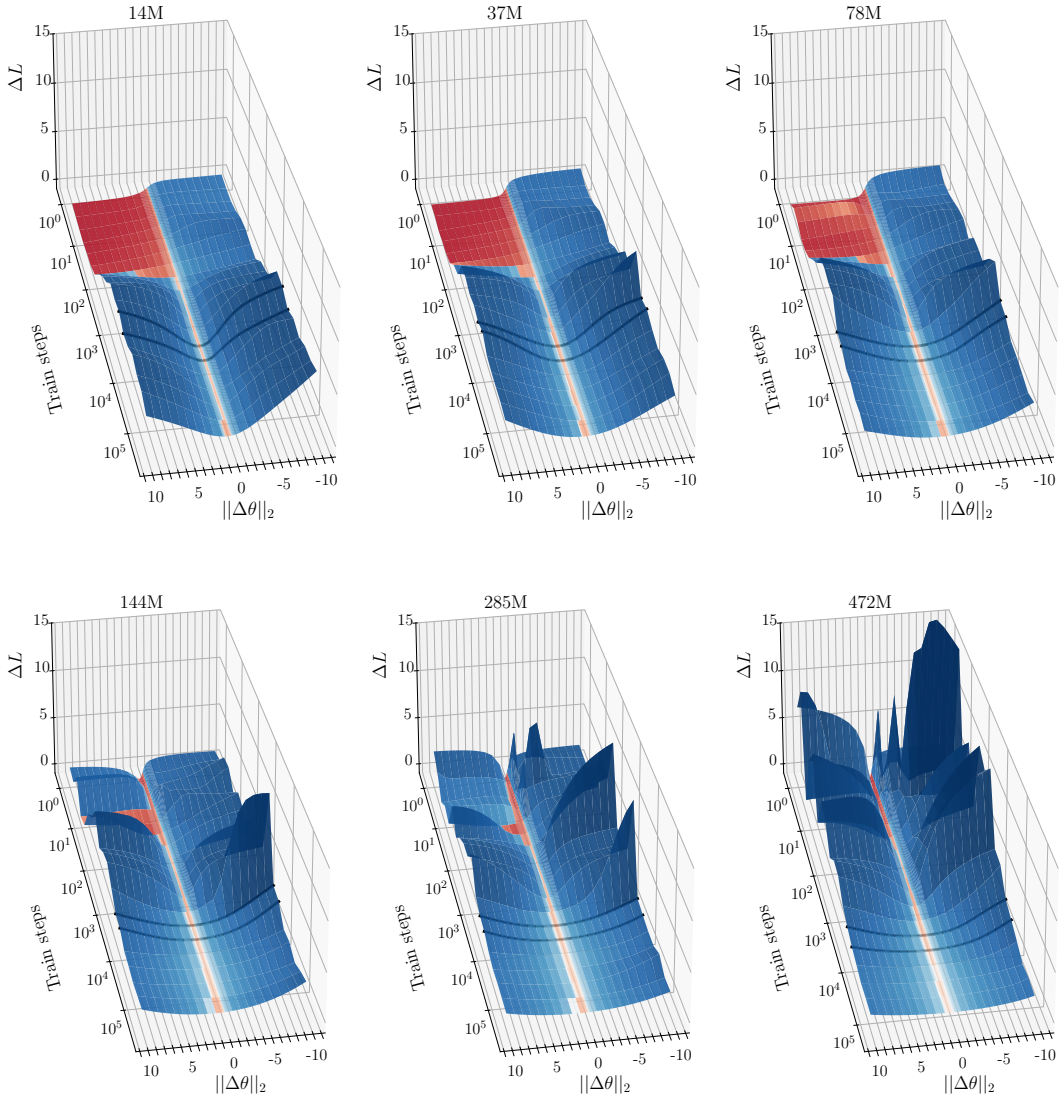


Fig. 9: Overall loss landscapes (cross section along $\Delta\theta$), visualized throughout training
We plot overall loss landscape cross sections across model sizes and train steps. Similar to Appendix C.5, we plot ΔL which has equivalent geometry to L but allows better distinguishing loss improvements from loss degradations. ΔL is additionally indicated with a symlog colorscale, with loss improvements being red. Loss deceleration is approximately indicated with two lines at $t = 4096$ and $t = 8192$. We observe that loss landscapes sharpen leading up to deceleration, but flatten significantly afterwards; with this trend being more pronounced in larger models. Furthermore, loss landscapes along $\Delta\theta$ appear much sharper in the beginning of training for larger models.

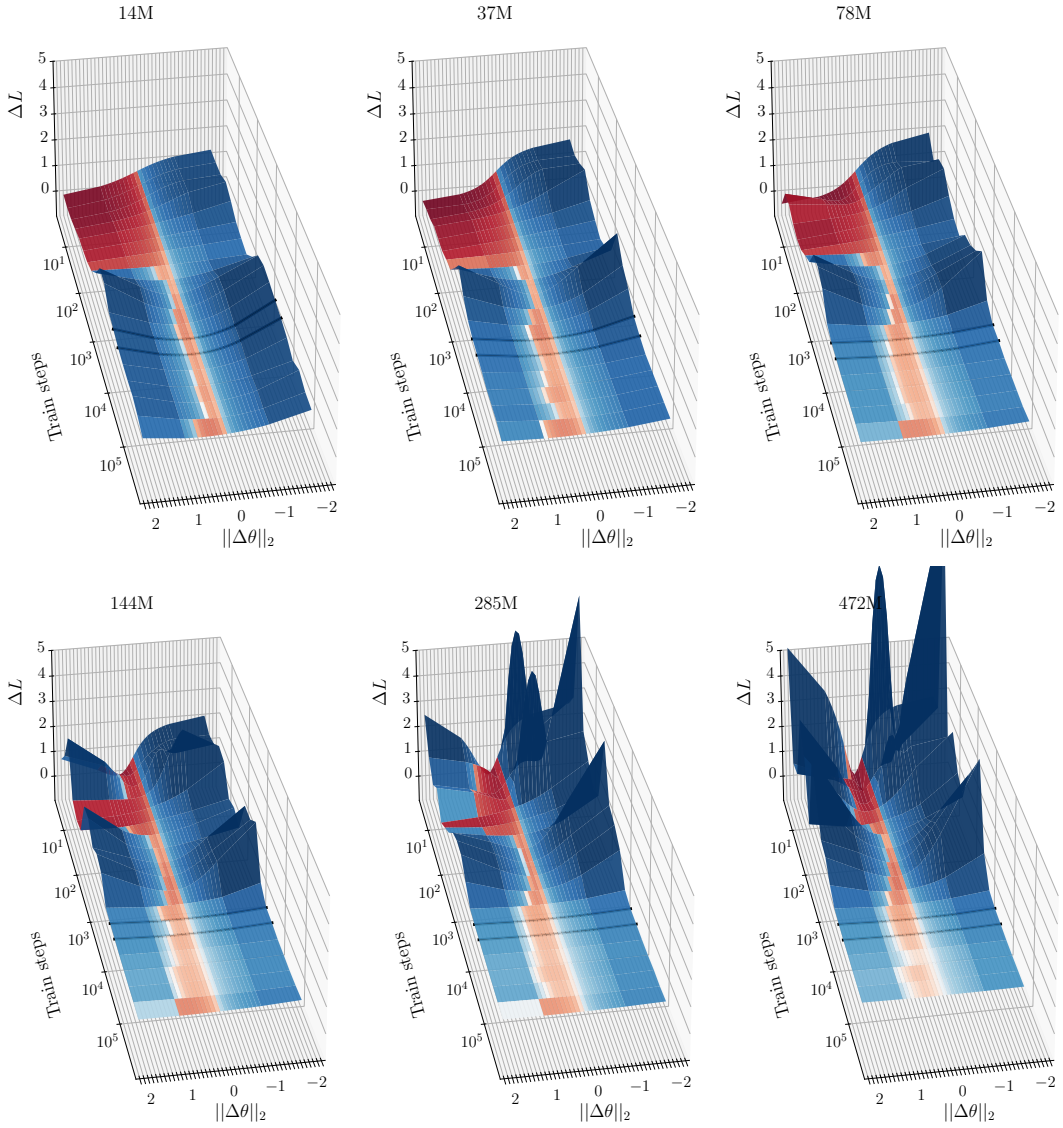


Fig. 10: Overall loss landscapes (cross section along $\Delta\theta$), visualized throughout training (zoomed in) We plot the same data as in 9, but zoomed into a narrower range.