Understanding Language Model Scaling Laws in Terms of Training Dynamics via Loss Deceleration and Zero-Sum Learning

Anonymous ACL submission

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 can be tied back to loss deceleration: an abrupt slowdown in the rate of loss improvement early in training, characterized by piece-wise linear behaviour in log-log space. Smoothly broken power laws allow us to parametrically measure this transition and express scaling improvements as a function of (1) decreasing the loss at which deceleration occurs, and (2) improving the log-log rate of loss improvement after deceleration. We hypothesize and validate *zero-sum learning* (ZSL) as a mechanistic explanation of loss deceleration that sheds new light on how scaling improves loss by mitigating this transition. In ZSL, per-token gradients become systematically opposed, leading to degenerate training dynamics where loss can't improve on one set of tokens without degrading on another; bottlenecking the rate at which overall loss can improve. In contrast to previous work on explaining scaling laws, ZSL is grounded in training dynamics and might potentially be targeted directly to improve loss independent of scale.

1 Introduction

What mechanisms underlie scaling laws? Increasing language model (LM) size empirically improves cross-entropy loss with power-law behaviour, accurately extrapolated across several orders of magnitude with scaling laws (Kaplan et al., 2020). Despite their predictive capabilities, scaling laws offer limited explanatory power as to the underlying mechanism (Stumpf and Porter, 2012); i.e. they do not explain *how* scaling improves loss. This question is of particular interest because, by identifying and understanding such a mechanism (Glennan and Illari, 2017), we may become able to target it directly and improve models independent of scale. While several recent works have sought to explain scaling laws based on notions of e.g. intrinsic model capacity (Sharma and Kaplan, 2022), data distribution properties (Michaud et al., 2023), or asymptotic behaviour (Bahri et al., 2024), mechanistic explanations that can inform new approaches and drive principled progress (beyond resource-intensive scaling) remain under-explored. In particular, little is known about the changes in training dynamics that underlie scaling improvements, which our work addresses.

Loss deceleration underlies scaling laws. We find that scaling improvements can be explained in terms of training dynamics via *loss deceleration*, a phenomenon where rates of loss improvement slow down abruptly with piecewise-linear loss curves in log-log space. Importantly, deceleration can be measured parametrically and used to decompose scaling improvements in terms of deceleration mitigation, specifically **(1)** decreases in loss at deceleration, and **(2)** increases in log-log rates of loss improvement after deceleration. A mechanistic explanation of deceleration (and of the mitigating effects of scale) therefore becomes an explanation for how scaling improves loss. The piecewise linear nature of deceleration suggests and indeed turns out to be a qualitative transition in training dynamics that explains deceleration. To the best of our knowledge, deceleration and the underlying transition in training dynamics has not been identified or addressed in relevant prior works on e.g. loss plateaus (Yoshida and Okada, 2020), learning curve shapes (Hutter, 2021; Viering and Loog, 2022), or LM saturation (Godey et al., 2024; Mircea et al., 2024). In light of this, we propose a mechanistic explanation of loss deceleration (and of the mitigating effects of scale) to shed new light on how scaling improves language models in terms of training dynamics.

A mechanistic explanation of deceleration. We hypothesize that deceleration occurs when per-example gradients become systematically opposed, leading to degenerate zero-sum learning training dynamics; i.e. where loss can't be improved on one set of examples without degrading on another, thus bottlenecking the rate at which overall loss can improve. We verify this hypothesis against alternative explanations, characterizing and validating the proposed mechanism with a complementary empirical and theoretical results. As a mechanistic explanation (Kaplan and Craver, 2011), zero-sum learning describes how the training dynamics of individual examples (i.e. their loss and gradients) behave and interact with one another to produce loss deceleration. This approach of understanding learning dynamics from the perspective of per-example gradient alignment and opposition is similar to Mircea et al. (2024), but otherwise under-explored outside of tangential areas of research on e.g. improving multi-task learning (Liu et al., 2021), or characterizing outliers in SGD (Rosenfeld and Risteski, 2023). Importantly, we believe zero-sum learning and systematic gradient opposition can potentially be mitigated directly to improve loss independent of scale. To better guide future work in this direction, we build on our findings and analyses to gain new understanding into how scaling improves loss by mitigating deceleration.

Summary of findings and contributions In Section 2 we identify loss deceleration as a novel qualitative transition in LM training dynamics. In particular, we show how scaling improvements can be explained in terms mitigating deceleration. In Section 3, we propose and validate a mechanistic explanation of deceleration based on destructive interference between per-example gradients and loss improvements. Lastly, in Section 4, we connect these mechanisms to scaling improvements, showing how they are mitigated in ways that could be targeted directly and independent of scale.

Methodology We adapt the training setup of Groeneveld et al. (2024) and scaling experiments of Kaplan et al. (2020), training and analyzing models between 14M and 472M parameters[1]. Details on training and model analyses are in Appendix A.

2 Loss deceleration in language models

Characterizing loss deceleration.

We find that LM loss curves typically exhibit an abrupt slow down in the rate of loss improvements early during training, in a transition we refer to as *loss deceleration*. Notably, we see in Fig. 1 that loss deceleration is characterized by piecewise-linear behaviour in log-log space, consistent across different model scales and training setups, suggesting a qualitative transition in training dynamics.

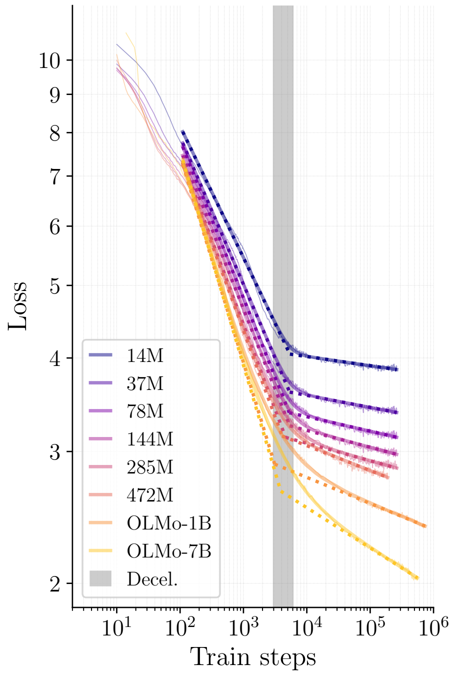

Fig. 1: Loss curves exhibit deceleration early in training (grey fill), and can be parametrically described with a one-break BNSL (Eqn. 1). The resulting BNSL fits are shown in bold, with the underlying piecewise-linear components shown as dashed lines. We also include the OLMo 1B and 7B models (Groeneveld et al., 2024), showcasing similar behaviour at larger scales.

An important observation from Fig. 1 is that loss improvements from scaling can be framed in terms of mitigating this transition, i.e. by improving:

(1) the loss at which deceleration occurs; and

(2) the log-log loss slope after deceleration.

This suggests that, by understanding the mechanism underlying loss deceleration (and the mitigating effects of scale), we can shed light on *how* scaling improves loss in terms of training dynamics. Such an understanding could in turn inform methods that directly target and mitigate deceleration independent of scale. However, to study how scale mitigates deceleration, we must first measure it.

Measuring loss deceleration with BNSL.

In measuring loss deceleration, we want to capture the log-log piecewise-linear behaviour observed in Fig. 1 and quantify how it changes with scaling.

[1] Code and artefacts, particularly model and optimizer checkpoints and logs across training, will be made available at the following https url to enable future work.

Luckily, this type of function can be parametrically described and fit with smoothly broken power laws such as BNSL (Caballero et al., 2023), particularly in the simplified one-break form:

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

where $L(t)$ is the loss at step t, and the remaining variables are the parameters being fit: a represents the irreducible loss; b the y-axis intercept $L(0)$; c_0 the log-log slope of the first linear segment; c_1 the difference between the slope of the second segment and the first; d_1 the step at which the break occurs; and f_1 the smoothness of the transition between segments. However, these parameters provide limited insight into how deceleration relates to loss.

Quantifying the effect of deceleration on loss. For a more interpretable but nevertheless quantifiable connection between deceleration and loss, we can tease out the linear segments underlying Eqn. 1. Concretely, an estimate $\hat{L}_T$ of the loss L_T at step $T > d_t$ can be expressed[2] in terms of three measurements grounded in BNSL parameters:

$$\boldsymbol{\log(\hat{L}_T)} = \log(L_d) - r_d \log(T/t_d) \quad (2)$$

$$\boldsymbol{\hat{L}_T} = L_d \, (t_d/T)^{r_d}$$

$\boldsymbol{t_d}$ $= d_1$, the step where deceleration occurs, or where the two segments intersect;

$\boldsymbol{L_d}$ $= b{d_t}^{-c_0}$, the loss where deceleration occurs, or where the two segments intersect;

$\boldsymbol{r_d}$ $= c_0 + c_1$, the log-log rate of loss improvement after deceleration, or the negative log-log slope of the second segment.

Intuitively, we see that final loss is fundamentally a function of L_d, r_d, t_d; such that scaling improvements can be explained solely in terms of increased r_d and decreased L_d, t_d. These measurements are reported in Table 1, where we indeed observe monotonic improvements in L_d, r_d and t_d with increased model size[3]. We also confirm that $\hat{L}_T$ is a valid approximation, typically within 1% of L_T.

Crucially, these are interpretable measurements of loss deceleration, allowing us to naturally describe and reason about scaling improvements in terms of training dynamics. For example, Eqn. 2 forms the basis of a novel scaling law functional form, with improved explanatory power as a result of being grounded in these interpretable quantities. While beyond the scope of this paper, we include preliminary results in Appendix C.2.

Table 1: Loss deceleration measurements from Eqn. 2: larger models have lower L_d, t_d and higher r_d.

Model	↓ L_d	↓ t_d	↑ r_d	$\hat{L}_T$	L_T
14M	4.05	5900	0.013	3.86	3.88
37M	3.60	5900	0.016	3.39	3.40
78M	3.38	5900	0.020	3.14	3.15
144M	3.25	6000	0.023	2.98	2.99
285M	3.14	5300	0.025	2.85	2.87
472M	3.16	4600	0.035	2.77	2.80
OLMo-1B	2.86	3700	0.034	2.39	2.40
OLMo-7B	2.64	4600	0.053	2.04	2.03

More generally, these results suggest that a mechanistic explanation of loss deceleration as a transition in training dynamics (and of what determines the values of L_d, r_d, and t_d; i.e., when does this transition happen and how severe is it?) can also account for final loss and thus shed light on *how* scaling improves performance in language models.

3 Explaining Loss Deceleration

The log-log piecewise-linear behaviour of loss deceleration suggests that a qualitative change in training dynamics underlies the abrupt slowdown in loss improvements. Our goal in this section is to characterize this transition in training dynamics as a mechanistic explanation for loss deceleration. By "mechanistic explanation", we mean identifying and formalizing an underlying mechanism as defined in Glennan and Illari (2017) and Section 1. To this end, we propose and verify the hypothesis that loss deceleration is a transition in training dynamics characterized by and resulting from zero-sum learning and systematic gradient opposition.

Zero-sum learning (ZSL) describes degenerate training dynamics where loss improvements in one set of examples are increasingly offset by loss degradation in another set of examples, bottlenecking overall loss improvements. Intuitively, this is a mechanistic explanation of how per-example changes in loss interact to produce the abrupt slowdown in overall loss improvements seen in deceleration. An alternative (but not mutually-exclusive) explanation is that the magnitude of per-example changes in loss decreases across examples. We show in Section 3.1 that ZSL is indeed responsible for loss deceleration.

[2] See Appendix A.2 in (Caballero et al., 2023).

[3] One notable outlier is t_d in OLMo-7B, likely attributable to OLMo-7B using a warmup of 5,000 rather than 2,000 steps.

Systematic gradient opposition (SGO) describes degenerate training dynamics where per-example gradients become increasingly opposed. As a result, under first-order training dynamics[4], a step of gradient descent fundamentally cannot improve loss on one set of examples without harming it on another. Intuitively, this is a mechanistic explanation of how per-example gradients interact to produce ZSL. We show in Section 3.2 that SGO is indeed responsible for ZSL.

Notation Let ℓ_i denote loss for token i in dataset $\mathcal{D}$, with overall loss $L = \sum_i \ell_i / |\mathcal{D}|$. Conversely, change in loss between steps t_1, t_2 is denoted as $\Delta_{t_1}^{t_2} L = \sum_i \Delta_{t_1}^{t_2} \ell_i / |\mathcal{D}|$. To reduce notation clutter, t_1, t_2 are sometimes omitted when evident from context or not relevant.

3.1 Zero-Sum Learning (ZSL)

Measuring ZSL with destructive interference To measure zero-sum learning, we define destructive interference in per-token loss improvements $\Delta\ell_i$ as the proportion with which they cancel out in overall loss improvements $\Delta L = \sum_i \Delta\ell_i / |\mathcal{D}|$, with respect to an ideal scenario where there is no interference $\Delta L^* = \sum_i |\Delta\ell_i| / |\mathcal{D}|$:

$$D(\Delta\ell) = \frac{\Delta L^* - |\Delta L|}{\Delta L^*} = 1 - \frac{|\sum_i \Delta\ell_i|}{\sum_i |\Delta\ell_i|} \quad (3)$$

Intuitively, as ZSL increases and per-token loss improvements cancel out in larger proportions, $D(\Delta\ell)$ increases and approaches 1 with complete ZSL. Conversely, as ZSL decreases, $D(\Delta\ell)$ decreases and approaches 0 with no ZSL.

Validating that ZSL occurs with deceleration. In Fig. 2, we measure $D(\Delta_t^{2t}\ell)$ throughout training. We observe that ZSL exhibits a sharp increase, beginning just before deceleration, then converging towards its maximum. One important consideration is that these measurements are based on $\Delta_t^{2t}\ell$ to smooth out noise from loss oscillations on too-small timescales. In practice, we find that $D(\Delta_{t_1}^{t_2}\ell)$ is mitigated as the number of steps $t_2 - t_1$ increases, such that Fig. 2 is effectively under-reporting the rate at which ZSL increases (Appendix C.3).

These results confirm that ZSL indeed occurs with loss deceleration, but are not sufficient evidence that ZSL is the underlying mechanism. The following sections will demonstrate how, in terms of per-token loss behaviour, deceleration is driven by ZSL rather than the alternative explanation.

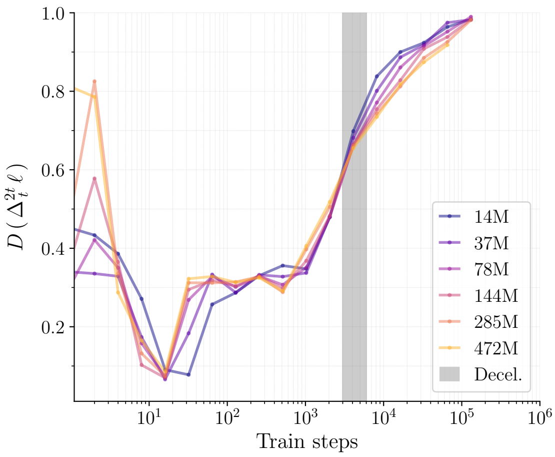

Fig. 2: ZSL throughout training, as measured by destructive interference in per-token loss improvements. Deceleration co-occurs with a sharp increase in ZSL.

Quantifying the role of ZSL in deceleration. In terms of per-token loss improvements $\Delta\ell_i$, loss deceleration can occur for two (non mutually exclusive) reasons: (1) $\Delta\ell_i$ increasingly cancel one another out due to ZSL; or (2) $\Delta\ell_i$ increasingly shrink in magnitude across tokens. Destructive interference $D(\Delta\ell)$ in Eqn. 3 captures (1); while average magnitude $M(\Delta\ell)$ in Eqn. 4 captures (2):

$$M(\Delta\ell) = \frac{\sum_i |\Delta\ell_i|}{|\mathcal{D}|} \quad (4)$$

Importantly, we can express the absolute change in loss $|\Delta L|$ in terms of these two quantities:

$$|\Delta L| = \frac{|\sum_i \Delta\ell_i|}{|\mathcal{D}|} = M(\Delta\ell)(1 - D(\Delta\ell)) \quad (5)$$

If loss is monotonically decreasing, $|\Delta L|$ corresponds to overall loss improvements, such that we can effectively quantify and disentangle the relative contributions of increasing $D(\Delta\ell)$ from decreasing $M(\Delta\ell)$ in loss deceleration.

Showing ZSL is responsible for deceleration. In Fig. 3, we plot model training trajectories with respect to the terms in Eqn. 5. This allows us to visually determine and quantify how increases in $D(\Delta\ell)$ map to decreases in $|\Delta L|$; i.e. the contribution of ZSL to loss deceleration. Notably, we see that during and after deceleration, reductions in $|\Delta L|$ are largely attributable to changes in D rather than M. Concretely, we know from Eqn. 5 that the observed reduction in M during deceleration, from 0.75 to 0.5, corresponds to a 1.5x reduction

[4] i.e. well approximated by a first-order Taylor expansion.

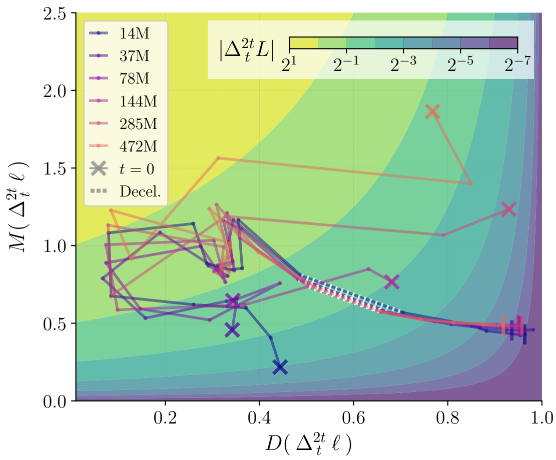

Fig. 3: Disentangling the relative contributions of increased ZSL ($D(\Delta\ell)$) and decreased token-level loss improvements ($M(\Delta\ell)$) towards decreased overall loss improvements ($|\Delta L|$). Model training trajectories, plotted with respect to these values, show that ZSL dominates decreases in $|\Delta L|$ after deceleration.

in $|\Delta L|$. 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 $|\Delta L|$ explodes such that ZSL effectively bottlenecks loss improvements and leads to deceleration. These results corroborate that, while decreasing magnitude across per-token loss improvements plays a role in deceleration, the effect of ZSL is almost an order of magnitude greater and effectively bottlenecks loss improvements.

3.2 Systematic Gradient Opposition (SGO)

Measuring SGO with destructive interference To measure gradient opposition, we can adapt Eqn. 3 for coordinate-level destructive interference between per-token gradient vectors:

$$\vec{D}\,(\nabla_\theta \ell) = 1 - \frac{|\sum_i \nabla_\theta \ell_i|}{\sum_i |\nabla_\theta \ell_i|} \tag{6}$$

Typically, we report $D(\nabla_\theta \ell)$ as the average over parameters. Similarly to ZSL and Eqn. 3, $D(\nabla_\theta \ell)$ approaches 1.0 as SGO increases, and approaches 0.0 as SGO decreases. $D(\nabla_\theta \ell)$ can equivalently be viewed as destructive interference between gradient vectors as measured by L^1-norm: $1 - \|\nabla_\theta L\|_1 / \sum_i \|\nabla_\theta \ell_i\|_1$. While we could define a similar measure based on L^2-norm, we found that this penalizes orthogonality and becomes biased towards 1.0 at high dimensions, even in cases of no gradient opposition.

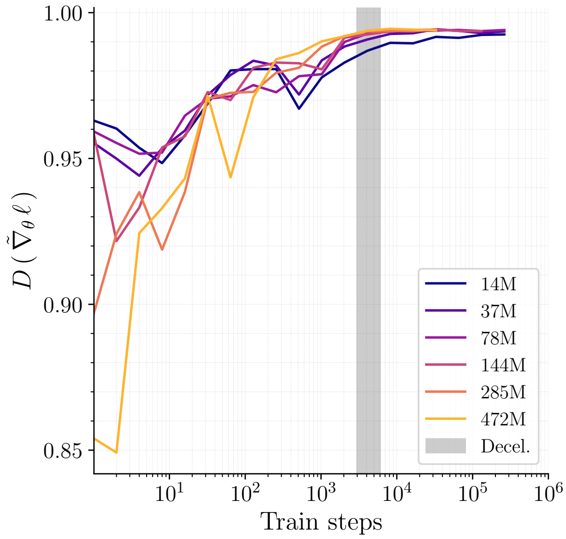

Fig. 4: Gradient destructive interference (averaged over parameters) converges to a maximum with deceleration.

Validating that SGO occurs with ZSL. As in Section 3.1, we first attempt to falsify our hypothesis that SGO is responsible for ZSL by verifying if they co-occur. We find in Fig. 4 that SGO (as measured by $D(\nabla_\theta \ell)$ in Eqn. 6) converges to a maximum at the same time as deceleration and the previously observed increase in ZSL. Surprisingly, we find that gradient opposition is quite high even at the start of training. However, as we saw in Eqn. 5 and Fig. 3, increasing $D(\nabla_\theta \ell)$ beyond 0.9 can rapidly decrease the magnitude of $\nabla_\theta L$ by several orders of magnitude.

Relating SGO to ZSL in first-order dynamics. While we now know that SGO converges with ZSL, this does not necessarily imply that SGO is responsible for ZSL. To address this, we will show how near-complete SGO where $D(\nabla_\theta \ell) \to 1.0$ fundamentally causes ZSL under first-order training dynamics. By first-order dynamics, we mean that weight updates $\Delta\theta$ are sufficiently small such that changes in overall or per-token loss are approximable by first-order Taylor expansions $\tilde{\Delta}L$, $\tilde{\Delta}\ell_i$:

$$\tilde{\Delta}L = \Delta\theta \cdot \nabla_\theta L = \textstyle\sum_i \tilde{\Delta}\ell_i \,/\, |D| \tag{7}$$

$$\tilde{\Delta}\ell_i = \Delta\theta \cdot \nabla_\theta \ell_i$$

In such cases, ZSL is intrinsically a result of destructive interference in $\Delta\theta \cdot \nabla_\theta \ell_i$ across tokens:

$$D(\tilde{\Delta}\ell) = 1 - \frac{|\sum_i \Delta\theta \cdot \nabla_\theta \ell_i|}{\sum_i |\Delta\theta \cdot \nabla_\theta \ell_i|} \tag{8}$$

Unfortunately, Eqn. 8 does not necessarily imply that gradient opposition results in ZSL. For instance, directions of high opposition in per-token gradients $\nabla_\theta \ell_i$ may be orthogonal to a weight update $\Delta\theta$, such that they are nullified when $\nabla_\theta \ell_i$ is projected onto $\Delta\theta$. Conversely, two gradients $\nabla_\theta \ell_i$, $\nabla_\theta \ell_j$ with no destructive interference may result in ZSL if e.g. $\Delta\theta$ is aligned with $\nabla_\theta \ell_i - \nabla_\theta \ell_j$. In light of this, we want to disentangle ZSL in Eqn. 8 that is attributable to update-gradient alignment independent of gradient opposition, from ZSL attributable to gradient opposition specifically.

To this end, we adapt Eqn. 5 and Eqn. 6 to decompose the overall gradient as $\nabla_\theta L = \pm\vec{M}(1 - \vec{D})$, where $\vec{D}$ is a vector of gradient coordinate-wise destructive interference, and $\vec{M}$ is a vector of average coordinate-wise gradient magnitude (with $\pm\vec{M}$ as a compact notation for $\vec{M}\,\mathrm{sign}(\nabla_\theta L)$). This lets us rewrite Eqn. 8 while factoring out dependence on gradient opposition via $\vec{D}$:

$$D(\tilde{\Delta}\ell) = 1 - \frac{|\Delta\theta \cdot \pm\vec{M}(1 - \vec{D})|}{\sum_i |\Delta\theta \cdot \nabla_\theta \ell_i|} \tag{9}$$

$$= 1 - \frac{|\Delta\theta \cdot \pm\vec{M} - \Delta\theta \cdot \pm\vec{M}\vec{D}|}{\sum_i |\Delta\theta \cdot \nabla_\theta \ell_i|}$$

Because $\vec{D} \in [0, 1]$ we can rewrite the numerator as $|\Delta\theta \cdot \pm\vec{M}| - |\Delta\theta \cdot \pm\vec{M}\vec{D}|$. We can then decompose dot products into their corresponding vector norms and cosine similarities to obtain Eqn. 10—an interpretable decomposition of Eqn. 8 that isolates the effect of update-gradient alignment into C_u, and the effect of gradient opposition into C_g:

$$D(\tilde{\Delta}\ell) = 1 - C_u + C_g \tag{10}$$

$$C_u = \frac{\|\pm\vec{M}\| \cos(\Delta\theta, \pm\vec{M})|}{\sum_i \|\nabla \ell_i\| |\cos(\Delta\theta, \nabla \ell_i)|} \in [0, 1]$$

$$C_g = \frac{\|\pm\vec{M}\vec{D}\| |\cos(\Delta\theta, \pm\vec{M}\vec{D})|}{\sum_i \|\nabla \ell_i\| |\cos(\Delta\theta, \nabla \ell_i)|} \in [0, C_u]$$

Intuitively, $1 - C_u$ capture destructive interference in the case there is no coordinate-level gradient opposition (i.e. $\nabla_\theta L = \pm\vec{M}$) such that ZSL is entirely attributable to update-gradient alignment.

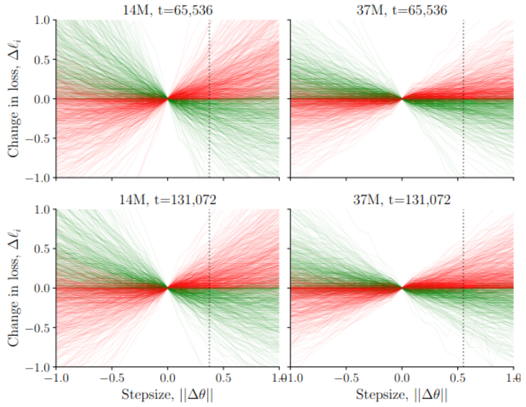

Fig. 5: Per-token loss landscapes at step t along $\Delta_t\theta$. Dashed vertical lines indicate $\theta_{t+1} = \theta_t + \Delta_t\theta$. Tokens which improve in loss after the update are indicated in green, and tokens which degrade are indicated in red

This quantity is an upper bound on C_g, which in turn captures the effect of gradient opposition on ZSL not already accounted for by C_u. In particular, as $\vec{D}$ approaches $\mathbf{1}$, we see that C_g approaches C_u, and $D(\tilde{\Delta}\ell)$ approaches 1.0 irrespective of C_u. In other words, systematic gradient opposition as $D(\nabla_\theta \ell) \to 1.0$ across parameters implies ZSL in a first-order approximation of loss improvements.

Testing the first-order dynamics assumption. Our hypothesis relies on the assumption that $\tilde{\Delta}\ell$ is a valid approximation of $\Delta\ell$ such that destructive interference in $\tilde{\Delta}\ell$ is reflective of destructive interference in $\Delta\ell$. To validate our hypothesis, we must therefore validate this assumption. 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.

Instead, to empirically measure $\tilde{\Delta}\ell$, we compute 1D cross-sections of per-token loss landscapes by evaluating model checkpoints along increments of their next 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 $\tilde{\Delta}\ell(\alpha) = \alpha\left(\frac{\ell_{\theta+\epsilon} - \ell_\theta}{\|\epsilon\|}\right)$. A sample of 1,000 such per-token loss landscapes is shown in Fig. 5, with the complete set in Appendix C.4. Generally, these appear linear in the vicinity of weight updates, suggesting that actual changes in per-token losses $\Delta\ell$ are well captured by their first-order approximation $\tilde{\Delta}\ell$.

However, to more quantifiably verify that this indeed is the case, we measure and plot the Pearson correlation coefficient between $\Delta\ell$ and $\tilde{\Delta}\ell$ throughout training in Fig. 6. We find strong correlation after deceleration where we observe ZSL and SGO, validating our hypothesis by validating the underlying assumption of first-order dynamics on which our reasoning depends.

Ruling out the role of progressive sharpening. As an alternative explanation for ZSL, one might consider progressive sharpening (Cohen et al., 2022; Rosenfeld and Risteski, 2023) where $\Delta\theta$ might overshoot local minima for some examples but not others. Surprisingly, and perhaps counter to conventional wisdom, we observe in Appendix C.5 that loss landscapes instead become significantly flatter with deceleration; following an initial phase of high sharpness before deceleration.

To quantify this observation, we measure the sharpness of loss landscapes along update directions, throughout training. Specifically, we fit a quadratic to the loss landscape cross-section, using the second order term as a measure of sharpness. In Fig. 7, we see the same trend, with sharpness peaking and immediately begin decreasing before deceleration. While the relationship between loss sharpness and zero-sum learning and systematic gradient opposition was outside the scope of this work, it appears there might be an interesting connection.

4 Explaining Scaling Improvements

In Section 2 we showed how scaling improves loss by mitigating loss deceleration, specifically by decreasing the loss L_d and step t_d at which it occurs, and increasing the subsequent log-log rate of loss improvement r_d (Table 1). Conversely, in Section 3 we proposed a mechanistic explanation of loss deceleration based on interactions at the levels of per-example loss improvements, and of per-example gradients. Specifically, we showed that loss deceleration is a transition in training dynamics characterized by the emergence of near-complete destructive interference in per-example gradients and loss improvements; i.e. SGO and ZSL. In this section, we will attempt to connect these findings, and shed light on *how* scaling mitigates deceleration based on the underlying mechanisms we identified.

Decomposing loss improvements. Similar to Section 3.2, we decompose the first-order Taylor expansion for changes in loss from Eqn. 7 into interpretable components that enable a finer-grained analysis of training dynamics, specifically the cosine similarity and L^2 norms of weight updates $\Delta\theta$ and gradients $\nabla_\theta L$:

$$\tilde{\Delta}L = \|\Delta\theta\|_2 \|\nabla_\theta L\|_2 \, \cos(\Delta\theta, \nabla_\theta L) \qquad (11)$$

We show these values across training steps and model scales in Fig. 8, and will discuss their interpretation in the following sections.

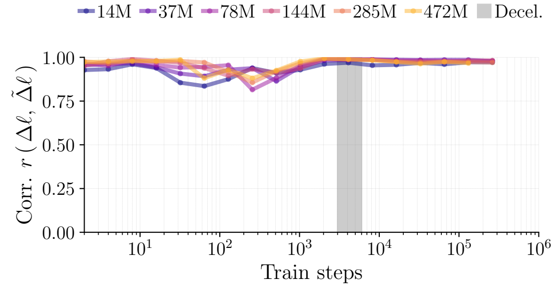

Fig. 6: Correlation between $\Delta\ell$ and it's first-order approximation $\tilde{\Delta}\ell$ is close to 1.0 at deceleration.

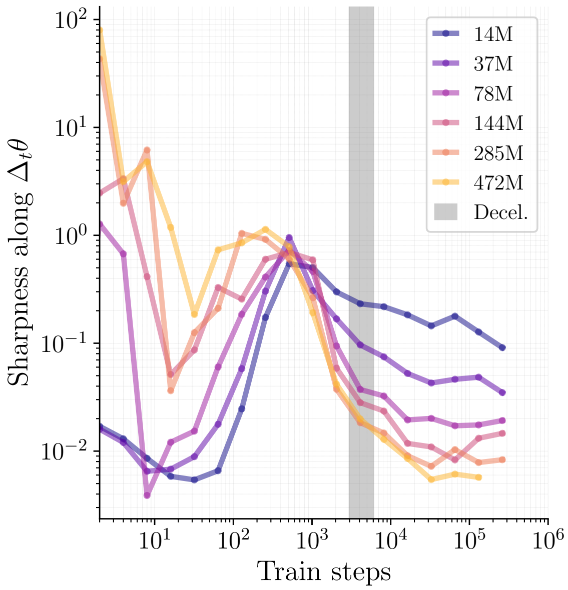

Fig. 7: Sharpness decreases with loss deceleration.

4.1 Improving Loss Before Deceleration (L_d)

Surprisingly, we find in Table 2 that most of the scaling improvements in loss at deceleration L_d are already established by step $t = 32$. From Eqn. 11

Table 2: Scaling improvements in loss at deceleration L_d are established early during training.

Loss Improvement	$t = 32$	$t = 4096$	$t = 8192$
14M → 37M	0.76	0.43	0.45
37M → 78M	0.29	0.20	0.21
78M → 144M	0.15	0.12	0.12
144M → 285M	0.15	0.11	0.11
285M → 472M	0.05	0.06	0.07

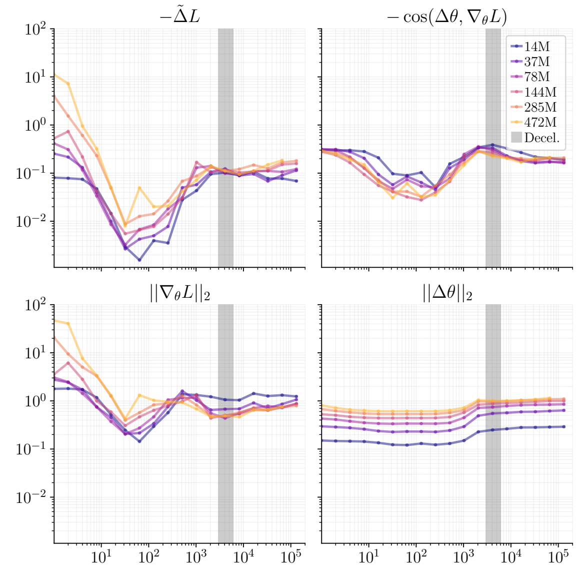

Fig. 8: First-order approximation of loss improvements with terms from Eqn. 11 plotted throughout training. Note that because $\log(\tilde{\Delta}L)$ is a sum of the log of its terms, the shared log-scale allows us to easily gauge how different terms contribute to changes in $\tilde{\Delta}L$.

and Fig. 8, the underlying reason becomes apparent. Scaling models improves $\tilde{\Delta}L$ primarily by improving gradient norms $||\nabla_\theta L||_2$ in the beginning of training. Beyond $t = 32$, the effects of scaling become less significant, with improvements in $\tilde{\Delta}L$ and $||\nabla_\theta L||_2$ orders of magnitude smaller and eventually reversed leading up to deceleration. In contrast, scaling degrades gradient-update alignment $-\cos(\Delta\theta, \nabla_\theta L)$, and results in consistent but relatively insignificant improvements in $||\Delta\theta||_2$. These effects are trivially explained by an increased number of parameters and appear unrelated to deceleration, however it remains an open question how similar effects can be achieved independent of scale.

4.2 Improving Loss After Deceleration (r_d)

We see in Fig. 2 that post-deceleration ZSL is mitigated by scaling model size, which we know results in greater loss improvements from Eqn. 5 that can explain how scaling improves r_d. Unfortunately, the way in which scaling reduces ZSL after deceleration is not as immediately obvious.

We see in Fig. 4 that gradient destructive interference (averaged across parameters) actually becomes more pronounced with larger models. However, 99% destructive interference in a 14M-dimensional gradient does not have the same effect as in a 144M-dimensional gradient. In particular, the latter will have more degrees of freedom along which shared gradient directions can exist between tokens. Indeed, we find in Fig. 9 that, especially after deceleration, larger models have more parameters with lower destructive interference. This can explain why larger models have lower ZSL after deceleration, and thus improved r_d.

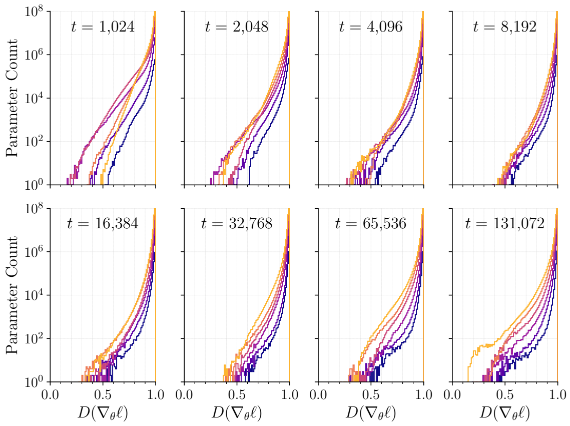

Fig. 9: Histograms of gradient destructive interference. Deceleration happens between steps 4096 and 8192.

5 Conclusion and outlook

In this work we proposed and validated a mechanistic explanation of scaling laws grounded in training dynamics. Specifically, we identified loss deceleration as a novel transition in training dynamics that can explain scaling improvements in quantifiable but interpretable terms, such that an explanation of deceleration becomes an explanation of scaling improvements. To this end, we proposed zero-sum learning and systematic gradient opposition as the mechanisms underlying deceleration, validating these against alternate hypotheses with empirical and theoretical analyses. Lastly, we revisit scaling improvements from the perspective of these mechanisms and show how scaling improves loss by mitigating zero-sum learning and systematic gradient opposition.

Our analyses and results suggest that these could potentially be mitigated directly to improve loss independent of scale, laying a foundation for future research. Furthermore, our approach of studying per-example gradient dynamics is an underexplored area of research that can shed new light on learning dynamics, scaling, and generalization more broadly.

Limitations

Comprehensiveness of experimental settings. Scaling laws are a general phenomenon observed across tasks, model architectures, parameters, and evaluation measures. However, this work only considers the scaling of cross-entropy loss with model size in transformer-based language models on typical webscale text. While we replicate our experiments across variants of this general setting (e.g. with and without learning rate decay in Appendix C.1), we do not generalize our findings to different settings. While this lies beyond the scope of our original research question and the prior works on which we build, verifying how our findings generalize across different settings is an important area of future work.

Characterizing the effects of hyperparameters on deceleration. Our model training runs make use of the optimal hyperparameter configurations identified by Kaplan et al. (2020) and Groeneveld et al. (2024) on which we base our experiments and training setups. This was motivated by two factors: limiting the computational cost of hyperparameter search, and conducting experiments and analyses consistent with prior works. However, an important gap that results from this is an unclear understanding of how loss deceleration and the associated measurements in Table 1 change with different hyperparameter configurations. For example, we found in Section 2 that differences in number of warmup steps between publicly available OLMo-1B and 7B models seem to have an important effect. Furthermore, OLMo-1B and 7B use different sequence lengths and batch sizes to (Kaplan et al., 2020) and our experiments which prioritize computational efficiency over downstream performance. While deceleration appears consistent across these variations, it is not clear to what extent improvements in the 1B and 7B models are due to increased scale as opposed to these differences in hyperparameters. While beyond the scope of our original research question, these questions present an important opportunity for future research.

Accounting for SGO in both $M(\Delta\ell)$ and $D(\Delta\ell)$. In Section 3.1 and Eqn. 5 we showed that ZSL as measured by $D(\Delta\ell)$ is primarily responsible for deceleration, while decreases in average per-token loss improvements $M(\Delta\ell)$ played an non-negligble but less significant role. However, our analysis of SGO (Section 3.2) only considers $D(\Delta\ell)$ and ZSL, while it likely also has an effect on $M(\Delta\ell)$ via its effect on optimizer steps $\Delta\theta$. However, these effects are likely highly dependent on the optimizer and its configuration, and likely not generalizable in the scope of our research question; hence why we chose to abstract away $\Delta\theta$ in our analysis. Nevertheless, this a salient gap in our analysis that should be further explored.

Reconciling single step and multi step training dynamics. The connection between the behaviour of gradients (SGO) and loss (ZSL and loss deceleration) can be made more precise. In particular, our gradient analysis only reflects single-step training dynamics, while ZSL and loss improvements appear to depend on interactions across multiple optimization steps (see Appendix C.3). Understanding the effect of multi step interaction is a natural next step for this research.

Negative societal impacts or ethical concerns. Our work focuses on understanding existing and well-established methods, and does not meaningfully contribute to any negative societal impacts or ethical concerns beyond what is typically associated with language modeling research. In principle, by focusing our analysis on a single metric (cross-entropy loss), this could lead to over-optimizing that metric at the expense of other real-world concerns. While this work is at too early a stage for this to pose a meaningful risk, it is important to keep in mind as a limitation in interpreting our findings and building new methods on top of them.

References

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

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

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

Blake Bordelon, Alexander Atanasov, and Cengiz Pehlevan. 2024. A Dynamical Model of Neural Scaling Laws. *arXiv preprint*. ArXiv:2402.01092 [cond-mat, stat].

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

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

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. 2024. Scaling Exponents Across Parameterizations and Optimizers. *arXiv preprint*. ArXiv:2407.05872 [cs] version: 1.

Stuart Glennan and Phyllis Illari, editors. 2017. *The Routledge Handbook of Mechanisms and Mechanical Philosophy*. Routledge, London.

Nathan Godey, Éric de la Clergerie, and Benoît Sagot. 2024. Why do small language models underperform? Studying Language Model Saturation via the Softmax Bottleneck. *arXiv preprint*. ArXiv:2404.07647 [cs].

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. 2024. OLMo: Accelerating the Science of Language Models. *arXiv preprint*. ArXiv:2402.00838.

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. 2022. Training Compute-Optimal Large Language Models. *arXiv preprint*. ArXiv:2203.15556 [cs].

Marcus Hutter. 2021. Learning Curve Theory. *arXiv preprint*. ArXiv:2102.04074 [cs, stat].

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

David Michael Kaplan and Carl F. Craver. 2011. The Explanatory Force of Dynamical and Mathematical Models in Neuroscience: A Mechanistic Perspective*. *Philosophy of Science*, 78(4):601–627. Publisher: [The University of Chicago Press, Philosophy of Science Association].

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

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

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. 2023. Paloma: A Benchmark for Evaluating Language Model Fit. *arXiv preprint*. ArXiv:2312.10523.

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

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

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

Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. 2020. Learning explanations that are hard to vary.

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

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

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

Utkarsh Sharma and Jared Kaplan. 2022. Scaling Laws from the Data Manifold Dimension. *Journal of Machine Learning Research*, 23(9):1–34.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. 2022. Beyond neural scaling laws: beating power law scaling via data pruning. In *Advances in Neural Information Processing Systems*, volume 35, pages 19523–19536. Curran Associates, Inc.

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

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. 2024. Gemma 2: Improving Open Language Models at a Practical Size. *arXiv preprint*. ArXiv:2408.00118 [cs].

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

Tom Viering and Marco Loog. 2022. The Shape of Learning Curves: a Review. *arXiv preprint*. ArXiv:2103.10948 [cs].

Kaiyue Wen, Zhiyuan Li, Jason Wang, David Hall, Percy Liang, and Tengyu Ma. 2024. Understanding Warmup-Stable-Decay Learning Rates: A River Valley Loss Landscape Perspective. ArXiv:2410.05192 [cs, stat].

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

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

A Methodology

A.1 Language model pretraining

For our experiments, we adapt the OLMo codebase (licensed under Apache-2.0) and train variants of OLMo with the publicly available training dataset of OLMo-7B-0724 (Groeneveld et al., 2024). Model dimensions and learning rates are based on (Kaplan et al., 2020) and shown in Table 3, labeled with (rounded) total parameter counts. For pretraining, we again adapt the experimental setup of (Kaplan et al., 2020), 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 (Hägele et al., 2024) with a learning rate warmup to the values in TableFig. 3 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 (Groeneveld et al., 2024) and could not included in our analysis of ZSL because of insufficient checkpointing frequency before deceleration.

Code and artefacts, particularly model and optimizer checkpoints and logs across training, will be made available at the following https url under Apache-2.0 license to enable future research in this direction. In our experiments, we used a variety of computational resources which are recorded in the logs we make available. Generally, we performed distributed training 4-32 L40 GPUs or 4 H100 GPUs, with smaller models pretraining requiring on the order of 10 GPU hours, and the largest 472M requiring on the order of 1000 GPU hours.

A.2 Language model analyses

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 (Magnusson et al., 2023), using the tokenizer from (Groeneveld et al., 2024), consistent with pretraining.

A.3 Additional Details on Fitting BNSL to Loss Curves

Fitting We adapt the methodology for fitting Eqn. 1 published by Caballero et al. (2023) at `https://github.com/ethancaballero/broken_neural_scaling_laws`. We include the code implementation below. Empirically, we had to implement the following changes to improve stability:

- Assume $a = 0$ and remove it from the optimization procedure.
- Fit the function in log-log space instead of manually scaling b and d_1 (note that datapoints sampled uniformly along x will result in a data imbalance when fitting in log-log space; to mitigate this we also subsample datapoints uniformly in log space). also made it necessary to subsample our fitting data uniformly in log-space to limit skewing from resudata imbalances)
- Estimate initial parameters instead of running a bruteforce gridsearch.

Smoothing The loss curves we fit are batch losses logged at every step during training. Because training is single-epoch, i.e. online, these losses are effectively a noisy measurement of the true validation loss. However, we found that this noise (characterized by oscillations in loss at too-small timescales) leads to severe instability with the original methodology published by (Caballero et al., 2023). To smooth these curves, we use LSMA$_k$, a logarithmic variant of the simple moving average that we found to work well for fitting noisy log-log loss curves with high fidelity. Notably, LSMA naturally handles the increasing timescales at which loss oscillations occur as number of training steps increase. We found $k = 1.2$ to work sufficiently well as shown in Fig. 10.

$$\mathrm{LSMA}_k\left(L_t\right) = \frac{1}{t - p(t)} \sum_{s=p(t)}^{t} L_s \qquad (12)$$

$$p(t) = \mathrm{floor}(t/k)$$

Results and validation We report the resulting parameters and error measurements from fitting Eqn. 1 in Table 4, finding that parameter standard deviation is typically on the order of 1%, while root standard log error (RSLE) is on the order of 0.01, comparable with values reported by Caballero et al. (2023). These results suggest that loss deceleration is reliably measurable with BNSL.

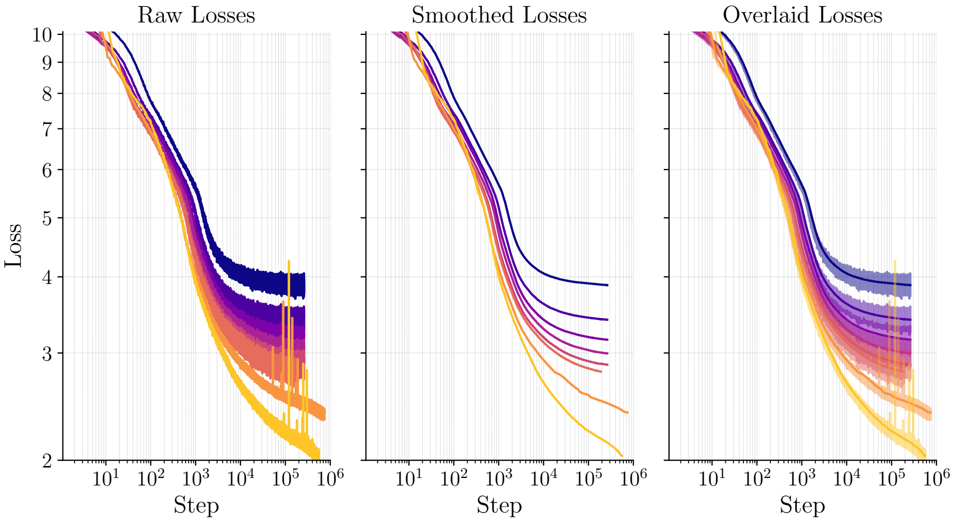

Fig. 10: $\text{LMSA}_k\ (L_t)$ smoothing with $k = 1.2$.

Table 3: 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
warmup	2,000	2,000	2,000	2,000	2,000	2,000	2,000	5,000

Table 4: BNSL parameters and root-standerd log error if resulting fit (RSLE).

Model	b	c_0	c_1	$\log(d_1)$	f_1	RSLE
14M	18.42 ± 0.16	0.17 ± 0.00	-0.16 ± 0.00	8.68 ± 0.02	0.20 ± 0.03	0.011
37M	19.64 ± 0.23	0.20 ± 0.00	-0.18 ± 0.00	8.68 ± 0.03	0.24 ± 0.03	0.015
78M	20.66 ± 0.25	0.21 ± 0.00	-0.19 ± 0.00	8.69 ± 0.03	0.29 ± 0.03	0.014
144M	20.31 ± 0.26	0.21 ± 0.00	-0.19 ± 0.00	8.71 ± 0.03	0.34 ± 0.03	0.015
285M	20.85 ± 0.30	0.22 ± 0.00	-0.20 ± 0.00	8.57 ± 0.03	0.44 ± 0.03	0.013
472M	21.16 ± 0.32	0.23 ± 0.00	-0.19 ± 0.00	8.44 ± 0.03	0.39 ± 0.04	0.014
OLMo-1B	25.97 ± 0.38	0.27 ± 0.00	-0.23 ± 0.00	8.22 ± 0.03	0.76 ± 0.02	0.008
OLMo-7B	27.49 ± 0.48	0.28 ± 0.00	-0.22 ± 0.00	8.44 ± 0.04	0.76 ± 0.03	0.008

```python

import numpy as np
import scipy

def log_1b_bnsl(xlog, b, c0, c1, d1log, f1):
    ylog_pred = np.log(b) - c0*xlog - (c1*f1)*np.log(1+np.exp((xlog-d1log)/f1))
    return ylog_pred

def fit_1b_bnsl(x: np.ndarray, y: np.ndarray, d1_est: float = 6000):
    # initialize parameters with reasonable values (for stability)
    d1log = np.log(d1_est)
    xlog = np.log(x)
    ylog = np.log(y)
    d1_idx = np.argmin(np.abs(xlog - d1log))
    c0 = -np.mean((ylog[0:d1_idx] - ylog[1:d1_idx+1]) \
            / (xlog[0:d1_idx] - xlog[1:d1_idx+1]))
    c1 = -np.mean((ylog[d1_idx:-2] - ylog[d1_idx+1:-1]) \
            / (xlog[d1_idx:-2] - xlog[d1_idx+1:-1]))
    c1 = c1 - c0
    b = ylog[0] + c0*xlog[0]

    # fit parameters with scipy
    p0 = [b, c0, c1, d1log, 0.3]
    popt, pcov = scipy.optimize.curve_fit(
        log_1b_bnsl,
        xlog, ylog,
        p0=p0,
        method='dogbox',
    )

    return popt, pcov
```

Code 1: Code for fitting one-break BNSL.

B Related works

This work connects several existing areas of research. In particular, several recent works attempt to explain scaling laws, typically from the perspective of intrinsic model capacity, long-tailed data distributions, and asymptotic behaviour (e.g. Hutter, 2021; Sharma and Kaplan, 2022; Michaud et al., 2023; Bahri et al., 2024; Bordelon et al., 2024). In contrast, our goal is to identify a mechanism grounded in training dynamics that can be targeted independent of scale. The mechanism we identify, loss deceleration, is to the best of our knowledge not addressed in relevant prior works on e.g. loss plateaus (Yoshida and Okada, 2020), learning curve shapes (Viering and Loog, 2022), or LM saturation (Godey et al., 2024; Mircea et al., 2024). Lastly, the study of training dynamics based on per-example gradient interactions remains under-explored, with related tangential works on e.g. improving multi-task learning (Liu et al., 2021), or characterizing outliers in SGD (Rosenfeld and Risteski, 2023).

Explaining scaling laws Several works have proposed different explanations for neural scaling laws such as (Kaplan et al., 2020; Hoffmann et al., 2022; Caballero et al., 2023; Hägele et al., 2024; Tissue et al., 2024; Everett et al., 2024). Notably, (Bahri et al., 2024) explain scaling laws in terms of asymptotic behaviour, 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 (Sharma and Kaplan, 2020)). (Atanasov et al., 2024) analytically explain power-law scaling in high-dimensional ridge regression with tools from random matrix theory. (Michaud et al., 2023) 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, (Hutter, 2021) 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 (Marion et al., 2023; Sorscher et al., 2022) model distillation (Allen-Zhu and Li, 2023; Team et al., 2024) and model pruning (Raposo et al., 2024) 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, Rosenfeld and Risteski, 2023 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. (Parascandolo et al., 2020; Yu et al., 2020; Liu et al., 2021). Gradient opposition between tokens in language modeling has, to the best of our knowledge, not been characterized. Related but distinct, is the work of (Mircea et al., 2024) 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 (Viering and Loog, 2022) for a comprehensive review of learning curve shapes, as well as (Hutter, 2021) and (Yoshida and Okada, 2020) as examples of attempting to explain features in a learning curve.

C Additional Results

C.1 Consistency of Loss Deceleration Findings with Learning Rate Decay

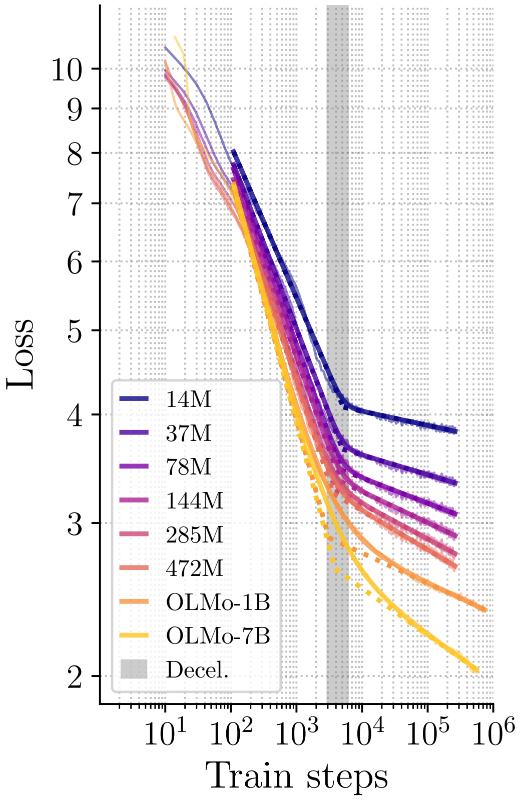

Fig. 11: Loss curves and BNSL fits when training with cosine lr decay.

Our main results are for training runs where learning rate was warmed up and held constant, in line with Hägele et al. (2024) and Wen et al. (2024). However, typically scaling experiments have been conducted with learning rate decay. In particular, Hoffmann et al. (2022) note that consistently decaying to 0.1 of the peak learning rate as an important difference to Kaplan et al. (2020), leading to different compute-optimal scaling. To rule out this potential confound, we replicate our experiments with a cosine learning rate decay in line with Hoffmann et al. (2022) (and Groeneveld et al. (2024)), leaving all else equal.

Fig. 1 is replicated in Fig. 11, with similar results and quality of fits. Table 4 is replicated in Table 6 with again similar results, and generally smaller values for c_1, $\log(d_1)$, and f_1. Lastly, Table 1 is replicated in Table 5, where we see that L_d resulting from the BNSL fit is increased, but this is offset by improved r_d and t_d, leading to better final loss. This improvement in final loss appears to increase with model size, suggesting a complementary mechanism by which scale improves loss under learning rate decay, which is not accounted for by our principal findings.

Table 5: Deceleration measurements with lr decay.

Model	L_d	r_d	t_d	$\hat{L}_T$	L_T
14M	4.08	0.016	5198	3.83	3.86
37M	3.65	0.023	5029	3.34	3.36
78M	3.45	0.029	4808	3.07	3.09
144M	3.35	0.036	4712	2.90	2.92
285M	3.28	0.040	3921	2.77	2.78
472M	3.24	0.045	3653	2.68	2.69
OLMo-1B	2.89	0.035	3106	2.39	2.38
OLMo-7B	2.66	0.054	3885	2.03	2.02

C.2 Language Model Scaling Law Grounded in Loss Deceleration

Defining and fitting a scaling law grounded in loss deceleration Let $L(N,T)$ be a scaling law for language model loss L, where N is the number of model parameters and T is number of training steps (with dataset size $D = T \cdot B$ for batch size B, i.e. single-epoch training). Recall from Eqn. 2 that an estimate of the loss L can be expressed in terms of the following parameters: **(1)** the number of steps at which deceleration occurs t_d; **(2)** the loss at which deceleration occurs L_d; and **(3)** the log-log rate of loss improvement after deceleration r_d. These parameters, shown in Table 1, are dependent on N, such that we can define a scaling law $L(N,T)$ grounded in loss deceleration as follows:

$$L(N,T) = L_d(N) \cdot t_d(N)^{r_d(N)} \cdot T^{\text{-}r_d(N)} \quad (13)$$

In Fig. 12, we observe, with the admittedly limited datapoints from our experiments, that L_d and r_d seem to exhibit power law scaling. In contrast, t_d appears to scale linearly if the outlier value for OLMo-7B, which is likely a result of being trained with 5,000 warmup steps instead of 2,000, is omitted. This suggests that warmup steps, among potentially other hyperparameters, have an important role not accounted for here. However, these results are preliminary and intended more as an exploratory proof of concept, included here for completeness, rather than a key result or claim of the paper. We leave the costly task of conducting sufficient training runs to more adequately validate this functional form for future work.

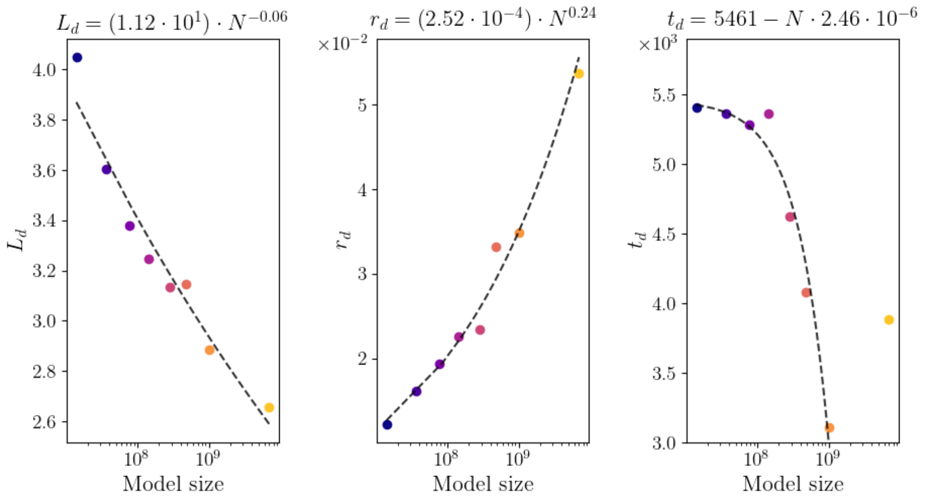

Fig. 12: Power law and linear scaling in deceleration parameters.

Table 6: BNSL parameters and error when training with cosine lr decay.

Model	b	c_0	c_1	$\log(d_1)$	f_1	RSLE
14M	18.32 ± 0.16	0.18 ± 0.00	-0.16 ± 0.00	8.56 ± 0.02	0.16 ± 0.03	0.012
37M	19.60 ± 0.22	0.20 ± 0.00	-0.17 ± 0.00	8.52 ± 0.02	0.18 ± 0.03	0.014
78M	20.67 ± 0.24	0.21 ± 0.00	-0.18 ± 0.00	8.48 ± 0.02	0.22 ± 0.03	0.014
144M	20.31 ± 0.25	0.21 ± 0.00	-0.18 ± 0.00	8.46 ± 0.03	0.24 ± 0.03	0.014
285M	20.87 ± 0.28	0.22 ± 0.00	-0.18 ± 0.00	8.27 ± 0.03	0.31 ± 0.03	0.013
472M	21.30 ± 0.29	0.23 ± 0.00	-0.18 ± 0.00	8.20 ± 0.03	0.31 ± 0.03	0.013
OLMo-1B	26.53 ± 0.42	0.28 ± 0.00	-0.24 ± 0.00	8.04 ± 0.03	0.76 ± 0.02	0.008
OLMo-7B	28.14 ± 0.54	0.29 ± 0.00	-0.23 ± 0.00	8.26 ± 0.04	0.78 ± 0.03	0.008

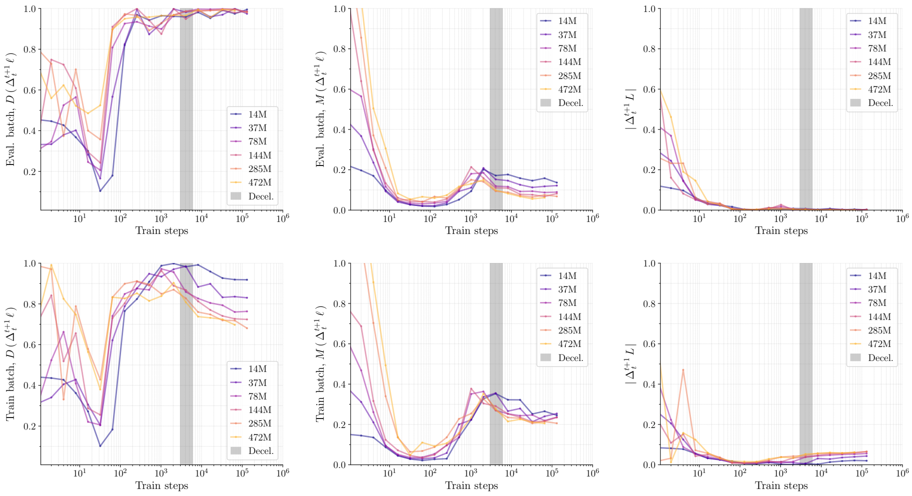

Fig. 13: Single-step ZSL in Train and Eval. batches.

C.3 Effect of Increasing Steps on ZSL

Destructive interference is mitigated by increasing number of steps. While the experiments and results in Section 3.1 consider the change in loss between steps t and $2t$, our initial experiments were based on checkpoints for steps $[1, 2, \ldots, 10, 20, \ldots, 100, 200, \ldots, 1000]$ and so on. When plotting $D(\Delta\ell_i)$ between these checkpoints in Fig. 14, we can see that $D(\Delta\ell_i)$ increases much more rapidly leading up to deceleration, when compared to Fig. 2. However, we also observe abrupt drops and subsequent rises in $D(\Delta\ell_i)$ after the number of steps between checkpoints is increased by a factor of 10. These results highlight that ZSL actually increases leading up to rather than after deceleration, but is mitigated by increasing number of steps.

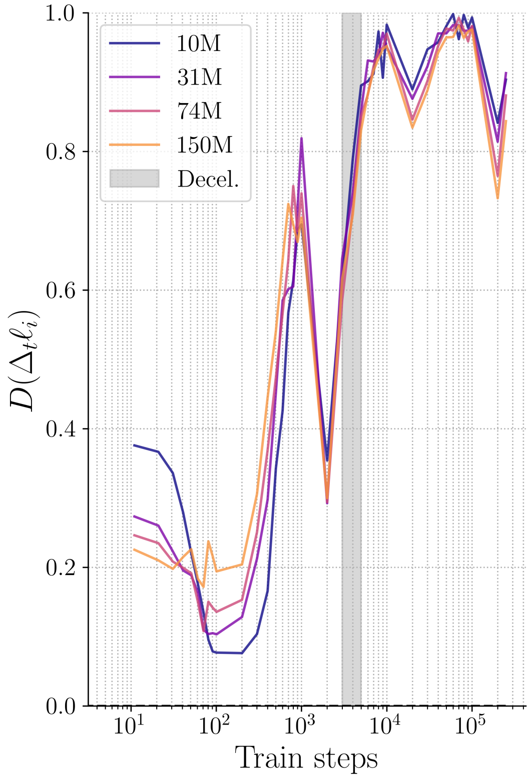

Fig. 14: Effect of number of steps (between changes in loss) on ZSL. Measurements are based on steps $[1, 2, \ldots, 10, 20, \ldots, 100, 200, \ldots, 1000]$ and so on. Drops in $D(\Delta\ell_i)$ correspond to points where steps between checkpoints increases by an order of magnitude.

C.4 Per-token loss landscape cross-sections

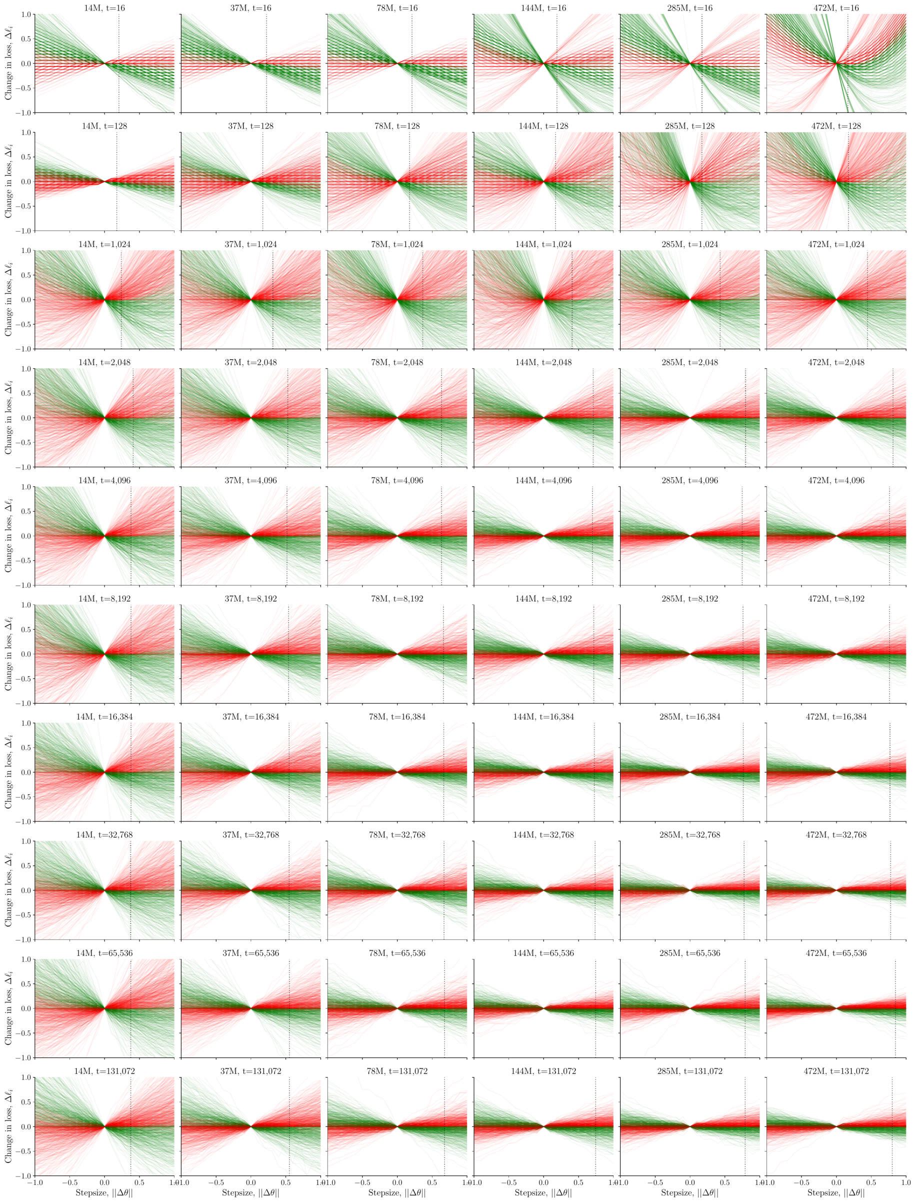

Fig. 15: Sampled per-token loss landscape cross-sections across model sizes and train steps Across model sizes (columns) and train steps (rows), we plot loss landscape cross-sections along increments of the weight update $\Delta\theta$ at step t. The actual stepsize is indicated with a dotted vertical line. We plot ΔL rather than L, which has the same geometry but allows more easily distinguishing loss improvements from degradations. Lines are colored in green or red depending on whether the loss (respectively) improved or deteriorated at the actual stepsize.

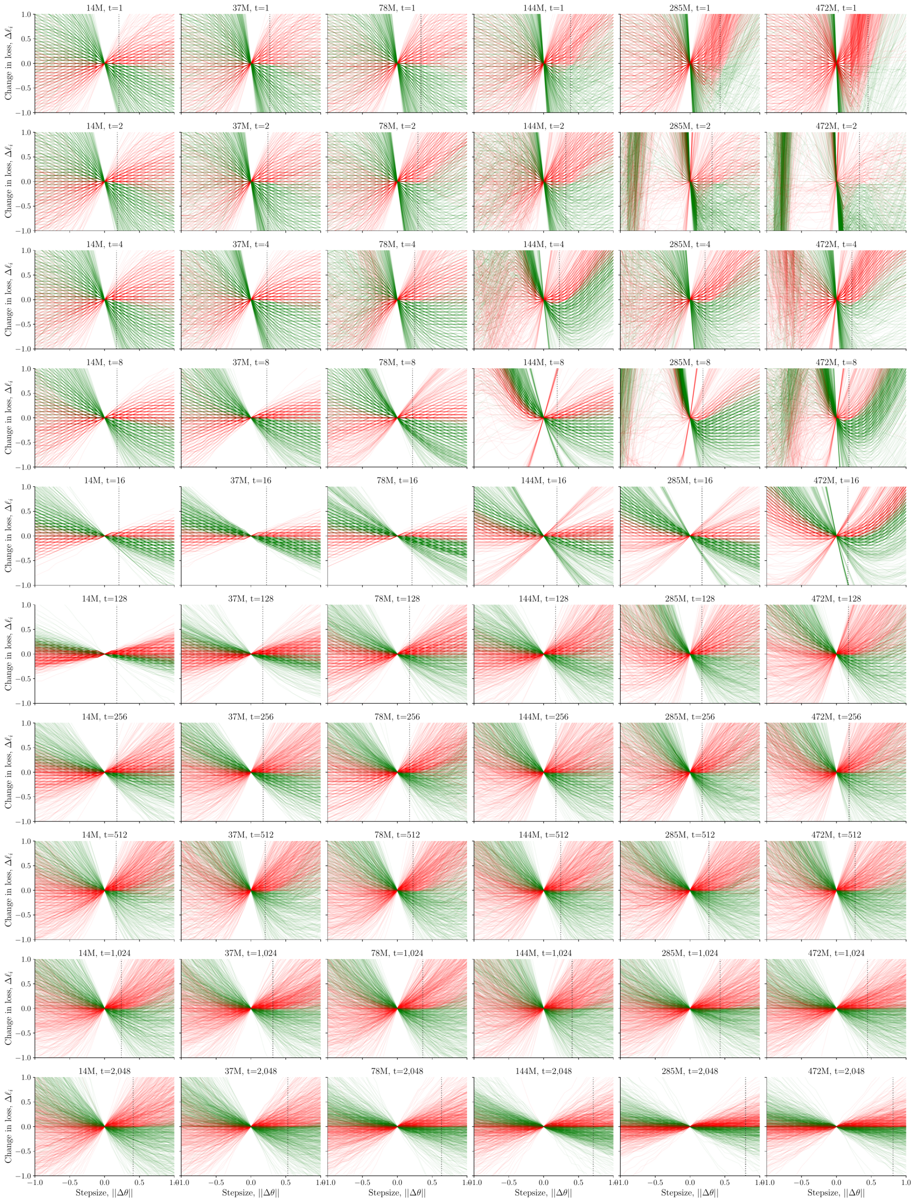

Fig. 16: Sampled per-token loss landscape cross-sections across model sizes at the start of training We plot the same data as in Fig. 15, but focused on the beginning of training (before deceleration).

C.5 Overall loss landscape cross-sections throughout training

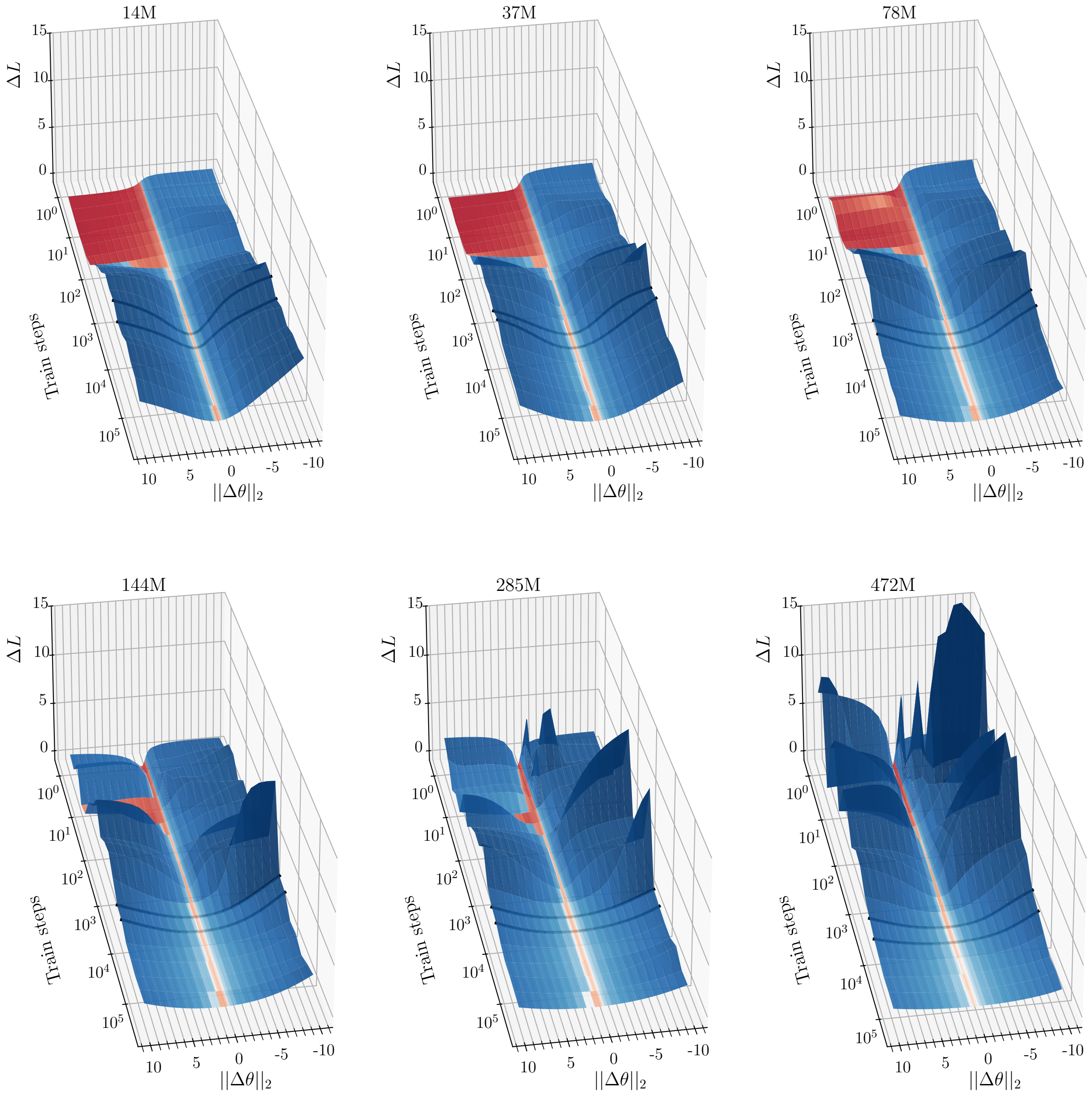

Fig. 17: Overall loss landscapes (cross section along $\Delta\theta$), visualized throughout training
We plot overall loss landscape cross sections across model sizes and train steps. Similar to Appendix C.4, we plot ΔL which has equivalent geometry to L but allows better distinguishing loss improvements 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.

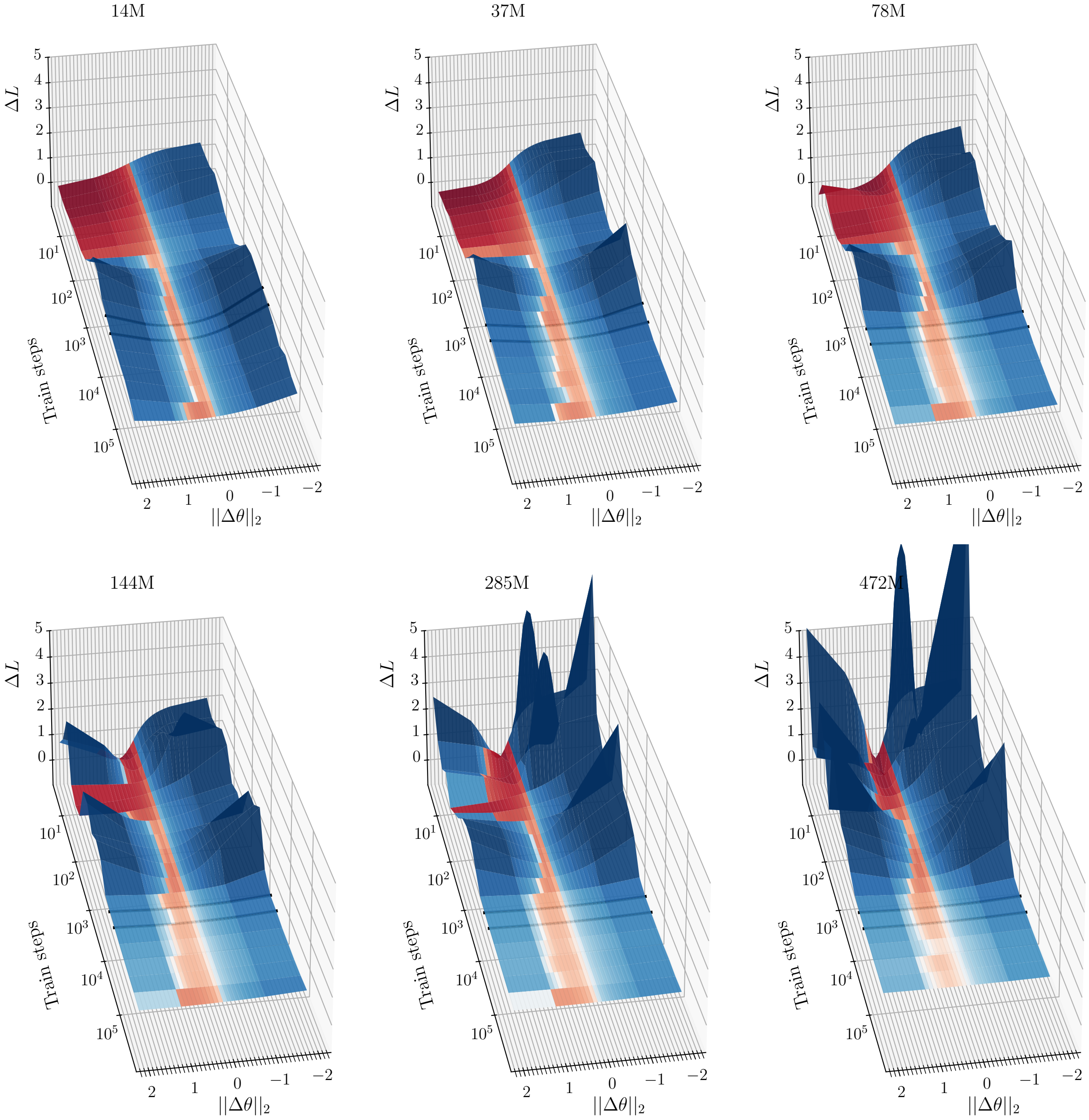

Fig. 18: Overall loss landscapes (cross section along $\Delta\theta$), visualized throughout training (zoomed in) We plot the same data as in Fig. 17, but zoomed into a narrower range.