Signal Frequency Imbalance and Ill-Conditioning

Tianyue H. Zhang TIANYUE.ZHANG@MILA.QUEBEC
Mila Quebec AI Institute, Université de Montréal, Canada

Francis Bach FRANCIS.BACH@INRIA.FR
Inria, France

Frederik Kunstner FREDERIK.KUNSTNER@INRIA.FR
Inria, France

Abstract

The source of the ill-conditioning addressed by Adam- and Muon-like optimizers remains poorly understood, making it unclear when and why they outperform SGD. We introduce a generalization of signal frequency imbalance that captures effective low-rank structure arising from correlations between weight-space directions and data subsets. Empirically, we show that this structure appears in the inner layers of language models as semantically meaningful gradient clusters, helping explain why Muon can outperform Adam. On a simplified problem, we show that signal frequency imbalance induces ill-conditioning only at large batch sizes, where minibatch averaging suppresses progress along rare directions, explaining why Adam and Muon outperform SGD only in this regime.

1. Introduction

The success of large language models has led to renewed attention to the optimization methods used to train them. Adam variants (Kingma and Ba, 2015) have been the default choice for training language models, but spectral normalization methods (Carlson et al., 2015) such as Muon (Jordan et al., 2024) have been shown to improve performance at scale (Liu et al., 2025; Shah et al., 2025; Bai et al., 2026). The benefit of spectral normalization is often attributed to the fact that the gradient and Hessian of the objective functions have a low effective rank, where a few top singular values contribute most of the mass, and spectral normalization should lead to more progress in directions with low signal (Davis and Drusvyatskiy, 2025). But many questions remain open about where this low-rank structure comes from and how it interacts with other design choices in large-scale training, as recent empirical observations have shown that Adam and Muon are often unnecessary at a small batch size and SGD performs well in this regime (Marek et al., 2025; Srećković et al., 2025).

We have evidence that Adam's benefit on language models stems specifically from the imbalance in word frequencies. This imbalance in word frequencies induces an imbalance in the gradient and Hessian magnitudes along directions associated with particular words. (Kunstner et al., 2024). Ablation studies have shown that one of the primary benefits of Adam is in correcting this imbalance in the first and last layer of the network (Zhao et al., 2025; Zhang et al., 2025b). This imbalance is axis-aligned, as the rows/columns of the weights of the first/last layer correspond to specific input/output words, and the sign-like normalization of Adam speeds up training in directions associated with low-frequency words (Kunstner and Bach, 2025). Muon-related algorithms provide similar benefits

© T.H. Zhang, F. Bach & F. Kunstner.

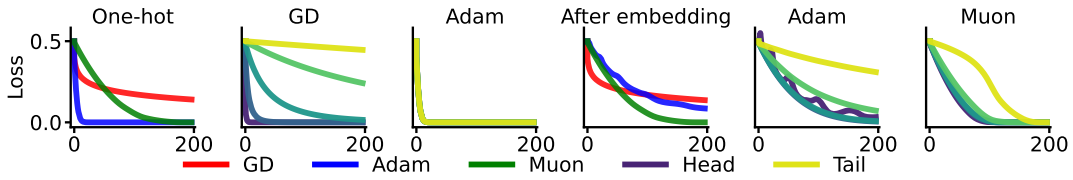

Figure 1: **Adam improves under axis-aligned frequency imbalance, Muon does not depend on axis alignment** Left: Under axis-aligned frequency imbalance (Example 1), Adam outperforms GD by converging faster on low-frequency directions (plot 1 shows overall train loss, plots 2 and 3 show the loss for different subsets of the data ordered by frequency). Right: After applying an embedding layer, the imbalance is no longer axis-aligned (Example 2). Adam is less effective while Muon still provides a significant improvement.

for inner layers of the network. Wang et al. (2025) show that Muon learns co-occurrence of words faster than Adam when the distribution of word pairs is imbalanced, and that this effect comes from using Muon in inner layers. This result suggests that data imbalance is also a problem in inner layers, where the coordinate-wise rescaling of Adam is less effective because the imbalance is not axis-aligned (Zhang et al., 2025a). But the picture is less clear than in the first/last layer since the lack of axis alignment also makes it harder to visualize differences across directions.

Contribution. This work is motivated by two questions: (i) why Muon outperforms Adam in the inner layers of a network, and (ii) why gaps between Muon, Adam, and SGD appear only at large batch sizes. To answer these questions, we propose a generalization of signal frequency imbalance to capture correlations between directions in weight space and subsets of the data. We show empirically that signal frequency imbalance appears in the inner layers of a language model, with semantically meaningful clusters of samples which share similar outputs or co-occurrence patterns. We show on a simplified problem that this form of imbalance leads to ill-conditioning only when the batch-size is large, explaining observations that Muon, Adam and SGD perform similarly at a small batch size.

2. Why the gap between SGD, Adam and Muon? Imbalance and axis-alignment

To illustrate frequency imbalance, consider the linear bigram model of Kunstner and Bach (2025).

Example 1 *Given data* $(\mathbf{x}, \mathbf{y}) \in \{0,1\}^{c \times c}$ *where* $\mathbf{x}, \mathbf{y}$ *are one-hot vectors over a vocabulary of size* c *drawn from* $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y} \mid x)$ *where* $p(\mathbf{x} = \mathbf{e}_i) \propto \pi_i$*, fit a linear model with the squared loss,*

$$\mathcal{L}(\mathbf{W}) = \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}} \left[\tfrac{1}{2} \|\mathbf{W}\mathbf{x} - \mathbf{y}\|_2^2 \right].$$

The Hessian for row $\mathbf{w}_j$ is $\nabla^2_{\mathbf{w}_j} \mathcal{L}(\mathbf{W}) = \mathrm{Diag}(\boldsymbol{\pi})$, the eigenvalues are the frequencies, so the problem is ill-conditioned if the π_i are imbalanced. The largest frequency limits the stable step-size for GD, while low frequencies lead to slow progress. Sign-like methods do not suffer from this problem and provably improve performance, as we observe in Figure 1 (left). But the benefit of sign normalization relies on the imbalance being axis-aligned. Suppose we remove this alignment by using an orthogonal matrix, representing an embedding layer.

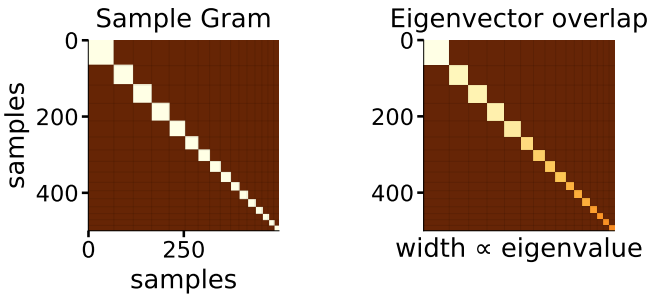

Figure 2: **Under frequency imbalance, the data Gram matrix has a block-diagonal structure and the eigenvectors of the weight Gram matrix correspond to clusters of samples.** In Example 1 at initialization, the sample Gram matrix exhibits a block-diagonal structure, with imbalanced clusters of samples with similar gradients (left, 10 largest blocks). Their projection onto the top 10 eigenvectors of the weight Gram matrix show that the largest directions of variation in the gradients correspond to the largest clusters of samples (right).

Example 2 *Let $E \in \mathbb{R}^{c\times c}$ be an orthogonal matrix applied to the inputs of Example 1:*

$$\mathcal{L}(\mathbf{W}) = \mathop{\mathbb{E}}_{x,y}\left[\tfrac{1}{2}\|\mathbf{W}\mathbf{E}\mathbf{x} - \mathbf{y}\|^2\right].$$

On this problem, the Hessian is $\nabla^2_{\mathbf{w}_j}\mathcal{L}(\mathbf{W}) = \mathbf{E}^\top \operatorname{Diag}(\boldsymbol{\pi})\mathbf{E}$. The problem still exhibits imbalance, but it is no longer axis-aligned due to the rotation $\mathbf{E}$. Sign normalization does not help, and instead, we have to rely on Muon to make more progress on low-frequency words, see Figure 1 (right).

3. Signal Frequency Imbalance

We propose the following simplified model to formalize signal frequency imbalance.

Definition 1 *A problem exhibits signal frequency clustering in weight $\mathbf{w}$ if the gradients for each of the n samples w.r.t. $\mathbf{w}$ can be clustered into distinct subsets $\mathcal{S}_1, ..., \mathcal{S}_k \subset [n]$ based on their cosine similarity. Let $\mathbf{g}_i, \mathbf{g}_j$ be the gradients of the loss for samples i, j with respect to $\mathbf{w}$. Then,*

$$|\cos(\mathbf{g}_i, \mathbf{g}_j)| \geq 1 - \epsilon \quad \text{if } i, j \text{ belong to the same cluster,} \quad |\cos(\mathbf{g}_i, \mathbf{g}_j)| \quad \leq \epsilon \text{ otherwise.}$$

We have signal frequency imbalance if the relative size of the clusters $\pi_i = |\mathcal{S}_i|/n$ are imbalanced.

For problems with this structure, we expect the following properties to hold, at least approximately.

Block-diagonal data Gram matrix and mapping between significant directions and clusters. Let $\mathbf{G} : \mathbb{R}^{n\times d}$ be the matrix of per-sample gradients for n samples on a weight $\mathbf{w} \in \mathbb{R}^d$. The data Gram matrix $\mathbf{G}\mathbf{G}^\top \in \mathbb{R}^{n\times n}$ can be reordered by a permutation matrix $\mathbf{P}$ such that $\mathbf{P}\mathbf{G}\mathbf{G}^\top\mathbf{P}^\top$ is approximately block diagonal, where the blocks correspond to clusters. The eigenvectors of the weight Gram matrix $\mathbf{G}^\top\mathbf{G} \in \mathbb{R}^{d\times d}$ will correspond to the blocks of the data Gram matrix $\mathbf{G}\mathbf{G}^\top$ and thus to clusters of samples. This structure holds for Example 1 at initialization.

Proposition 2 *Consider the linear bigram model of Example 1 at $\mathbf{W} = \mathbf{0}$ with a finite number of samples, where $n_{i,j}$ is the number of occurrences of $(x, y) = (i, j)$. The data Gram matrix is*

block-diagonal up to a permutation $\mathbf{P}$*, with block sizes proportional to the frequency of each bigram,*

$$\mathbf{P}\mathbf{G}\mathbf{G}^\top\mathbf{P}^\top = \mathrm{Diag}\left(\mathbf{J}_{n_{11}}, \mathbf{J}_{n_{12}}, \ldots, \mathbf{J}_{n_{dd}}\right), \quad \mathbf{J}_m \coloneqq \mathbf{1}_m\mathbf{1}_m^\top.$$

As $\mathbf{G}\mathbf{G}^\top$ *is exactly block-diagonal, the eigenvectors of the weight Gram matrix* $\mathbf{G}^\top\mathbf{G}$ *can also be partitioned into blocks corresponding to the clusters of samples.*

This structure is illustrated in Figure 2 and also applies to Example 2 up to a change of basis.

3.1. Impact of Frequency vs. Other Imbalance on Batch Size

The simplified model of signal frequency imbalance can provide some intuition for the recent empirical observation that although SGD performs poorly at a large batch size, it is competitive with Adam and Muon at small batch sizes (Marek et al., 2025; Srećković et al., 2025). Stochastic second-order methods are known to require large batches to accurately estimate a good preconditioner and provide little benefit at small batch sizes if gradient noise is the bottleneck and the optimizer reaches a noise floor (Bottou et al., 2018; Bollapragada et al., 2019; Zhang et al., 2019). Signal frequency imbalance points to another issue: ill-conditioning only becomes a problem when we have a large batch size because of *averaging* the gradients over a batch. To illustrate, we construct two problems with no noise floor and the same ill-conditioned Hessian that differ in their noise structure, and compare how the optimal fixed preconditioner depends on the batch size.

Example 3 *Consider the following quadratic problem with weights* $\mathbf{w} \in \mathbb{R}^d$ *and data* $\mathbf{x} \in \mathbb{R}^d$*,*

$$\mathcal{L}(\mathbf{w}) = \mathop{\mathbb{E}}_{\mathbf{x}}\left[\tfrac{1}{2}\|\mathbf{x}^\top\mathbf{w} - 1\|^2\right] \quad \textit{with} \quad \begin{array}{ll} \textit{Frequency imbalance:} & \mathbf{x} = \mathbf{e}_i \textit{ with probability } \pi_i \propto 1/i, \\ \textit{Magnitude imbalance:} & \mathbf{x} = d\sqrt{\pi_i}\mathbf{e}_i \textit{ with probability } 1/d. \end{array}$$

Both problems have Hessian $\nabla^2\mathcal{L}(\mathbf{w}) = \mathrm{Diag}(\boldsymbol{\pi})$, and have $\mathbb{E}_{\mathbf{x}}\left[\|\nabla\mathcal{L}(\mathbf{w})\|^2\right] = 0$ at the minimum. There is no noise floor and SGD converges even with a constant step-size. However, the behavior of the best preconditioner as a function of batch size depends heavily on the data distribution.

Proposition 3 *The optimal fixed preconditioner for the problem in Example 3, as a function of batch size* b*,* $\mathbf{P}^\star_{\mathrm{Freq}}$ *and* $\mathbf{P}^\star_{\mathrm{Magn}}$ *for the frequency and magnitude imbalance problems, respectively, are*

$$\mathbf{P}^\star_{\mathrm{Freq}}(b) = \mathrm{Diag}\left(\frac{1}{b} + \left(1 - \frac{1}{b}\right)\boldsymbol{\pi}\right)^{-1}, \qquad \mathbf{P}^\star_{\mathrm{Magn}}(b) = \mathrm{Diag}\left(\boldsymbol{\pi}\right)^{-1}.$$

Under magnitude imbalance, the conditioning does not depend on batch size, and can be fixed by a single preconditioner. Under frequency imbalance, with batch size one, the loss for a given sample $\mathbf{x}$ is perfectly conditioned; its eigenvalues are either 0 or 1. It is only as the batch size increases that the loss over a minibatch becomes ill-conditioned, as averaging suppresses updates in rare directions.

Interestingly, the usual Adam step-size scaling $\eta \propto \sqrt{b}$ (Malladi et al., 2022) is not observed in our experiments on this problem. Instead, the best step-size is independent of b (Section B). Understanding these regimes helps reconcile apparently contradictory observations (Li et al., 2024).

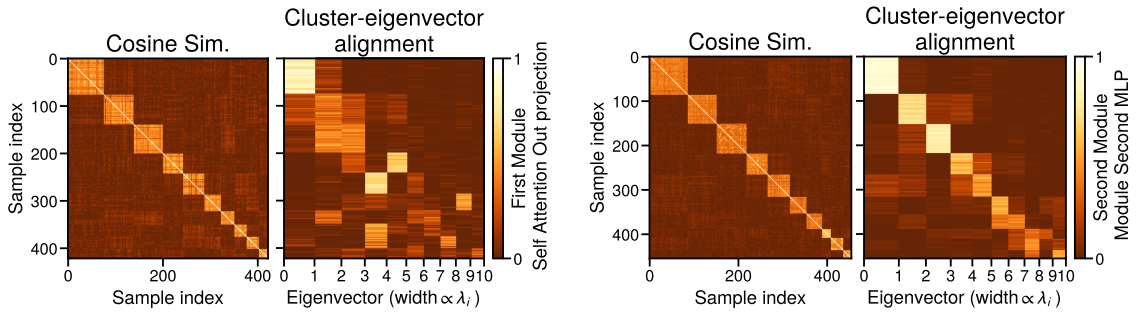

Figure 3: **Frequency imbalance occurs in inner layers.** Repeating the observation of Figure 2 for the inner layers of a 2-layer transformer trained on a language model. Left panels: after permutation, the sample Gram matrix of the gradients has a block-diagonal structure. Blocks of different sizes correspond to clusters of gradients pointing in almost-orthogonal directions. Right panels: the top eigenvectors of the weight Gram matrix $G^\top G$ of the gradients are primarily influenced by the largest sample clusters.

4. Signal frequency imbalance in inner layers of language models

To see whether signal frequency imbalance occurs in the inner layers of language models, we check whether the data and weight Gram matrices of the gradient exhibit the expected structure on a simple 2-layer transformer. Technical details for this section are detailed in Section C.

Identifying cluster structure. To reorder the data Gram matrix and reveal its block-diagonal structure, we use DBScan to cluster the gradients based on the absolute cosine similarity as the distance metric, and reorder the samples by their cluster assignment, ordered by cluster size.

Clusters and clusters-directions alignment. In Figure 3, we show the reordered data Gram matrix and the cluster-eigenvectors alignment for the output projection weight of the first attention module and the last linear layer of the last MLP block. While we only expect an exact block-diagonal structure on simplified problems (Figure 2), the block-diagonal structure is clear and each of the largest eigenvectors of the feature Gram matrix captures the direction of each of the largest clusters.

Semantic meaning of clusters and differences across layers. The clusters vary by layer and over time. The clusters in the last layer depend primarily on the output words, while clusters in layers closer to the input depend on co-occurrence patterns (Tables 1 to 4). We show examples of the data across the clusters and over time in Section C. The block-diagonal structure appears in the gradients of the linear layers of the MLP blocks, the value and output projection matrix of the attention blocks and the normalization layers. But the block-diagonal structure is not as obvious for the key and query matrices of the attention blocks. Interestingly, this observation matches the observation of Wang et al. (2025) that Muon does not provide as large an improvement on the key and query matrices.

5. Conclusion

We propose a definition of frequency imbalance that allows us to show the presence of frequency imbalance beyond the first and last layers of language models. Our preliminary results suggest that this frequency imbalance could be a good explanation for empirical observations on the performance

of normalized optimizers such as Adam and Muon. Their benefits over gradient descent come from a better performance in the presence of frequency imbalance, which otherwise suppresses progress along infrequent directions. This benefit is limited to axis-aligned imbalance for Adam, which explains the benefit of Muon in the inner layers observed by Wang et al. (2025). Frequency imbalance also leads to ill-conditioning at a large batch size, which explains that SGD, Adam and Muon all perform well at a small batch size. Further work is needed to formalize these arguments beyond simplified models, understand how it interacts with optimizer design choices, and investigate whether the frequency imbalance structure appears in tasks beyond language, such as in vision tasks.

Acknowledgment

Tianyue H. Zhang acknowledges the support of the Hector Foundation and the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). Frederik Kunstner is supported by a Marie Skłodowska-Curie Fellowship from the European Union's Horizon Europe Research and Innovation program under Grant Agreement No. 101210427. This work has received support from the French government, managed by the National Research Agency, under the France 2030 program with the reference "PR[AI]RIE-PSAI" (ANR-23-IACL-0008), and is supported in part by the EC under the grant No. 101195233: OpenEuroLLM.

References

Yifan Bai et al. (2026). *Kimi K2: Open Agentic Intelligence*. Technical Report.

Raghu Bollapragada, Richard H. Byrd, and Jorge Nocedal (2019). "Exact and inexact subsampled Newton methods for optimization". In: *IMA Journal of Numerical Analysis* 39.2, pp. 545–578.

Léon Bottou, Frank E. Curtis, and Jorge Nocedal (2018). "Optimization methods for large-scale machine learning". In: *SIAM Review* 60.2, pp. 223–311.

David E Carlson, Edo Collins, Ya-Ping Hsieh, Lawrence Carin, and Volkan Cevher (2015). "Preconditioned spectral descent for deep learning". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Damek Davis and Dmitriy Drusvyatskiy (2025). "When do spectral gradient updates help in deep learning?" arXiv preprint.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu (1996). "A density-based algorithm for discovering clusters in large spatial databases with noise". In: *International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 226–231.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein (2024). *Muon: An Optimizer for Hidden Layers in Neural Networks*. https://kellerjordan.github.io/posts/muon/. Accessed: 2026-04-15.

Diederik P. Kingma and Jimmy Ba (2015). "Adam: A method for stochastic optimization". In: *International Conference on Learning Representations (ICLR)*.

Frederik Kunstner and Francis R. Bach (2025). "Scaling laws for gradient descent and sign descent for linear bigram models under Zipf's law". In: *Neural Information Processing Systems (NeurIPS)*.

Frederik Kunstner, Alan Milligan, Robin Yadav, Mark Schmidt, and Alberto Bietti (2024). "Heavy-tailed class imbalance and why Adam outperforms gradient descent on language models". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Shuaipeng Li, Penghao Zhao, Hailin Zhang, Xingwu Sun, Hao Wu, Dian Jiao, Weiyan Wang, Chengjun Liu, Zheng Fang, Jinbao Xue, Yangyu Tao, Bin Cui, and Di Wang (2024). "Surge phenomenon in optimal learning rate and batch size sscaling". In: *Neural Information Processing Systems (NeurIPS)*.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang (2025). *Muon is scalable for LLM training*. Technical Report.

Sadhika Malladi, Kaifeng Lyu, Abhishek Panigrahi, and Sanjeev Arora (2022). "On the SDEs and scaling rules for adaptive gradient algorithms". In: *Neural Information Processing Systems (NeurIPS)*.

Martin Marek, Sanae Lotfi, Aditya Somasundaram, Andrew Gordon Wilson, and Micah Goldblum (2025). *Small batch size training for language models: When vanilla SGD works, and why gradient accumulation is wasteful*. Neural Information Processing Systems (NeurIPS).

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher (2017). "Pointer sentinel mixture models". In: *International Conference on Learning Representations (ICLR)*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay (2011). "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research (JMLR)* 12, pp. 2825–2830.

Rico Sennrich, Barry Haddow, and Alexandra Birch (2016). "Neural machine translation of rare words with subword units". In: *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ishaan Shah, Anthony M. Polloreno, Karl Stratos, Philip Monk, Adarsh Chaluvaraju, Andrew Hojel, Andrew Ma, Anil Thomas, Ashish Tanwer, Darsh J Shah, Khoi Nguyen, Kurt Smith, Michael Callahan, Michael Pust, Mohit Parmar, Peter Rushton, Platon Mazarakis, Ritvik Kapila, Saurabh Srivastava, Somanshu Singla, Tim Romanski, Yash Vanjani, and Ashish Vaswani (2025). *Practical efficiency of Muon for pretraining*. Technical Report.

Teodora Srećković, Jonas Geiping, and Antonio Orvieto (2025). *Is your batch size the problem? Revisiting the Adam-SGD gap in language modeling*. arXiv preprint.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5998–6008.

Shuche Wang, Fengzhuo Zhang, Jiaxiang Li, Cunxiao Du, Chao Du, Tianyu Pang, Zhuoran Yang, Mingyi Hong, and Vincent Y. F. Tan (2025). *Muon outperforms adam in tail-end associative memory learning*. Preprint. arXiv/2509.26030.

Guodong Zhang, Lala Li, Zachary Nado, James Martens, Sushant Sachdeva, George E. Dahl, Christopher J. Shallue, and Roger B. Grosse (2019). "Which algorithmic choices matter at which batch sizes? Insights from a noisy quadratic model". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Tianyue H. Zhang, Lucas Maes, Alan Milligan, Alexia Jolicoeur-Martineau, Ioannis Mitliagkas, Damien Scieur, Simon Lacoste-Julien, and Charles Guille-Escuret (2025a). "Understanding Adam requires better rotation dependent assumptions". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Yushun Zhang, Congliang Chen, Ziniu Li, Tian Ding, Chenwei Wu, Diederik P. Kingma, Yinyu Ye, Zhi-Quan Luo, and Ruoyu Sun (2025b). "Adam-mini: Use fewer learning rates to gain more". In: *International Conference on Learning Representations (ICLR)*.

Rosie Zhao, Depen Morwani, David Brandfonbrener, Nikhil Vyas, and Sham M. Kakade (2025). "Deconstructing What makes a good optimizer for autoregressive language models". In: *International Conference on Learning Representations (ICLR)*.

Appendix

- A: Experimental details on small models 10
- B: Derivations 13
- C: Experimental details for language model and clustering 16
- D: Additional observations on the cluster structure in language models 18

Appendix A. Experimental details on small models

In this section, we provide additional details on the toy problems used throughout the paper to study the interaction between optimization methods and signal frequency imbalance. These controlled settings isolate specific forms of imbalance and help explain the empirical behaviors observed in larger neural networks, including the dependence on axis alignment and batch size.

We study a collection of toy problems designed to isolate different forms of imbalance and their interaction with optimization methods. In particular, we vary whether the imbalance is axis-aligned or non-axis-aligned, as well as row- versus column-wise. We begin with a balanced setting as a sanity check, then progressively introduce these structural effects. These settings loosely mimic different parts of neural networks: input imbalance corresponds to axis-aligned feature imbalance in early layers, the rotated setting captures non-axis-aligned intermediate representations after embedding layers and feature mixing, while output imbalance resembles class-frequency imbalance in the final classifier layer.

A.1. Experimental Details

We evaluate three optimizers: SGD, Adam with $(\beta_1, \beta_2) = (0, 0.999)$, and Muon. Unless otherwise specified, we use constant learning rates selected from log-spaced grids, with no weight decay. For each configuration, we report results at the best-performing learning rate, selected by final training loss within stable runs. We use $d = 10000$ and $T = 500$ iterations. For output imbalance, we fix $d = 1000$, input dimension $k = 20000$, dataset size $n = 10000$, and use batch sizes $B \in \{10, 500, 10000\}$. For the other problems, we sweep a small, medium and large (full) batch sizes $B \in \{10, 1000, \text{full}\}$. All experiments use a fixed random seed. We log the full training loss at each iteration, as well as per-class losses grouped into five bins of equal probability mass under the sampling distribution. Within each group, we report the weighted average loss.

A.2. Problem Setup

No imbalance Given data $(\mathbf{x}, \mathbf{y}) \in \{0, 1\}^{c \times c}$ which are one-hot encodings of words $\mathbf{x}, \mathbf{y}$ in vocabulary of size c. drawn from $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y} \mid \mathbf{x})$ where $p(\mathbf{x} = i) = \frac{1}{c}$, fit a linear model with the squared loss,

$$\mathcal{L}(\mathbf{W}) = \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}} \left[\tfrac{1}{2} \|\mathbf{W}\mathbf{x} - \mathbf{y}\|_2^2 \right].$$

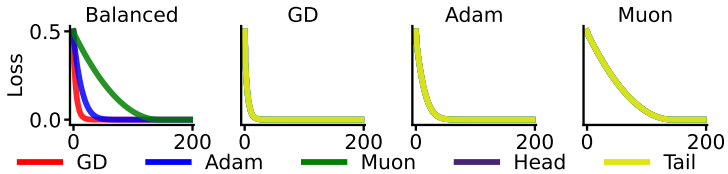

Figure 4: Under the balanced matrix bigram problem and full batch training, all methods converge uniformly across frequency groups, with nearly identical head and tail dynamics.

Input (row-wise) imbalance Given data $(\mathbf{x}, \mathbf{y}) \in \{0,1\}^{c \times c}$ which are one-hot encodings of words $\mathbf{x}, \mathbf{y}$ of vocabulary size c drawn from $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y} \mid \mathbf{x})$ where $p(\mathbf{x} = i) \propto \pi_i$, fit a linear model with the squared loss,

$$\mathcal{L}(\mathbf{W}) = \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}} \left[\tfrac{1}{2} \|\mathbf{W}\mathbf{x} - \mathbf{y}\|_2^2 \right].$$

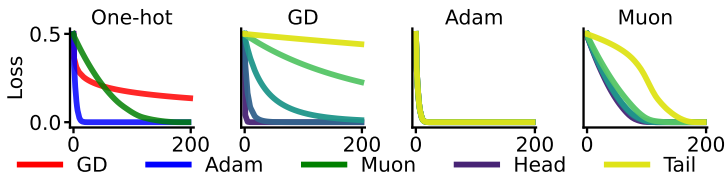

Figure 5: Under axis-aligned input frequency imbalance and full batch training, GD learns frequent directions quickly but struggles on rare directions, whereas Adam nearly equalizes convergence across groups through adaptive rescaling. Muon also accelerates tail convergence, though its advantage is less pronounced in this axis-aligned setting.

Non-axis aligned imbalance Given data $(\mathbf{x}, \mathbf{y}) \in \{0,1\}^{c \times c}$ which are one-hot encodings of words $\mathbf{x}, \mathbf{y}$ of vocabulary size c drawn from $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x})p(\mathbf{y} \mid \mathbf{x})$ where $p(\mathbf{x} = i) \propto \pi_i$. Let $\mathbf{E} \in \mathbb{R}^{c \times c}$ be an orthogonal matrix, and applied to the inputs, and fit a linear model with the squared loss,

$$\mathcal{L}(\mathbf{W}) = \mathop{\mathbb{E}}_{\mathbf{x},\mathbf{y}} \left[\tfrac{1}{2} \|\mathbf{W}\mathbf{E}\mathbf{x} - \mathbf{y}\|^2 \right].$$

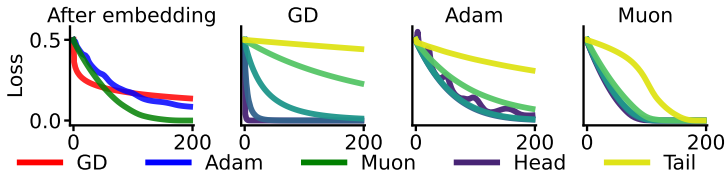

Figure 6: After applying a random rotation to the input imbalance problem, in full batch training, although Adam still partially accelerates convergence on low-frequency directions, its advantage deteriorates once the imbalance is no longer axis-aligned. In contrast, Muon continues to improve convergence across all groups, highlighting its robustness to rotated and non-axis-aligned structures.

Output (column-wise) imbalance Given a dataset $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ where $\mathbf{x}_i \sim \mathcal{N}(1, I_d)$ and $\mathbf{y}_i \in \{0,1\}^c$ is a one-hot label vector with $p(\mathbf{y}_{i,j} = 1) \propto \frac{1}{j}$, we fit a linear model with cross-entropy loss,

$$\mathcal{L}(\mathbf{W}) = \frac{1}{n} \sum_{i=1}^{n} \left(-\mathbf{y}_i^\top \mathbf{W}\mathbf{x}_i + \mathrm{LSE}(\mathbf{W}\mathbf{x}_i) \right), \qquad \text{where} \qquad \mathrm{LSE}(\mathbf{z}) = \log \left(\sum_{i=1}^{c} \exp(\mathbf{z}_i) \right).$$

Note that, unlike the previous input-imbalance setting, where imbalance appears through the sampling frequency of input directions, here the imbalance appears in the output labels while the input features

are dense and shared across classes. Here, we consider a fixed finite dataset rather than sampling fresh examples at every iteration, making the problem closer to practical classification settings. As a result, the imbalance is no longer axis-aligned in parameter space, and the gradients from different classes interact through shared input representations and the softmax normalization.

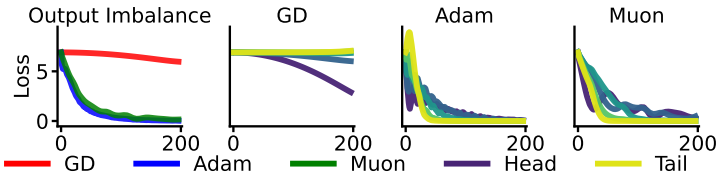

Figure 7: Under output-side (column-wise) frequency imbalance, GD fails to make substantial progress, while Adam and Muon significantly improve convergence across all groups. ($b = 10000$)

Effect of batch size To study the role of minibatch averaging under frequency imbalance, we compare optimization dynamics across small- and large-batch regimes. At small batch sizes, stochasticity dominates the dynamics, and all methods behave similarly. In contrast, large batches expose the underlying frequency imbalance, amplifying the separation between frequent and rare directions. As shown in Figure 8, GD becomes increasingly bottlenecked by rare directions at large batch sizes, while Adam substantially improves convergence under axis-aligned imbalance. After rotations, this advantage weakens, whereas Muon remains effective.

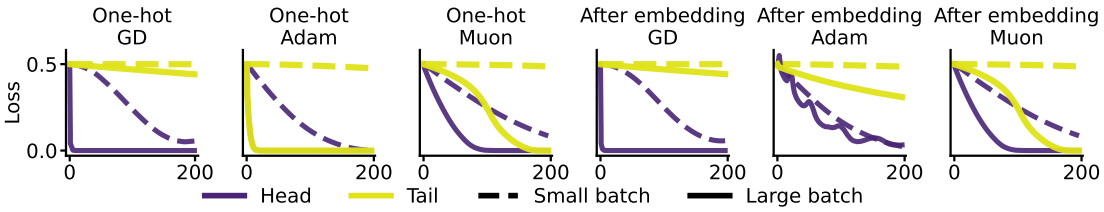

Figure 8: Effect of batch size under frequency imbalance. Dashed lines denote small-batch training ($b = 10$), and solid lines denote large (full) batch training. Large batches suppress stochasticity and expose the underlying frequency imbalance, amplifying the separation between frequent and rare directions. Adam improves over GD under axis-aligned imbalance (left), while Muon remains effective after rotations (right).

Appendix B. Derivations

B.1. Gram matrix on quadratics

Proposition 4 (Gradient Gram structure) *Consider the linear bigram model of Example 1 with a finite number of samples, where $n_{i,j}$ is the number of occurrences of $(x, y) = (i, j)$. Let $\mathbf{G}$ denote the gradient matrix obtained by stacking $\mathrm{vec}(g_{i,j})^\top$ over all samples. For both $\mathbf{W} = \mathbf{0}$, the gradient Gram matrix $\mathbf{G}\mathbf{G}^\top$ is block diagonal up to a permutation $\mathbf{P}$, with blocks*

$$\mathbf{P}\mathbf{G}\mathbf{G}^\top\mathbf{P}^\top = \mathrm{Diag}\left(\mathbf{J}_{n_{11}}, \mathbf{J}_{n_{12}}, \ldots, \mathbf{J}_{n_{dd}}\right), \quad \textit{where} \quad \mathbf{J}_m \coloneqq \mathbf{1}_m\mathbf{1}_m^\top.$$

As $\mathbf{G}\mathbf{G}^\top$ is exactly block-diagonal, the eigenvectors of the weight Gram matrix $\mathbf{G}^\top\mathbf{G}$ can also be partitioned into blocks corresponding to the clusters of samples.

Proof For two examples $(x_1, y_1) = (e_i, e_j)$ and $(x_2, y_2) = (e_k, e_\ell)$, the gradient inner product is

$$\begin{aligned}\langle g_{i,j}, g_{k,\ell}\rangle_F &= \left\langle (\mathbf{W}e_i - e_j)e_i^\top, (\mathbf{W}e_k - e_\ell)e_k^\top \right\rangle_F \\ &= \langle \mathbf{W}e_i - e_j, \mathbf{W}e_k - e_\ell\rangle \cdot \langle e_i, e_k\rangle = \mathbb{1}_{i=k}\langle \mathbf{W}e_i - e_j, \mathbf{W}e_i - e_\ell\rangle.\end{aligned}$$

Thus gradients corresponding to different inputs are orthogonal, so $\mathbf{G}\mathbf{G}^\top$ decomposes into blocks indexed by the input coordinate. At $\mathbf{W} = \mathbf{0}$, $g_{i,j} = (-e_j)e_i^\top$, and therefore

$$\langle g_{i,j}, g_{k,\ell}\rangle_F = (e_j^\top e_\ell)(e_i^\top e_k) = \mathbb{1}_{j=\ell}\mathbb{1}_{i=k}.$$

Hence, examples corresponding to the same bigram have identical gradients, while different bigram types are orthogonal. After permuting examples by bigram type,

$$\mathbf{P}\mathbf{G}\mathbf{G}^\top\mathbf{P}^\top = \mathrm{Diag}(\mathbf{J}_{n_{11}}, \mathbf{J}_{n_{12}}, \ldots, \mathbf{J}_{n_{dd}}).$$

The same block structure transfers to the nonzero eigenspaces of the weight Gram matrix $\mathbf{G}^\top\mathbf{G}$. Indeed, if $\mathbf{u}$ is an eigenvector of $\mathbf{G}\mathbf{G}^\top$ with eigenvalue $\lambda > 0$, then

$$\mathbf{G}^\top\mathbf{G}(\mathbf{G}^\top\mathbf{u}) = \mathbf{G}^\top(\mathbf{G}\mathbf{G}^\top\mathbf{u}) = \lambda\mathbf{G}^\top\mathbf{u}.$$

Thus $\mathbf{G}^\top\mathbf{u}$ is an eigenvector of $\mathbf{G}^\top\mathbf{G}$ with the same nonzero eigenvalue. Since $\mathbf{G}\mathbf{G}^\top$ is block diagonal after the permutation $\mathbf{P}$, its eigenvectors can be chosen to have support on a single block, and the eigenvectors of $\mathbf{G}^\top\mathbf{G}$ are the eigenvectors of each block. ■

B.2. Optimal preconditioner scaling with batch size

Proposition 5 *The optimal fixed preconditioner for the problem Example 3, as a function of batch size b, $\mathbf{P}^\star_{\mathrm{Freq}}$ and $\mathbf{P}^\star_{\mathrm{Magn}}$ for the frequency and magnitude imbalance problems, respectively, are*

$$\mathbf{P}^\star_{\mathrm{Freq}}(b) = \mathrm{Diag}\left(\frac{1}{b} + \left(1 - \frac{1}{b}\right)\boldsymbol{\pi}\right)^{-1}, \qquad \mathbf{P}^\star_{\mathrm{Magn}}(b) = \mathrm{Diag}\left(\boldsymbol{\pi}\right)^{-1}.$$

Proof

Frequency imbalance The sample $\mathbf{x} = \mathbf{e}_i$ appears with probability π_i, so the coordinate i is only updated on iterations where $\mathbf{x} = \mathbf{e}_i$ appears in the minibatch. The optimum is $\mathbf{w}^\star = \mathbf{1}$, so define the coordinate residual

$$r_i(t) \coloneqq w_i(t) - 1.$$

For a sample $\mathbf{x} = e_i$, the stochastic gradient is

$$\nabla \tfrac{1}{2}\|\langle \mathbf{x}, \mathbf{w}\rangle - 1\|^2 = (w_i(t) - 1)\mathbf{e}_i = r_i(t)\mathbf{e}_i.$$

Thus, every occurrence of $\mathbf{x} = e_i$ shrinks the error $r_i(t)$ multiplicatively.

Now consider a minibatch of size b. Let $N_i(t) \sim \text{Binomial}(b, \pi_i)$ be the number of occurrences of $\mathbf{x} = \mathbf{e}_i$ in minibatch t. The gradient in coordinate i is then $\frac{N_i(t)}{b} r_i(t)$ when averaged over the current minibatch. Using a diagonal preconditioner $\mathbf{P} = \mathrm{Diag}(p_1, \ldots, p_d)$, the update is

$$r_i(t+1) = \left(1 - p_i \frac{N_i(t)}{b}\right) r_i(t).$$

Taking the conditional expectation gives

$$\mathbb{E}\big[r_i(t+1)^2 \mid r_i(t)\big] = \mathbb{E}\left[\left(1 - p_i \frac{N_i(t)}{b}\right)^2\right] r_i(t)^2 = \left((1 - \pi_i p_i)^2 + \frac{\pi_i(1-\pi_i)}{b} p_i^2\right) r_i(t)^2,$$

where we used

$$\mathbb{E}\left[\frac{N_i(t)}{b}\right] = \pi_i, \qquad \text{Var}\left[\frac{N_i(t)}{b}\right] = \frac{\pi_i(1-\pi_i)}{b}.$$

Unrolling over t gives

$$\mathbb{E}\big[r_i(t)^2\big] = \left((1 - \pi_i p_i)^2 + \frac{\pi_i(1-\pi_i)}{b} p_i^2\right)^t r_i(0)^2.$$

Therefore, the expected error decomposes as

$$\mathbb{E}[\mathcal{L}(\mathbf{w}_t) - \mathcal{L}^\star] = \sum_{i=1}^{d} \pi_i \left((1 - \pi_i p_i)^2 + \frac{\pi_i(1-\pi_i)}{b} p_i^2\right)^t r_i(0)^2.$$

Since the loss is separable across coordinates, for fixed t it suffices to minimize

$$f_i(p_i) \coloneqq (1 - \pi_i p_i)^2 + \frac{\pi_i(1-\pi_i)}{b} p_i^2$$

independently for each coordinate i. Differentiating gives

$$f_i'(p_i) = -2\pi_i(1 - \pi_i p_i) + 2\frac{\pi_i(1-\pi_i)}{b} p_i.$$

Setting $f_i'(p_i) = 0$ yields

$$p_i^\star(b) = \frac{1}{\pi_i + \frac{1-\pi_i}{b}} = \left(\frac{1}{b} + \left(1 - \frac{1}{b}\right)\pi_i\right)^{-1}.$$

Magnitude imbalance Under magnitude imbalance, every coordinate is observed equally often and the imbalance in π_i appears through the curvature. Using the same reasoning,

$$\mathbb{E}\big[r_i(t)^2\big] = \mathbb{E}\left[\left(1 - \frac{p_i \pi_i}{d}\right)^2\right] r_i(t)^2.$$

The minimizer is $p_i^\star \propto 1/\pi_i$. and $\mathbf{P}^\star_{\text{Magn}}(b) \propto \text{Diag}(\boldsymbol{\pi})^{-1}$, independent of the batch size b. ■

A note on SGD critical batch size in frequency imbalance In frequency imbalance, the optimal preconditioner transitions between two regimes depending on which term dominates the denominator

$$p_i^\star(b) = \frac{1}{\frac{1}{b} + \left(1 - \frac{1}{b}\right)\pi_i}.$$

The denominator contains two competing effects: the stochastic minibatch noise scale $1/b$ and the curvature scale π_i. The behavior of the optimal preconditioner depends on which term dominates. When $\frac{1}{b} \gg \pi_i$, the minibatch noise term dominates and $p_i^\star(b) \approx b$, so the preconditioner is approximately isotropic. In contrast, when $\frac{1}{b} \ll \pi_i$, the curvature term dominates and $p_i^\star(b) \approx \frac{1}{\pi_i}$, recovering inverse-frequency scaling. The transition occurs when the two terms are comparable $\frac{1}{b} \approx \pi_i$.

Under the Zipf distribution $\pi_i = \frac{1}{iH_d}$ the largest frequency is $\pi_1 = \frac{1}{H_d}$. Using the asymptotic approximation $H_d \sim \log d$, the transition for the dominant coordinates occurs at $b_{\text{crit}} \sim H_d \sim \log d$.

This transition is also visible empirically in Figure 9: the optimal learning rate initially scales approximately linearly with batch size, corresponding to the noise-dominated regime, before saturating once the dynamics become curvature-dominated.

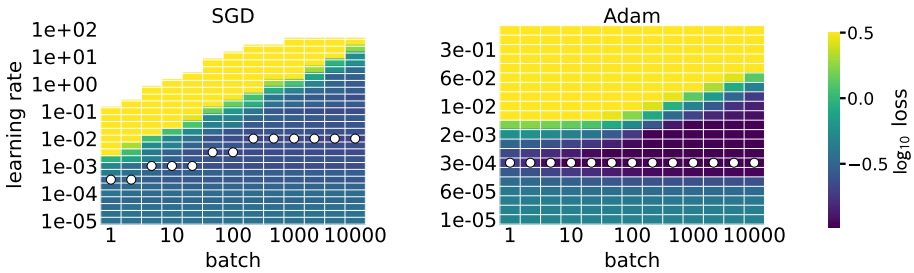

Figure 9: **Learning rate scaling differs between SGD and Adam.** We sweep the learning rate and batch size for the input imbalance problem and report the final training loss after a fixed number of iterations. White markers denote the best-performing learning rate at each batch size. For SGD, the optimal learning rate increases with batch size until a critical batch size is reached. In contrast, Adam remains effective over a much narrower range of learning rates, with the optimal learning rate remaining nearly constant across batch sizes.

Appendix C. Experimental details for language model and clustering

This section gives the experimental details for the visualization in Figure 3.

Model and dataset. We use a small transformer model (Vaswani et al., 2017) with depth 2 (two transformer blocks), sequence length of 32, embedding width of 256 (and 1024 for the hidden dimension of the MLP), 2 heads and dropout of 0.1. We use the implementation from the word language model example from PyTorch.[1] We use the WikiText-103 dataset (Merity et al., 2017), tokenized using Byte-Pair Encoding (Sennrich et al., 2016) with a vocabulary size of $2^{14} = 16\,384$.

Training. The visualization in Figure 3 is taken at initialization, but we show the structure over the course of training in the appendix. Training uses Adam with default hyperparameters expect for the step-size, which is set to $10^{-2.5}$, selected by grid-search over half-powers of 10 and a batch size (number of sequences) of 512. We do not use a learning rate schedule.

Computing the per-sample gradients. By a sample, we mean an input sequence and its corresponding output, such that the overall loss is the average over samples. The batch size in language tasks typically refers to the number of sequences, but each sequence contains multiple prediction tasks, one for each element of the sequence. We call each of these a sample; that is, the sequence $abcd$ leads to the samples (a, b), (ab, c), (abc, d). Due to the large dimensionality of the model d and the size of the number of samples n, we cannot store all gradient matrices as an $[n \times d]$ matrix in memory. For the figures in the paper, we subsample a fixed set of 128 sequences of length 32 and select every second prediction, leading to $n = 128 \times 16 = 2\,048$ samples. For most layers, we reduce the dimensionality by vectorizing the gradients and projecting them down to $d = 1\,000$ dimensions using a random Gaussian matrix, which preserves distances and inner products. For the first and last layer, we use the fact that the gradients are sparse and low-rank respectively to store them without projection.

Clustering. To find the permutation matrix to visualize the block-diagonal structure, we use the Scikit-learn (Pedregosa et al., 2011) implementation of DBScan (Ester et al., 1996) using (absolute) cosine similarities as the metric. We form a permutation matrix $\mathbf{P}$ by sorting the clusters in decreasing order of a combination of size and intensity of within-cluster similarity. For a cluster $\mathcal{S} \subseteq [n]$ and a similarity matrix $\mathbf{S} \in [0, 1]^{n \times n}$, the score of a cluster is given by

$$\text{Score}(\mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{i,j \in \mathcal{S}} |\mathbf{S}[i, j]|.$$

We tune the ϵ parameter controlling the neighborhood size automatically by a heuristic, minimizing a score over the fixed grid of $\epsilon = [1/1000, 1/5, 2/5, 3/5, 4/5, 1]$. Given a permutation matrix $\mathbf{P}$ induced by the clusters $\mathcal{S}_1, \ldots, \mathcal{S}_k$ obtained by DBScan with a given ϵ, we score the value of ϵ using

$$\text{Score}(\epsilon) = \frac{1}{n-1} \sum_{i=1}^{n-1} 1 - (\mathbf{P}\mathbf{S}\mathbf{P}^\top)[i, i+1] + \frac{|\mathcal{S}_1|^2}{n^2}.$$

The first term is the normalized path length as a measure of the quality of the seriation and the second is the percentage of the area occupied by the largest cluster to avoid solutions where the first cluster contains too many samples. Beyond those intuitions, we do not have a strong motivation for the heuristic other than that we found the results to be consistent across layers.

1. https://github.com/pytorch/examples/tree/acc295dc7b90714f1bf47f06004fc19a7fe235c4/word_language_model

Cluster-eigenvector alignment. To verify that the largest clusters in the data Gram matrix $\mathbf{G}\mathbf{G}^\top$ correspond to the top eigenvectors of the weight Gram matrix $\mathbf{G}^\top\mathbf{G}$, we use the responsibilities of eigenvectors for each sample, which is shown in Figures 2 and 3 and further figures in the appendix. To define responsibilities, let $\mathbf{G}^\top\mathbf{G} = \mathbf{V}\Lambda\mathbf{V}^\top$ be the eigendecomposition of the weight Gram matrix. and let $\mathbf{Z} = \mathbf{G}\mathbf{V}$. be the projection of the gradients onto the eigenvectors. The responsibilities $\mathbf{R}[i,j] = \mathbf{Z}[i,j]^2/\|\mathbf{Z}[i,:]\|^2$, which sums to 1, are the fraction of the Euclidean norm of each gradient captured by the projection onto eigenvector j, as a measure of the alignment between the gradient of sample i and eigenvector j. The resulting matrix $\mathbf{P}\mathbf{R}$, indicate that the largest clusters in the data Gram matrix $\mathbf{G}\mathbf{G}^\top$ correspond to the top eigenvectors of the weight Gram matrix $\mathbf{G}^\top\mathbf{G}$, suggesting that the leading directions in weight space are still primarily associated with specific clusters of samples.

Appendix D. Additional observations on the cluster structure in language models

This section gives more visualizations of the structure of the sample Gram matrix and their projection on the leading eigenvectors of the feature Gram matrix.

The main observations are that

- Not all weight matrices follow the same structure. The block-diagonal structure appears to be a good model for some weights at initialization, specifically for the V, out projection, and the linear layers of the MLP part of the attention module.
- The clustering structure evolves over time. The size of the largest cluster decreases over training for all layers, capturing more refined correlations. The samples start clustered by input words, short contexts or output word, but this structure gets refined as training progress to depend on more fine-grained patterns such as bigrams or different subset of the same sequence.

D.1. Difference in patterns across parameters

The block-diagonal structure is less present in some layers. We recall the architecture used, which uses 2 modules transformer model.

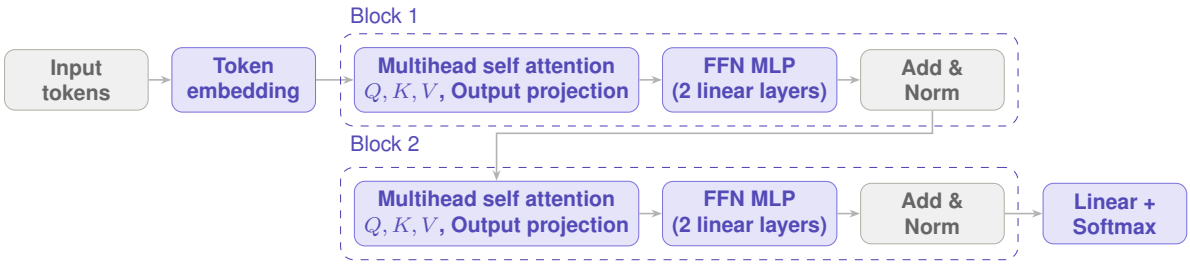

Figure 10: Two modules transformer model architecture.

We show the same matrices as Figure 3 for all layers at initialization in Figure 11 and Figure 12 for the first and second module, respectively. The main observation is that our clustering methodology for the gradients with respect to the Q and K appears somewhat less reliable. The points in the identified clusters are less uniformly uniform, and the projection. Whether this is due to a real difference is unclear. It is possible that a strong cluster structure exists for those matrices exists as well, but our heuristic is less suited to those parameters and did not find it. Although our results are not conclusive on the Q, K matrices, it is interesting to note that those coincides with the observations of Wang et al. (2025) that using Muon for those layers is less critical for performance.

D.2. Difference in patterns over time and semantic meaning

The cluster structure and the semantic meaning of the clusters—what feature samples in a cluster have in common—evolves over time. We focus on two examples to showcase this behavior, the gradients w.r.t. the Q, K matrices of the first module and the w.r.t. the linear layers of the second MLP module. We show the cluster structre in Figure 13 and Figure 14 respectively, and show some samples from each of the top 10 clusters in Table 2 and Table 4.

Figure 11: **Signal imbalance at initialization: First module**

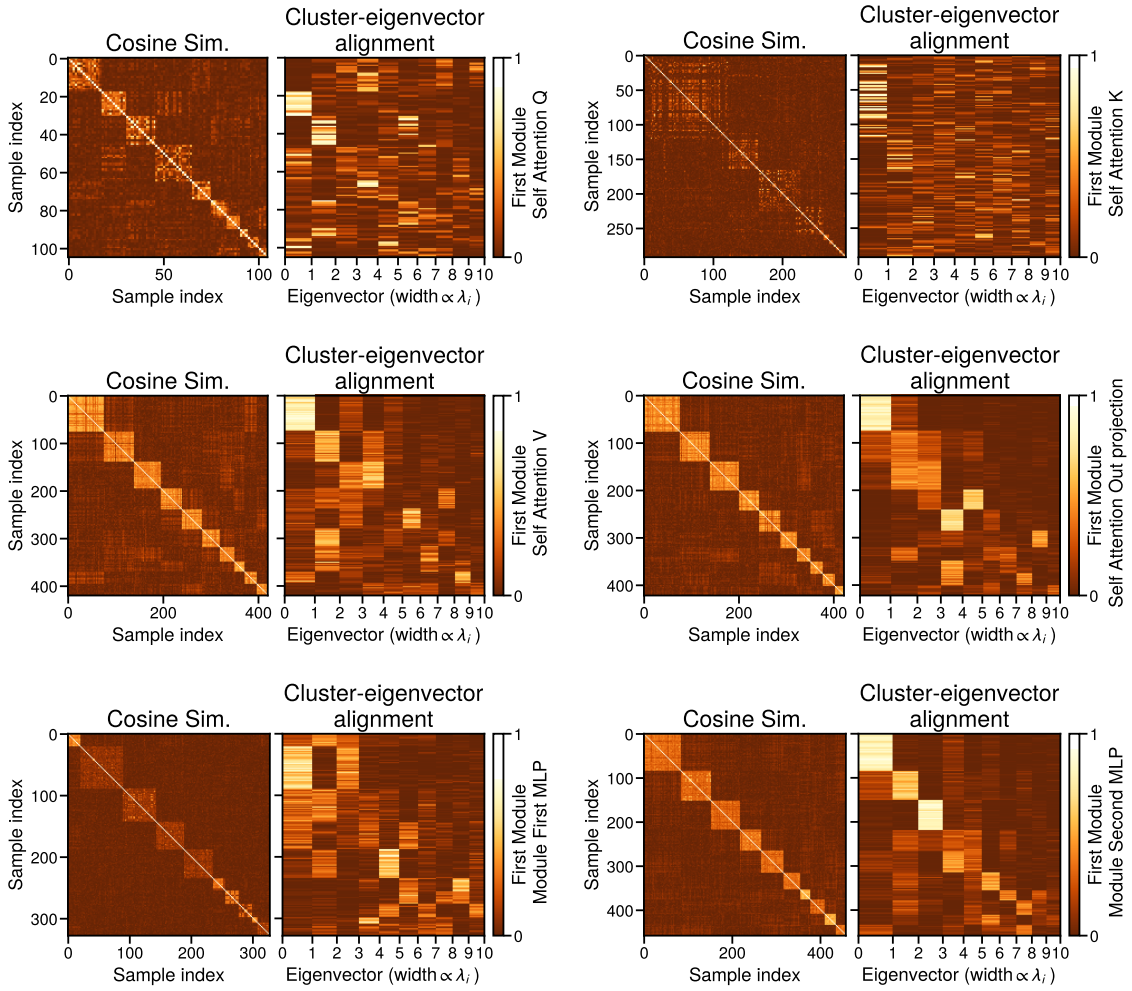

Figure 12: **Signal imbalance at initialization: Second module**

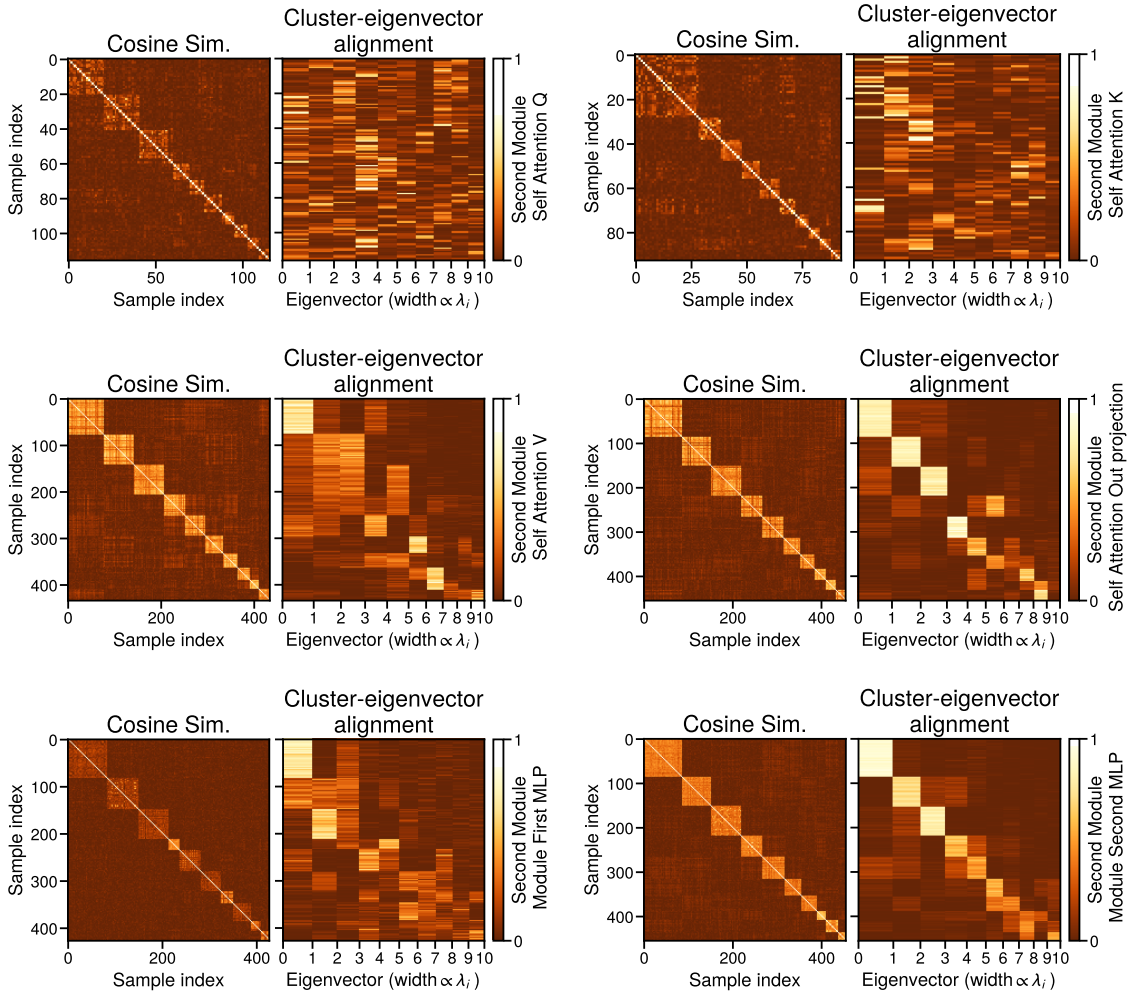

Figure 13: **Evolution of the cluster structure over training for the K, Q matrices.** Although the cluster structure is weak at initialization, it appears stronger after a few optimization steps before becoming weaker again. At the end of training, we observe strong small clusters which correspond to similar sentences (see Table 2).

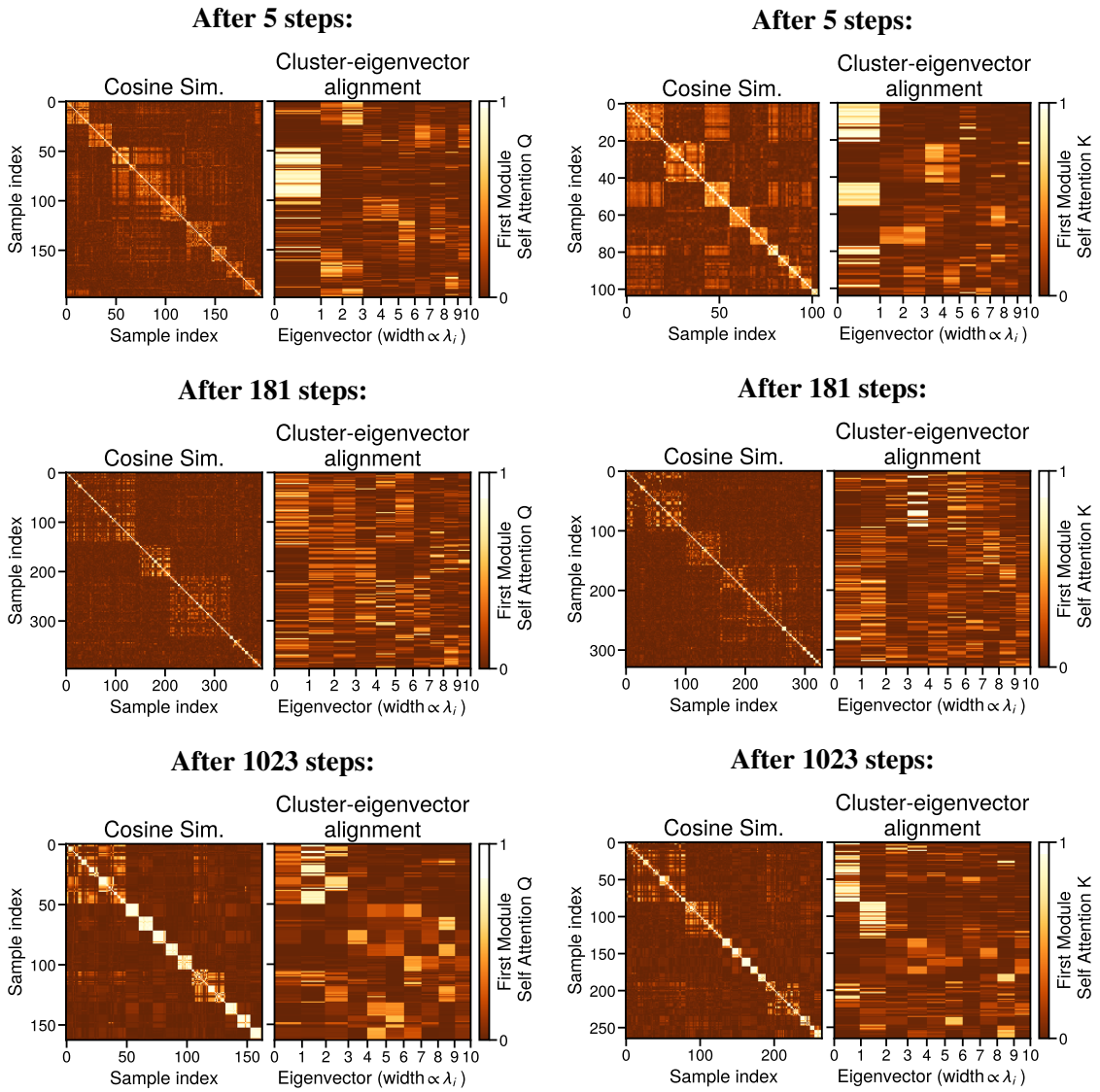

Figure 14: **Evolution of the cluster structure over training for the two linear layers of the last MLP block.** The size of the clusters decreases over training. Early clusters primarily reflect the output word, while later clusters appear to capture bigram co-occurrences (see Table 4).

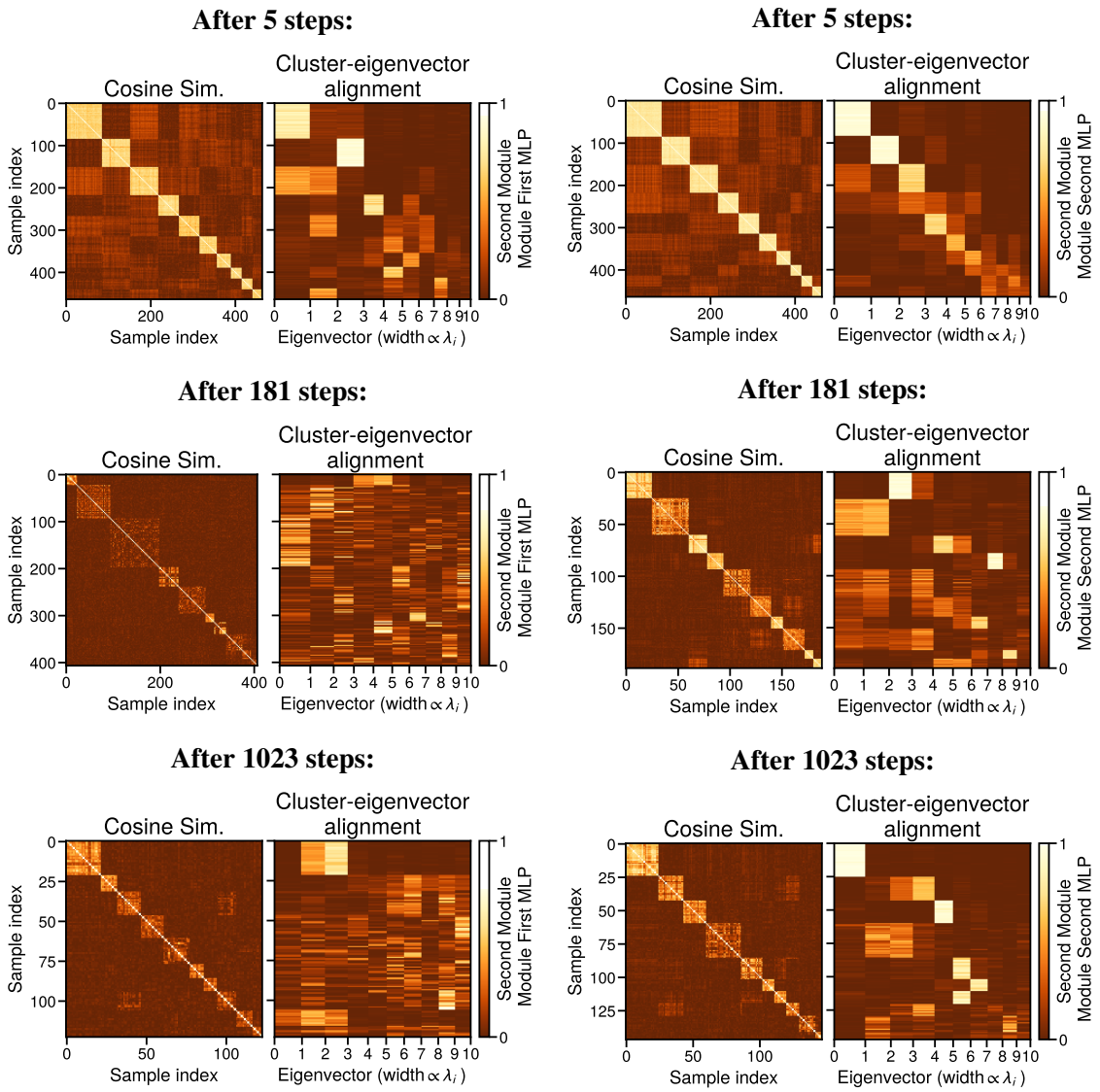

Table 1: **Large clusters in Q matrix at the start of training are short contexts with similar tokens.** Text samples from each of the top 5 clusters of the Q matrix of the first module at the start of training.

Cluster	Input	Output
1	| ,|	| and|
	| first| ,|	| and|
	| lot| "| ,|	| but|
	| ,|	| K|
	| ,|	| notably|
	| ,|	| killing|
	| ,|	| Wals|
2	| the|	| 2005|
	| album| .| According| to| the|	| producer|
	| album| .| According| to| the| producer| ,| few| days| after| he| had| finished| the|	| composition|
	|iting| occurs| in| the|	| spring|
	| the|	| population|
	| while| revenue| from| the|	| Federal|
	| R|otten| Tom|atoes| gives| the|	| film|
3	|ite| bed|rock| .|	| Whit|
	| .|	| |
	| .|	| Lind|
	| .|	| |
	|s| .|	| The|
	| .|	| Its|
	| album| .|	| According|
4	| of|	| the|
	| of| the|	| United|
	| of| the| United| States| '| declaration| of| war| on| 6| April| 1917| .| Tucker| joined| the|	| fleet|
	| a| writer| for| anime| who| had| previously| collaborated| with| Tak|ah|ashi| on| adaptations| of| the|	| first|
	| the| population| of|	| the|
	| the| population| of| the|	| metropolitan|
	| the| population| of| the| metropolitan| area| (| the| administrative| area| of| the|	| City|
5	| ,| and|	| headed|
	| ,| and| headed|	| back|
	| ,| and| headed| back|	| to|
	| first| ,| and|	| for|
	| game| developed| and|	| published|
	|house| and|	| weight|
	| and|	| retained|

Table 2: **Large clusters in Q matrix at the end of training are different predictions tasks on the same sequence, with different context lenghts.** Text samples from each of the top 5 clusters of the Q matrix of the first module at the end of training.

Cluster	Input	Output
1	| of| the|	| United|
	| of| the| United|	| States|
	| of| the| United| States|	| '|
	| one| million| copies| .|	| In|
	| one| million| copies| .| In|	| an|
	| one| million| copies| .| In| an|	| experience|
	| one| million| copies| .| In| an| experience|	| similar|
2	|is|abeth|	| '|
	|is|abeth| '|	|s|
	|is|abeth| '|s|	| funeral|
	|is|abeth| '|s| funeral|	| ,|
	|is|abeth| '|s| funeral| ,|	| Victoria|
	|is|abeth| '|s| funeral| ,| Victoria|	| removed|
	|is|abeth| '|s| funeral| ,| Victoria| removed|	| her|
3	| most| common| forms|	| include|
	| most| common| forms| include|	| the|
	| most| common| forms| include| the|	| inc|
	| most| common| forms| include| the| inc|	|ense|
	| most| common| forms| include| the| inc|ense|	| stick|
	| most| common| forms| include| the| inc|ense| stick|	| and|
	| most| common| forms| include| the| inc|ense| stick| and| inc|	|ense|
4	|s| offense| again|	| failed|
	|s| offense| again| failed|	| to|
	|s| offense| again| failed| to|	| advance|
	|s| offense| again| failed| to| advance|	| the|
	|s| offense| again| failed| to| advance| the|	| ball|
	|s| offense| again| failed| to| advance| the| ball|	| .|
	|s| offense| again| failed| to| advance| the| ball| .| After|	| a|
5	| .| Lind|er|f|elt| left|	| Mil|
	| .| Lind|er|f|elt| left| Mil|	|w|
	| .| Lind|er|f|elt| left| Mil|w|	|au|
	| .| Lind|er|f|elt| left| Mil|w|au|kee|	| that|
	| .| Lind|er|f|elt| left| Mil|w|au|kee| that| day|	| for|
	| .| Lind|er|f|elt| left| Mil|w|au|kee| that| day| for| a|	| promised|
	| .| Lind|er|f|elt| left| Mil|w|au|kee| that| day| for| a| promised| job|	| at|

Table 3: **Large clusters in last linear layer at the start of training depend on output token.** Text samples from each of the top 5 clusters.

Cluster	Input	Output
1	| they| never| discussed| the| other| fe|ather| @-@| bearing| specimens|	| ,|
	|enced|	| ,|
	|enced| ,| single| @-@| handed|	| ,|
	|ite| bed|rock| .| Whit|em|ore| creek|	| ,|
	|ite| bed|rock| .| Whit|em|ore| creek| ,| a| tribut|ary| of| the| Me|ander| River| (| Tasman|ia|)| ,| flows| past| Whit|em|ore| .| The| creek| is| seasonal|	| ,|
	| .| |\n| The| museum| '|s| hold|ings| number| over| 2| million| rocks|	| ,|
	| .| |\n| The| museum| '|s| hold|ings| number| over| 2| million| rocks| ,| min|erals| ,| fossils|	| ,|
2	|ite| bed|rock|	| .|
	|ite| bed|rock| .| Whit|em|ore| creek| ,| a| tribut|ary| of| the| Me|ander| River| (| Tasman|ia|)| ,| flows| past| Whit|em|ore|	| .|
	| .| |\n| Other| companies| followed| with| their| own| b|ubble| light| versions| and| other| lighting| nov|elt|ies|	| .|
	| .| |\n| Other| companies| followed| with| their| own| b|ubble| light| versions| and| other| lighting| nov|elt|ies| .| In| 1946| ,| S|ylvania| introduced| flu|ores|cent| past|el| lights|	| .|
	|idian| Bank| replaced| M|ell|on| Bank| as| the| lead| ten|ant|	| .|
	| '|m| in| trouble|	| .|
	| ,| and| headed| back| to| New| York| on| 30| September|	| .|
3	| they| never| discussed|	| the|
	|idian| Bank| replaced| M|ell|on| Bank| as|	| the|
	| likely| to| be| a| preliminary| sketch| for| a| now| @-@| unknown| work| .| |\n| =| =| Reception| =| =| |\n| Youth| on| the| Pro|w| ,| and| Ple|asure| at|	| the|
	| of|	| the|
	| brigade| and| ,| during|	| the|
	| in| assets| and| $| 65|4| million| in| l|i|abilities| .| During| bankrupt|cy| ,|	| the|
	|u| ranked| Super| Mario| 64| DS|	| the|
4	| Cl|in|ic| '|s| knowledge|	| and|
	| ,|	| and|
	| near| Gil|ford| .| It| had| been| travelling| from| Ban|bridge| to| |¡unk¿| with| nine| people| on| board| ;| all| were| Catholics|	| and|
	| brigade|	| and|
	| when| Alex|ios| III| married| his| two| daughters| I|rene|	| and|
	| when| Alex|ios| III| married| his| two| daughters| I|rene| and| Anna| to| Alex|ios| Pal|ai|olog|os|	| and|
	| in| assets|	| and|
5	|ite| bed|rock| .| Whit|em|ore| creek| ,| a| tribut|ary|	| of|
	| "| A| Canter|l|ot| W|edding| "| to| viewers| who| may| be| ske|pt|ical| about| the| popularity|	| of|
	| ,| and| headed| back| to| New| York| on| 30| September| .| She| took| on| a| load|	| of|
	| of| the| United| States| '| declaration|	| of|
	| brigade| and| ,| during| the| Valley| Campaign|s|	| of|
	| brigade| and| ,| during| the| Valley| Campaign|s| of| 1864| he| was| briefly| placed| in| command|	| of|
	| brigade| and| ,| during| the| Valley| Campaign|s| of| 1864| he| was| briefly| placed| in| command| of| a| division|	| of|

Table 4: **Large clusters in last linear layer at the end of training depend on bigrams between the last token of the input sequence and the output token.** Text samples from each of the top 5 clusters.

Cluster	Input	Output
1	| .| |	|\n|
	| .| |	|\n|
	| '|m| in| trouble| .| '| "| |	|\n|
	| likely| to| be| a| preliminary| sketch| for| a| now| @-@| unknown| work| .| |	|\n|
	|u| ranked| Super| Mario| 64| DS| the| 29|th| most| wanted| title| .| |	|\n|
	| 22| was| then| redesign|ated| as| NY| 164| .| |	|\n|
	| have| been| familiar| .| |	|\n|
2	| .|	| |
	| "| A| Canter|l|ot| W|edding| "| to| viewers| who| may| be| ske|pt|ical| about| the| popularity| of| the| series| .|	| |
	| .| Lind|er|f|elt| left| Mil|w|au|kee| that| day| for| a| promised| job| at| the| Library| Bureau| in| Boston| .|	| |
	| .|	| |
	| '|m| in| trouble| .| '| "|	| |
	| 22| was| then| redesign|ated| as| NY| 164| .| |\n| =| =| Major| intersections| =| =| |\n| The| entire| route| is| in| Pat|ters|on| ,| Put|nam| County| .|	| |
	| have| been| familiar| .|	| |
3	| Cl|in|ic| '|	|s|
	| .| |\n| The| museum| '|	|s|
	| of| the| United| States| '|	| declaration|
	| small| proportion| of| their| surface| ,| but| most| or| all| of| Hy|ak|ut|ake| '|	|s|
	|ounded| by| C|att|ell| Road| ,| Cov|entry| Road| ,| T|ilton| Road| ,| Gar|rison| Lane| and| the| railway| ,| and| near| St| Andrew| '|	|s|
	| lot| "| ,| but| disliked| Mon|a| '|	|s|
	| eastern| reaches| of| the| N|esc|op|eck| Creek| watershed| are| near| the| border| of| the| App|al|ach|ian| Plate|au| region| .| |\n| N|esc|op|eck| Creek| '|	|s|
4	| of|	| the|
	| brigade| and| ,| during|	| the|
	| a| massive| flooding| of|	| the|
	| first| ,| and| for| years| the| only| ,| city| in|	| the|
	| Roman| Catholic| Church| on| Malta| .| Within| a| week| ,| Bon|ap|arte| had| res|up|pl|ied| his| ships| ,| and| on| 19| June| ,| his| fleet| departed| for| Alexandria| in|	| the|
	| have| been| familiar| .| |\n| Don|ne| '|s| cart|ographic| references| in|	| the|
	|iting| occurs| in|	| the|
5	| .| |\n|	| The|
	| likely| to| be| a| preliminary| sketch| for| a| now| @-@| unknown| work| .| |\n| =|	| =|
	| 22| was| then| redesign|ated| as| NY| 164| .| |\n| =|	| =|
	|nder| uncertainty| .| |\n|	| =|
	|nder| uncertainty| .| |\n| =|	| =|
	| Person| of| the| Year| recognition| in| 2012| .| |\n|	| =|
	| Person| of| the| Year| recognition| in| 2012| .| |\n| =| =|	| Early|