Detecting Learning Dynamics in Graph Transformers via Spectral Deviations

Sydney Dolan
Technical University of Munich
sydney.dolan@tum.de

Max Z. Li*
University of Michigan
maxzli@umich.edu

Abstract

Spectral anomalies in graph shift operators may expose meaningful deviations from expected behavior, offering insights into learned structures, overfitting, or instability in associated graph transformers. We leverage tools from random matrix theory to identify statistically significant deviations in spectral distributions of graph transformer architectures, using the GraphGPS graph transformer architecture as a case study. Matrices extracted from various stages of the training process, such as attention maps, layer outputs, or learned weights, are analyzed to assess whether statistically significant spectral deviations correspond to high-information components or key learning dynamics. We comment on our preliminary work applying random matrix theory in this domain, which reveals distinct spectral signatures across different phases of model learning, and highlights open challenges in extending symmetric random matrix theory frameworks to the inherently non-symmetric matrices found in graph transformers.

1 Introduction

Graph-structured data is ubiquitous in various disciplines, from multi-agent traffic systems to chemical bonds to social networks [1]. Thus, graph representation learning has the potential to provide a huge impact. There are various types of graph neural networks (GNNs), the majority of which are based on a message passing scheme where node representations are computed iteratively by aggregating the embeddings of neighboring nodes. Yet this mechanism in its basic form has limited theoretical expressive power, as these message passing algorithms are not more powerful than the first-order Weisfeiler-Lehman test [2]. Further intrinsic limitations like over-smoothing [3] and over-squashing [4] make it challenging for GNNs to capture long-range dependencies within graphs.

Transformer models have gained popularity in graph learning, due to their potential to capture long-range dependencies. Graph Transformers (GTs) adapt the Transformer architecture [5] to operate on fully connected computational graphs. GT's integrate graph-specific information indirectly using modified attention mechanisms and positional encodings. Despite their success [6, 7], understanding how graph transformer models encode and propagate information is challenging, particularly as they scale in depth and complexity. One promising approach to gain deeper insight is the application of random matrix theory. In deep neural networks, random matrix theory (RMT) has been successfully applied to identify statistically significant deviations from the theoretical asymptotic spectral distributions [8]. Deviations from RMT predictions indicate where feature learning occurs, as opposed to lazy learning [9] where weights remain close to their initialized state.

As demonstrated in [10], the initialization of weights in a deep neural network will also follow RMT predictions. Building on these insights, we use RMT to evaluate the self-attention weight matrices in a classical graph transformer architecture, GraphGPS [11]. We evaluate the statistical significance of spectral deviations of architectural elements across multiple datasets and comment on the differences. By analyzing the spectral and RMT properties of graph transformer weight matrices at the beginning and end of training, we identify patterns that offer insights into learning dynamics and functional characteristics. As a preliminary work, there are many interesting areas to explore. Thus, in Section 5, we highlight these areas of future work.

Preprint. Preliminary work.

2 Related Work

Graph Transformer models use a basic form of node positional encoding to capture graph elements by encoding the eigenvectors of the graph Laplacian. This approach equips the structure-agnostic Transformer with spatial awareness of node positions within the graph [12]. Expanding on this idea, SAN [13] introduced a more refined method by aggregating Laplacian eigenvectors in a permutation-invariant manner for PE. They further enhanced the model with a conditional attention mechanism that distinguishes between real and virtual edges, leading to notable performance gains. In subsequent work [14], a hybrid architecture was proposed that combined message-passing neural networks (MPNNs) with Transformer layers in a modular design. Each layer consists of an MPNN component to encode edge features, a Transformer block to model global interactions, and a linear readout module to integrate their outputs.

RMT has been extensively utilized as a statistical framework for analyzing machine learning models, particularly through the spectral analysis of weight matrices and loss landscapes. Initial applications of RMT, such as in [15], investigated the eigenvalue spectra of loss surfaces, yielding valuable insights into optimization dynamics and generalization. Expanding on this foundation, [16] characterized universal features of outliers within these spectra. More recent studies have applied RMT to trained weight matrices [17], with [8] examining spectral dynamics across training in image classification models. This line of research led to findings such as those in [18], where large singular value outliers were identified as markers of well-trained representations. Further empirical studies [19, 20] have reinforced the idea that deviations from classical RMT predictions signal non-trivial feature learning. Gueddari et al. address non-symmetric RMT, particularly the complexity arising from complex-valued eigenvalues, and propose an approximate message passing algorithm [21].

3 Preliminaries

Graph Transformers (GTs) utilize Transformer architecture [5] within graphs. Their core component is multi-head self-attention, which is a map from the input $X \in \mathcal{R}^{n\times d}$ to $\mathcal{R}^{n\times d}$ as:

$$\text{Attn}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V, \quad \text{selfAttention}(X) = \text{Attn}(XW^Q, XW^k, XW^V) \quad (1)$$

Here, Q, K, V are linear projections of the input, and computed using learned weight matrices $W^Q, W^K, W^V \in \mathbb{R}^{d\times d}$. The attention mechanism captures pairwise similarities between input tokens. In the GT architecture studied in this work, GraphGPS [11], the graph transformer architecture is combined with one round of local neighborhood aggregation via a MPNN layer. This hybrid architecture is intended to reduce the initial representation bottleneck and enables iterative local and global interactions. Assuming efficient representation encoding by the MPNN, the node features can implicitly encode edge information. Thus, when concatenated with the transformer architecture, edges can play a role in the key, query, or value matrices. In our experiments, we focus on analyzing the output of the self-attention matrix.

Marčenko-Pastur Distribution. The Marčenko–Pastur distribution models the asymptotic eigenvalue spread of random matrices. As described by [22], if $R = (1/T)X^\intercal X$, where X is a $T \times N$ matrix with i.i.d. entries of zero mean and fixed variance ν^2, then at the limit T, $N \to \infty$ with ratio $Q = T/N \geq 1$, the eigenvalue probability density function converges to the Marčenko-Pastur distribution:

$$P(\lambda) = \frac{Q}{2\pi\nu^2}\frac{\sqrt{(\lambda_{\max}-\lambda)(\lambda-\lambda_{\min})}}{\lambda}, \quad \lambda_{\min} \leq \lambda \leq \lambda_{\max}, \quad \lambda_{\max\,/\,\min} = \nu^2\left(1 \pm \sqrt{\frac{1}{Q}}\right)^2 \quad (2)$$

The MP distribution defines the theoretical bounds λ_{min} and λ_{max} for the eigenvalues λ_i of purely random matrices. Eigenvalues outside of these bounds indicate a deviation from the expected random matrix behavior. Building on this idea, we can count the number of spikes that significantly deviate from the bulk of this spectrum to estimate, MP, as an indicator of structured signal.

Empirical Spectral Density and Heavy Tailed Distributions. While classical random matrices often exhibit well-known spectral laws (e.g., Marčenko-Pastur), matrices arising in learning systems often display heavy-tailed spectral densities. This means that the distribution of eigenvalues decays slowly, typically following a power law $P(\lambda) \propto \lambda^{-\alpha}$, where α is a parameter that governs the "heaviness" of the tail, and λ_i is the eigenvalues of the weight matrix. In heavy tailed self regularization theory, it is hypothesized that α models a networks ability to generalize, with high alpha values being an indicator of poor learning [18]. To characterize the tail behavior of the power law fit, a cut-off σ is introduced,

and only the eigenvalues $\lambda_i > \sigma$ are used to estimate the tail exponent α. A pronounced heavy tail in the spectrum suggests the presence of structured correlations or long-range dependencies in the data or model, which are often indicative of meaningful learning or hierarchical representations in the system. We also count the number of spikes in the PL distribution, PL. These spikes can signal structures or anomalies that standard spectral statistics might miss.

4 Experiments

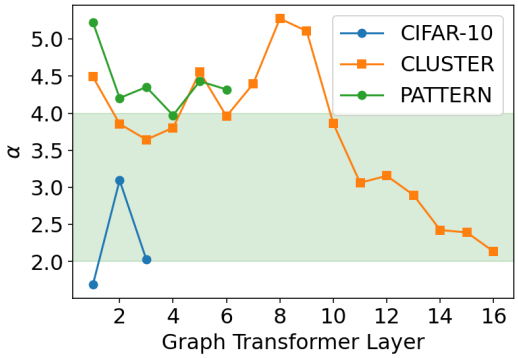


Figure 1: Comparison of fitted power law exponent α for the eigenvalue density of the self-attention weight matrix in GraphGPS.

Heavy Tailed Distribution of Graph Transformer Layers. First, we analyze the power law exponent α fitted to the spectral density of the singular value spectrum for each layer in the self-attention matrix in the graph transformer. This analysis aims to assess whether the heavy tailed spectral properties observed in other deep neural network architectures also manifest in graph transformer self-attention layers. The exponent α serves as an indicator heavy tailed the layer's weight matrix spectrum is, which in turn reflects the degree of correlation and information structure captured by the layer. According to the heavy tailed self regularization theory proposed by [18], deep neural networks that generalize well tend to exhibit spectral densities with power law behavior, where $\alpha \in [2, 4]$ indicates a regime of strong correlation and efficient learning. This region is highlighted in green in Figure 1. Specifically, $\alpha \approx 2$ corresponds to a maximally heavy-tailed stable regime whose layers are highly expressive, capturing meaningful long-range correlations without diverging variance. In contrast, $\alpha \geq 6$ suggests weak or absent heavy-tailed structure, often associated with overfitting, whereas $\alpha \leq 2$ may indicate overfitting or memorization. We evaluate the performance of GraphGPS across three datasets: CIFAR-10, CLUSTER, and PATTERN. Additional information about each dataset is provided in [23]. All results generated are based on a single seed, as in experimental runs across multiple seeds, we did not observe significant variations across results. In Figure 1, we observe that GraphGPS exhibits the most stable and theoretically favorable architecture on the CLUSTER dataset. Its deepest layers trend toward $\alpha \approx 2$, implying effective information propagation. Conversely, for models trained on CIFAR-10, early layers persistently show low α values, suggesting overfitting and poor generalization. Additional experiments exploring a shallower transformer structure (e.g. 2 layers) found the same overfitting issue in the initial layer, with subsequent layers exhibiting α values of ≥ 4, and similar final trained accuracy values in evaluation (0.72 vs 0.71). These results suggest a trade-off in transformer model depth for this dataset: while the deeper architectures did not provide a significant performance improvement, they did lead to more favorable spectral behavior. More experiments are required to determine if this improved spectral behavior leads to generalizability in other evaluations.

The performance on the PATTERN dataset presents an interesting case, as the observed range $4 \leq \alpha \leq 6$ corresponds to a weakly heavy-tailed regime. While this range suggests a less pronounced correlation structure than the optimal heavy-tailed regime, it does not necessarily imply poor learning. Rather, it indicates that the layer weight matrices still exhibit some non-trivial spectra decay, capturing moderately structured features with high-rank, highly correlated representations. Models operating in this regime may still generalize reasonably well (confirmed by GraphGPS's strong performance in evaluation on this dataset), particularly on tasks where the input structure does not demand deep hierarchical abstraction. This suggests that the PATTERN dataset benefits from models that capture mid-level feature interactions, making it well-suited for graph transformers that can leverage such moderately heavy-tailed spectra without requiring deep hierarchical abstractions.

Difference in Spectral and RMT Distributions Across Training. Next, we evaluate the difference in spectral and RMT characteristics at the start and end of training to determine the shift from random (e.g. Marčenko-Pastur distribution) to learned behavior (e.g. power law). We use four metrics: α, MP, PL, and σ. As in the previous analysis, α represents the power law exponent fitted to the spectral density. MP is the number of spikes that occur within the Marčenko-Pastur distribution and PL is the number of spikes outside of the power law distributions. The MP spikes represent

Table 1: self-attention Layer metrics for layers 1–8 of the GraphGPS model applied to the CLUSTER dataset.

Metric	1	2	3	4	5	6	7	8
α_0	30.342	5.900	5.387	15.245	10.709	8.982	5.997	4.559
α_{100}	4.490	3.857	3.644	3.804	4.557	3.959	4.398	5.275
MP_0	0	0	0	0	0	0	0	0
MP_{100}	0	0	0	0	0	0	0	10
PL_0	5	17	19	7	13	14	16	19
PL_{100}	12	15	15	17	13	18	12	10
σ_0	13.122	1.188	1.006	5.384	2.693	2.133	1.249	0.816
σ_{100}	1.008	0.738	0.683	0.680	0.986	0.697	0.981	1.352

Table 2: self-attention Layer metrics for layers 9–16 of the GraphGPS model applied to the CLUSTER dataset.

Metric	9	10	11	12	13	14	15	16
α_0	7.453	21.984	2.773	19.048	8.550	8.958	12.378	13.947
α_{100}	5.107	3.859	3.063	3.157	2.897	2.426	2.393	2.134
MP_0	0	0	0	0	0	0	0	0
MP_{100}	0	0	2	2	0	3	2	4
PL_0	17	8	28	6	15	11	8	7
PL_{100}	13	14	15	21	24	22	23	27
σ_0	1.565	7.419	0.335	7.368	1.949	2.399	4.023	4.893
σ_{100}	1.139	0.764	0.533	0.471	0.387	0.304	0.290	0.218

deviations from the behavior expected of random matrices, and can be interpreted as the number of learned features. The PL spikes represent deviations from the power law fit and may reflect either meaningful learned features or potential mismatches in the accuracy of the α estimate. σ represents the threshold at which the singular values of the weight matrices begin to exhibit a power law behavior. A smaller σ indicates that more of the values are captured by the α power law distribution..

The high α values at the start of training are indicative of the random weight distribution. After training, most layers converge to the range $2 \leq \alpha \leq 5$. The structure of the final layer suggests feature flow, whereas the middle layers show a mild increase in over training, indicating regularization and information compression. The increase in the number of power law (PL) spikes from initialization to convergence indicates the development of dominant outlier modes in the singular value spectrum. These spikes represent directions in weight space that capture strong, high-variance features learned during training. The evolution of the σ metric further supports this interpretation. In the early layers, σ values decrease sharply from initialization to convergence, signaling a regularization of the initially over-parameterized layers. Meanwhile, deeper layers exhibit either stable or increasing σ, consistent with the consolidation of high-signal modes critical for task-specific generalization. We also evaluated the fit of Marčenko-Pastur distribution to the spectral density spread of the self-attention weight matrices. As expected at initialization, there were no significant spikes above the MP bulk. At the end of training, deeper layers possess more spikes, indicating learned signal. Metrics like α and PL are useful in this case, as they can provide deeper insight into the role of each layer and their associated information flow. Future work will investigate the deviations between the Marčenko–Pastur fit and the power-law behavior more systematically, to better understand how each technique captures different aspects of learned structure in self-attention layers.

5 Discussion

Our preliminary experiments show that observing spectral deviations in graph transformer weight matrices could help dissect learning dynamics. Early layers often remain closer to a random matrix theory-predicted spectra, whereas deeper layers show heavy-tailed regimes and outliers, potentially signaling emergence of effective, identified training features. Moving forward, we aim to extend sparse random matrix theory [24] to better align with dimensions of the attention matrices commonly used in graph transformers. We aim to develop metrics that formally relate input graph structure to random matrix theory, offering deeper insight into how graph properties affect learning. In the short term, we will explore additional architectures and datasets (e.g. [25]) to study the relationship between α and problem generalizability. Future work may also examine symmetric graph shift operators, whose randomized spectra should follow Wigner or semicircle laws.

References

[1] Ahsan Shehzad, Feng Xia, Shagufta Abid, Ciyuan Peng, Shuo Yu, Dongyu Zhang, and Karin Verspoor. Graph transformers: A survey. *arXiv preprint arXiv:2407.09777*, 2024. 1

[2] Muhammet Balcilar, Pierre Héroux, Benoit Gauzere, Pascal Vasseur, Sébastien Adam, and Paul Honeine. Breaking the limits of message passing graph neural networks. In *International Conference on Machine Learning*, pages 599–608. PMLR, 2021. 1

[3] T Konstantin Rusch, Michael M Bronstein, and Siddhartha Mishra. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*, 2023. 1

[4] Jhony H Giraldo, Konstantinos Skianis, Thierry Bouwmans, and Fragkiskos D Malliaros. On the trade-off between over-smoothing and over-squashing in deep graph neural networks. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pages 566–576, 2023. 1

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017. URL https://arxiv.org/pdf/1706.03762.pdf. 1, 2

[6] Michael Elrod, Niloufar Mehrabi, Rahul Amin, Manveen Kaur, Long Cheng, Jim Martin, and Abolfazl Razi. Graph based deep reinforcement learning aided by transformers for multi-agent cooperation. *arXiv preprint arXiv:2504.08195*, 2025. 1

[7] Sydney Dolan, Siddharth Nayak, Jasmine Jerry Aloor, and Hamsa Balakrishnan. Asynchronous cooperative multi-agent reinforcement learning with limited communication, 2025. URL https://arxiv.org/abs/2502.00558. 1

[8] Charles H Martin and Michael W Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021. 1, 2

[9] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019. 1

[10] Maciej Skorski, Alessandro Temperoni, and Martin Theobald. Revisiting weight initialization of deep neural networks. In *Asian conference on machine learning*, pages 1192–1207. PMLR, 2021. 1

[11] Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer, 2023. URL https://arxiv.org/abs/2205.12454. 1, 2

[12] Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs, 2021. URL https://arxiv.org/abs/2012.09699. 2

[13] Devin Kreuzer, Dominique Beaini, William L. Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention, 2021. URL https://arxiv.org/abs/2106.03893. 2

[14] Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer, 2023. URL https://arxiv.org/abs/2205.12454. 2

[15] Jeffrey Pennington and Yasaman Bahri. Geometry of neural network loss surfaces via random matrix theory. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 2798–2806. JMLR.org, 2017. 2

[16] Nicholas P Baskerville, Jonathan P Keating, Francesco Mezzadri, Joseph Najnudel, and Diego Granziol. Universal characteristics of deep neural network loss surfaces from random matrix theory. *J. Phys. A Math. Theor.*, 55(49):494002, December 2022. 2

[17] Vardan Papyan. Traces of class/cross-class structure pervade deep learning spectra. *Journal of Machine Learning Research*, 21(252):1–64, 2020. 2

[18] Charles H Martin, Tongsu Serena Peng, and Michael W Mahoney. Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data. *Nat. Commun.*, 12(1):4122, July 2021. 2, 3

[19] Matthias Thamm, Max Staats, and Bernd Rosenow. Random matrix analysis of deep neural network weight matrices. *Physical Review E*, 106(5):054124, 2022. 2

[20] Noam Levi and Yaron Oz. The underlying scaling laws and universal statistical structure of complex datasets. *arXiv preprint arXiv:2306.14975*, 2023. 2

[21] Mohammed-Younes Gueddari, Walid Hachem, and Jamal Najim. Approximate message passing for general non-symmetric random matrices, 2025. URL https://arxiv.org/abs/2503.20409. 2

[22] Laurent Laloux, Pierre Cizeau, Marc Potters, and Jean-Philippe Bouchaud. Random matrix theory and financial correlations. *International Journal of Theoretical and Applied Finance*, 03(03):391–397, July 2000. ISSN 1793-6322. doi: 10.1142/s0219024900000255. URL http://dx.doi.org/10.1142/S0219024900000255. 2

[23] Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *Journal of Machine Learning Research*, 24(43):1–48, 2023. 3

[24] Timothy Rogers. *New results on the spectral density of random matrices*. PhD thesis, King's College London, 2010. 4

[25] Vijay Prakash Dwivedi, Ladislav Rampášek, Michael Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. *Advances in Neural Information Processing Systems*, 35:22326–22340, 2022. 4

[26] Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, November 2009. ISSN 1095-7200. doi: 10.1137/070710111. URL http://dx.doi.org/10.1137/070710111. 6

A Appendix

A.1 Code

All code affiliated with this project is open source and anonymized for peer review. It can be found in this repository: https://anonymous.4open.science/r/spectral_assess-8572/README.md.

A.2 α Power Law Fitting Process

To estimate the power-law exponent α, we follow the maximum likelihood estimation (MLE) approach described by Clauset et al. (2009) [26]. The estimator is given by:

$$\hat{\alpha} = 1 + n \left[\sum_{i=1}^{n} \ln \left(\frac{x_i}{x_{\min}} \right) \right]^{-1} \tag{3}$$

where $x_i \geq x_{\min}$, and n is the number of data points in the tail (i.e., those satisfying $x_i \geq x_{\min}$). In the notation used in this paper, x_{min} is equal to σ.

To determine the appropriate lower bound $x_{\min}$, we select the value that minimizes the Kolmogorov–Smirnov (KS) statistic, which quantifies the maximum distance between the empirical and model cumulative distribution functions (CDFs). Specifically, we define:

$$D = \max_{x \geq x_{\min}} |S(x) - P(x)| \tag{4}$$

where $S(x)$ is the empirical CDF of the data for $x \geq x_{\min}$, and $P(x)$ is the CDF of the best-fit power law model over the same region. The optimal threshold $\hat{x}_{\min}$ is chosen to minimize this distance D, ensuring the best possible agreement between the empirical data and the model in the power law region.

A.3 Effect of Model Depth on α in CIFAR-10 dataset

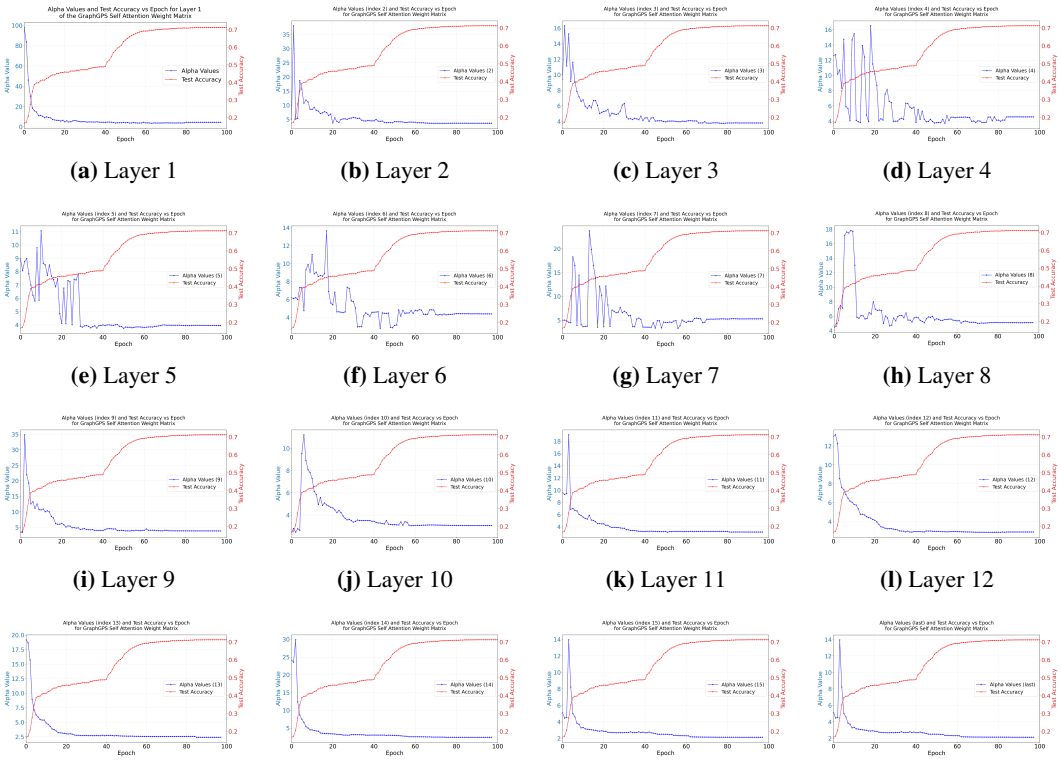
For the two-layer variant, since the αs are increasing as we move down the model, this means that the information is not flowing well through the model, and the network is not fully correlated. When the number of model layers increases to three layers, the α parameters stabilizes, remaining in the optimal$\alpha \in [2, 4]$ regime.

	Two Layer		Three Layer		
Metric	Layer 1	Layer 2	Layer 1	Layer 2	Layer 3
α	1.775	4.629	1.689	3.095	2.027
PL	24	7	19	12	22
MP	3	1	4	2	1
σ	0.158	1.37	0.158	0.604	0.219

Table 3: The effect of model depth on performance for the CIFAR-10 dataset. α is the power-law exponent fitted to the spectral density. PL is the number of spikes in the power law distribution. MP is the number of spikes that occur within the Marcenko-Pastur distribution. σ is the threshold at which the singular values of the weight matrix begin to exhibit power law behavior. The three layer model was used to generate the results for CIFAR-10 on Figure 1.

A.4 α across Training for the CLUSTER dataset.

Figure 2 has the plot of the model accuracy vs alpha variable over training for each layer in the graph transformer self-attention matrix in the CLUSTER dataset.



(a) Layer 1 **(b)** Layer 2 **(c)** Layer 3 **(d)** Layer 4

(e) Layer 5 **(f)** Layer 6 **(g)** Layer 7 **(h)** Layer 8

(i) Layer 9 **(j)** Layer 10 **(k)** Layer 11 **(l)** Layer 12

(m) Layer 13 **(n)** Layer 14 **(o)** Layer 15 **(p)** Layer 16

Figure 2: The α parameter across training epochs for the self-attention matrices of layers 1–16 in the GraphGPS transformer on the CLUSTER dataset. The blue lines represent α values across training, and the red lines show model accuracy during evaluation.