OPT2025: 17th Annual Workshop on Optimization for Machine Learning

Evolution of the Spectral Dimension of Transformer Activations

Andy Zeyi Liu ANDY.LIU@YALE.EDU
Yale University

Elliot Paquette ELLIOT.PAQUETTE@MCGILL.CA
McGill University

John Sous JOHN.SOUS@YALE.EDU
Yale University

Abstract

Transformers learn from high-dimensional representations, making it challenging to interpret how training shapes the process. In this work, we reveal that hidden activations exhibit consistent power-law heavy-tail spectral decay with exponents $\alpha < 1$, gradually increasing across layers and fine-tuning. This spectral evolution offers a compact signature of training dynamics, with larger α values empirically correlating with better generalization. Complementing this, we further find that the gradient SVD spectrum has exponents decreasing over depth, suggesting that gradients become increasingly isotropic as they backpropagate. Together, these spectral signals offer an alternative lens in examining the hidden structure in transformers, which potentially inspire new ways to optimize pre-training and push the scaling-law frontier inward.

1. Introduction

Large language models (LLMs) exhibit a striking macroscopic regularity known as the *neural scaling law*: that is, as model size N, dataset size D, or computational budget C are increased, the model's loss decreases as a power law, $\mathcal{L} \propto N^{-\alpha_N}$ or $D^{-\alpha_D}$ [1, 2, 3]. This empirical law underpins the rapid scaling of model size in recent years [4, 5, 6, 7] and motivates the search for a better understanding of the underlying mechanisms. Recent theoretical works suggest that these scaling exponents are not arbitrary: rather, they have strong connections with the spectral decay of data or kernel matrices [8]. A solvable random-feature model was also proposed to demonstrate how the power-law spectrum in the data translates into power-law scaling of the loss [9]. These indicate that the macroscopic behavior of scaling laws is tied to spectral properties of the data and model.

However another aspect is how the training modifies power-exponents. Meanwhile, empirical studies have discovered consistent spectral patterns within deep networks. Instead of following the Marchenko-Pastur distribution predicted by random matrix theory, the weight matrices, hidden activations, and gradients of trained models often exhibit heavy-tailed spectra [10, 11, 12, 13, 14]. These power-law behaviors appear across architectures and training regimes, and have been linked to generalization performance. Proposed explanations range from implicit regularization and reduced training temperature to the entropy of eigenvectors [15, 16]. Even the noise in stochastic gradients has been shown to follow heavy-tailed α-stable distributions, potentially aiding exploration during optimization [17]. Collectively, these results point toward spectral exponents as rich

descriptors of training dynamics.

Yet transformers remain particularly difficult to probe, as their training takes place in high-dimensional spaces, making it hard to observe the training dynamics directly. In this work, we explore the use of the heavy-tail exponent α of the eigenvalue (or singular value) spectrum, as a low-dimensional signature. We are particularly motivated by theoretical results that relate the eigenvalue decay rate $\lambda_i \sim i^{-\alpha}$ to the scaling-law exponent itself [8, 18, 9]. We conjecture that for nonlinear transformers a similar relationship holds: if the covariance eigenvalues λ_i decay as $i^{-\alpha}$, then α may serve as a proxy for the neural scaling-law exponent. Spectral exponents can be estimated from a single model without training across multiple scales, providing a cheap diagnostic for training efficiency and optimization dynamics. Such insights could offer practical guidance for improving model performance under limited resources, and help push the scaling-law curve inward. We therefore pose the following questions:

1. **Activation spectra evolution**: How do the eigenvalue spectra of hidden activations evolve during training? Does fine-tuning change the spectral decay compared to using plain pre-trained weights?
2. **Gradient spectra evolution**: How do the singular value spectra of per-example gradients change across layers? Is there a consistent relationship between gradient and activation spectra?
3. **Proxy for scaling**: Can the slope α of these spectra act as a proxy for the neural scaling-law exponent? While a rigorous theoretical link remains open, we discuss evidence and limitations.

1.1. Contributions and Outlines

In order to investigate, we perform experiments of spectral dynamics in GPT2 models of different sizes and datasets. Our main contributions are: (i) We characterize the layer- and sublayer- activation covariance spectra and show that, α increases gradually as we progress from layer to layer. Moreover, apart from the embedding and final residuals, the heavy-tail exponents cluster tightly in the range 0.65-0.90 across most depths and training conditions. This narrow band points to a consistent heavy-tailed regime in deep transformer representations. (ii) We analyze gradient singular value spectra across layers. Opposite to the activation spectrum, the gradient singular value spectra decrease almost monotonically, but within an even tighter range.

2. Evolution of Spectra for Pre-trained Models

In order to investigate the heavy-tailed spectra, we examine from two complementary perspectives: the activation covariance spectrum, which captures correlations in hidden representations, and the gradient SVD spectrum, which probes the anisotropy of gradients across examples.

2.1. Methodology

2.1.1. ACTIVATION COVARIANCE SPECTRUM

Consider an autoregressive transformer with L layers. For each layer $l = 0, \ldots, L-1$, we collect activations $X_l \in \mathbb{R}^{m \times d}$ across m token positions, where d is the feature dimension. Unless

otherwise specified (as in the sublayer analysis), X_l refers to the output context vectors of the l-th transformer block. The empirical variance is

$$C_l = \frac{1}{N} X_l^T X_l \in \mathbb{R}^{d \times d}$$

and we sort its spectrum $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$. To avoid storing all vectors in memory, we adopt Welford's algorithm [19], a numerically stable online covariance estimation method. Note that this algorithm provides the exact covariance (within numerical precision) and does not introduce approximation errors relative to the batch statistics. Specifically, given a batch of activations X_l, we update the empirical mean μ and second moment matrix M_2 as

$$\mu_{\text{new}} = \mu + \frac{m}{n+m}(\bar{X} - \mu) \quad M_{2,\text{new}} = M_2 + (X - \bar{X})^T(X - \bar{X}) + \frac{nm}{n+m}(\bar{X} - \mu)(\bar{X} - \mu)^T$$

where n is the total number of samples seen so far and $\bar{X}$ the batch mean. At the end of the pass, we form the covariance matrix $C_l = M_2/(n-1)$. A heavy-tail manifests as a power-law decay $\lambda_i \propto i^{-\alpha}$ for ranks i in an intermediate range. We estimate α by performing linear regression to $\log \lambda_i$ against $\log i$ curve over a window $[i_s, i_e]$ (we take it to be 10-100 in our experiments) and take $\alpha = -\text{slope}$.

2.1.2. GRADIENT SVD SPECTRUM

In this case, we select a weight matrix $W_{l,k}$ in layer l (e.g. the output projection of self-attention). We then collect per-sample gradient by processing 1000 sequences of length 256 from the dataset with batchsize 1. For each example x, we perform a forward pass to compute the loss $\mathcal{L}(x)$, followed by a backward propagation to obtain the gradient of the loss with respect to the weights $W_{l,k}$. We then flatten this gradient into a vector $g_x \in \mathbb{R}^p$ and store it. We stack the collected gradients g_x into a matrix $G \in \mathbb{R}^{n \times p}$. We then perform SVD to capture the singular value spectrum. The exponent α of this spectrum is computed similarly as before.

2.2. Experimental setup

We consider two model-dataset pairs. The base experiment uses the 12-layer GPT2-small [20] with weights fetched from the HuggingFace [21] pre-trained model and evaluated on the Wikitext-2 corpus [**merity2016pointer**]. A randomly initialized model with the same architecture serves as the baseline. We collect spectra under 5 conditions; (i) fine-tuned for 2 epochs (FT epoch-2) evaluated on a subset of the training split; (ii) FT epoch-2 evaluated on the test set; (iii) No FT, and evaluated on a subset of training set (iv) No FT, and evaluated on test set (v) random initialization, evaluated on a subset of the training set. Experiments are repeated with 5 random seeds and we report the averages.

The second larger experiment uses the 24-layer GPT2-medium [20] with Wikitext-103 corpus [**merity2016pointer**]. We set the sequence length to 1024 and activation spectra are computed on 10k sampled sequences with dropout disabled. We also perform a sublayer analysis in this case, by extracting activations from the embedding output and nine intermediate sublayers per transformer block (layer-norm 1, attention output, attention residual, layer-norm 2, fully connected MLP, MLP activation, MLP projection, dropout, MLP residual) as well as the final layer output. Gradient

spectra are collected for the attention output projection weight matrices of each layer from 1000 per-example gradients. These analyses allow us to compare how heavy-tail exponents vary across sublayers, layers, various training conditions and datasets.

3. Main Results

In this section, we present our empirical results and summarize key findings.

3.1. Results for GPT2-small + Wikitext-2

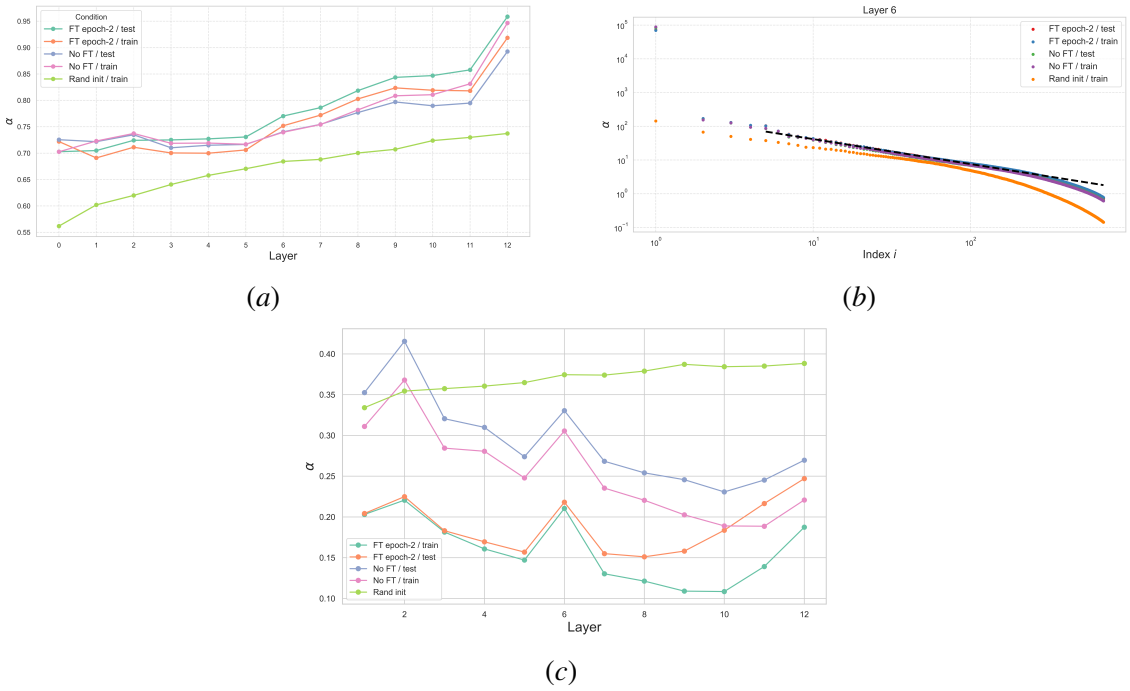


Figure 1: Spectral properties of activations and gradients in GPT2-small.(a) Layer-wise activation spectral exponent α for five conditions: random initialization, no fine-tuning + spectrum computed by passing in a subset of training/the full test set, and fine-tuned (passing in train/test). (b) Activation eigenvalue spectra at layer 6 under the same five conditions. (c) Gradient singular value spectral exponent α across layers for the attention output projection matrix under the same 5 conditions.

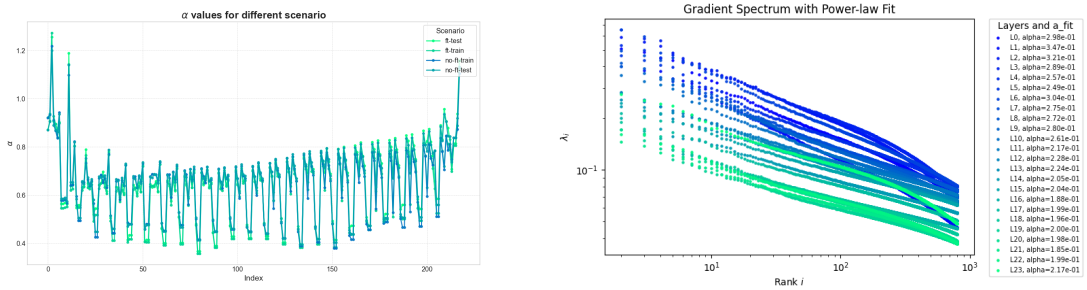
We first present the result for the activation covariance spectrum. In Figure 1a we observe the following trends: (i) Shallow layers have similar slopes. For layers 0-5 the exponents concentrate between 0.70 and 0.75 for all trained models, indicating a relatively slow spectral decay. (ii) Deeper layers compress activations. Beyond layer 6 the slopes increase consistently. In the fine-tuned model the exponent increases to around 0.90 at the final layer, implying a much faster decay of eigenvalues and hence a lower effective dimensionality. This phenomenon aligns with observations in other architectures where deeper layers have been shown to compress representations [22].

This monotonic increase suggests that transformer layers gradually distill information and suppress noise. (iii) Fine-tuning sharpens the spectrum. The effect is subtle compared with the difference between random initialization and pre-trained models.
Overall, the activation spectra exhibit power-law behaviour across a substantial range of ranks. Apart from the embedding and final residual layers, the fitted exponents lie in a tight interval of roughly 0.65-0.90, increasing with depth and fine-tuning. To give a more explicit visualization, in Figure 1b we plot the full spectrum for layer 6 under the same 5 conditions. α is the slope of the black fitted dashed line. It highlights how training steepens the spectral decay relative to random initialization.

Next we examine the singular values of per-example gradients for the attention output projection matrix. Figure 1c plots the evolution of α under different conditions. Opposite to the activation spectra, the gradient spectra in general present a decreasing pattern. And there is a noticeable difference between fine-tuned curves and those without fine-tuning. Fine-tuned models (cyan and orange curves) show smaller α values overall and they resemble U-shaped, with exponents being large in the shallowest and final layers. Moreover, as with activation, the spectra on the training and test splits are almost identical.

3.2. Results for GPT2-medium + Wikitext-103

In addition to GPT2-small, we analyze the spectra for a 24-layer GPT2-medium model. While the small model aggregates activations at the block level, here we decompose each block into finer components and for each sublayer we estimate the heavy-tail α. In Figure 2a we show the activa-



(*a*) (*b*)

Figure 2: Spectral behavior of GPT2-medium with Wikitext-103. (a) Heavy-tail exponent α for activation spectra across sublayers in the 24-layer model under 4 conditions: no fine-tuning + spectrum computed by passing in a subset of training/the full test set, and fine-tuned (passing in train/test). (b) Gradient spectral exponent α for the attention output projection weights under no-fine-tuning (train set).

tion spectra. Several patterns emerge. Firstly by zooming in, the exponent presents a bowl-shaped trajectory. The embedding output and the final layers exhibit the largest exponents ($\alpha \approx 1.1 - 1.3$). Moving deeper into the network, the exponents drop sharply and then steadily increases, mirroring the monotonic increase observed in GPT2-small. Besides, the fine-tuned models have slightly

higher heavy-tail exponents than those of pre-trained models, particularly in later sublayers, suggesting that fine-tuning accelerates spectral decay. Within a sublayer, different components have distinct exponents. Layer-norm outputs and attention residuals typically have smaller α than MLP activations and residuals, implying that attention mechanisms retain a broader range of directions while MLPs compress more aggressively. A more detailed exposition can be found in Appendix 5.1.

We also computed singular value spectra of per-example gradients for the attention output projection across all 24 layers of GPT2-medium for no-FT-train condition, as shown in Figure 2b. Here the heavy-tail exponent decreases almost monotonically from about 0.30 in the shallowest layer to roughly 0.20 in the later layers. The behaviour is consistent with our previous observation from GPT2-small, implying the universality of the pattern.

3.3. Interpretation and hypotheses

1. **Compression by training**. From previous experiments, fine-tuning sharpens activation spectra and raises α, thereby reducing the effective dimension. SGD, combined with layer normalization and weight decay, selectively amplifies a few principal components and suppresses the tail. This explains why model with random initialization yields a shallower decay than trained models. *Hypothesis*: increasing dropout or weight decay should further increase α.

2. **Train-test consistency**. Activation and gradient spectra are nearly identical when passing through train and test datasets because both splits sample the same language distribution. However, this may be due to the limited size of the dataset. Nonetheless, it implies that the spectral geometry is governed more by model architecture and training than by the particular corpus.

3. **Distinct spectra for sublayers**. From our experiments, we observe that attention outputs often exhibit shallower tails than MLP activations, while dropout and MLP residuals show steeper decay. This agrees with the intuition that attention combines multiple token contexts via softmax kernels, producing a broad spectrum, whereas MLPs act as a filter that prunes dimensions. *Hypothesis*: sharing query/key projections or reducing the diversity of attention heads (e.g. through sliding-window attention [23]) should increase α in attention sublayers.

4. **Gradient spectra**. Larger values indicate that the gradients are dominated by a few directions, while smaller values indicate more uniform gradients. In both sets of experiments, under the condition of no-FT, the gradient spectra decrease almost monotonically. This suggests that deeper networks gradually isotropize gradients as error signals propagate backward. *Hypothesis*: switching to more effective optimizer (e.g. Muon [24, 25]) or increased depth should flatten the gradient tail.

4. Discussion and Outlook

Through our analysis, we find that transformer activations and gradients exhibit remarkably consistent spectral patterns: aside from the embeddings and final outputs, heavy-tail exponents predominantly cluster between 0.65 and 0.90, with deeper layers and fine-tuning sharpening the decay, while gradient spectra fall almost monotonically with depth. These spectral slopes provide a compact, low-dimensional signature of high-dimensional training dynamics, offering a complementary

diagnostic to loss curves. Intriguingly, we observe that larger α in late layers and flatter gradient spectra correlate with improved generalization under fixed compute, suggesting that shaping spectra via normalization schemes, attention mechanisms, or optimization strategies may effectively shift the scaling-law frontier inward. This hints at a new avenue for improving training efficiency—not by scaling up, but by reshaping internal dynamics. However, a theoretical justification linking α to scaling exponents in the case of transformers is still lacking. Our empirical findings, drawn from GPT2 variants and curated corpora, serve as a first step. Future research should investigate whether targeted interventions (e.g., adjusting dropout, QK normalization, or optimizer temperature) can steer α in a predictable direction, validate these trends in larger and multimodal models, and establish a formal connection between spectral structure and compute-optimal training.

References

[1] Jared Kaplan et al. *Scaling Laws for Neural Language Models*. 2020. arXiv: 2001.08361 [cs.LG]. URL: https://arxiv.org/abs/2001.08361.

[2] Jordan Hoffmann et al. *Training Compute-Optimal Large Language Models*. 2022. arXiv: 2203.15556 [cs.CL]. URL: https://arxiv.org/abs/2203.15556.

[3] Ben Sorscher et al. *Beyond neural scaling laws: beating power law scaling via data pruning*. 2023. arXiv: 2206.14486 [cs.LG]. URL: https://arxiv.org/abs/2206.14486.

[4] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL]. URL: https://arxiv.org/abs/2005.14165.

[5] DeepSeek-AI. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. 2025. arXiv: 2501.12948 [cs.CL]. URL: https://arxiv.org/abs/2501.12948.

[6] Aaron Grattafiori et al. *The Llama 3 Herd of Models*. 2024. arXiv: 2407.21783 [cs.AI]. URL: https://arxiv.org/abs/2407.21783.

[7] Darioush Kevian et al. *Capabilities of Large Language Models in Control Engineering: A Benchmark Study on GPT-4, Claude 3 Opus, and Gemini 1.0 Ultra*. 2024. arXiv: 2404.03647 [math.OC]. URL: https://arxiv.org/abs/2404.03647.

[8] Yasaman Bahri et al. "Explaining neural scaling laws". In: *Proceedings of the National Academy of Sciences* 121.27 (June 2024). ISSN: 1091-6490. DOI: 10.1073/pnas.2311878121. URL: http://dx.doi.org/10.1073/pnas.2311878121.

[9] Alexander Maloney, Daniel A. Roberts, and James Sully. *A Solvable Model of Neural Scaling Laws*. 2022. arXiv: 2210.16859 [cs.LG]. URL: https://arxiv.org/abs/2210.16859.

[10] Brian Richard Olsen et al. *From SGD to Spectra: A Theory of Neural Network Weight Dynamics*. 2025. arXiv: 2507.12709 [cs.LG]. URL: https://arxiv.org/abs/2507.12709.

[11] Blake Bordelon, Alexander Atanasov, and Cengiz Pehlevan. *A Dynamical Model of Neural Scaling Laws*. 2024. arXiv: 2402.01092 [stat.ML]. URL: https://arxiv.org/abs/2402.01092.

[12] Mert Gurbuzbalaban, Umut Şimşekli, and Lingjiong Zhu. *The Heavy-Tail Phenomenon in SGD*. 2021. arXiv: 2006.04740 [math.OC]. URL: https://arxiv.org/abs/2006.04740.

[13] Charles H. Martin and Michael W. Mahoney. *Implicit Self-Regularization in Deep Neural Networks: Evidence from Random Matrix Theory and Implications for Learning*. 2018. arXiv: 1810.01075 [cs.LG]. URL: https://arxiv.org/abs/1810.01075.

[14] Charles H. Martin and Michael W. Mahoney. *Traditional and Heavy-Tailed Self Regularization in Neural Network Models*. 2019. arXiv: 1901.08276 [cs.LG]. URL: https://arxiv.org/abs/1901.08276.

[15] Liam Hodgkinson, Zhichao Wang, and Michael W. Mahoney. *Models of Heavy-Tailed Mechanistic Universality*. 2025. arXiv: 2506.03470 [stat.ML]. URL: https://arxiv.org/abs/2506.03470.

[16] Vignesh Kothapalli et al. *From Spikes to Heavy Tails: Unveiling the Spectral Evolution of Neural Networks*. 2025. arXiv: 2406.04657 [cs.LG]. URL: https://arxiv.org/abs/2406.04657.

[17] Mert Gurbuzbalaban, Umut Simsekli, and Lingjiong Zhu. "The Heavy-Tail Phenomenon in SGD". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 3964–3975. URL: https://proceedings.mlr.press/v139/gurbuzbalaban21a.html.

[18] Elliot Paquette et al. *4+3 Phases of Compute-Optimal Neural Scaling Laws*. 2025. arXiv: 2405.15074 [stat.ML]. URL: https://arxiv.org/abs/2405.15074.

[19] B. P. Welford. "Note on a method for calculating corrected sums of squares and products". In: *Technometrics* 4.3 (1962), pp. 419–420.

[20] Alec Radford et al. "Language Models are Unsupervised Multitask Learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[21] Thomas Wolf et al. "HuggingFace's Transformers: State-of-the-art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 2020, pp. 38–45.

[22] Wojciech Masarczyk et al. "The Tunnel Effect: Building Data Representations in Deep Neural Networks". In: *Advances in Neural Information Processing Systems*. 2023.

[23] Iz Beltagy, Matthew E. Peters, and Arman Cohan. *Longformer: The Long-Document Transformer*. 2020. arXiv: 2004.05150 [cs.CL]. URL: https://arxiv.org/abs/2004.05150.

[24] Jeremy Bernstein and Laker Newhouse. *Old Optimizer, New Norm: An Anthology*. 2024. arXiv: 2409.20325 [cs.LG]. URL: https://arxiv.org/abs/2409.20325.

[25] Jingyuan Liu et al. *Muon is Scalable for LLM Training*. 2025. arXiv: 2502.16982 [cs.LG]. URL: https://arxiv.org/abs/2502.16982.

5. Appendix

5.1. Detailed sublayer spectra analysis

To further investigate the spectra behaviour, Figure 3 displays the full activation spectra for each sublayer group across all 24 layers. This is under no-FT-train condition. Each subplot shows the eigenvalues λ_i versus index i on a loglog scale together with fitted power-law exponents on ranks 10-100.

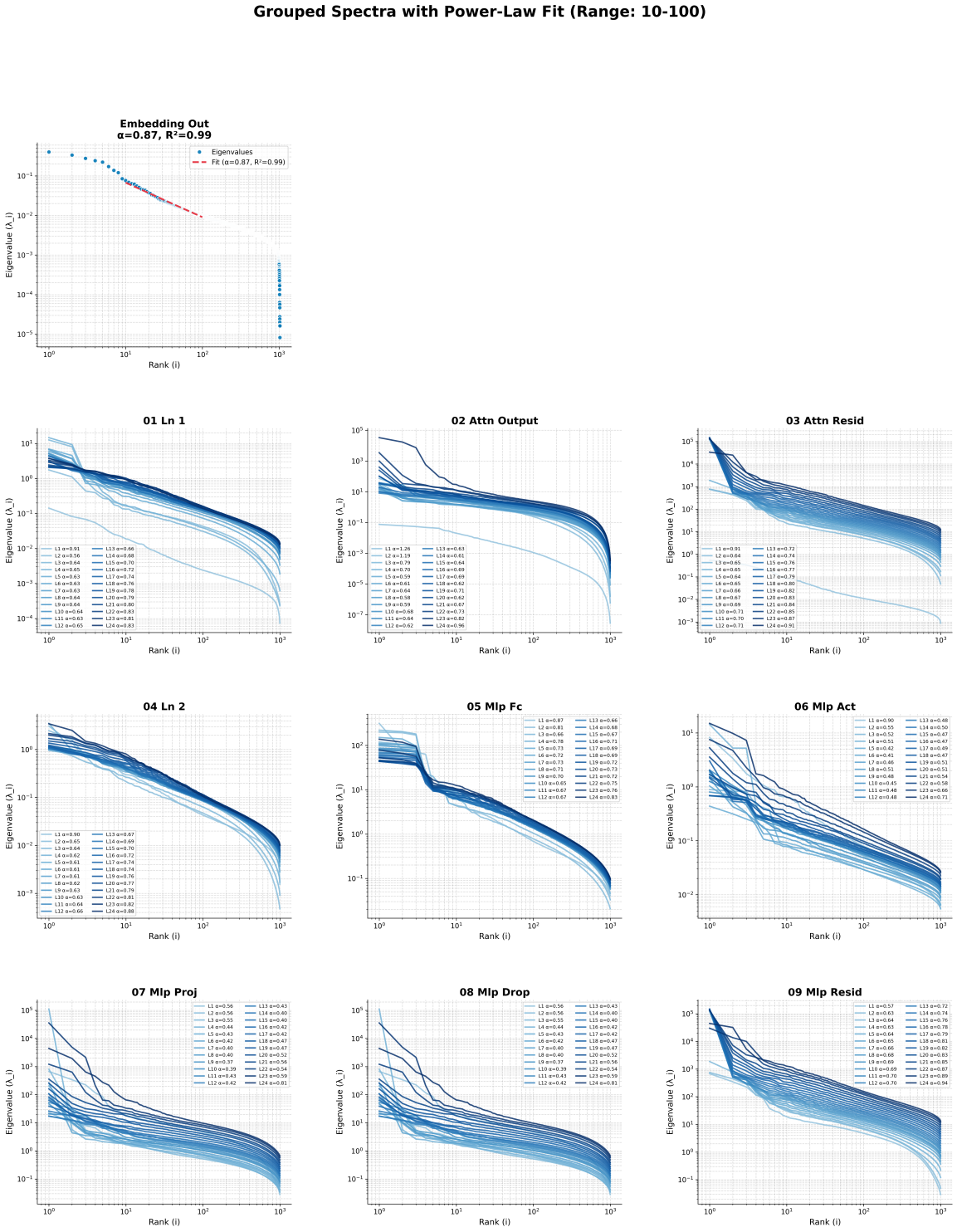


Figure 3: Full activation covariance spectra across all 24 layers and sublayers of GPT2 medium evaluated on subset of the training set of Wikitext-103 without fine-tuning. Each subplot shows eigenvalues λ_k of the activation covariance matrix for a specific sublayer (Norm, attention or MLP) plotted against rank index k in loglog scale. The heavy-tail exponents α are fitted with ranks 10-100.