Gradient Dissent in Language Model Training and Saturation

Andrei Mircea, Ekaterina Lobacheva, Irina Rish
Université de Montréal, Montréal, Canada; Mila – Quebec AI Institute, Montréal, Canada
{mirceara,ekaterina.lobacheva,irina.rish}@mila.quebec

Abstract

We seek to shed light on language model (LM) saturation from the perspective of learning dynamics. To this end, we define a decomposition of the cross-entropy gradient, which forms a shared low-dimensional basis for analyzing the training dynamics of models across scales. Intuitively, this decomposition consists of attractive and repulsive components that increase the logit of the correct class and decrease the logits of incorrect classes, respectively. Our analysis in this subspace reveals a phenomenon we term *gradient dissent*, characterized by gradient components becoming systematically opposed such that loss cannot be improved along one component without being degraded along the other. Notably, we find that complete opposition, which we term *total dissent*, reliably occurs in tandem with the saturation of smaller LMs. Based on these results, we hypothesize that gradient dissent can provide a useful foundation for better understanding and mitigating saturation.

1. Introduction

Understanding how language models (LMs) learn from data promises a foundation for progress driven by principled approaches. This paper considers the opposite but closely related question of *how LMs fail to learn from data*. Specifically, we aim to characterize the phenomenon of saturation, i.e., when LM validation loss stops improving with additional training data. On one hand, saturation may arise in smaller LMs due to intrinsic constraints on capacity (e.g., softmax bottleneck [12, 23] as proposed by Godey et al. [11]). On the other hand, we hypothesize saturation can result from degenerate but preventable training dynamics which constrain learning, as in gradient starvation [19].

To test this hypothesis, we consider the Pythia suite of LMs [3]. We show that improvement in validation loss is limited to a subset of tokens which correspond to the Zipfian long tail of the model's vocabulary. Surprisingly, smaller models abruptly saturate as performance stops improving for these tokens specifically. A significant challenge in exploring the learning dynamics of this phenomenon is the lack of shared interpretable basis along which different model sizes and token frequencies can be characterized and compared. To overcome this, we define one such low-dimensional latent subspace based on the linear decomposition of cross-entropy gradients into "attractive" and "repulsive" components, similar to [13]. Intuitively, these respectively correspond to increasing the logit of the correct prediction and decreasing the logits of incorrect predictions for a given example. Crucially, we can measure and compare how learning for different models and tokens aligns with these two objectives by projecting parameters, activations, or even gradients onto these directions throughout training. Notably, our analysis reveals a phenomenon which reliably occurs with saturation and may shed light on how it develops: *gradient dissent*. Dissent arises when attractive and repulsive components become systematically opposed such that loss cannot be improved along one component without being degraded along the other, suggesting a significant but mitigatable role in LM saturation.

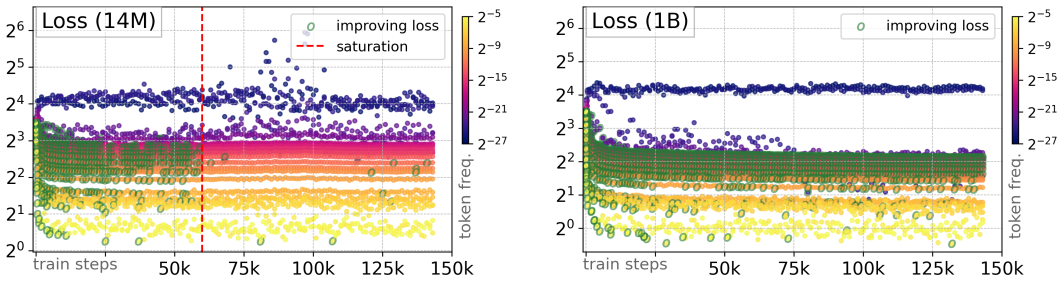

Figure 1: Validation losses on $\mathcal{V}$ throughout training, stratified by token frequency and plotted for 14M (left) and 1B (right) Pythia LMs. Highlighted points are improving, while the other are saturated. The red dotted line corresponds to the approximate *overall* model saturation threshold.

2. Saturation in Pythia LMs

We start our analysis by characterizing saturation in Pythia LMs [3]. In this work, we use the checkpoints made publicly available by the authors. In the main text, we provide the results for two models: a small 14M parameter LM, which exhibits saturation during training, and a large 1B parameter LM, which does not. The results for other model sizes can be found in Appendices C, D. We compute all measurements on a consistent set of context-target pairs $(\boldsymbol{c}, t) \in \mathcal{V}$, $|\mathcal{V}| = 2^{21} \approx 2\text{M}$ sampled from the validation set of The Pile [10] (see Appendix A for more details).

We informally define saturation as validation loss no longer improving with training. Instead of looking at the overall loss, we stratify our analysis by token frequency to show a more comprehensive picture of saturation across tokens. To achieve this, we bin tokens by the base-2 logarithm of their normalized frequency in the training dataset. In Figure 1, we show the behavior of the validation loss on the resulting token groups during training, highlighting the points that are still improving in green. We determine whether the point is improving by comparing it to all the previous points for the same token group and checking if the current loss value is lower than the best previous one plus 1% (for stability). We refer to all non-improving points as saturated.

The qualitative behavior of validation loss differs drastically across token frequencies but in a way that is mostly consistent across model scales. For instance, the most frequent tokens are characterized by a rapid decrease in validation loss, which remains lower than that of the less frequent tokens throughout training, but effectively saturates after only a few thousand steps. In contrast, the rarest tokens just as rapidly increase in loss, converging to a negative log-likelihood of ~ 16 (i.e., a probability on the order of one in ten million). However, the main difference between small and large models lays in the behavior of the loss on the tokens in the Zipfian long-tail (i.e., tokens of medium frequencies between $\sim 2^{-12} - 2^{-20}$). Large models consistently improve the loss on these tokens throughout the training process, while small models saturate on them in the middle of training. As a result, small models stop improving on almost all the data and exhibit the *overall* model saturation, which we mark approximately with the dotted red line on the plots. The occurrence and timing of the model saturation we observe closely match the transitions reported in Godey et al. [11].

Overall, the stratified analysis reveals that saturation in LMs is not a homogeneous phenomenon, and in fact, is characterized by qualitatively different behaviors across token frequencies. Moreover, our results suggest that the *overall* model saturation is primarily attributable to saturation in the tokens from the Zipfian long tail of the model vocabulary.

3. Gradient decomposition for training dynamics analysis

Since model saturation is a phenomenon occurring during the training process, we propose to analyze the dynamics of the training itself to shed more light on *how* and *why* the saturation appears. Both model parameters and gradients are extremely high-dimensional, hence, we can either measure some general scalar properties during the training process, such as validation loss we looked at in Section 2, or choose a low-dimensional subspace to project them to. In this Section, we describe the linear decomposition of the gradient similar to [13], which allows us to analyze the gradient training dynamics in terms of relative behavior of the overall gradient and its two components.

Generally, an LM is a model which takes *context* $\boldsymbol{c} = (c_1, ..., c_s)$, $c_i \in [1, v]$ indexes the model's v-token vocabulary, as an input and outputs the probability distribution $\boldsymbol{p} \in \mathbb{R}^v$ for the next token using the following computations on the forward pass:

$$\boldsymbol{h} = \mathcal{F}(\boldsymbol{c}, \theta), \quad \boldsymbol{\ell} = U^T \boldsymbol{h}, \quad p_i = \exp(\ell_i)/Z\ , \qquad Z = \sum\nolimits_{j=1}^{v} \exp(\ell_j), \tag{3.1}$$

where $\mathcal{F}$ is a feature-extracting part of the model, $\boldsymbol{h} \in \mathbb{R}^d$ is a final hidden representation, $\boldsymbol{\ell} \in \mathbb{R}^v$ is a vector of model logits, θ and $U \in \mathbb{R}^{d \times v}$ are the learnable model parameters. During training, the cross-entropy loss $\mathcal{L}(\boldsymbol{c}, t) = -\ell_t + \log(Z)$ averaged over a set of training examples $(\boldsymbol{c}, t) \in \mathcal{X}$, where $t \in [1, v]$ is the true next token observed for $\boldsymbol{c}$, is usually minimized.

For a given training example, the gradient of the cross-entropy with respect to the logits vector $\boldsymbol{\ell}$ can be linearly decomposed into two components, corresponding to the correct token t and to all other incorrect ones respectively:

$$\frac{\partial \mathcal{L}(\boldsymbol{c}, t)}{\partial \boldsymbol{\ell}} = \nabla^{(+)}_{\boldsymbol{\ell}} + \nabla^{(-)}_{\boldsymbol{\ell}} = (\dots, 0, \nabla^{(+)}_{\ell_t}, 0, \dots) + (\dots, \nabla^{(-)}_{\ell_{t-1}}, 0, \nabla^{(-)}_{\ell_{t+1}}, \dots). \tag{3.2}$$

Due to the linearity of this decomposition, gradients with respect to any model parameter or hidden representation can also be decomposed into the same two parts $\nabla^{(+)}$ and $\nabla^{(-)}$ by the chain rule:

$$\nabla^{(+)}_{\boldsymbol{\ell}_i} = \boldsymbol{p}_i - 1; \quad \nabla^{(+)}_{U_i^T} = \boldsymbol{h}\nabla^{(+)}_{\ell_i}; \quad \nabla^{(+)}_{\boldsymbol{h}} = U_i^T \nabla^{(+)}_{\ell_i}; \quad \nabla^{(+)}_{\theta} = \tfrac{\partial \ell_i}{\partial \theta} \nabla^{(+)}_{\ell_i}; \qquad \text{for } i = t \tag{3.3}$$

$$\nabla^{(-)}_{\boldsymbol{\ell}_j} = \boldsymbol{p}_j; \qquad \nabla^{(-)}_{U_j^T} = \boldsymbol{h}\nabla^{(-)}_{\ell_j}; \quad \nabla^{(-)}_{\boldsymbol{h}} = \sum_{j \neq t} U_j^T \nabla^{(-)}_{\ell_j}; \quad \nabla^{(-)}_{\theta} = \sum_{j \neq t} \tfrac{\partial \boldsymbol{\ell}_j}{\partial \theta} \nabla^{(-)}_{\ell_j}; \quad \text{for } j \neq t \tag{3.4}$$

Intuitively, $\nabla^{(+)}$ tries to increase the logit for the correct token prediction, while $\nabla^{(-)}$ tries to decrease all other logits. Doing so, they influence the parameters and hidden representations differently due to the opposite signs of the $\nabla^{(+)}_{\ell_t}$ and $\nabla^{(-)}_{\ell_j}$. Component $\nabla^{(+)}$ pulls any parameter or hidden representation towards the corresponding gradient of the logits (Eq. 3.3), therefore, we call it an *attractive* gradient. In contrast, we call the second component $\nabla^{(-)}$ a *repulsive* gradient since it pushes any parameter or hidden representation from the corresponding gradient of the logits (Eq. 3.4). For example, the final hidden representation $\boldsymbol{h}$ and the correct token unembedding vector U_t^T are pulled closer together by their attractive gradients, while $\boldsymbol{h}$ and all other unembedding vectors $U_j^T, j \neq t$ are pushed away from each other by their repulsive gradients.

We note that our gradient decomposition applies to a single training example. In contrast, prior works [19, 21] consider gradient decompositions from the perspective of different latent features and training examples. Also, since the two gradient components have the same meaning for any training step and any model size, we can compare the checkpoints of models of different sizes at different training steps in terms of the relation between them and the overall gradient.

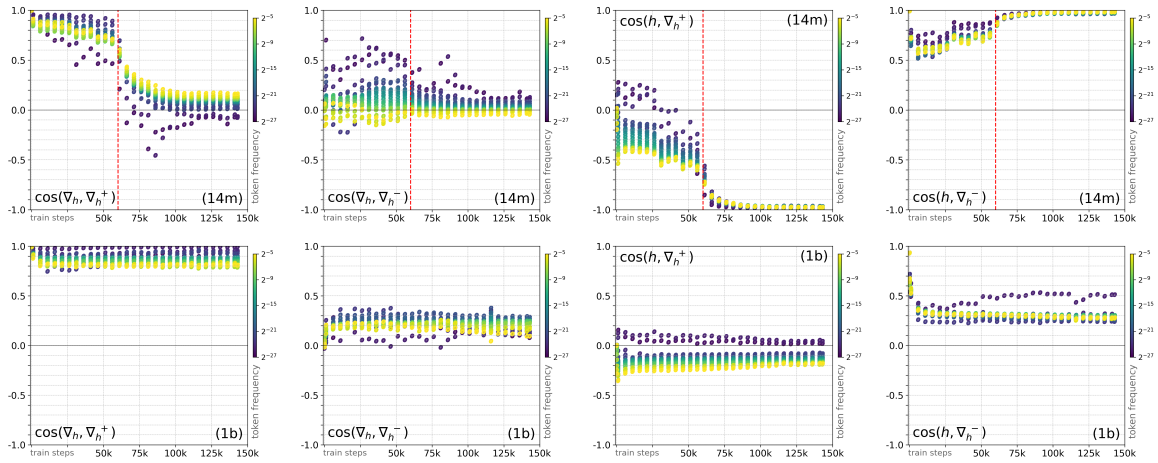

Figure 2: Training dynamics analysis for $\boldsymbol{h}$ for small 14M (top) and large 1B (bottom) models. Red dotted lines correspond to the approximate *overall* model saturation threshold from Figure 1.

4. Analysis of training dynamics of Pythia LMs

In this section, we focus our analysis on learning dynamics in final hidden representations $\boldsymbol{h}$, discussing how these results propagate to other model parameters in Appendix E. In Figure 2, we show how the overall gradient $\nabla_{\boldsymbol{h}}$ and the final hidden representation $\boldsymbol{h}$ itself progress in the attractive-repulsive space during training of a saturating 14M and non-saturating 1B parameter LMs, with similar plots shown in Appendix D for other model sizes. Consistent with Section 2, we stratify our analysis by token frequency to account for systematic differences in $\nabla_{\boldsymbol{h}}^{(+)}$ and $\nabla_{\boldsymbol{h}}^{(-)}$ which may arise due to frequency biases in token representations U_i^T (see Appendix B).

For the large model, the overall gradient is partially aligned with both components throughout training, hence, the optimization process can simultaneously increase and decrease the magnitude of correct and incorrect logits respectively. However, a much higher alignment with the attractive component indicates that optimization is dominated by the former at the expense of the latter. As a result, by going in the direction of the anti-gradient, optimization makes hidden representations $\boldsymbol{h}$ more opposed to the attractive gradient and more aligned with the repulsive gradient.

For the smaller model, initial training dynamics are similar but present a more pronounced difference between the attractive and repulsive components. Notably, there is less alignment between the overall gradient and its repulsive component, and even slight opposition for some token frequency groups. This structure in the gradient suggests that, in the case of limited model capacity, there is some level of opposition between attractive and repulsive gradients, which results in destructive interference between them. Strikingly, at some point, we observe a transition as the overall gradient loses its alignment with the attractive component too, and the hidden representation become strongly aligned with the attractive gradient and strongly opposed with the repulsive one. This alignment and opposition between hidden representations and their two gradient components increases our confidence in the possibility that opposition between attractive and repulsive gradients results in destructive interference. Moreover, the timing of the transition correlates well with the saturation transition previously identified in Section 2 for all model sizes (see Appendix D). Altogether, these observations lead us to the gradient dissent hypothesis, which we discuss in the next section in detail.

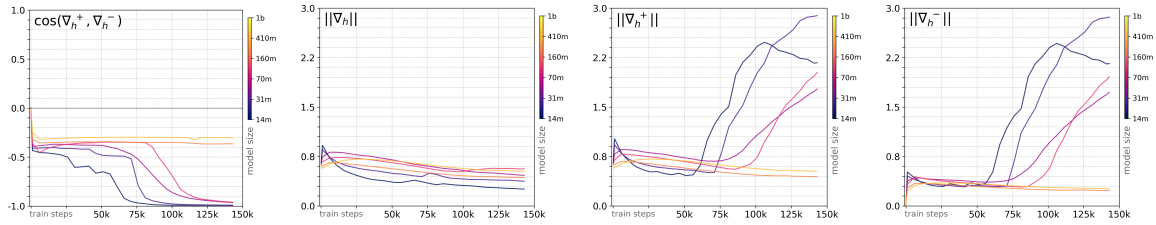

Figure 3: Gradient dissent for $\boldsymbol{h}$: for small models, attractive and repulsive gradients become large and completely opposed without the significant change in the norm of the overall gradient.

5. Gradient dissent hypothesis

The learning dynamics identified in Section 4 and Appendix D suggest that attractive and repulsive gradients $\nabla_{\boldsymbol{h}}^{(+)}$ and $\nabla_{\boldsymbol{h}}^{(-)}$ become systematically and fully opposed in smaller models, at the same points where the models saturate (see Section 2 and Appendix C). In light of these results, we define *gradient dissent* as a phenomenon where additive components of a gradient become systematically opposed during training and interfere destructively with one another. In the case of complete opposition, which we term *total dissent*, this can cause the overall gradient to approach zero despite individual components being non-zero and potentially even quite large. Intuitively, this corresponds to a suboptimal minimum where the loss cannot be improved along one gradient component without being degraded along another, similar to [19, 21].

In Figure 3, we confirm that total dissent arises simultaneously with saturation, characterized by $\nabla_{\boldsymbol{h}}^{(+)}$ and $\nabla_{\boldsymbol{h}}^{(-)}$ becoming fully opposed and exploding in norm while the overall gradient $\nabla_{\boldsymbol{h}}$ norm shrinks. As model size increases, saturation and total dissent are consistently delayed and do not appear for larger non-saturing LMs. Furthermore, in Appendix E, we show that total dissent simultaneously occurs in model parameters through backpropagation. However, the opposition between attractive and repulsive gradients (for both hidden states and parameters) remains surprisingly stable and negative for unsaturated models, suggesting that moderate degrees of gradient dissent are not always indicative of degenerate training dynamics. Whether learning in LMs occurs despite or on account of gradient dissent remains an open question. Based on these preliminary findings, we hypothesize that directly mitigating gradient dissent may reduce constraints on learning dynamics and delay saturation in smaller LMs, potentially even improving LM performance and training efficiency across scales.

6. Conclusion and future work

By decomposing LM gradients into interpretable additive terms, we create a unified low-dimensional latent subspace to study and compare LM learning dynamics across scales. In the resulting analysis, we uncover a phenomenon we term *gradient dissent*, which we hypothesize can lead to degenerate learning dynamics. We show that gradient dissent occurs systematically across training and model scales in Pythia LMs but that, in its most extreme form, it is strongly associated with the emergence of saturation in smaller LMs. While we characterize *how* dissent and saturation co-occur, *why* it happens remains an open question. Furthermore, our work provides a foundation for future work on mitigating saturation and studying training dynamics from the perspective of gradient dissent.

Acknowledgments

We acknowledge support from the Canada CIFAR AI Chair Program [I.R.] and the Canada Excellence Research Chairs Program [I.R.], as well as the NSERC post-graduate doctoral (PGS D) scholarship [A.M.] and the FRQNT Doctoral (B2X) doctoral scholarship [A.M.]. We would also like to thank Mila (mila.quebec) and its IDT team for providing and supporting the computing resources used in this work.

References

[1] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. 2017. URL https://openreview.net/forum?id=SyK00v5xx.

[2] Stella Biderman, Kieran Bicheno, and Leo Gao. Datasheet for the Pile, January 2022. URL http://arxiv.org/abs/2201.07311. arXiv:2201.07311 [cs].

[3] Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling, April 2023. URL http://arxiv.org/abs/2304.01373. arXiv:2304.01373 [cs].

[4] Daniel Biś, Maksim Podkorytov, and Xiuwen Liu. Too Much in Common: Shifting of Embeddings in Transformer Language Models and its Implications. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5117–5130, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.403. URL https://aclanthology.org/2021.naacl-main.403.

[5] Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. GPT-NeoX-20B: An Open-Source Autoregressive Language Model, April 2022. URL http://arxiv.org/abs/2204.06745. arXiv:2204.06745 [cs].

[6] Haw-Shiuan Chang and Andrew McCallum. Softmax Bottleneck Makes Language Models Unable to Represent Multi-mode Word Distributions. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8048–8073, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.554. URL https://aclanthology.org/2022.acl-long.554.

[7] 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. URL http://arxiv.org/abs/2103.00065. arXiv:2103.00065 [cs, stat].

[8] Matthew Finlayson, John Hewitt, Alexander Koller, Swabha Swayamdipta, and Ashish Sabharwal. Closing the Curious Case of Neural Text Degeneration, October 2023. URL http://arxiv.org/abs/2310.01693. arXiv:2310.01693 [cs].

[9] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. Representation Degeneration Problem in Training Natural Language Generation Models. September 2018. URL https://openreview.net/forum?id=SkEYojRqtm.

[10] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The Pile: An 800GB Dataset of Diverse Text for Language Modeling, December 2020. URL http://arxiv.org/abs/2101.00027. arXiv:2101.00027 [cs].

[11] Nathan Godey, Éric de la Clergerie, and Benoît Sagot. Why do small language models underperform? Studying Language Model Saturation via the Softmax Bottleneck, April 2024. URL http://arxiv.org/abs/2404.07647. arXiv:2404.07647 [cs].

[12] Sekitoshi Kanai, Yasuhiro Fujiwara, Yuki Yamanaka, and Shuichi Adachi. Sigsoftmax: Reanalysis of the Softmax Bottleneck, May 2018. URL http://arxiv.org/abs/1805.10829. arXiv:1805.10829 [cs, stat].

[13] Kian Kenyon-Dean, Andre Cianflone, Lucas Page-Caccia, Guillaume Rabusseau, Jackie Chi Kit Cheung, and Doina Precup. Clustering-Oriented Representation Learning with Attractive-Repulsive Loss, December 2018. URL http://arxiv.org/abs/1812.07627. arXiv:1812.07627 [cs, stat].

[14] Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. Datasets: A Community Library for Natural Language Processing. In Heike Adel and Shuming Shi, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-demo.21. URL https://aclanthology.org/2021.emnlp-demo.21.

[15] Tomas Mikolov, Kai Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. January 2013. URL https://www.semanticscholar.org/paper/Efficient-Estimation-of-Word-Representations-in-Mikolov-Chen/f6b51c8753a871dc94ff32152c00c01e94f90f09.

[16] Jiaqi Mu and Pramod Viswanath. All-but-the-Top: Simple and Effective Postprocessing for Word Representations. February 2018. URL https://openreview.net/forum?id=HkuGJ3kCb.

[17] Dwarak Govind Parthiban, Yongyi Mao, and Diana Inkpen. On the Softmax Bottleneck of Recurrent Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13640–13647, May 2021. ISSN 2374-3468. doi: 10.1609/aaai.v35i15.17608. URL https://ojs.aaai.org/index.php/AAAI/article/view/17608. Number: 15.

[18] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, December 2019. URL http://arxiv.org/abs/1912.01703. arXiv:1912.01703 [cs, stat].

[19] Mohammad Pezeshki, Sékou-Oumar Kaba, Yoshua Bengio, Aaron Courville, Doina Precup, and Guillaume Lajoie. Gradient Starvation: A Learning Proclivity in Neural Networks, November 2021. URL http://arxiv.org/abs/2011.09468. arXiv:2011.09468 [cs, math, stat].

[20] Steven T. Piantadosi. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21(5):1112–1130, October 2014. ISSN 1069-9384. doi: 10.3758/s13423-014-0585-6. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4176592/.

[21] Elan Rosenfeld and Andrej Risteski. Outliers with Opposing Signals Have an Outsized Effect on Neural Network Optimization, November 2023. URL http://arxiv.org/abs/2311.04163. arXiv:2311.04163 [cs, stat].

[22] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. HuggingFace's Transformers: State-of-the-art Natural Language Processing, July 2020. URL http://arxiv.org/abs/1910.03771. arXiv:1910.03771 [cs].

[23] Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. Breaking the Softmax Bottleneck: A High-Rank RNN Language Model, March 2018. URL http://arxiv.org/abs/1711.03953. arXiv:1711.03953 [cs].

[24] Sangwon Yu, Jongyoon Song, Heeseung Kim, Seong-min Lee, Woo-Jong Ryu, and Sungroh Yoon. Rare Tokens Degenerate All Tokens: Improving Neural Text Generation via Adaptive Gradient Gating for Rare Token Embeddings, June 2022. URL http://arxiv.org/abs/2109.03127. arXiv:2109.03127 [cs].

Appendix A. Experimental setup details

Pythia LMs We use model checkpoints made publicly available by Biderman et al. [3] through the `transformers` library [22] and under the MIT License [2]. Specifically, we consider 14m, 31m, 70m, 160m parameter LMs which exhibit saturation during training, as well as 410m and 1b parameter LMs which do not. Each model shares the same vocabulary size $v = 50,254$ and have hidden state dimension $d = 128, 256, 512, 768, 1024, 2048$ respectively.

Data sampling and distributional properties Measurement throughout this work are computed on a consistent set of context-target pairs $(\boldsymbol{c}, t) \in \mathcal{V}$, $|\mathcal{V}| = 2^{21} \approx 2\text{M}$ sampled from the validation set of The Pile [10], made available through the `datasets` library [14] and under the MIT License [2].

In Figure 4, we measure the frequency distribution of tokens in $\mathcal{V}$ relative to the `pile-train` dataset on which Pythia LMs were trained. We find that just over 90% of tokens are represented in $\mathcal{V}$, with unsampled tokens largely belonging to a long tail of rare tokens (with less than 0.001% occurrence frequency throughout training). Interestingly, we find that while tokens initially follow a long-tailed Zipfian distribution typical of word occurrence in language [20], the use of byte-pair encoding tokenization [5] seems to result in a markedly increased rate of frequency decay for the bottom $\sim 5\%$ of tokens with normalized frequency $\leq 2^{-20}$.

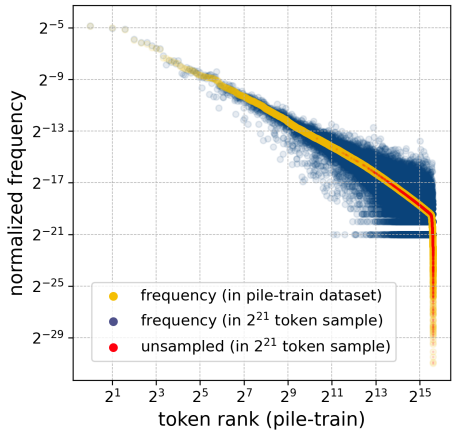

Figure 4: Frequency distribution of tokens in the `pile-train` dataset and validation set sample $\mathcal{V}$.

Token groups by frequency To account for the distributional properties highlighted in the previous section, we stratify our measurements and analysis by token frequency, binning tokens by the base-2 logarithm of their normalized frequency in the training dataset (Figure 4). This enables us to distinguish trends in saturation and learning dynamics for tokens which are frequent, in the Zipfian long-tail, or part of the non-Zipfian outlier subset of rarest tokens. Furthermore, based on prior work on representation degeneration other frequency biases in LM unembeddings (see Appendix B.2), we expect qualitatively different behavior for rare and frequent tokens, driven primarily by systematic differences in their unembedding vectors U_i^T.

Computing gradient components To compute attractive and repulsive gradients, we use pytorch [18], injecting a hook in the backward pass of the cross-entropy loss which masks logit gradients $\nabla_{\boldsymbol{\ell}_t}^{(+)}$ and $\nabla_{\boldsymbol{\ell}_j}^{(-)}, j \neq t$ respectively. Regular gradients are computed as usual using an unaltered cross-entropy loss. Hidden state gradients are computed and cached for each token individually, with associated measurements performed on the gradient of each token individually unless otherwise specified. In contrast and for reasons of tractability, parameter gradients are computed and averaged once over all tokens before caching, with associated measurements performed on the final gradient of individual parameter tensors unless otherwise specified.

Resources We use a single A100 GPU with experiments taking on average between 5 minutes and 2-3 days for one line on the plot for the smallest 14m and the largest 1b models respectively.

Appendix B. Related works

B.1. Softmax bottleneck

Since hidden size d is generally smaller than the vocabulary size v, the rank of U is at most d and bottlenecks the degrees of freedom in logits $\boldsymbol{\ell}$. Yang et al. [23] and Kanai et al. [12] show that the corresponding log probabilities $\log(\boldsymbol{p}) = \boldsymbol{\ell} - \log(Z) \cdot \mathbf{1}$ share the same low-rank bottleneck despite the non-linearity of softmax; preventing LMs from accurately representing the presumably high-rank matrix of "true" log probability distributions $P^*_{t,i} = p(t|c_i),\ P^* \in \mathbb{R}^{v \times |\boldsymbol{c}|}$ over all possible contexts $\boldsymbol{c}$. While several methods have been proposed to address this issue and increase the rank of LM log-probabilities, Parthiban et al. [17] show in extensive experiments that this increased rank is not sufficient nor necessary for previously reported performance improvements, attributing these to implicit regularization instead. Nevertheless, recent work has also linked the softmax bottleneck to more specific limitations, including the inability to model multi-mode token distributions [6], a propensity for neural text degeneration [8], and notably saturation in small language models [11].

B.2. Frequency bias in token representations

Arora et al. [1], Mu and Viswanath [16] found that word embedding models [e.g. 15] learn frequency-biased representations. Specifically, their first principal components are found to minimize the training objective by encoding frequency, but actually harm performance on downstream tasks. Gao et al. [9] identify a similar "representation degeneration" problem in LMs, whereby cross-entropy loss is minimized when output layer embeddings of rare tokens cluster in a narrow cone, limiting their representational capacity. Similarly, Biś et al. [4], Yu et al. [24] show that cross-entropy gradients in LMs lead to output layer embeddings clustering together for rare tokens.

B.3. Gradient starvation and opposition

Pezeshki et al. [19] introduce the concept of gradient starvation, whereby spurious features which correctly classify many examples are learned rapidly, limiting gradient signal and the model's ability to learn new more generalizable features on these examples. Similarly, Rosenfeld and Risteski [21] identify "opposing signals" as a phenomenon where groups of outlier examples have gradients which point in opposite directions, leading to oscillations as the model alternates between optimizing one group at the expense of another, eventually leading to loss spikes. In contrast to these works, gradient dissent is a fundamentally different phenomenon which considers an input-invariant decomposition of the gradient for a single example but generalizes to batch gradients as shown in Appendix E.

Appendix C. Saturation in Pythia LMs: all model sizes

In Figure 5, we show the behavior of validation loss on different frequency token groups for Pythia LMs of all considered model sizes, complementing Figure 1 from the main text. The main results, such as 1) variation of the loss behavior for the frequency token groups, and 2) the main role of the middle frequencies for the model saturation, are valid for all model sizes. The largest two models, 410M and 1B parameter ones, do not saturate. In contrast, all smaller models, 14M, 31M, 70M, and 160M parameter ones, exhibit saturation during training. Moreover, the larger the model, the later this transition occurs, closely matching the transition timings reported in Godey et al. [11].

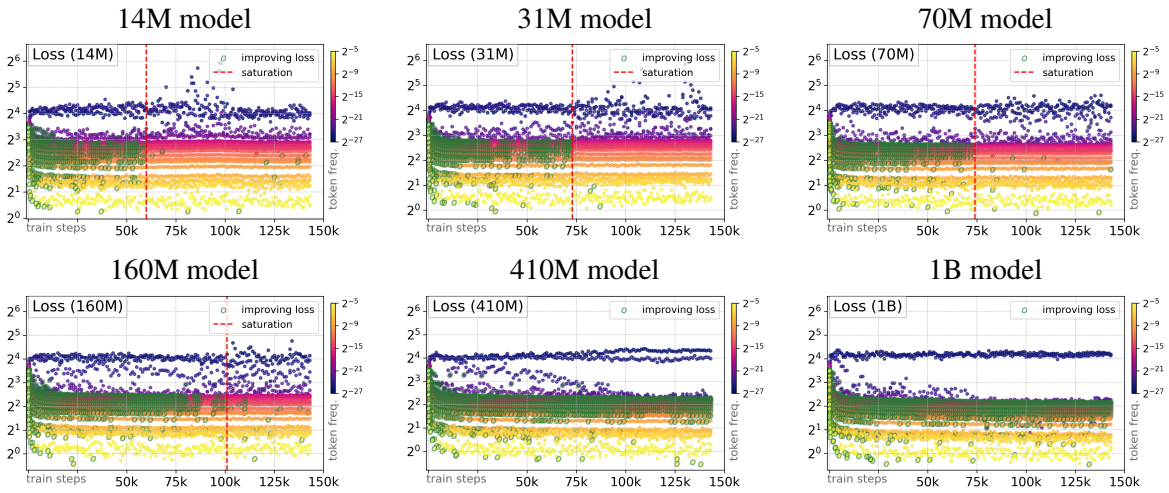

Figure 5: Validation losses on $\mathcal{V}$ throughout training, stratified by token frequency and plotted for Pythia LMs of different sizes. Highlighted points are improving, while the other are saturated. Red dotted lines correspond to the approximate *overall* model saturation threshold.

Appendix D. Gradient analysis for final hidden representations: all model sizes

In Figure 6, we show how the full gradient $\nabla_{\boldsymbol{h}}$ and the final hidden representations $\boldsymbol{h}$ evolve during training in the attractive-repulsive space for models of all considered sizes, complementing Figure 2 from the main text. The main differences between the small and large models behaviors described in the main text occur gradually with increasing model size. Notably, the transition in the gradient behavior appears only for the models which saturate (14M - 160M models). Moreover, the timing of the transition is closely related to the saturation threshold for the corresponding models in Figure 5.

Appendix E. Gradient analysis for model parameters

In the main text, we show that the attractive and repulsive gradients w.r.t. the final hidden representation $\boldsymbol{h}$ are opposed to some extend during training, with the emergence of total gradient dissent for small models. Intuitively, if the gradient dissent emerges in the final hidden states, then any parameter or activation θ of which $\boldsymbol{h}$ is a function of will have its gradient similarly affected via the chain rule:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \nabla_\theta^{(+)} + \nabla_\theta^{(-)} = \frac{\partial \boldsymbol{h}}{\partial \theta}\nabla_{\boldsymbol{h}}^{(+)} + \frac{\partial \boldsymbol{h}}{\partial \theta}\nabla_{\boldsymbol{h}}^{(-)}, \qquad \text{s.t. } \nabla_{\boldsymbol{h}}^{(+)} \approx -\nabla_{\boldsymbol{h}}^{(-)} \implies \nabla_\theta^{(+)} \approx -\nabla_\theta^{(-)}. \tag{E.1}$$

In Figure 7 we show that dissent in hidden states $\boldsymbol{h}$ indeed propagates to and in fact seems to become amplified in model parameters θ, with all models rapidly converging to almost complete opposition within the first thousand steps. Nevertheless, we observe the same transition as in hidden states with gradients in smaller saturating models becoming truly completely opposed with increasing norms relative to the overall gradient, again co-occurring with previous transitions in hidden states and in validation loss.

One key difference which may explain the discrepancy in magnitudes between hidden states and model parameters is the fact that hidden state gradients are computed individually for different

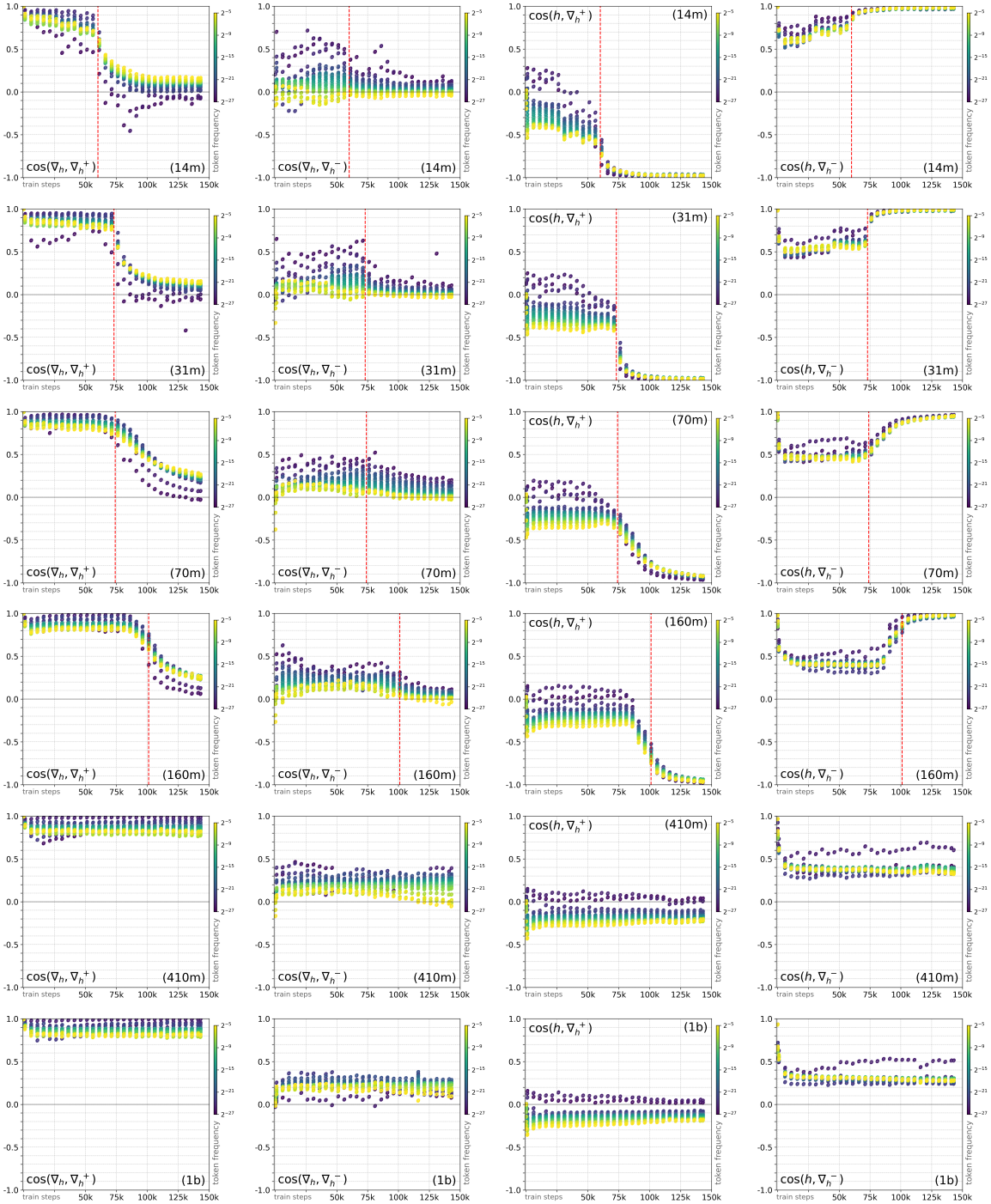

Figure 6: Training dynamics analysis for $\boldsymbol{h}$ for models of different sizes. Red dotted lines correspond to the approximate *overall* model saturation threshold form Figure 5.

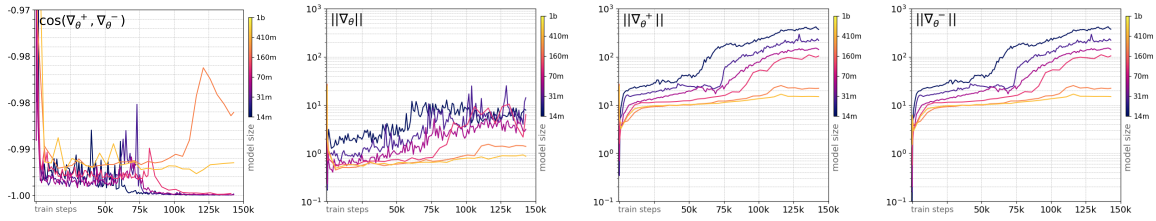

Figure 7: Gradient dissent for model parameters θ: the gradient opposition is even more pronounced than for the final hidden representation $\boldsymbol{h}$.

predictions, while model parameter gradients are pooled across all $\sim$ 2M examples in our validation set. In addition, the multiplicative contributions of parameter gradients during backpropagation may be magnifying gradients along these directions. However, the sheer magnitude of the opposition observed in gradients even for larger non-saturating models suggests that models can still learn despite having effectively dissenting gradients. Whether or not learning would improve or suffer from mitigating dissent in these cases is not clear and remains an open question for future work. Intuitively, dissenting gradients mean that the model cannot learn features which improve the loss along one gradient without harming the loss along the other. As a result, we hypothesize that gradient dissent induces training dynamics similar to the edge-of-stability phenomena identified by Cohen et al. [7] with learning defined by local instability as gradient dissent makes overall gradients inherently noisy; but global convergence as optimization still finds gradient signal in the noise.