CHARACTERIZING SPARSITY IN TRANSFORMERS

Reece Shuttleworth[*]
Massachusetts Institute of Technology
rshuttle@mit.edu
https://github.com/reeceshuttle/958

ABSTRACT

We look for sparse features in large language models (LLMs) in order to identify if sparsity emerges in high performing systems. We have two reasons, biological and theoretical, to believe that sparsity is important in intelligence: the human brain is sparse and sparse representations in the form of compositional functions help avoid the curse of dimensionality. We focus our study on the attention mechanism inside these models and look at both the raw weight matrices and the attention scores generated during inference. We find that weight matrices involved in attention, particularly the matrix product $W_Q W_K^T$, have very low stable rank. We find that the entropy of attention scores is low, implying that the attention scores are sparse.

1 INTRODUCTION

The Merriam-Webster dictionary defines sparse as *"of few and scattered elements."* (Merriam-Webster). We know that the brain is sparse in many ways. The brain is activation sparse because at any one moment, only a small percentage of neurons are firing (Barth & Poulet, 2012). The brain is connection sparse because neurons are sparsely interconnected between each other (Hunter et al., 2021). The brain is stimuli sparse because only a small portion of neurons are activated by any specific stimuli (Vonderschen & Chacron, 2011). From this, we can see that sparsity presents itself in many different ways in the brain.

The curse of dimensionality is a well-known problem in machine learning which, in its simplest form, states that the number of parameters required to model an arbitrary function well grows exponentially in the input dimension (Poggio et al., 2017). However, there are ways to escape this curse. Important work has shown that if sparse representations in the form of compositional functions are used instead of a dense representation, the parameters required grow linearly instead of exponentially (Poggio, 2023).

The curse of dimensionality is relevant in machine learning because many current models operate within high dimensional input spaces, which means that an extremely large number of parameters would be required if using a dense representation. One class of models that escape the curse of dimensionality are Convolutional Neural Networks (CNNs) (Krizhevsky et al., 2012), which use sparse compositional functions in the form of convolution filters and max pooling layers. Because they use sparse compositional functions, CNNs enjoy the guarantee that they can scale linearly with the input dimension.

One class of models that do not appear to escape the curse of dimensionality are transformers (Vaswani et al., 2017), which are the basis of Large Language Models (Brown et al., 2020) and Vision Transformers (Dosovitskiy et al., 2021). These models do not use sparse compositional functions and instead use dense matrix operations in the form of multi-layer perceptron or multi-head attention layers. Remarkably, these models optimize and perform impressively well, reaching or exceeding human performance on many tasks (OpenAI, 2023), even though they lack theoretical support to do so. Nothing reiterates this conflict more that ChatGPT(OpenAI, 2023), which has led to paradigm shifts in huge sectors of human employment and everyday life. It operates in a

[*] A special thanks to Akshay Rangamani for mentoring me throughout this project. Without his guidance, this project would not have been possible.

high dimensional input space but has not run into the problems of exponentially growing parameter requirements that the curse of dimensionality implies.

How are Large Language Models able to escape the curse of dimensionality without explicitly using sparse compositional functions? *What if, instead of being explicitly constrained to use these sparse compositional functions, these models implicitly learned sparse representations and behavior*? As discussed above, we have biological and theoretical reasons to expect to find sparsity in neural networks. If it is the case that optimization implicitly leads LLMs to sparse behavior, this would provide an explanation for the performance of LLMs in lieu of the consequences threatened by the curse of dimensionality.

We examine LLMs in order to determine if, and what, sparse behavior they have. The discovery of sparsity would, as explained earlier, help us understand why LLMs and transformers in general perform so well. Another key question building off the existence of sparsity is *if there is sparsity in these models, is there some specific structure or pattern to the sparsity within or across models?* The discovery that there is a structure or pattern to the sparsity would be interesting because this would mean that all models are optimizing to a similar sparse structure without explicit constraints to do so. These patterns could provide the basis for new methods of, or tools for, training.

2 Background & Methodology

Prior work has looked into the multi-layer perceptron(MLP) layers of transformers for sparsity, and found that the activations after a RELU activation function were very sparse(Li et al., 2023b). For example, on average 3% of these activations were nonzero in the T5-Base model(Li et al., 2023b). They also found that sparsity emerged during training, emerged on both NLP and vision tasks and even random inputs/labels, and that explicitly enforcing sparsity in the form of Top-k thresholding led to several improvements in their experiments (Li et al., 2023b). This work supports the hypothesis that sparsity should emerge in neural networks as they become more capable, even if they are not constrained to do so.

Instead of examining the MLP layers of transformers like (Li et al., 2023b), we instead examine the attention mechanism. We do this because the attention mechanism is a key part of the transformer architecture and is what differentiates it from other models. The attention mechanism in transformers for a specific can be formulated by the equation $Attention(X) = Softmax(\frac{XW_Q W_K^T X^T}{\sqrt{d_{emb}}})XW_V$[1], where X is the input, d_{emb} is the dimension of the embedding space, and W_Q, W_K, and W_V are, respectively, the weight matrices for the queries, keys, and values for a specific head. While most models use Multi-Head Attention (MHA) (Vaswani et al., 2017), which means that this above equation is executed numerous times in parallel across different 'heads', newer models have adapted new versions of attention, such as Multi-Query Attention (MQA) (Shazeer, 2019) and Grouped-Query Attention (GQA) (Touvron et al., 2023). For our investigation, we focus on models that only use MHA. While this decision may be a drawback because the newest and state-of-the-art models like LLaMA-2(Touvron et al., 2023) and MistralJiang et al. (2023) will not be included, this simplifies the comparison across models because models using MQA frequently differ in the number of groups they use.

Two models that use MHA, are open-source, and have strong performance are the Phi-1.5(Li et al., 2023a) and MPT-7B(Team, 2023) models. Phi-1.5 is a 1.3 billion parameter model trained on textbook quality data (Li et al., 2023a). MPT-7B is a 6.7 billion parameter model that was trained on 1 trillion tokens(Team, 2023). We select these two models to investigate because of the reasons mentioned above and because while they have impressive performance, they are small enough that they can be run using our limited compute resources. Inside these two models, we study their attention mechanisms by examining both the raw weight matrices inside their attention layers and the attention scores they generate during inference.

We examine the raw weight matrices in the attention mechanism of these models by calculating their stable rank. Stable rank can be defined by the equation $StableRank(M) = \frac{\sigma_1 + ... + \sigma_n}{\sigma_1}$, where M is the matrix, $(\sigma_1, ..., \sigma_n)$ are the singular values of the matrix, and σ_1 is the biggest singular value

[1] adapted from (Vaswani et al., 2017), where the queries are $Q = XW_Q$, keys are $K = XW_K$, and values are $V = XW_V$.

in M. This function ranges from 1 to n, with its value being close to 1 if there is one large singular value, and close to n if all singular values have similar magnitude. Because of this, we can think of this value as describing the number of 'important' dimensions in a weight matrix. Therefore, we can claim that *the lower the stable rank, the sparser the matrix*. While prior work has used the singular values of weight matrices to do spectral analysis in order to determine if a weight matrix is under- or over-trained (Martin et al., 2021), to our knowledge no one has calculated the stable rank of weight matrices in order to calculate their sparsity.

We examine the attention scores generated by these models during inference by calculating the entropy of the attention scores of the last token across the entire input. We define attention scores to be the output of the equation $AttentionScores(X) = Softmax(\frac{XW_QW_K^TX^T}{\sqrt{d_{emb}}})$. Note that this is part of the attention equation described above, but without being multiplied by XW_V. We define entropy using by the equation $H(A) = -\sum_{i=1}^{n} p(a_i)log(p(a_i))$, where $A = AttentionScores(X)_n$ is of size n and contains the attention scores of token n across the entire input. Importantly, because of the softmax function the attention scores sum to 1, ensuring that our attention scores can be treated as a probability distribution as required.

Entropy is a useful measure here because it is an effective proxy for sparsity, and we use this to claim that *the lower the entropy, the sparser the attention scores*. This is because the maximum the entropy can be for a probability distribution is when there is an equal probability assigned to every possible outcome (in our case, this is the n tokens). As probabilities become more and more concentrated, and therefore sparser, the entropy of the probability distribution decreases. Since our attention scores are our 'probabilities', we can see here that this is the behavior we want, since the more concentrated the attention scores become, the sparser they are.

	layers	heads	d_{emb}	Parameters
Phi-1.5	24	32	64	1.3 Billion
MPT-7B	32	32	128	6.7 Billion

Table 1: Details about the two models we examine, Phi-1.5 (Li et al., 2023a) and MPT-7B (Team, 2023).

3 Stable Rank of Attention Matrices

For each layer and each head of MHA, we calculate the stable rank of four matrices: W_Q, W_K, W_V, and $W_QW_K^T$. We calculate the product $W_QW_K^T$ because although this value is not stored as an actual matrix or is ever even calculated during inference, it can be seen as the crucial matrix multiplication for self-attention, since the self-attention operation is $XW_QW_K^TX^T$. Our calculated values are provided in Table 2&3.

As we can see in these tables, the Stable Rank of W_V is high. This makes sense, because this matrix is used to represent the tokens in the embedding space and therefore should be expected to span that space reasonably well. Interestingly, however, is the fact that W_Q and W_V appear to have lower stable rank and plummet in the last layer for both models. Even more interestingly, the matrix product $W_QW_K^T$ has *extremely low stable rank* for all layers and in both models. This indicates sparsity in the attention mechanism of these models.

Importantly, this sparsity is not present during initialization: initializing a matrix the same size as the individual matrices of each model across 10 random seeds resulted in a stable rank of 46.70 for Phi-1.5 and 93.29 for MPT-7B. This indicates that these attention matrices become sparse in their stable rank *during optimization*.

4 Entropy of Attention Scores

In order to calculate attention scores, we need to do inference with text data points. For our dataset, we elect to use TinyStories (Eldan & Li, 2023), a set of simple stories that were generated from GPT-4 (OpenAI, 2023). In order to be consistent when comparing across data points, we select a constant number of tokens to use for each data point that is used for inference. We choose this number to be 200 tokens by eliminating examples with less than 200 tokens and truncating examples with more than 200 tokens. This truncation should not effect model performance, because the models are trained autoregressively and therefore would have been trained to do next token prediction on truncated sentences. We sample 1000 sentences, do inference, and for each layer and each head, we calculate the entropy of the attention scores of the *last token only*. We use the last token only because, due to autoregressive property of these models, other tokens do not pay attention to the whole input due to masking. Our calculated entropy values can be viewed in Figure 1.

It is difficult to scrutinize entropy values on their own. Because of this, it is helpful to use analogous sparsity levels in our analysis. We define these sparsity levels to be the entropy of a probability distribution which has x% zero values and an even probability distribution over the remaining (100-x)% non-zero values. For example, for the 90% sparsity level, we calculated the entropy of a probability distribution in which 90% of its values were zero and an even probability distribution across the remaining 10%.

Given these analogous sparsity levels, we can see that most of the averages and indeed many of the heads fall between the 90% and 99% sparsity levels. This suggests that most of the entropy values calculated are analogous to being 90-99% sparse. This is a high degree of sparsity. However, we must reiterate that these sparsity levels are merely analogies that were created to understand the entropies more easily. They should not be treated as causal evidence that the attention scores have a certain sparsity.

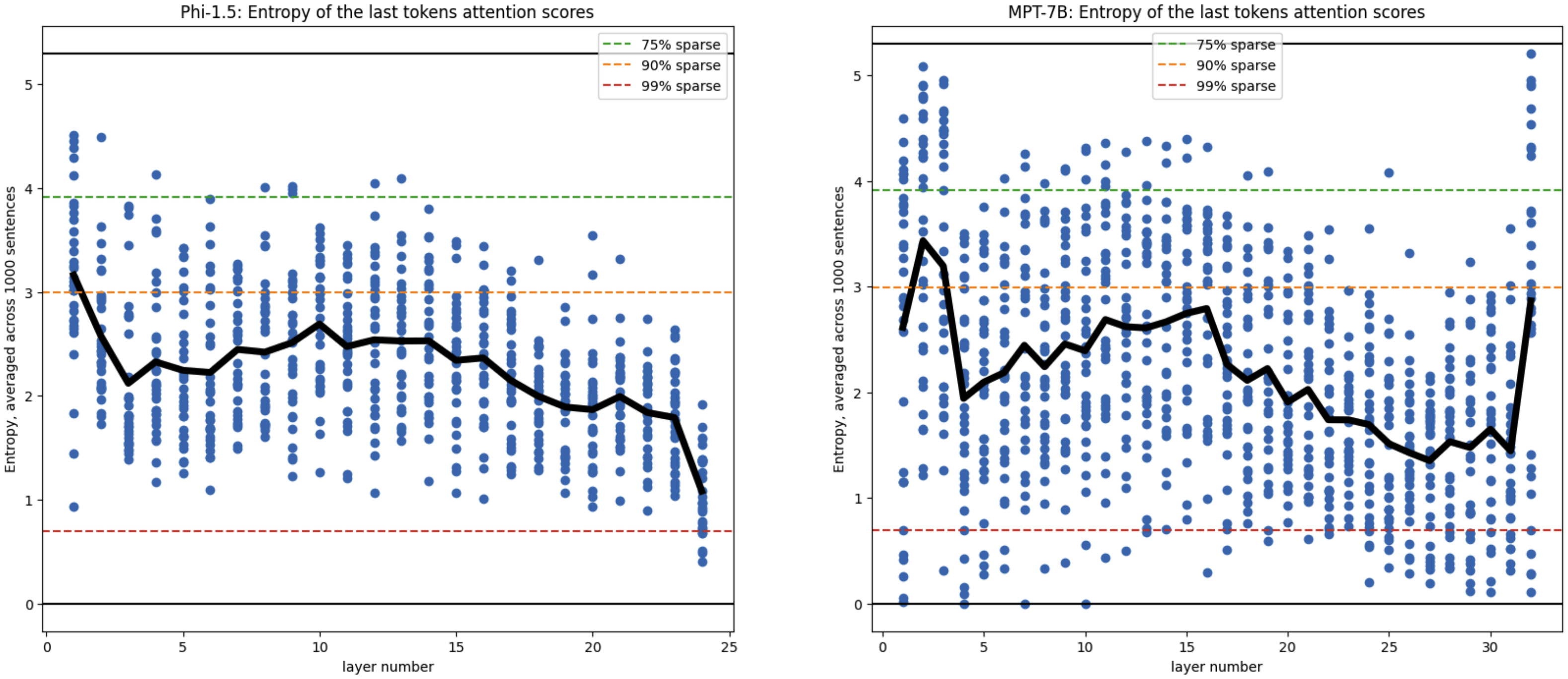

Figure 1: Graphs containing calculated entropy values, averaged over 1000 data points, for Phi-1.5(left)(Li et al., 2023a) and MPT-7B(right)(Team, 2023). Blue dots represent a certain head within that layer, and the black line is the average across heads. The dotted lines correspond to their respective analogous sparsity levels, which assume an even probability distribution over non-zero values.

5 Conclusions

We examined the raw weight matrices and the attention scores generated during inference within the attention mechanisms of two models, Phi-1.5 and MPT-7B. We found both to be sparse in the stable rank of their attention matrices, particularly the matrix product $W_Q W_K^T$. We found that the entropies of the attention scores generated during inference can be interpreted as falling between 90% and 99% sparsity levels by using analogous sparsity levels.

These results suggest that sparse features do appear in transformers and that they appear during optimization. Because of this, these results suggest a possibility of how transformers escape the curse of dimensionality: *by optimizing into sparsity*.

5.1 Future Work

There are several interesting directions to take this work. One obvious direction would be to scale these experiments up to more models, including new state-of-the-art models that use things like GQA. There are also a few interesting experiments that could be conducted: since we have observed that the attention matrices have low stable rank and entropy, what effect would forcing the attention matrices to be low rank approximations of themselves cause on the overall loss and performance of the model? What about using the Sparsemax (Martins & Astudillo, 2016) function instead of Softmax in attention? Also, could we train new models with constrained low rank approximations for their attention matrices and reach similar performance while also using fewer parameters?

Acknowledgments

A special thanks to Akshay Rangamani for mentoring me throughout this project. Without his guidance, this project would not have been possible.

References

Alison L Barth and James F A Poulet. Experimental evidence for sparse firing in the neocortex. *Trends in neurosciences*, 35(6):345—355, June 2012. ISSN 0166-2236. doi: 10.1016/j.tins.2012.03.008. URL https://doi.org/10.1016/j.tins.2012.03.008.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english?, 2023.

Kevin Lee Hunter, Lawrence Spracklen, and Subutai Ahmad. Two sparsities are better than one: Unlocking the performance benefits of sparse-sparse networks, 2021.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report, 2023a.

Zonglin Li, Chong You, Srinadh Bhojanapalli, Daliang Li, Ankit Singh Rawat, Sashank J. Reddi, Ke Ye, Felix Chern, Felix Yu, Ruiqi Guo, and Sanjiv Kumar. The lazy neuron phenomenon: On emergence of activation sparsity in transformers, 2023b.

Charles H. Martin, Tongsu Peng, and Michael W. Mahoney. Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data. *Nature Communications*, 12 (1), July 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-24025-8. URL `http://dx.doi.org/10.1038/s41467-021-24025-8`.

André F. T. Martins and Ramón Fernandez Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification, 2016.

Merriam-Webster. Sparsity. In *Merriam-Webster.com dictionary*. URL `https://www.merriam-webster.com/dictionary/sparsity`.

OpenAI. Gpt-4 technical report, 2023.

Tomaso Poggio. How deep sparse networks avoid the curse of dimensionality: Efficiently computable functions are compositionally sparse, 2023.

Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep – but not shallow – networks avoid the curse of dimensionality: a review, 2017.

Noam Shazeer. Fast transformer decoding: One write-head is all you need, 2019.

MosaicML NLP Team. Introducing mpt-7b: A new standard for open-source, commercially usable llms, 2023. URL `www.mosaicml.com/blog/mpt-7b`.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

Katrin Vonderschen and Maurice Chacron. Sparse and dense coding of natural stimuli by distinct midbrain neuron subpopulations in weakly electric fish. *Journal of neurophysiology*, 106:3102–18, 09 2011. doi: 10.1152/jn.00588.2011.

A Appendix

Phi-1.5 (64 is maximum here)				
Layer	W_Q	W_K	W_V	$W_Q W_K^T$
1	16.38	14.42	37.75	3.52
2	23.36	19.11	39.09	4.26
3	25.04	21.84	37.81	5.97
4	23.43	20.98	33.18	7.76
5	22.39	21.07	37.21	6.24
6	25.35	23.43	36.27	8.74
7	25.23	22.14	34.37	7.12
8	24.5	22.69	33.75	7.49
9	22.84	20.77	37.01	7.06
10	22.63	19.86	38.14	5.93
11	22.53	20.48	37.3	4.8
12	21.67	19.71	35.67	3.94
13	21.34	20.55	36.88	5.45
14	20.56	19.26	38.9	4.02
15	20.54	20.09	39.57	4.31
16	18.96	19.53	39.36	4.4
17	18.32	20.05	39.96	4.31
18	18.32	20.13	38.88	4.0
19	16.15	17.22	40.0	3.24
20	14.62	16.29	38.33	3.4
21	16.69	16.07	42.16	3.5
22	14.8	14.49	41.6	3.01
23	7.27	10.59	42.69	2.57
24	1.87	4.73	39.73	1.6

Table 2: Stable rank for each of $\{W_Q, W_K, W_V, W_Q W_K^T\}$ in each layer averaged across heads for Phi-1.5 (Li et al., 2023a). Note that the maximum value stable rank could be is d_{emb}, which is 64 here.

MPT-7B (128 is maximum here)				
Layer	W_Q	W_K	W_V	$W_Q W_K^T$
1	4.83	4.75	32.93	2.61
2	17.1	15.36	29.51	9.23
3	14.12	14.48	26.83	6.49
4	15.34	18.47	32.22	6.49
5	19.97	25.39	39.85	9.8
6	20.24	27.72	45.38	8.23
7	19.85	25.87	45.46	8.91
8	22.24	32.65	49.82	10.51
9	24.63	35.41	52.05	10.6
10	26.22	39.53	53.6	11.6
11	30.98	42.31	58.59	14.26
12	30.07	43.08	57.72	12.73
13	30.58	45.68	56.78	14.04
14	30.93	47.89	57.03	12.71
15	30.9	42.23	62.22	10.28
16	31.74	41.9	68.04	9.77
17	30.72	42.31	66.6	8.58
18	31.44	42.07	63.62	7.64
19	31.63	39.41	63.02	7.39
20	31.85	39.96	63.83	7.68
21	32.17	42.23	58.94	6.82
22	33.5	40.98	60.4	7.46
23	32.35	39.59	53.18	7.42
24	32.53	36.54	55.63	7.62
25	29.86	32.35	53.91	6.56
26	31.69	33.79	54.62	7.74
27	27.86	28.15	54.56	7.02
28	30.67	32.78	55.9	7.32
29	28.36	32.37	54.83	6.96
30	31.64	37.98	51.59	8.37
31	31.47	41.44	47.18	9.93
32	4.03	4.18	24.24	2.27

Table 3: Stable rank for each of $\{W_Q, W_K, W_V, W_Q W_K^T\}$ in each layer averaged across heads for MPT-7B (Team, 2023). Note that the maximum value stable rank could be is d_{emb}, which is 128 here.

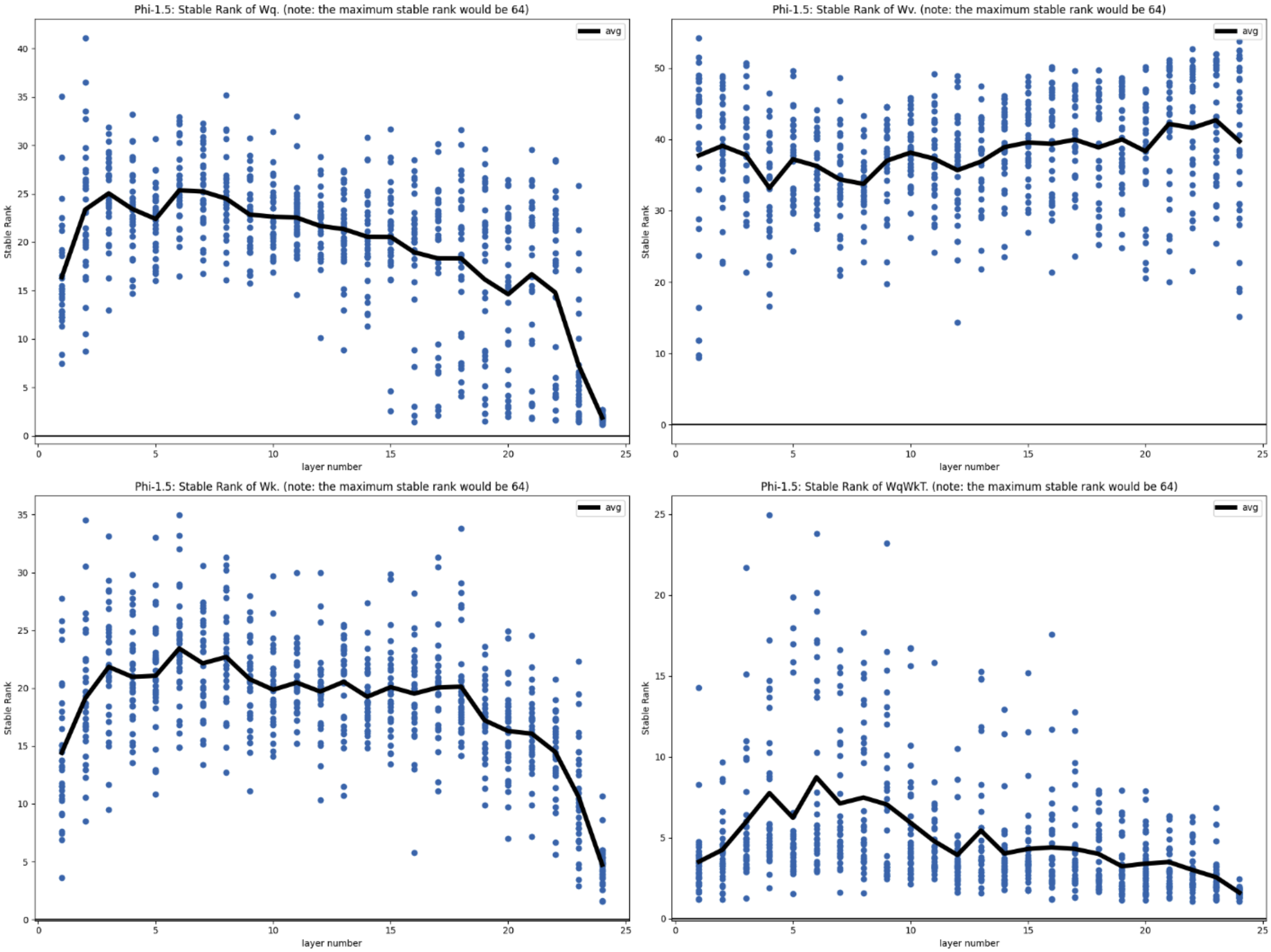

Figure 2: Graphs containing calculated stable rank values for Phi-1.5(Li et al., 2023a). Blue dots represent a certain head within a layer, and the black line is the average across heads.

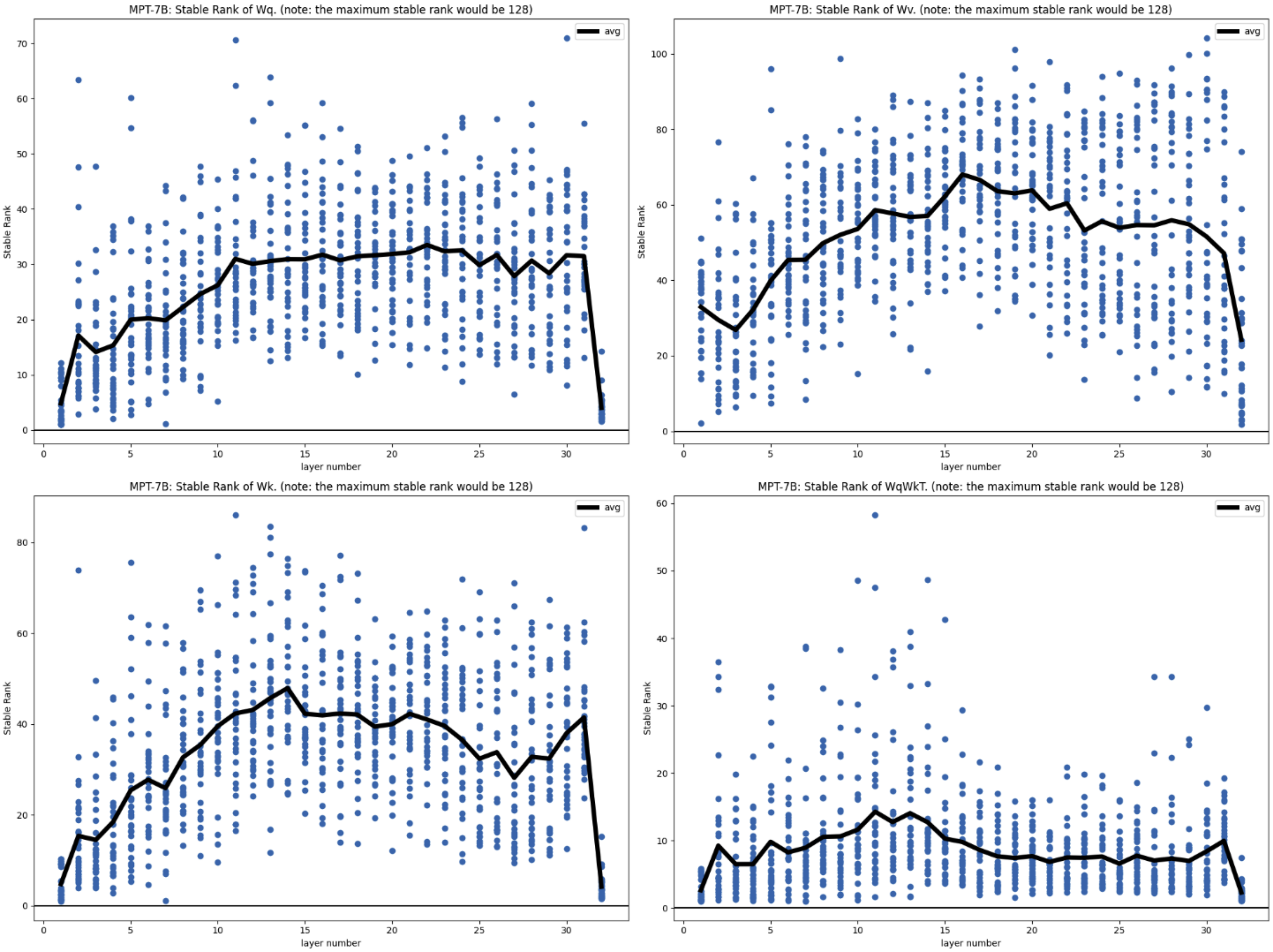

Figure 3: Graphs containing calculated stable rank values for MPT-7B(Team, 2023). Blue dots represent a certain head within a layer, and the black line is the average across heads.