TASK-LEVEL INSIGHTS FROM EIGENVALUES ACROSS SEQUENCE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Although softmax attention drives state-of-the-art performance for sequence models, its quadratic complexity limits scalability, motivating linear alternatives such as state space models (SSMs). While these alternatives improve efficiency, their fundamental differences in information processing remain poorly understood. In this work, we leverage the recently proposed dynamical systems framework to represent softmax, norm and linear attention as dynamical systems, enabling a structured comparison with SSMs by analyzing their respective eigenvalue spectra. Since eigenvalues capture essential aspects of dynamical system behavior, we conduct an extensive empirical analysis across diverse sequence models and benchmarks. We first show that eigenvalues influence essential aspects of memory and long-range dependency modeling, revealing spectral signatures that align with task requirements. Building on these insights, we then investigate how architectural modifications in sequence models impact both eigenvalue spectra and task performance. This correspondence further strengthens the position of eigenvalue analysis as a principled metric for interpreting, understanding, and ultimately improving the capabilities of sequence models.

1 INTRODUCTION

Deep sequence models have emerged as the backbone of modern artificial intelligence applications, demonstrating remarkable capabilities through their ability to learn complex patterns in large-scale datasets (Bommasani et al., 2021). The transformer (Vaswani et al., 2017), with softmax attention as its backbone, has been the dominant paradigm driving these advances. However, its quadratic complexity with respect to sequence length represents a critical bottleneck, limiting the practical deployment of these models for long-context applications (Tay et al., 2021). To address this limitation, numerous linear attention variants, including state space models (SSMs), have been proposed, which achieve linear complexity while attempting to preserve the expressive capabilities of softmax attention (Katharopoulos et al., 2020; Gu et al., 2022; Smith et al., 2023; Orvieto et al., 2023b; Dao & Gu, 2024; Beck et al., 2024; Schlag et al., 2021). Despite these advances, a fundamental question remains: *how do different model classes process and retain information?* The recently proposed dynamical systems framework (DSF) (Sieber et al., 2024) provides a unified perspective for analyzing masked attention and its linear alternatives as dynamical systems. This framework builds the foundation for this paper, enabling the investigation of the sequence model eigenvalues. This is motivated by the fact that eigenvalues are known to govern the stability, memory retention, and information flow within dynamical systems, yet their empirical behavior in sequence models remains unexplored. While the eigenvalues of the state transition matrix in SSMs are often explicitly constrained during model design, attention-based models do not impose spectral constraints. Understanding whether characteristic eigenvalue patterns still emerge can reveal (a) fundamental insights into how different sequence models balance stability, expressiveness, and long-range dependency modeling, and (b) whether the transition of eigenvalues from initialization to the fully trained model aligns with the demands of the task. To achieve this, we leverage the DSF to (i) conduct a comprehensive empirical analysis of eigenvalue spectra across different attention mechanisms and SSMs over a variety of tasks, and (ii) introduce design choices informed by our findings from (i). We then analyze their influence on performance, as well as the corresponding eigenvalue spectra. As a result of the conducted studies, we reveal a link between the eigenvalue distribution and good performance on tasks with specific memory requirements. Specifically, we observe a high concentration of eigenvalues close to one when long memory is important, and similar peaks close to zero when memory selectiveness is required by the task. These insights highlight the potential of the proposed metric as a tool for understanding sequence model behavior and guiding architectural decisions.

Notation: We denote with N the hidden state size, with d the model size, and with L the sequence length. We use subscripts, e.g., $\cdot_i$, to denote the time index (or input dependency). Specifically, v_i represents the value of vector v at time i, and $|v_i|_\infty$ its infinity norm. We use u_i to denote the i-th input and use bold notation to indicate sequences, i.e., $\mathbf{v}_i = [v_1, \ldots, v_i]$, the identity matrix of size $\mathbb{R}^{n\times n}$ is denoted as $\mathbb{I}_n$, and $\boldsymbol{\lambda}(A)$ denotes all n eigenvalues of a matrix $A \in \mathbb{R}^{n\times n}$.

2 Related Work

The challenge of modeling long-range dependencies has been a central theme in sequence modeling across recurrent, state space, and attention-based approaches. To better understand the strengths and limitations of each model class from the linear system point of view, researchers have so far employed two primary analytical tools: eigenvalue spectra and memory functions.

Eigenvalue-based analyses. Eigenvalue spectra provide a principled way to characterize stability, and the relationship between remembering and forgetting, i.e., memory retention, and information decay in dynamical systems. This perspective has a long history in recurrent neural networks (RNNs), where eigenvalue normalization was introduced to mitigate vanishing gradients while preserving controlled memory decay (Helfrich & Ye, 2019). Later analyses investigated how different eigenvalue spectra encode solutions to temporal tasks, revealing that seemingly diverse eigenvalue distributions can correspond to functionally equivalent memory behaviors (Jarne, 2022). Additionally, Naiman & Azencot (2023) show that the eigenvalues of an approximate Koopman operator matrix of single-layer RNNs reflect and encode task-specific structures. More recently, similar ideas have been extended to SSMs, where the eigenvalues of the transition matrix are shown to govern stability and memory. Early works on SSMs studied how careful initialization, motivated by the eigenvalue placement, can provide models with long-range dependency, enabling them to compete with RNNs and transformers on synthetic benchmarks (Gu et al., 2020; Fu et al., 2023). Beyond initialization, subsequent works focused on structural re-parameterizations. In particular, the eigenvalues of the transition matrix have been shown to play a crucial role in governing the stability and memory length of SSMs (Wang & Li, 2024; Grazzi et al., 2025). This perspective directly connects to earlier analyses of RNNs, where eigenvalue spectra determine the decay rates of information.

Memory function analyses. A complementary line of work analyzes long-term dependency through the lens of memory functions (Oppenheim et al., 1997), which have a strong relation to the eigenvalues of the system. These functions quantify how much influence past inputs have on current outputs, typically formalized via norms of system response functions or spectral measures. For SSMs, memory functions provide a principled way to characterize expressivity and effective memory horizons (Wang & Xue, 2023). For RNNs, memory functions are used to prevent rapid decay of state memory (Su & Kuo, 2019).

In attention-based architectures, however, analyses have largely focused on mechanisms for extending memory. Early work demonstrated that even simple feed-forward architectures equipped with attention can solve long-memory tasks, outperforming classical recurrent networks (Raffel & Ellis, 2016). Later, augmenting self-attention with persistent memory slots was proposed as a way to extend the memory context (Sukhbaatar et al., 2019). Other works analyze the statistical properties of the attention score matrices themselves (Bao et al., 2024; Bhojanapalli et al., 2021), revealing certain regularities in how attention distributes over inputs. While insightful, these approaches remain tied to the score-matrix formulation and do not naturally translate into a framework that allows systematic comparison with SSMs. By contrast, memory functions have been successfully applied to RNNs and SSMs, but they require architecture-specific design choices and do not generalize easily to linear parameter-varying (LPV) systems, such as attention-based models and Mamba (Dao & Gu, 2024). The recently proposed DSF addresses this gap by recasting masked attention itself as a dynamical system, thereby making eigenvalue analysis applicable. This unifying perspective enables systematic cross-architectural comparisons of memory behavior, which is leveraged in this work to provide the first comprehensive eigenvalue-based analysis of attention mechanisms, enabling a direct comparison with SSMs. Moreover, casting all these models into the eigenvalue perspective not only unifies their study, but also makes it possible to draw on the vast body of results from linear system theory to analyze and interpret their behavior in order to guide the model design choices.

3 Preliminaries

Before presenting our analysis, we detail how the DSF allows for eigenvalue computations of both attention and state space models, and then review the connection between eigenvalues and memory in dynamical systems.

3.1 The dynamical systems framework

The dynamical systems framework (Sieber et al., 2024) represents *causal* (i.e., masked) attention-based models or SSMs as dynamical systems. A discrete-time dynamical system models how an output $y_i \in \mathbb{R}^d$ is achieved given inputs $u_i \in \mathbb{R}^d$ through an internal hidden state $h_i \in \mathbb{R}^N$, which evolves over time according to a difference equation. In particular, the DSF formulates a sequence model as a discrete-time LPV dynamical system, i.e.,

$$h_i = \Lambda_i h_{i-1} + B_i u_i, \quad y_i = C_i h_i + D_i u_i, \tag{1}$$

where h_i is initialized with $h_{-1} = 0$, $\Lambda_i \in \mathbb{R}^{N \times N}$ is a diagonal or diagonal plus low rank state transition matrix, $B_i \in \mathbb{R}^{N \times d}$ and $C_i \in \mathbb{R}^{d \times N}$ are the input and output matrices, respectively, and $D_i \in \mathbb{R}^{d \times d}$ is a scaled skip connection. In this formulation, Λ_i, B_i, C_i can be input-dependent, e.g., $\Lambda_i = f(u_i)$.

All SSMs investigated in this paper are natively written in the DSF and we can directly analyze their eigenvalues $\boldsymbol{\lambda}(\Lambda_i)$. For attention, we follow the reformulation provided in (Sieber et al., 2024), which uses the convolutional representation of equation 1, i.e., $\mathbf{y} = \mathbf{\Phi}\mathbf{u}$, where the convolutional kernel $\mathbf{\Phi}$ is defined in Appendix A. Note that $\mathbf{\Phi}$ has the same interpretation as the attention matrix, i.e., softmax$(\mathbf{q}\mathbf{k}^\top)$ for softmax attention. Hence, defining $q_i = W_Q u_i \in \mathbb{R}^m, k_j = W_K u_j \in \mathbb{R}^m$, with $W_Q \in \mathbb{R}^{m \times d}$, $W_K \in \mathbb{R}^{m \times d}$, and $W_V \in \mathbb{R}^{d \times d}$, and rewriting the latter in the form

$$y_i = \sum_{j=0}^{i} \frac{\phi(q_i)^\top \psi(k_j)}{\eta(q_i, \mathbf{k}_i)} W_V u_j, \tag{2}$$

where $\phi(\cdot) : \mathbb{R}^m \to \mathbb{R}^n$, $\psi(\cdot) : \mathbb{R}^m \to \mathbb{R}^n$, and $\eta(\cdot, \cdot) : \mathbb{R}^m \times \mathbb{R}^{m \times (i+1)} \to \mathbb{R}$, it has been shown that the transition dynamics Λ_i are solely governed by a scalar defined through the normalization terms and can be computed exactly using:

$$\Lambda_i = \frac{\eta(q_{i-1}, \mathbf{k}_{i-1})}{\eta(q_i, \mathbf{k}_i)} \mathbb{I}_N \in \mathbb{R}^{N \times N}. \tag{3}$$

This normalization term then follows as $\eta(q_i, \mathbf{k}_i) = (\mathrm{elu}(q_i) + 1) \sum_{j=0}^{i} (\mathrm{elu}(k_j) + 1)$ for linear attention (Katharopoulos et al., 2020) and as $\eta(q_i, \mathbf{k}_i) = \sum_{j=0}^{i} \exp(q_i k_j^\top)$ for softmax attention. In both cases, the normalization depends explicitly on the queries and keys. In contrast, norm attention resolves this dependence: its normalization can be defined directly as a general nonlinear function of u, for which we adopt the softplus function in our formulation. Interested readers are referred to (Sieber et al., 2024) for a more detailed discussion of the framework and Appendix A for a more detailed derivation.

3.2 Relationship of Eigenvalues and Memory

In linear system theory (Kailath, 1980), the eigenvalues of a system's dynamics matrix are fundamental to understanding the behavior of the system. For a linear system of the form $h_i = \Lambda_i h_{i-1} + B_i u_i$, the eigenvalues encode crucial information about the system's memory characteristics. Eigenvalues near zero induce rapid forgetting, i.e., features decay quickly and have minimal influence on future states. Conversely, eigenvalues close to the unit circle of the complex plane[1]

[1]A discrete-time dynamical system is stable if its eigenvalues are within the unit circle, i.e., $|\boldsymbol{\lambda}(\Lambda_i)|_\infty \leq 1$. To avoid numerical issues during training, SSMs thus constrain the eigenvalues to the unit circle.

enable long-term memory retention, allowing information to persist across many time steps. The placement of eigenvalues thus directly determines whether a model favors short-term or long-term memory retention.

4 Empirical Study

To investigate model- and task-specific patterns in the respective eigenvalue distributions, we conduct an ablation study spanning multiple benchmarks. In the DSF, we analyze eigenvalue distributions across different models, tasks, layers, and heads, and investigate whether particular spectral signatures align with the type of memory and processing required, or the performance obtained.

4.1 Methodology

Task selection: We select five tasks designed to probe distinct model capabilities. Specifically, we use a subset of the Long Range Arena (LRA) benchmarks (Tay et al., 2021), consisting of (i) *Long ListOps*, which tests reasoning over deeply nested structures where every token in the input sequence is essential; (ii) *Byte-level text classification* (IMDb), which evaluates compositional generalization and the ability to process long natural-language sequences with sparse informative signals; and (iii) *Image classification from pixel sequences* (CIFAR-10), which emphasizes learning sparse local and global spatial relationships. In addition, we include (iv) the *MQAR* task (Arora et al., 2023), which requires associative recall, stressing a model's ability to retain and retrieve specific elements of the input sequence with high fidelity, as well as (v) the next token prediction task on *WikiText-103* (Merity et al., 2016) dataset, which underscores the models capabilities relevant for natural language processing. Together, these tasks capture a spectrum of challenges: long-context reasoning, compositionality, spatial structure learning, and selective memory.

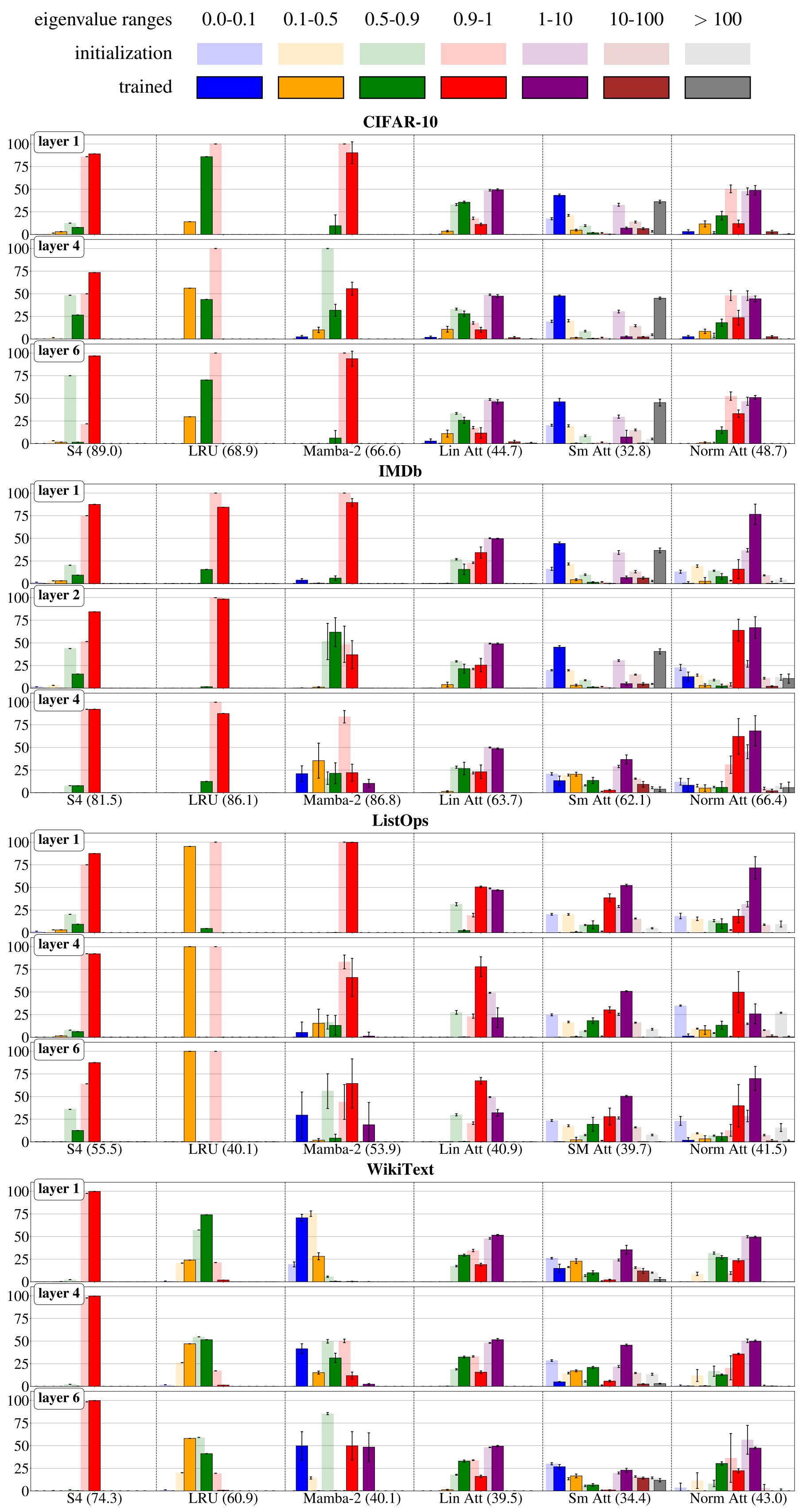
Model selection: We evaluate six representative architectures: two linear time-invariant (LTI) SSMs (S4 (Gu et al., 2021) and LRU (Orvieto et al., 2023a)), one LPV SSM (Mamba-2 (Dao & Gu, 2024)), and three causal attention mechanisms (softmax attention (Vaswani et al., 2017), norm attention (Sieber et al., 2024) and linear attention (Katharopoulos et al., 2020)). To ensure a fair comparison, for each task we fix the overall architecture and tune the remaining hyperparameters to achieve competitive performance relative to the best known baselines. The complete training details are provided in Appendix B.

4.2 Results and Discussion

Task performance and the corresponding eigenvalue distributions for all models are shown in Figure 1 (for complex eigenvalues, magnitudes are reported, however, a visualization of the complex plane is provided in Appendix C). Since the eigenvalues of Mamba-2 and attention-based models depend on the input , i.e., they emerge from an LPV system, we compute them over a batch of test data and report the batch-averaged distributions (for task-specific batch sizes, see Appendix B). For multi-headed models, we present results from a single head; remaining heads and additional random seeds, as well as results averaged over heads or seeds, are reported in Appendix C. Based on our observations, we make the following statement:

> Eigenvalues capture essential aspects of stability, memory, and long-range dependency modeling, not only for LTI SSMs, but also LPV models and attention mechanisms. In other words, downstream task requirements are reflected in spectral signatures of eigenvalues.

This claim is supported by the conducted ablation study, which shows that task-dependent eigenvalue distributions reveal distinct memory retention and gating behaviors. When long-term memory is critical, we consistently observe a strong concentration of eigenvalues near one, whereas tasks that require selective forgetting exhibit peaks near zero, which can also be observed in the correlation plots in Appendix C, i.e., Figures 30 and 31. This aligns with results derived for LTI SSMs using memory functions (Wang & Xue, 2023) and our assumption rooted in linear system theory, that eigenvalues close to zero induce selective forgetting, and eigenvalues close to one are responsible for long-term memory. On LRA tasks, which require long-term memory, all well-performing models


eigenvalue ranges
0.0-0.1
0.1-0.5
0.5-0.9
0.9-1
1-10
10-100
> 100
initialization
trained
CIFAR-10
layer 1
layer 4
layer 6
S4 (89.0)
LRU (68.9)
Mamba-2 (66.6)
Lin Att (44.7)
Sm Att (32.8)
Norm Att (48.7)
IMDb
layer 1
layer 2
layer 4
S4 (81.5)
LRU (86.1)
Mamba-2 (86.8)
Lin Att (63.7)
Sm Att (62.1)
Norm Att (66.4)
ListOps
layer 1
layer 4
layer 6
S4 (55.5)
LRU (40.1)
Mamba-2 (53.9)
Lin Att (40.9)
SM Att (39.7)
Norm Att (41.5)
WikiText
layer 1
layer 4
layer 6
S4 (74.3)
LRU (60.9)
Mamba-2 (40.1)
Lin Att (39.5)
Sm Att (34.4)
Norm Att (43.0)
0
25
50
75
100

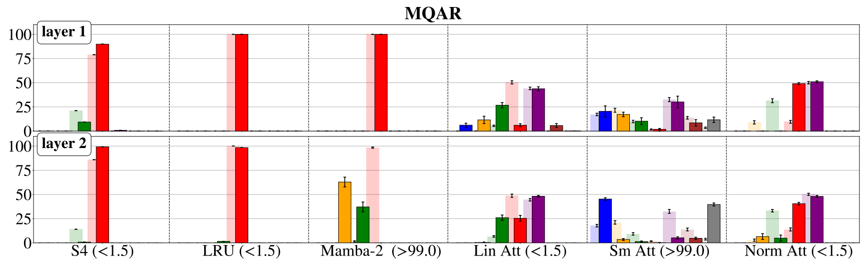


Figure 1: Eigenvalue distributions for one head, across models, selected layers, and tasks. Bars show the percentage of eigenvalues within discretized ranges (chosen to emphasize eigenvalues near zero and near one). Light and dark bars indicate the distribution at initialization and after training, respectively. Error bars denote standard deviation across input sequences. Model performance, measured as perplexity for WikiText (lower is better) and percentage of correct output sequences for the other tasks (higher is better), is indicated in parentheses. Complete plots for all layers, heads and multiple seeds are provided in Appendix C.

not only avoid placing eigenvalues close to zero, but show prominent peaks around one, at least in the first layer and often across all layers. By contrast, attention models distribute eigenvalues both near zero and well above one, which can be interpreted as gating, with low eigenvalues inducing selective forgetting, possibly contributing to poor performance on all LRA tasks. The existence of eigenvalues larger than one could potentially be problematic, as it causes unstable discrete-time dynamics of LPV systems. This might further contribute to the shortcomings of attention on LRA benchmarks. Softmax attention is especially affected: across all spectra, it shows a nearly balanced mix of very low and very high eigenvalues, which can be detrimental, particularly on CIFAR-10, and to a lesser degree on IMDb, where the gating effect is less pronounced. This is in line with our findings, as due to the textual analysis nature of the IMDb task, not all tokens are crucial, implying selective forgetting can be favorable. An extreme example of an LRA task is ListOps, where every token is crucial. Here, we find that all models, including attention, avoid placing eigenvalues close to zero. It is interesting that this behavior extends to attention models, despite the fact that their eigenvalues are not explicitly designed but instead emerge from learned weight matrices and can hence take any value. The fact that attention still avoids near-zero eigenvalues in this setting reinforces our hypothesis linking small eigenvalues to selective forgetting: when forgetting is harmful, the model tries not to, albeit imperfectly. In contrast, Mamba-2, as an LPV SSM, occupies a middle ground: its eigenvalue distribution avoids excessive gating, while still allowing selective placement near zero, enabling competitive performance not only on LRA benchmarks but also on MQAR and WikiText. There, selective forgetting proves beneficial, since only highly specific information needs to be retained. LTI SSMs, by comparison, rarely place any eigenvalues close to zero, and consistently underperform attention on both of these tasks.

In LTI SSMs, eigenvalue placement is known to influence training stability (Orvieto et al., 2023a; Gu et al., 2020), yet our analysis suggests that the initial eigenvalue configuration is not preserved during training and therefore cannot fully explain the final eigenvalue distributions. For instance, S4 is initialized with eigenvalues having the magnitude primarily in the range $(0.5, 1)$, but this distribution consistently shifts towards a sharp concentration around 1 across tasks and layers. LRU, though also initialized close to 1, often drifts away from the initialization, most notably on ListOps and CIFAR-10. A similar departure from initialization is observed in both Mamba-2 and softmax attention, where the resulting eigenvalue spectra differ substantially from the starting distributions. In contrast, linear attention largely retains the overall shape of its initialization, raising the question of the potential importance of a task-dependent initialization. For further illustration of the eigenvalue evolution during training, we refer to Figures 32 and 33 in Appendix C.

5 Architectural Modifications and Their Spectral Impact

In this section, we study how architectural modifications shape the eigenvalue spectrum and how these changes correlate with model performance. In particular, we consider five architectural variations: gating, convolution, varying the number of layers, an LTI-inspired version of Mamba-2, and alternative normalization functions in norm attention. In the remainder of this section, we provide details for each modification and the conclusions we draw after investigating its eigenvalue spectra, which can be summarized in the following:

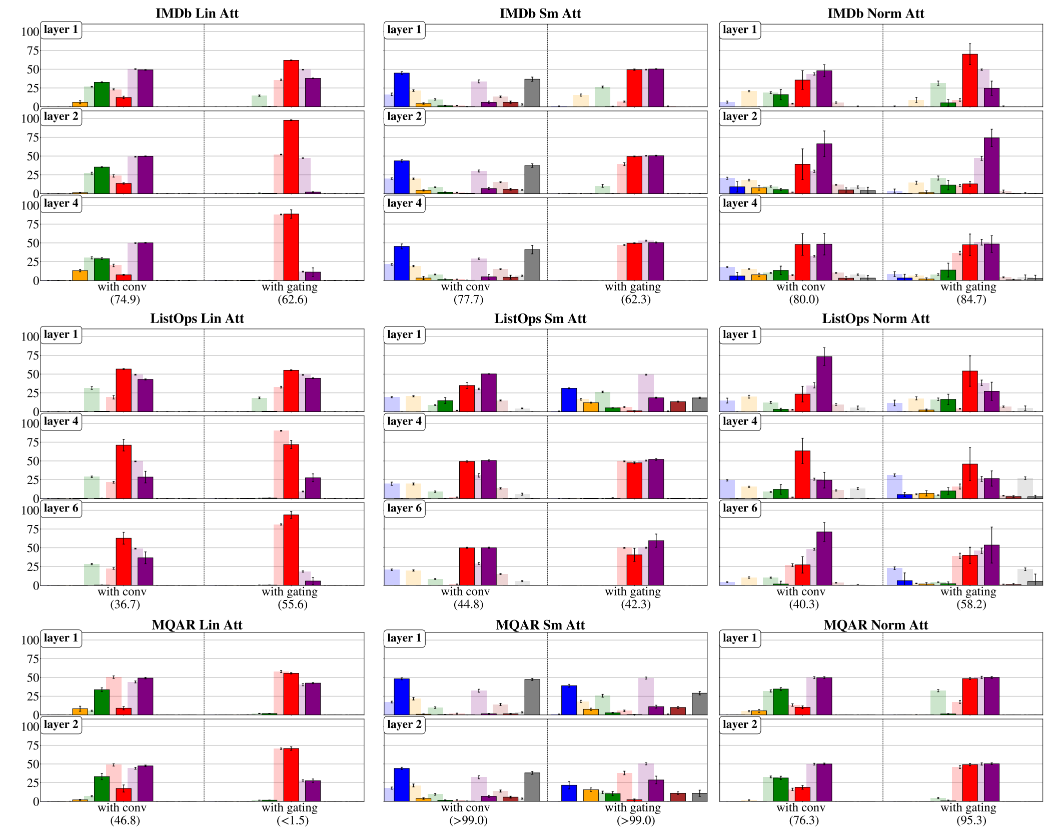


Figure 2: Comparison of the effects of gating and convolution on the eigenvalue spectra for one head, across selected tasks, models, and layers. Complete plots for all layers and tasks are provided in Appendix C.

Architectural changes are reflected not only in model performance but also in the eigenvalue spectra of trained models. Such modifications alleviate specific challenges in the underlying dynamical system, freeing capacity for other aspects of sequence modeling, i.e., selective forgetting when convolution is introduced, and memory retention when gating is added.

5.1 GATING

As discussed in Section 4, we observe a correlation between the gating behavior of a model and the selective-memory demands of the task. For example, in ListOps, where all input information is essential, well-performing models tend not to perform gating, whereas on other tasks, they exhibit diverse gating strategies with varying success. Motivated by this, we investigate how attention models behave when equipped with an explicit gating mechanism. Following Yang et al. (2024), we augment each layer with a gating mechanism: the input to each layer is linearly projected, passed through a SiLU activation, and used to multiplicatively gate the layer's output. This introduces a learnable, input-dependent modulation that can adaptively control the contribution of each layer to the forward pass. For comparison we implemented a second version of gating, in accordance with gating used in Mamba-2, which is placed before the output projection of each attention layer. The results of this implementation, yielding lower performance, are provided in Appendix C.

Figure 2 illustrates that introducing an explicit gating mechanism alleviates the need for the dynamical system to implement gating implicitly, while still allowing it when beneficial. This effect is particularly pronounced on IMDb, where the eigenvalue distribution shifts away from zero, indicating that the dynamical system is primarily leveraged for memory preservation and state space expansion once gating is explicitly handled. Viewed through the lens of our novel metric, this shift reflects a redistribution of eigenvalues away from the selectivity regime (near zero) toward the memory regime (near one). An exception arises on ListOps, which does not exhibit low eigenvalues prior to gating but develops them once gating is introduced (see e.g., softmax attention layer 1). This behavior may stem from the task's unique requirement to preserve all input information without selective suppression, causing explicit gating to interact with the dynamics in an atypical way. Another possibility is that the eigenvalues of the gated model have little influence on performance and can therefore drift freely. Additionally, as the performance of all models on this task is relatively low, especially attention-based ones, favourable eigenvalue spectra for solving this task may not have emerged during training. Therefore, the precise mechanism behind this effect remains unclear, and we leave a deeper investigation of this phenomenon to future work. Additionally, as reported in Section 4, models that perform extensive gating exhibit decreased performance on CIFAR-10, an effect that is also observed when softmax and linear attention models are augmented with gating in Appendix C.

5.2 Convolution

Motivated by the strong performance of Mamba-2 on LRA, MQAR, and WikiText, we adapt attention models to more closely mirror its architecture. Specifically, we prepend each layer with a 1D convolution along the sequence dimension, applied to all hidden channels with kernel size d_{conv} and appropriate padding to preserve length. This lightweight operation introduces local interactions among neighboring tokens prior to the main transformation, thereby enriching the input with short-range contextual information.

Introduction of convolution causes a shift in the eigenvalue spectrum, with eigenvalues occurring more frequently near zero and less frequently near one (see e.g., softmax attention on IMDb in Figure 2). Since, according to linear system theory, eigenvalues close to one correspond to slowly decaying modes that preserve information, while those near zero correspond to rapidly vanishing modes, this change in spectral distribution indicates that convolution alleviates the task of a memory preserving layer in the recurrent dynamics by providing local context directly. Consequently, the dynamical system allocates a larger portion of its capacity to selective processing. This change in the eigenvalues aligns with consistent performance gains on tasks requiring selective memory, underscoring a link between introducing convolution, spectral structure, and task-level outcomes. The only exceptions arise in linear attention and the ListOps task, where exhaustive memory preservation remains critical and the benefits of selectivity diminish.

5.3 Varying the Number of Layers

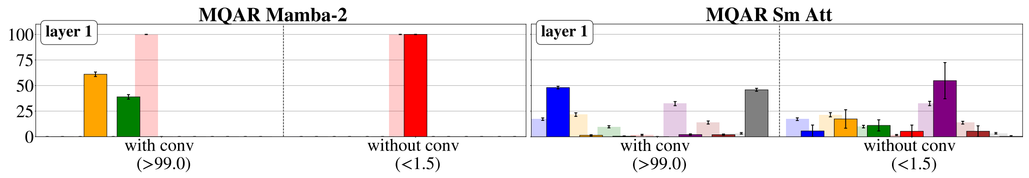


Figure 3: Eigenvalue distribution comparison for single-layer Mamba-2 and softmax attention models with and without convolution on MQAR. Results for one out of four heads are shown.

Given the observation suggesting that a layer-dependent task division for softmax attention is present in Figure 1), we investigate the effect of varying the number of layers on MQAR task. Specifically, when using two layers, the second layer exhibits clear gating-like behavior, following a first layer characterized by stronger memory retention. Prior work (Okpekpe & Orvieto, 2025) hypothesizes that, in single-layer models, attention undergoes a phase transition where it attempts to form induction heads, but the model lacks sufficient expressivity to fully exploit this mechanism, requiring greater depth to do so (Sanford et al., 2024). Adding convolution in that study alleviates this limitation and enables task completion.

Building on this insight and our observation that adding convolution takes over the task of a memory-retaining layer, we aim to explain these phenomena by analyzing the corresponding eigenvalue spectra shown in Figure 3. These provide a quantitative view of memory retention and gating dynamics in respective layers. As expected, with convolution offloading the memory-retaining layer, both single-layer softmax attention and Mamba-2 successfully learn the task, despite not being able to do it with one layer otherwise. In this setting, the sole role assumed by the dynamical system is gating, highlighting how architectural modifications can reallocate functional responsibilities across layers and how this is directly reflected in the eigenvalue distributions.

5.4 Normalization of Norm Attention

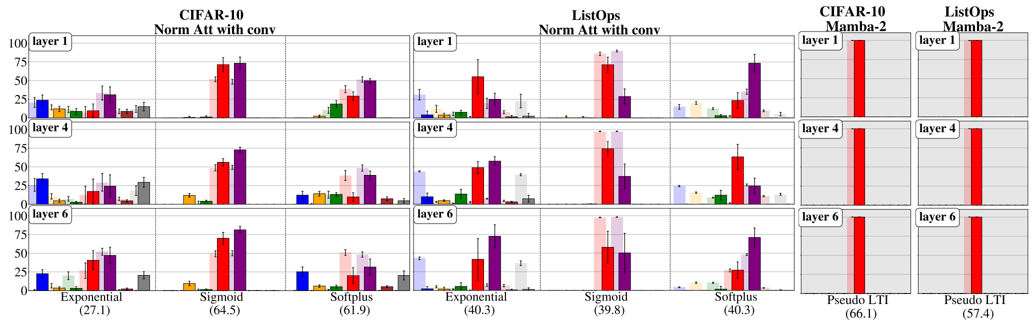


Figure 4: Eigenvalue distributions on CIFAR-10 and ListOps for one of the heads for: (left, with white background) norm attention with convolution and different normalization functions; (right, with grey background) Mamba-2 as LTI. Complete plots for all layers and tasks are provided in Appendix C.

In the norm attention model, we systematically replace the normalization function to test its influence on the distribution of the eigenvalues, and whether the incurred changes affect the model performance, according to the findings in Section 4. Specifically, for norm attention with explicit convolution, we substitute the original `softplus` normalization with two alternatives: `exponential` and `sigmoid`, as each choice affects the range of the normalized values differently. This directly affects the placement of eigenvalues, both at initialization and during training.

As illustrated by Figure 4, different normalization functions lead to different eigenvalue distributions, reflecting trade-offs between memory and selectivity. Exponential normalization places eigenvalues that promote selectivity, sigmoid keeps them close to one, emphasizing memory retention, and softplus exhibits an intermediate pattern combining both regimes. While these distributions are generally maintained during training, certain tasks, e.g., ListOps, show that exponential normalization can converge to a configuration that implies memory retention. Furthermore, the gating behavior induced by exponential normalization on CIFAR-10 results in a substantial performance drop, consistent with our observation in Section 4 that this task is adversely affected by gating.

5.5 Mamba-2 as a Pseudo-LTI

Motivated by the strong performance of SSMs with LTI structure on LRA tasks, and by the high variance observed in input-dependent eigenvalues (particularly pronounced in Mamba-2 on ListOps), we construct a pseudo-LTI variant of Mamba-2. Concretely, we modify the discretization step so that the state matrix Λ is computed with respect to a fixed sampling interval, rather than an input-dependent one. This ensures that Λ remains constant across inputs, while allowing input-dependencies to be absorbed into the input matrix B.

According to Figure 4, this modification of Mamba-2 yields a behavior analogous to S4, both in performance on ListOps and eigenvalue distribution, with all eigenvalues remaining close to one, while the time-varying dynamics are assumed to be captured through the input matrix B. More fine-grained plots are provided in Appendix C, revealing that eigenvalues still shift from their initial distribution, though only within a narrow range.

5.6 Summary

In summary, we show that specific architectural changes lead to the expected shift in eigenvalue spectra and, as a consequence, in some cases also improve the performance of the considered models. This especially applies to several specific architectural changes, such as MQAR with convolution (allowing for a single-layer architecture), MAMBA-2 as a pseudo-LTI, adding gating (especially to norm attention, which performs best of all considered models on the ListOps task), and an adjusted normalization function for norm attention, which leads to a significant improvement in performance on CIFAR-10.

6 Conclusion

Across diverse sequence modeling benchmarks, we leveraged a unified dynamical systems representation of SSMs and attention to conduct an extensive empirical analysis of eigenvalue distributions. This study revealed characteristic eigenvalue distributions distinguishing attention models from their linear counterparts and demonstrated the efficacy of eigenvalues in capturing essential aspects of selectivity and memory retention. We also identified correlations between spectral properties and task requirements or model selection. Furthermore, we highlighted how architectural modifications, such as gating, convolution, and layer variations, manifest in the eigenvalue spectrum. Taken together, our results show that eigenvalue analysis may serve as a systematic method for understanding the capabilities of sequence models on a given task, opening up new possibilities for initialization schemes and architectural design choices that encourage spectral properties well-suited to particular tasks.

Limitations: While analyzing the eigenvalues of the state space interpretation reflects effects of several architectural changes, other components of these models also shape their behavior, and a complete understanding requires further investigation. Additionally, we analyze eigenvalue magnitudes within predefined bins, but finer-grained analyses may provide supplementary insights. Extending these analyses to capture the full complexity of model dynamics, as well as other choices, such as positional embeddings, represents an important direction for future research.

Reproducibility Statement

We have made efforts to ensure the reproducibility of our results. The eigenvalue computations follow the procedure described in the DSF paper (Sieber et al., 2024), with full derivation and computation details referenced therein. In Appendix B, we provide training configurations, hyperparameter choices, and experimental settings, together with additional clarifications of our methodology. Furthermore, all datasets used are standard benchmarks, and are also described in Appendix B.

Acknowledgement of AI-assisted Tools

This paper has benefited from AI-assisted editing tools used for grammar and style polishing.

References

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and Improving Recall in Efficient Language Models. *arXiv:2312.04927*, 2023.

Han Bao, Ryuichiro Hataya, and Ryo Karakida. Self-attention networks localize when qk-eigenspectrum concentrates. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 2903–2922, 2024.

Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xLSTM: Extended Long Short-Term Memory. *arXiv preprint arXiv:2405.04517*, 2024.

Srinadh Bhojanapalli, Ayan Chakrabarti, Himanshu Jain, Sanjiv Kumar, Michal Lukasik, and Andreas Veit. Eigen analysis of self-attention and its reconstruction from partial computation. *arXiv preprint arXiv:2106.08823*, 2021.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized Models and Efficient Algorithms with Structured State Space Duality. In *ICML 2024*, 2024.

Daniel Y. Fu, Tri Dao, Khaled K. Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. Hungry Hungry Hippos: Towards Language Modeling with State Space Models, 2023. URL `https://arxiv.org/abs/2212.14052`.

Riccardo Grazzi, Julien Siems, Arber Zela, Jörg K. H. Franke, Frank Hutter, and Massimiliano Pontil. Unlocking state-tracking in linear rnns through negative eigenvalues, 2025. URL `https://arxiv.org/abs/2411.12537`.

Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. HiPPO: Recurrent Memory with Optimal Polynomial Projections. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1474–1487. Curran Associates, Inc., 2020.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently Modeling Long Sequences with Structured State Spaces. In *The International Conference on Learning Representations (ICLR)*, 2022.

Kyle Helfrich and Qiang Ye. Eigenvalue normalized recurrent neural networks for short term memory, 2019. URL `https://arxiv.org/abs/1911.07964`.

Cecilia Jarne. Different eigenvalue distributions encode the same temporal tasks in recurrent neural networks. *Cognitive Neurodynamics*, 17(1):257–275, April 2022. ISSN 1871-4099. doi: 10.1007/s11571-022-09802-5. URL `http://dx.doi.org/10.1007/s11571-022-09802-5`.

Thomas Kailath. *Linear systems*, volume 156. Prentice-Hall Englewood Cliffs, NJ, 1980.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

Ilan Naiman and Omri Azencot. An operator theoretic approach for analyzing sequence neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 9268–9276, 2023.

Destiny Okpekpe and Antonio Orvieto. When recalling in-context, transformers are not ssms. *arXiv preprint arXiv:2508.19029*, 2025.

Alan V Oppenheim, Alan S Willsky, and Syed Hamid Nawab. *Signals & systems*. Pearson Educación, 1997.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pp. 26670–26698. PMLR, 2023a.

Antonio Orvieto, Samuel L Smith, Albert Gu, Anushan Fernando, Caglar Gulcehre, Razvan Pascanu, and Soham De. Resurrecting Recurrent Neural Networks for Long Sequences. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 26670–26698. PMLR, 23–29 Jul 2023b.

Colin Raffel and Daniel P. W. Ellis. Feed-forward networks with attention can solve some long-term memory problems, 2016. URL https://arxiv.org/abs/1512.08756.

Clayton Sanford, Daniel Hsu, and Matus Telgarsky. One-layer transformers fail to solve the induction heads task. *arXiv preprint arXiv:2408.14332*, 2024.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning*, pp. 9355–9366. PMLR, 2021.

Jerome Sieber, Carmen Amo Alonso, Alexandre Didier, Melanie Zeilinger, and Antonio Orvieto. Understanding the differences in foundation models: Attention, state space models, and recurrent neural networks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=iF7MnXnxRw.

Jimmy T.H. Smith, Andrew Warrington, and Scott Linderman. Simplified State Space Layers for Sequence Modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Yuanhang Su and C.-C. Jay Kuo. On extended long short-term memory and dependent bidirectional recurrent neural network. *Neurocomputing*, 356:151–161, September 2019. ISSN 0925-2312. doi: 10.1016/j.neucom.2019.04.044. URL http://dx.doi.org/10.1016/j.neucom.2019.04.044.

Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Herve Jegou, and Armand Joulin. Augmenting self-attention with persistent memory, 2019. URL https://arxiv.org/abs/1907.01470.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long Range Arena : A Benchmark for Efficient Transformers. In *International Conference on Learning Representations (ICLR)*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Shida Wang and Qianxiao Li. Stablessm: Alleviating the curse of memory in state-space models through stable reparameterization, 2024. URL https://arxiv.org/abs/2311.14495.

Shida Wang and Beichen Xue. State-space models with layer-wise nonlinearity are universal approximators with exponential decaying memory. *Advances in Neural Information Processing Systems*, 36:74021–74038, 2023.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. *arXiv preprint arXiv:2412.06464*, 2024.

APPENDIX

A ADDITIONAL DETAILS ON THE DSF

The LPV dynamics equation 1 can also be written in a convolutional representation, i.e., $\mathbf{y} = \mathbf{\Phi u}$, where the kernel $\mathbf{\Phi}$ is defined as

$$\mathbf{\Phi} = \begin{bmatrix} C_0 B_0 + D_0 & & & \\ C_1 \Lambda_1 B_0 & C_1 B_1 + D_1 & & \\ \vdots & \ddots & \ddots & \\ C_L \prod_{k=1}^{L} \Lambda_k B_0 & \dots & C_L \Lambda_L B_{L-1} & C_L B_L + D_L \end{bmatrix}. \tag{4}$$

Note that the kernel $\mathbf{\Phi}$ has the same dimensions as the attention matrix, i.e., softmax$(\mathbf{qk}^\top)$ for softmax attention, and that the two matrices are equivalent, up to a scaling factor W_V. Using Lemma 1 in Sieber et al. (2024), all considered masked attention can be written in the form given by equation 1 as

$$\Lambda_i = \frac{\eta(q_{i-1}, \mathbf{k}_{i-1})}{\eta(q_i, \mathbf{k}_i)} \mathbb{I}_N \in \mathbb{R}^{N \times N}, \tag{5a}$$

$$B_i = \left(\frac{1}{\eta(q_{i-1}, \mathbf{k}_{i-1})} \mathbb{I}_d \otimes \psi(k_j) \right) W_V \in \mathbb{R}^{N \times d}, \tag{5b}$$

$$C_i = \mathbb{I}_d \otimes \phi(q_i)^\top \in \mathbb{R}^{d \times N}, \tag{5c}$$

where $\otimes$ denotes the Kronecker product and the transition dynamics Λ_i are solely governed by a scalar defined through normalization terms in equation 5a. The detailed derivation of softmax attention as an LPV system are provided in (Sieber et al., 2024, Appendix B).

B EXPERIMENTAL DETAILS

This section contains details about experimental results provided in Section 4. The experiments were performed on the long range arena (LRA) benchmark (Tay et al., 2021), the multi-query associative recall (MQAR) benchmark (Arora et al., 2023) and the WikiText benchmark (Merity et al., 2016). To obtain these results, we combined the LRA[2] and Zoology[3] code bases.

Model Details We evaluate the following three architecture classes on all tasks:

1. **Attention:** softmax attention (Vaswani et al., 2017), linear attention (Katharopoulos et al., 2020) and norm-attention (Sieber et al., 2024). In all three settings, we employ a standard GPT-2–style multi-headed Transformer, where the attention block is replaced by the respective attention mechanism. Each attention block is followed by a task-dependent multi-layer perceptron. For fairness, all three variants share the same overall architecture and hyperparameters, differing only in the attention function, learning rate, dropout, and batch size. Furthermore, softmax and linear attention use learnable positional token embeddings, while norm attention does not.
2. **Linear parameter-varying state space models:** Mamba-2 (Dao & Gu, 2024). We use the same GPT-2 style multi-headed Transformer backbone with learnable token embeddings but no positional encoding, substituting the attention block with the SSD block, where we additionally apply a 1D short-convolution, with a filter of dimension 4 across sequence length to the input of each layer. The implementation follows the official codebase.[4]
3. **Linear time-invariant state space models:** S4 (Gu et al., 2021) and LRU (Orvieto et al., 2023a). We use one-hot encoding of the input, followed by a stacked encoder model architecture, which consists of a linear encoder followed by SSM blocks corresponding to the

[2]`https://github.com/google-research/long-range-arena`
[3]`https://github.com/HazyResearch/zoology`
[4]`https://github.com/state-spaces/mamba`

specific model. We adapt the S4[5] and unofficial LRU[6] code bases and integrate them into our framework. To ensure a fair comparison, we keep the architecture and all hyperparameters of both models constant, except for the learning rate, dropout and batch size.

All models are trained and evaluated over five different tasks (3 LRA tasks, MQAR and WikiText-103), while their architectures were fixed for each task (with the architecture details provided in Table 1), with the aim to keep the models comparable in size. The number of parameters per model is provided in Table 2. In the remainder of the section, we provide task-wise details of experiments.

B.1 LRA EXPERIMENTS

Training details As outlined in Section 4.1, we restrict our experiments to a subset of LRA tasks: CIFAR-10, IMDb, and ListOps. For all LRA experiments, we follow the training protocol below:

- **Optimizer and schedule:** Linear warmup with cosine annealing and AdamW optimizer (Loshchilov & Hutter, 2019), trained for a constant number of epochs per task, as indicated in Table 1.
- **Position information:** Positional embeddings (Brown et al., 2020) are used only for linear and softmax attention.
- **Data:** Each model is trained on the subset of standard datasets from the LRA benchmark, summarized below; full details can be found in (Tay et al., 2021).
    1. Long List Operations (`ListOps`): This benchmark tests a model's ability to handle hierarchical dependencies across long input sequences. The task involves predicting the outcome of mathematical expressions built from nested *mean*, *median*, *max*, and *min* operations.[7] It is framed as a ten-class classification problem, with input lengths of up to 2k tokens.
    2. Byte-Level Text Classification (`IMDb`): This task measures a model's ability to identify sentiment in long tokenized texts. The dataset is composed of IMDb movie reviews, each labeled as either positive or negative. It is formulated as a binary classification problem with input sequences of up to 4k tokens.
    3. Image Classification from Pixel Sequences (`CIFAR-10`): This benchmark assesses a model's capacity to infer 2D spatial structure from linearized pixel sequences. The dataset contains vectorized images belonging to one of ten categories (e.g., horse, car). The task is framed as a ten-class classification problem with input sequences of up to 1k tokens.

Performed Experiments We run all six models with task-wise parameters given in Table 1 on the described subset of LRA tasks for four different random seeds. For each of the runs, we provide the eigenvalue distributions in the corresponding figures of the main text and Appendix C. Note that we do not optimize the hyperparameters and we use the comparable model sizes to make comparison sensible, and therefore the reported accuracies might be lower than in the original LRA paper (Tay et al., 2021).

B.2 MQAR EXPERIMENTS

Training Details For all MQAR runs, we use the following training protocol:

- **Optimizer and schedule:** Linear warmup with duration of 10% and cosine annealing, with AdamW optimizer (Loshchilov & Hutter, 2019). For each run, we sweep the learning rates in `logspace`$(-4, -2, 4)$ and train for 40 000 steps with a global batch size of 64. This is the same setup as in (Arora et al., 2023).

[5]`https://github.com/state-spaces/s4`
[6]`https://github.com/NicolasZucchet/minimal-LRU`
[7]For instance, `input: max(4, 3, min(2, 3), 1, 0, median(1, 5, 8, 9, 2), output: 5.`

- **Position information:** Positional embeddings (Brown et al., 2020) are used for linear and softmax attention models only, and not for the SSM architecture classes. This is the same setup as in (Arora et al., 2023).
- **Data:** Each model is trained on an MQAR dataset with 100,000 datapoints and evaluated on 3,000 datapoints. This is the same setup as in (Arora et al., 2023). The data and its order are constant for all runs, with the number of key-value pairs equal to 64 and an input sequence length of 512, corresponding to the most challenging task from (Arora et al., 2023).

Performed Experiments We run all six models on the above-specified MQAR task, for 4 different seeds. For each model, we apply the architecture settings in Table 1 and sweep over the learning rates. We only report the results of the best-performing learning rate in the corresponding figures of the main text and Appendix C.

B.3 WikiText experiments

Training Details For all WikiText experiments, we use the following training protocol:

- **Optimizer and schedule:** Linear warmup 3000 steps and cosine annealing, with AdamW optimizer (Loshchilov & Hutter, 2019). We train for 130 000 steps with a global batch size of 8.
- **Position information:** Positional embeddings (Brown et al., 2020) are used for softmax and linear attention models only.
- **Data:** We use Wikitext-103, which contains 103 million tokens extracted from the set of verified articles on Wikipedia. Aside from featuring a large vocabulary, it also retains the original case, punctuation and numbers. Given the fact that it is composed of full articles, this dataset is well-suited for evaluating a model's ability to take advantage of long term dependencies.

Performed Experiments We run all models on the above-specified WikiText task for four different seeds. For each model, we apply the architecture settings in Table 1 and report the results in the corresponding figures of the main text and Appendix C.

Table 1: Common architectures for different tasks. Eval BSZ corresponds to the size of the batch used for the computation of eigenvalues of attention models and Mamba-2.

	Depth	Model dim d	State dim N	Heads	Mixer dim	Iterations	Eval BSZ
CIFAR-10	6	512	64	4	128	50 epochs	64
IMDb	4	128	64	4	512	30 epochs	32
ListOps	6	128	64	4	256	50 epochs	32
MQAR	2	128	128	1	-	40k steps	64
WikiText	6	512	512	8	512	130k steps	8

Table 2: Total number of parameters (in millions) excluding the encoder, across different tasks.

	S4	LRU	Mamba-2	Lin Att	Sm Att	Norm Att
CIFAR-10	4.74	3.95	6.73	4.41	4.41	4.42
IMDb	0.40	0.27	0.34	0.79	0.79	0.79
ListOps	0.60	0.40	0.51	0.73	0.73	0.73
MQAR	1.39	1.26	1.26	1.18	1.18	1.18
WikiText	41.53	35.24	35.30	35.19	35.19	35.22

C Additional Results

With results in the main text being limited to a single head, we leverage this section to extend our empirical analysis to further head dimensions of all investigated multi-head models (Mamba-2 and attention models) and benchmark tasks. Note that since MQAR has been trained with a single head, no further results are displayed in the following. To further strengthen our empirical investigation, we subsequently display results for three additional random seeds, all models, and all benchmark tasks. The overview of all presented results is given below:

- Figure 6: contains all missing layers from Figure 1;
- Figures 7- 9: show remaining heads for the same models from Figure 1 on LRA tasks;
- Figures 10- 11: show remaining heads for the same models from Figure 1 on WikiText benchmark;
- Figures 12-19: illustrate eigenvalue spectra for one of the heads over three additional seeds for all models and tasks;
- Figure 20: comparison of attention models with explicit convolution and gating for one of the heads on LRA tasks (extension of Figure 2);
- Figure 21: comparison of attention models with explicit convolution and gating for one of the heads on WikiText benchmark (extension of Figure 2)
- Table 3 and Figure 22: summarized performance of nominal models from Figure 1 and the absolute change in performance after introducing the architectural changes (introduction of explicit gating and convolution, as well as the change of normalization function for norm attention);
- Figure 23: comparison of different normalization functions for norm attention models on all tasks (extension of Figure 4);
- Figure 24: contains eigenvalue distributions for all layers of one of the heads for Mamba-2 pseudo-LTI variation on all LRA tasks (extension of Figure 4), as well as fine-grained eigenvalue distribution plots showing that the eigenvalues shift after training, though almost negligibly.
- Figure 25: shows eigenvalue distributions averaged over all heads for all multi-headed models and all tasks, with the corresponding standard deviations.
- Figures 26-28: show eigenvalue distributions averaged over all seeds for one of the heads, all models and all tasks, with the corresponding standard deviations.
- Figure 29: shows the comparison of two different gating placements on the eigenvalue spectra for one head, across ListOps, IMDb, and MQAR, attention models, and all layers.
- Figure 30 shows the correlation of model performance against the percentage of eigenvalues contained in the corresponding bin averaged over all layers.
- Figure 31: shows correlation of model performance against the percentage of eigenvalues contained in the corresponding bin for MQAR, each layer, and all models.
- Figures 32-33: show evolution of eigenvalues from initialization, over training half-time, to the final distribution for CIFAR-10 and IMDb tasks.
- Figure 34: visualizes the complex eigenvalues across LRA task, SSMs, and all layers. The trained values are indicated in blue and the initialization in orange.

The provided figures showcase the robustness of our conclusions and their extension to different seeds, model heads, and tasks not directly provided in the main paper.

All figures make use of the same legend as Figure 1, once again displayed in Figure 5, below. Thereby, bars show the proportion of eigenvalues falling within discretized ranges. Lightly shaded and darker bars indicate the distribution at initialization and after training, respectively. Error bars denote standard deviation across input sequences. Model performance, measured as perplexity for WikiText (lower is better) and percentage of correct output sequences for the other (higher is better), is indicated in the parentheses.

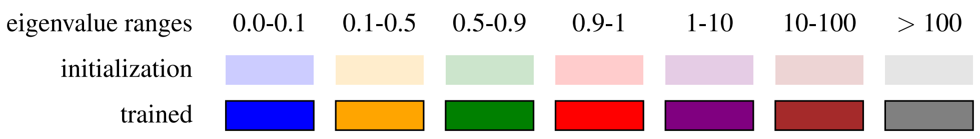
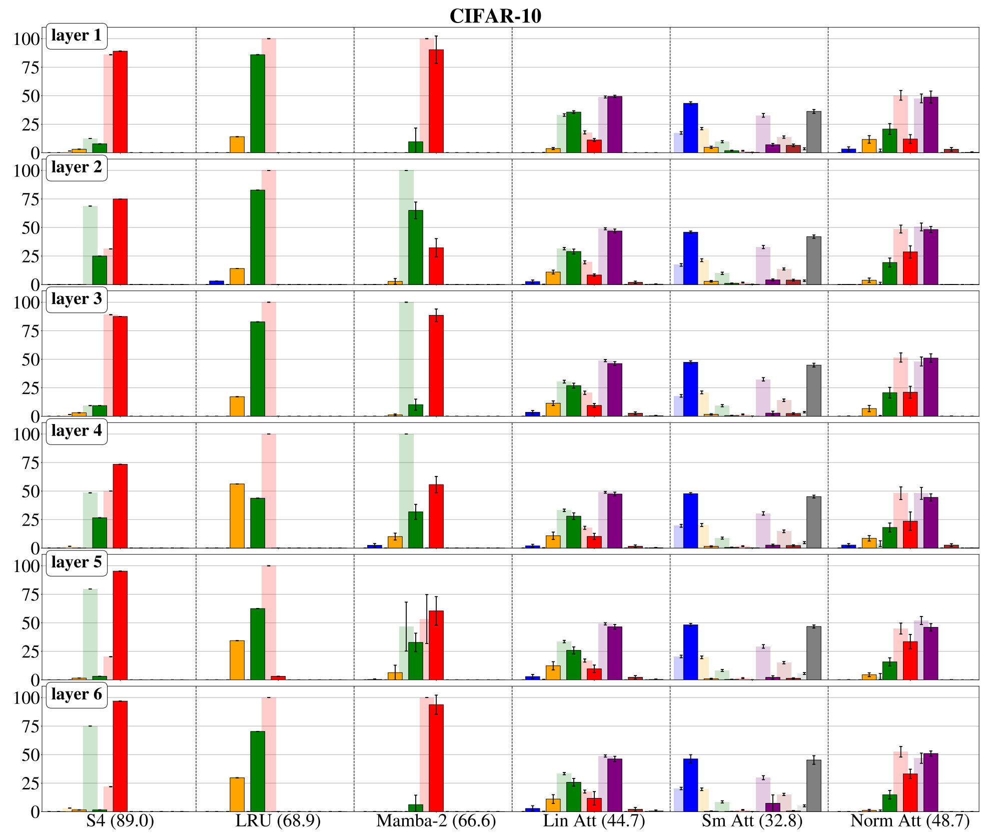
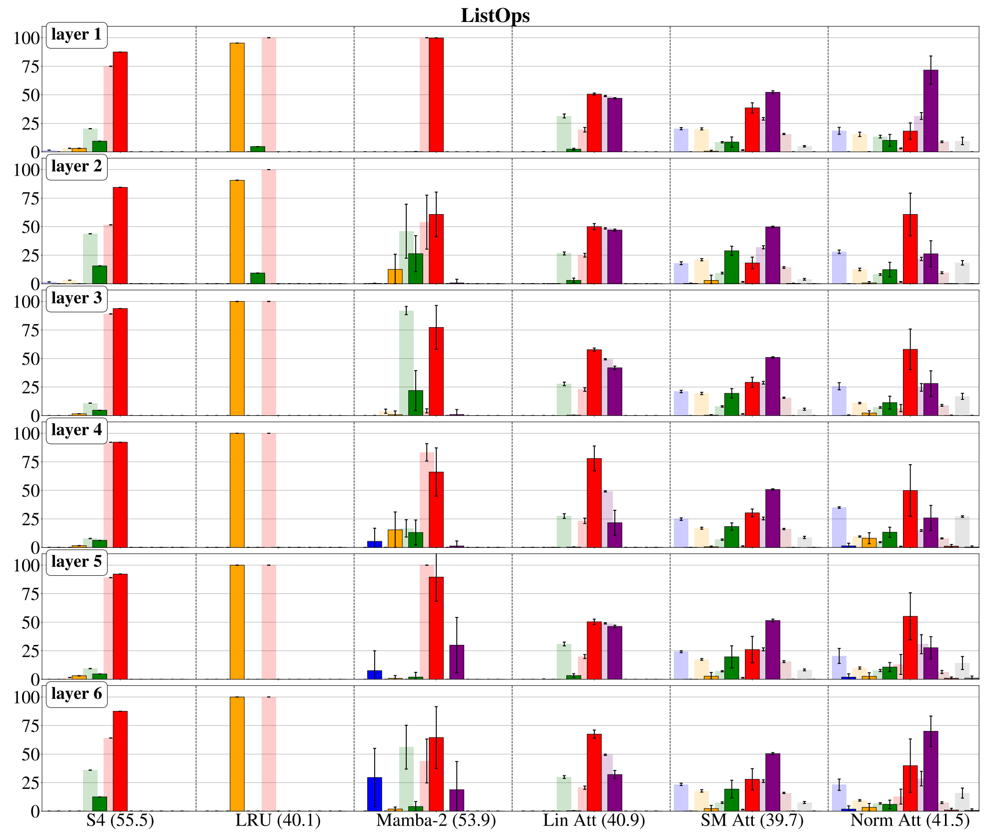


Figure 5: Plot legend of all subsequent figures.

Table 3: Performance of models across different tasks for the nominal seed shown in Figure 1.

	S4	LRU	Mamba-2	Lin Att	Sm Att	Norm Att
CIFAR-10	**89.0**	68.9	66.6	44.7	32.8	48.7
IMDb	81.5	86.1	**86.8**	63.7	62.1	66.4
ListOps	**55.5**	40.1	53.9	40.9	39.7	41.5
MQAR	< 1.5	< 1.5	$> \mathbf{99.0}$	< 1.5	$> \mathbf{99.0}$	< 1.5
WikiText	74.3	60.9	40.1	39.5	**34.4**	43.0


CIFAR-10
layer 1
layer 2
layer 3
layer 4
layer 5
layer 6
100
75
50
25
0
S4 (89.0)
LRU (68.9)
Mamba-2 (66.6)
Lin Att (44.7)
Sm Att (32.8)
Norm Att (48.7)


ListOps
layer 1
layer 2
layer 3
layer 4
layer 5
layer 6
100
75
50
25
0
S4 (55.5)
LRU (40.1)
Mamba-2 (53.9)
Lin Att (40.9)
SM Att (39.7)
Norm Att (41.5)

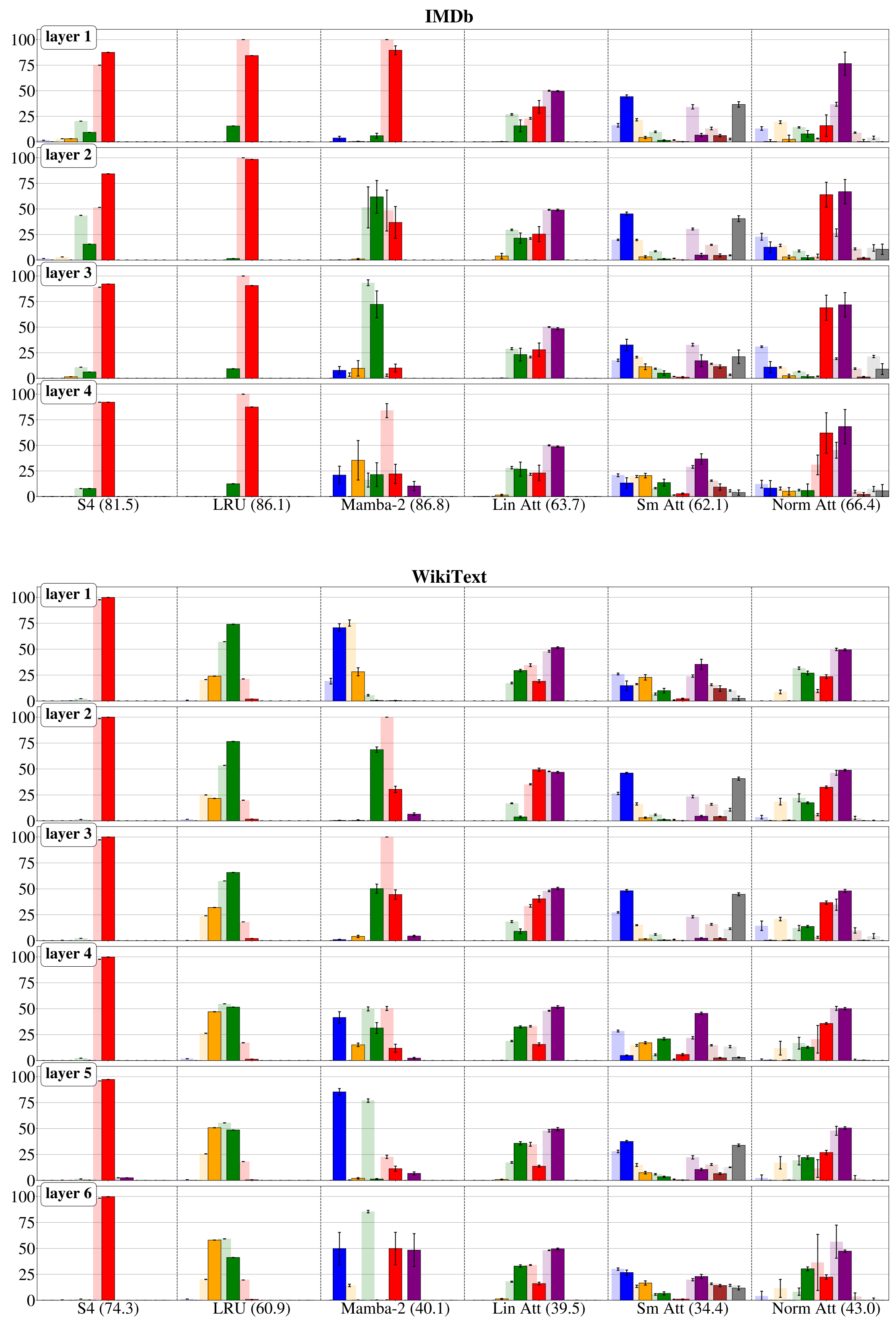
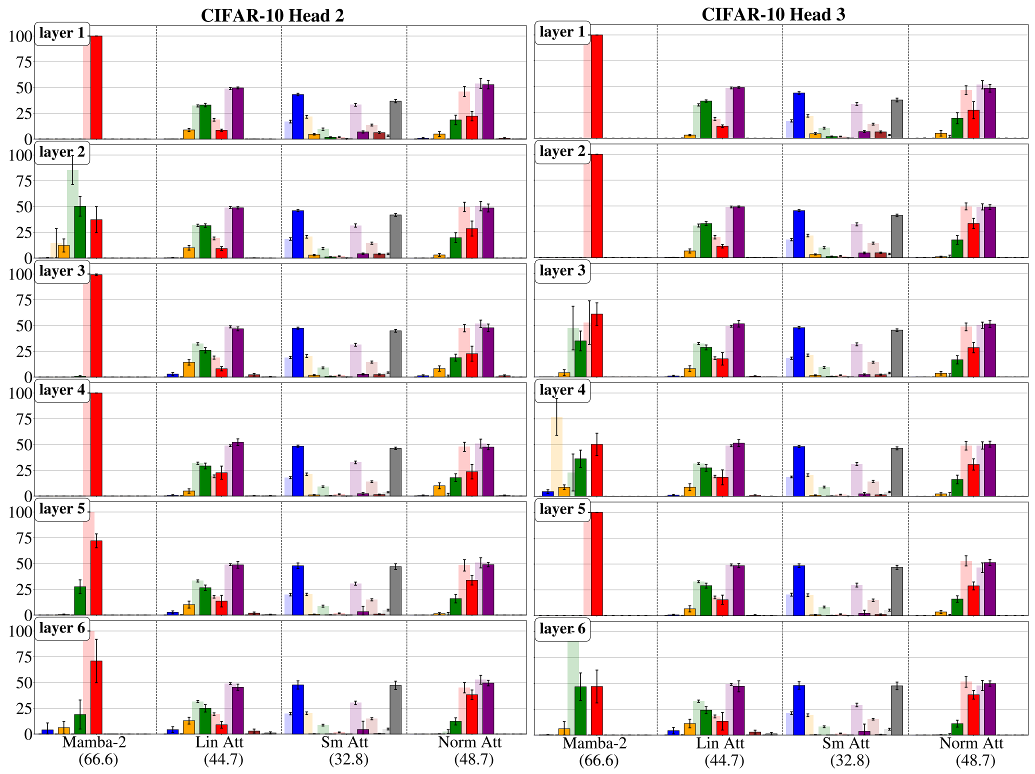


Figure 6: Eigenvalue distributions for one head, across all models, layers, and tasks (excluding MQAR).



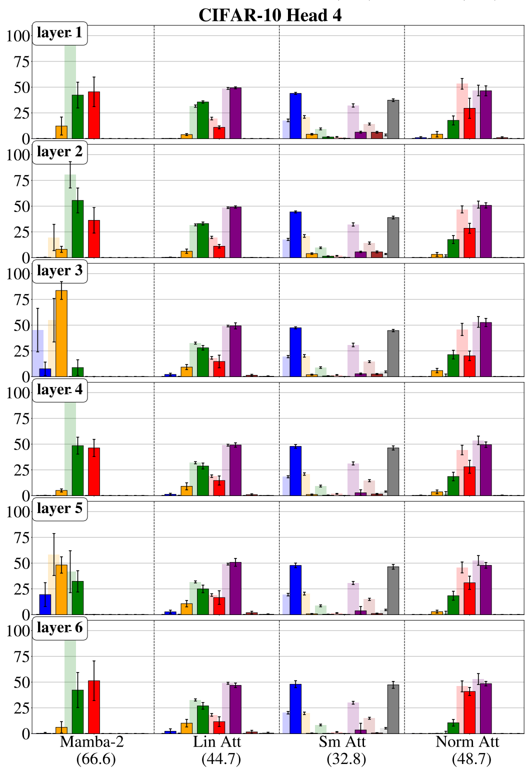
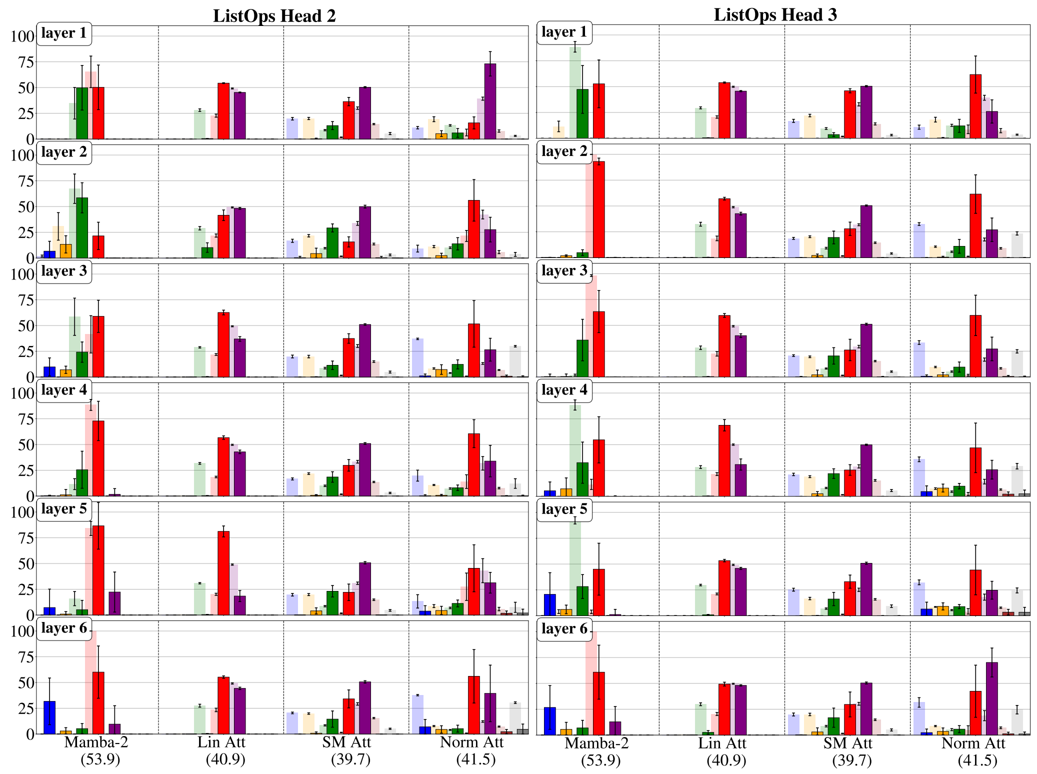


Figure 7: Eigenvalue distributions across remaining heads and all layers for CIFAR-10.



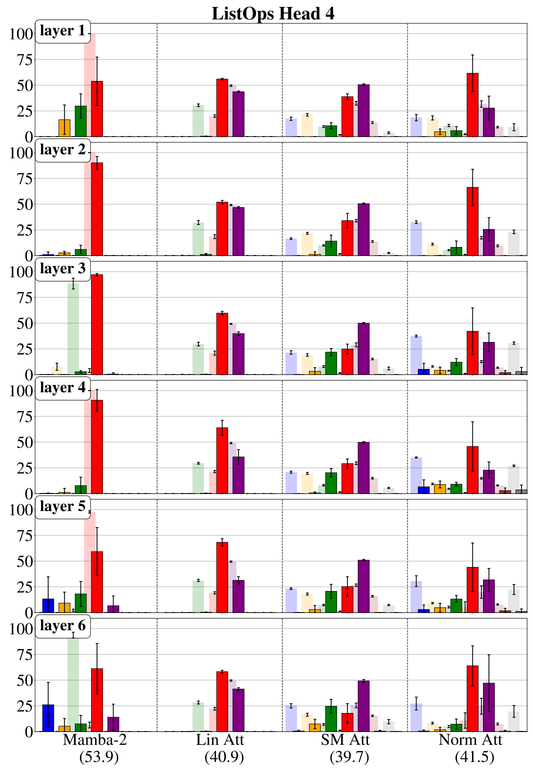


Figure 8: Eigenvalue distributions across remaining heads and all layers for ListOps.

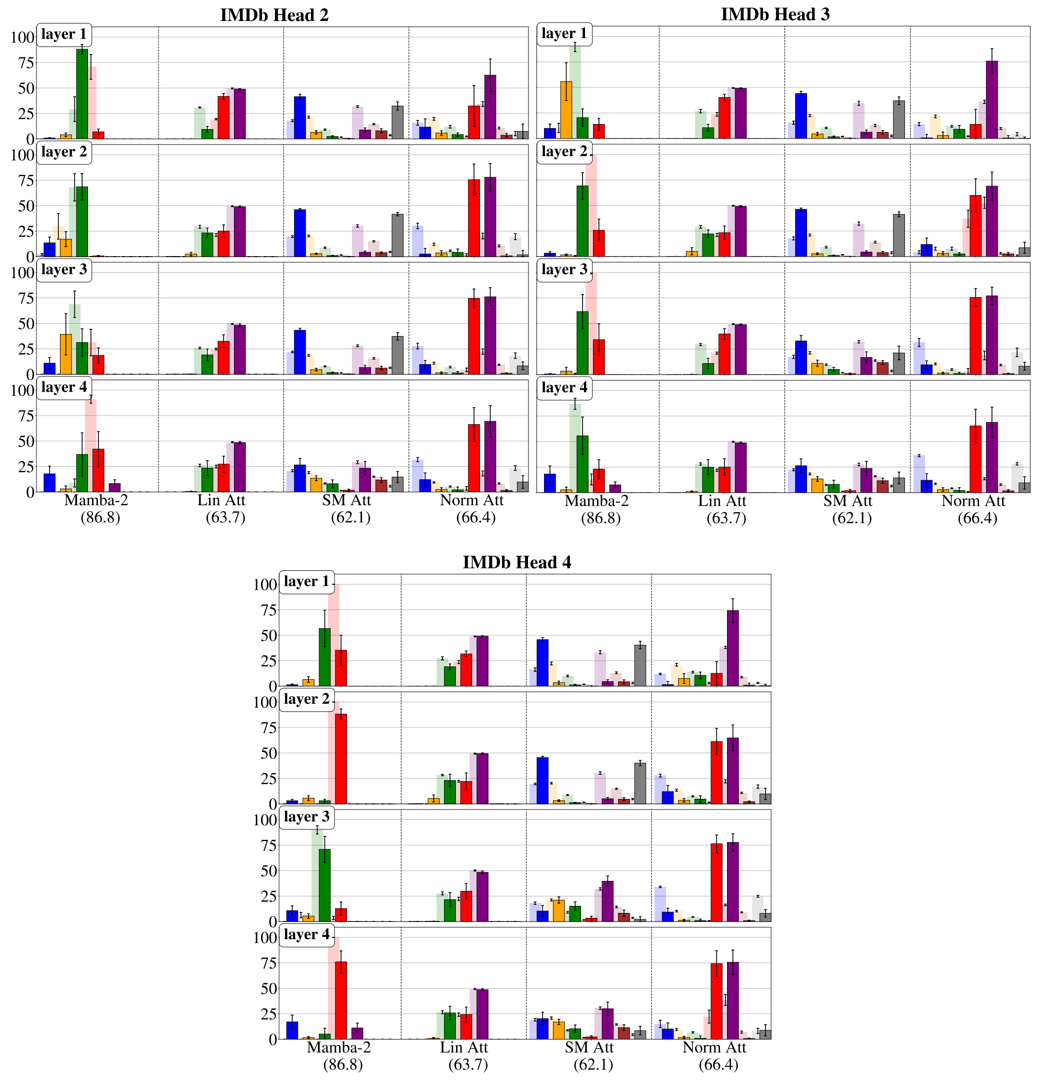


Figure 9: Eigenvalue distributions across remaining heads and all layers for IMDb.

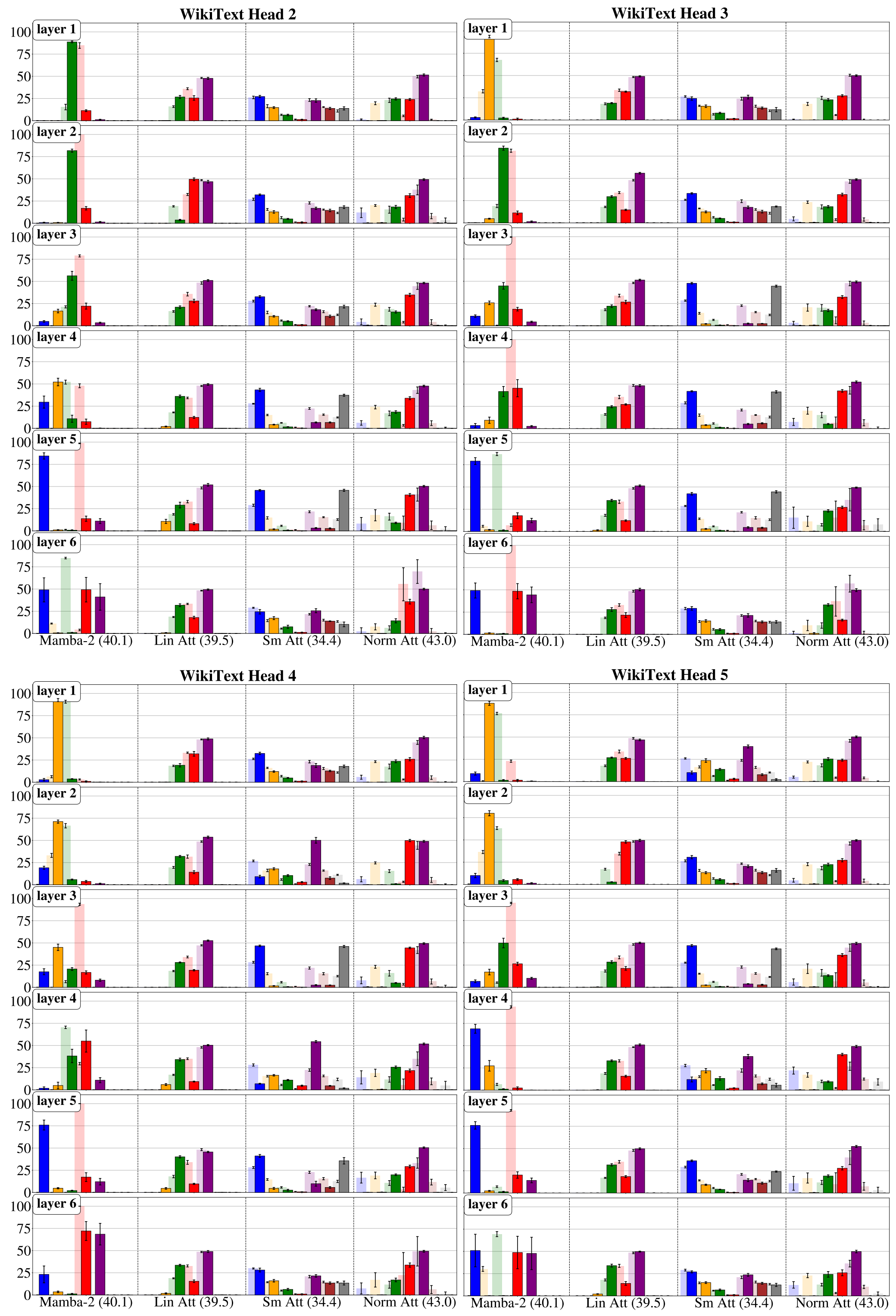


Figure 10: Eigenvalue distributions across 4 additional heads and all layers for WikiText.

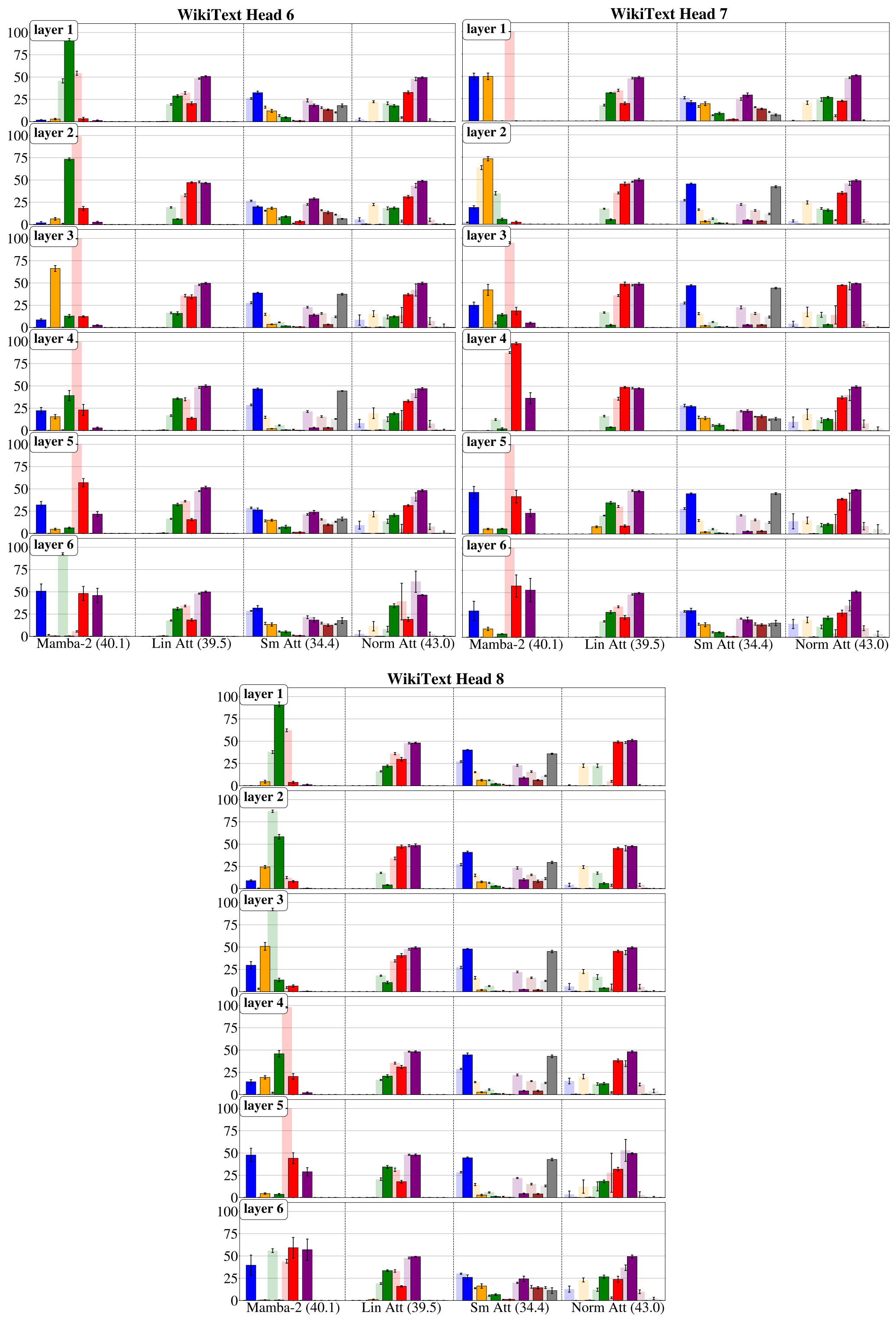


Figure 11: Eigenvalue distributions across remaining heads and all layers for WikiText.

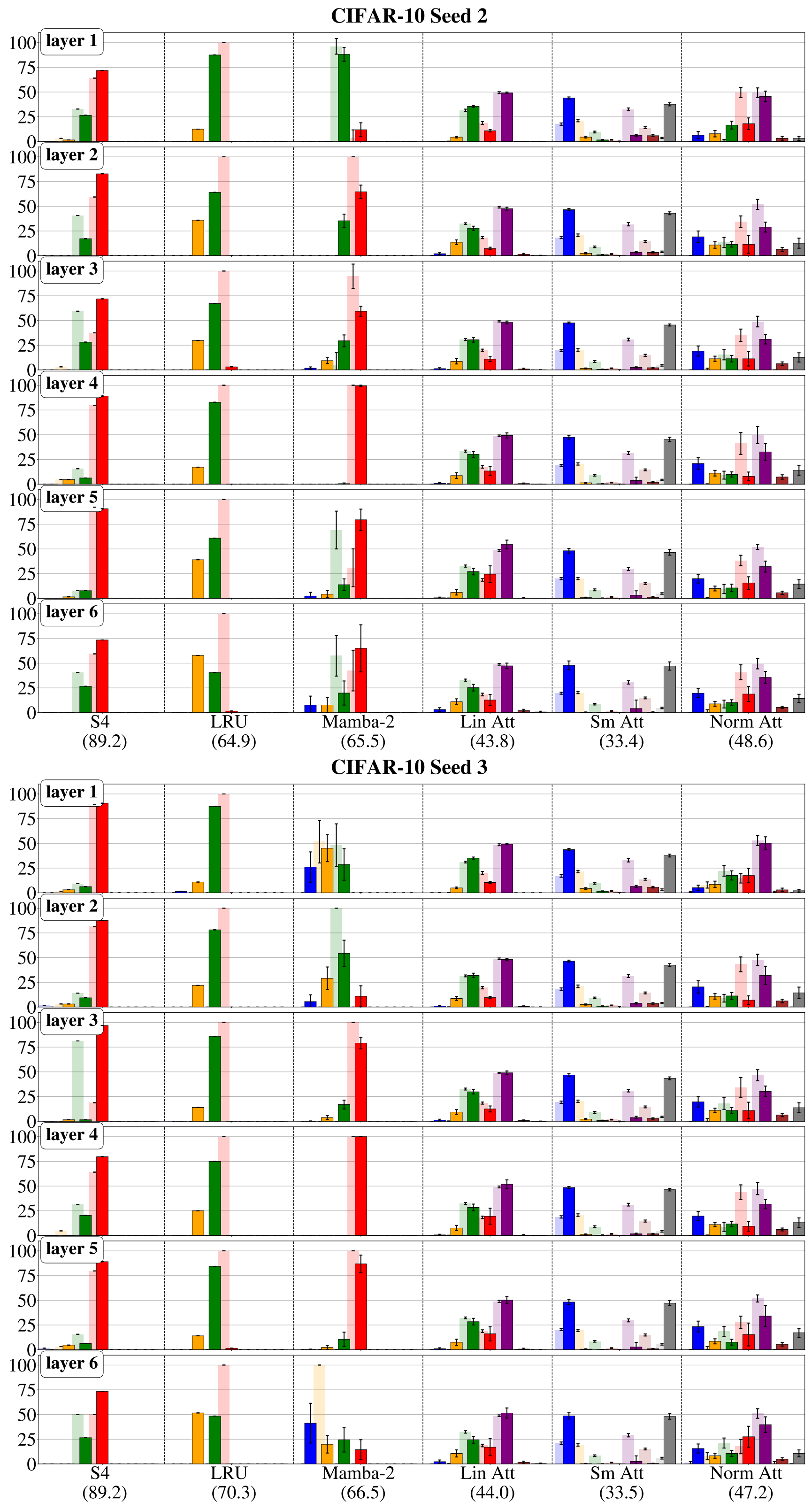


Figure 12: Eigenvalue distributions across models, layers, and two out of three additional random seeds for CIFAR-10.

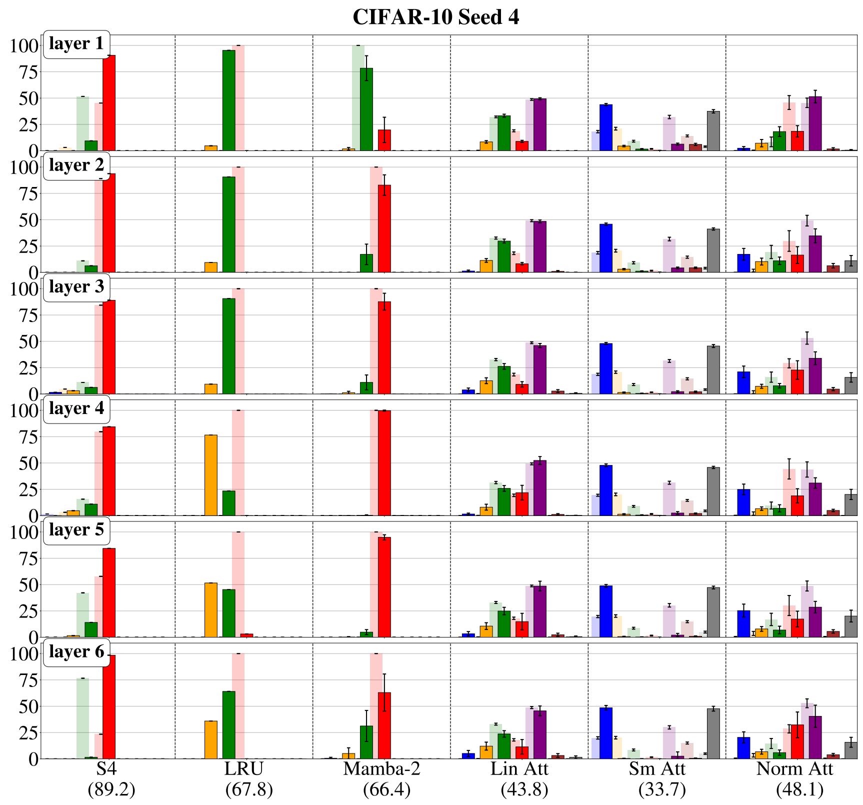
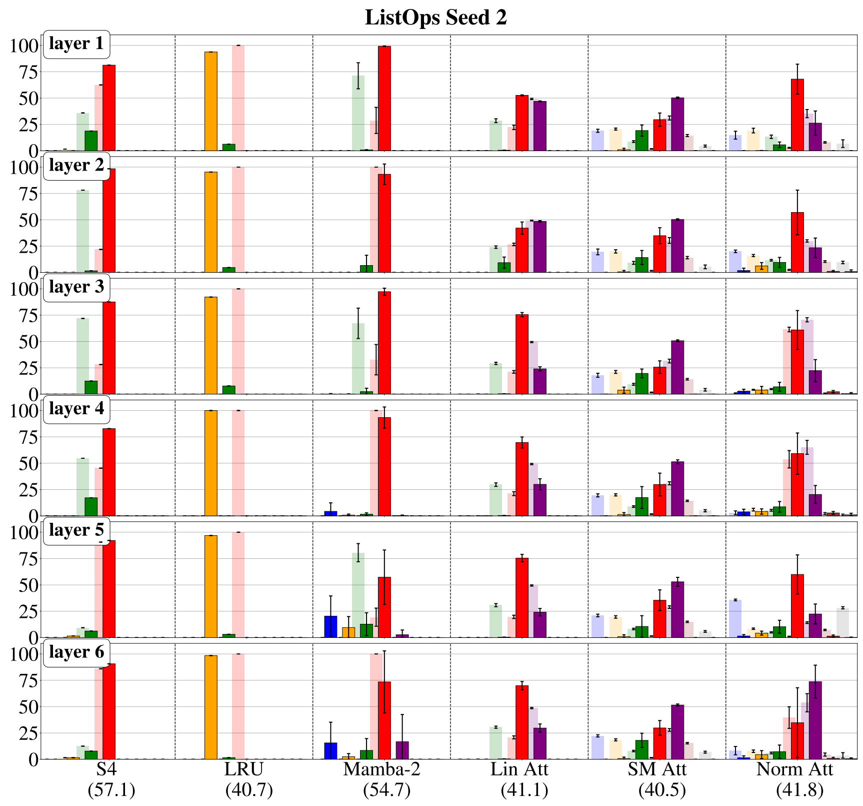


Figure 13: Eigenvalue distributions across models, layers, and one remaining additional random seed for CIFAR-10.



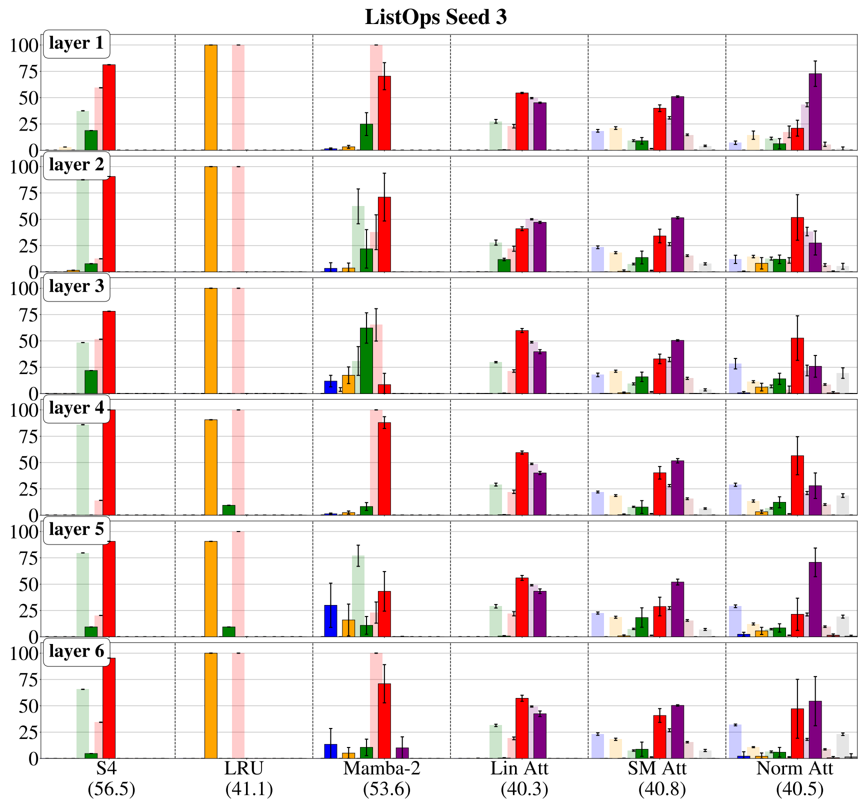


Figure 14: Eigenvalue distributions across models, layers, and two out of three additional random seeds for ListOps.

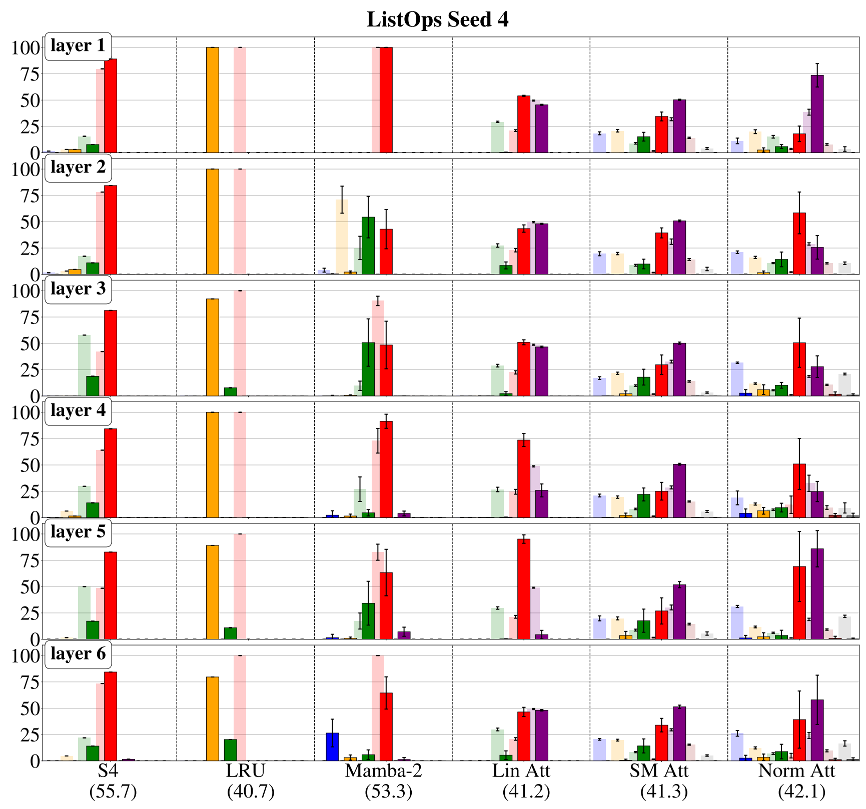


Figure 15: Eigenvalue distributions across models, layers, and one remaining additional random seed for ListOps.

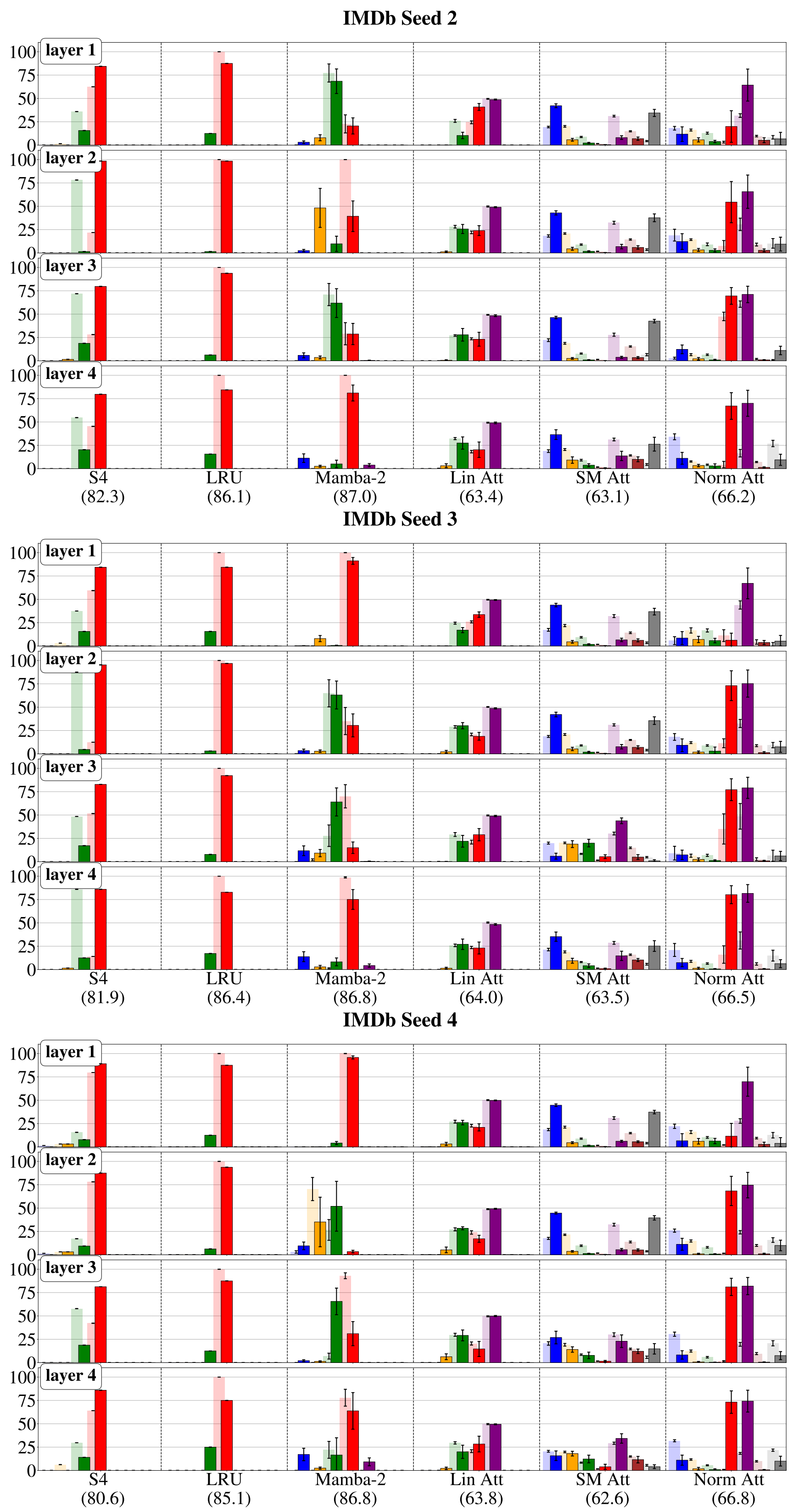
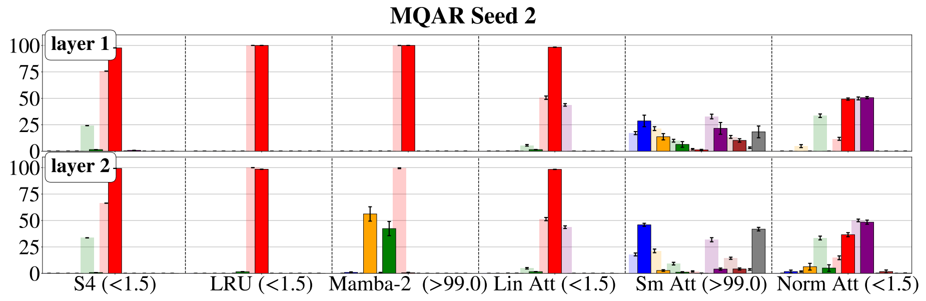
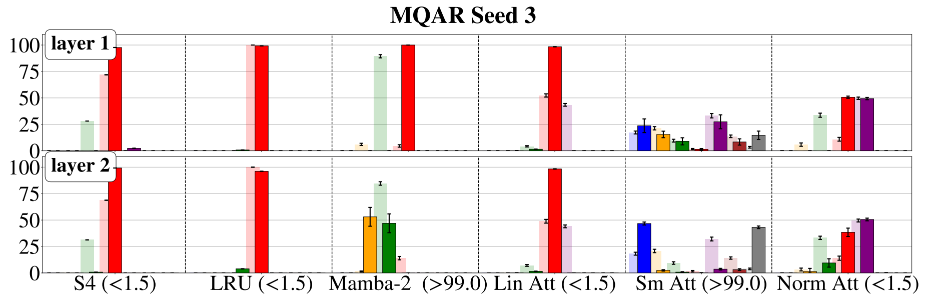


Figure 16: Eigenvalue distributions across models, layers, and three additional random seeds for IMDb.





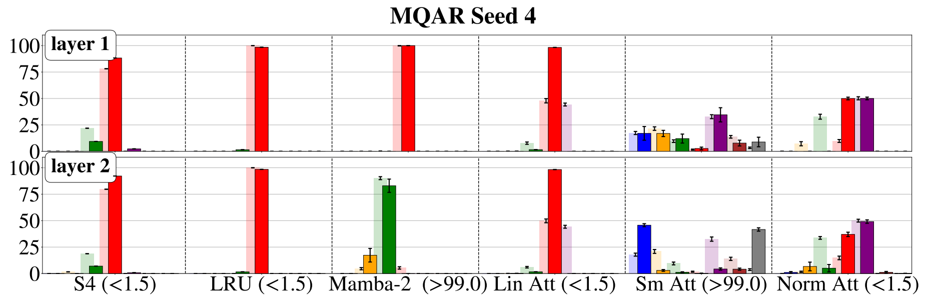


Figure 17: Eigenvalue distributions across models, layers, and three additional random seeds for MQAR.

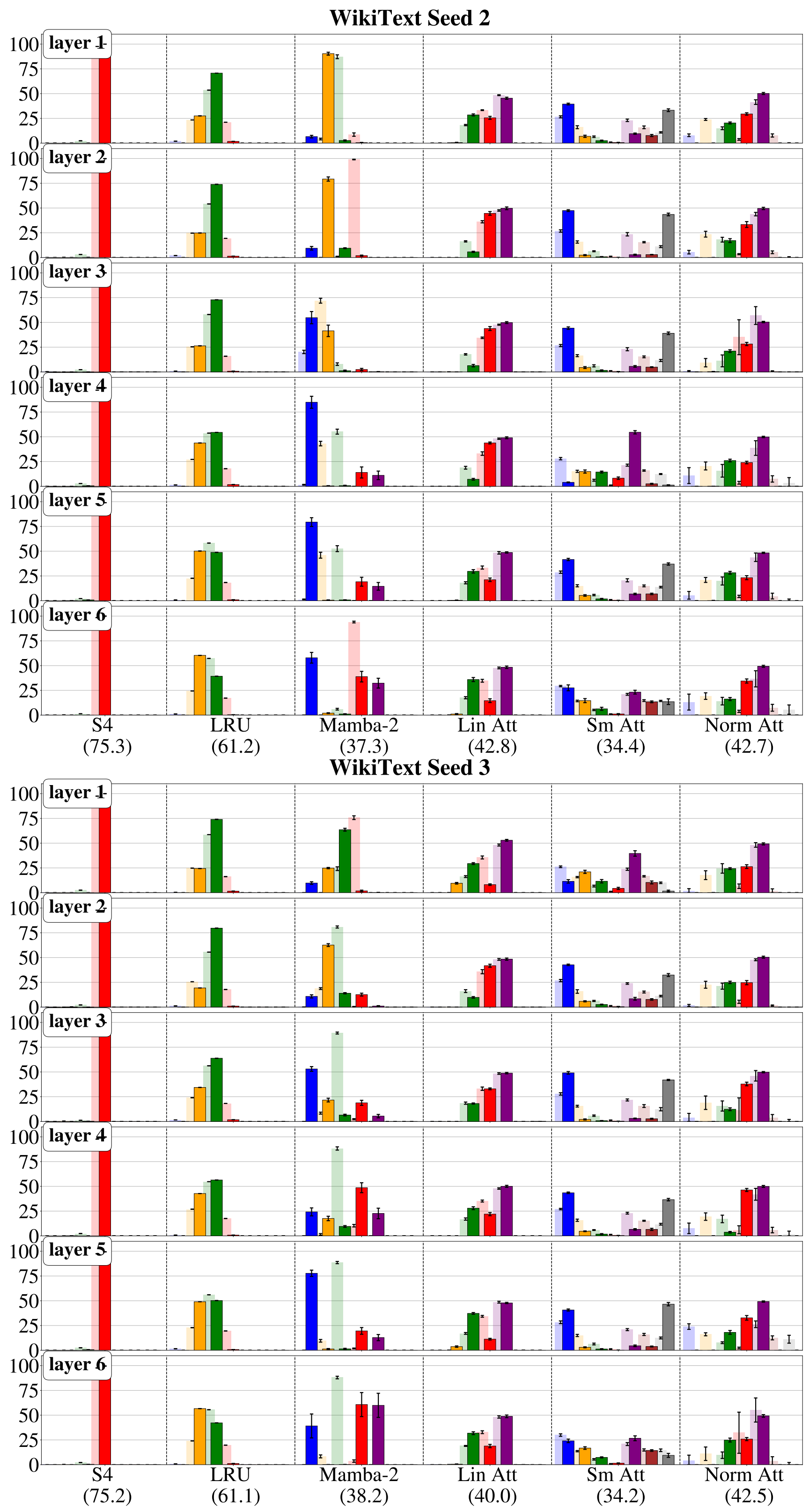


Figure 18: Eigenvalue distributions across models, layers, and two out of three additional random seeds for WikiText.

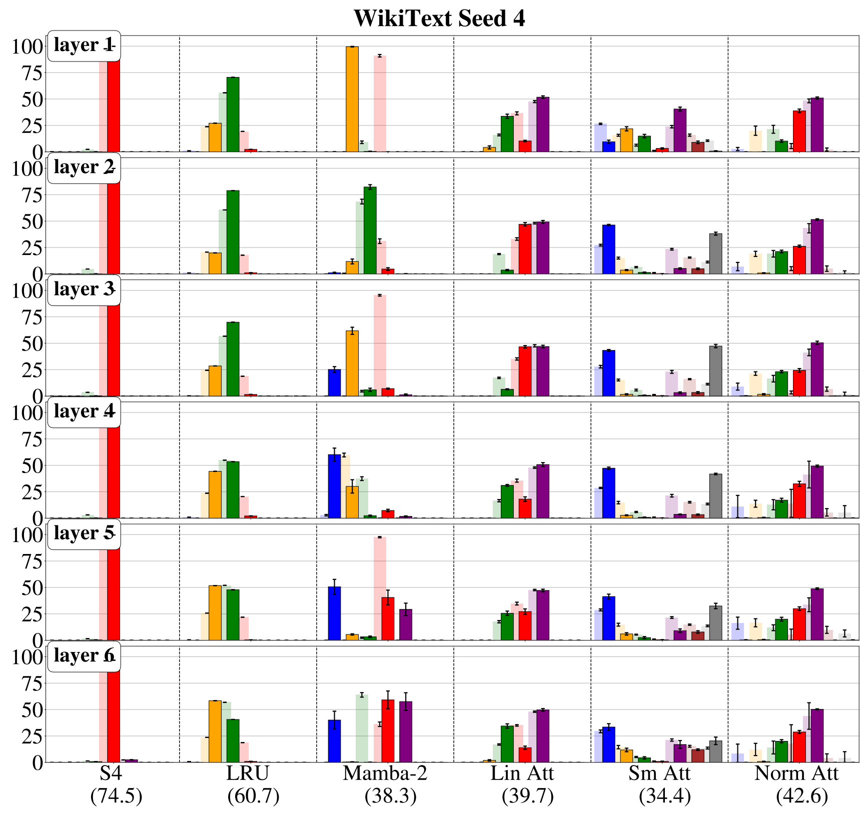


Figure 19: Eigenvalue distributions across models, layers, and one remaining additional random seed for WikiText.

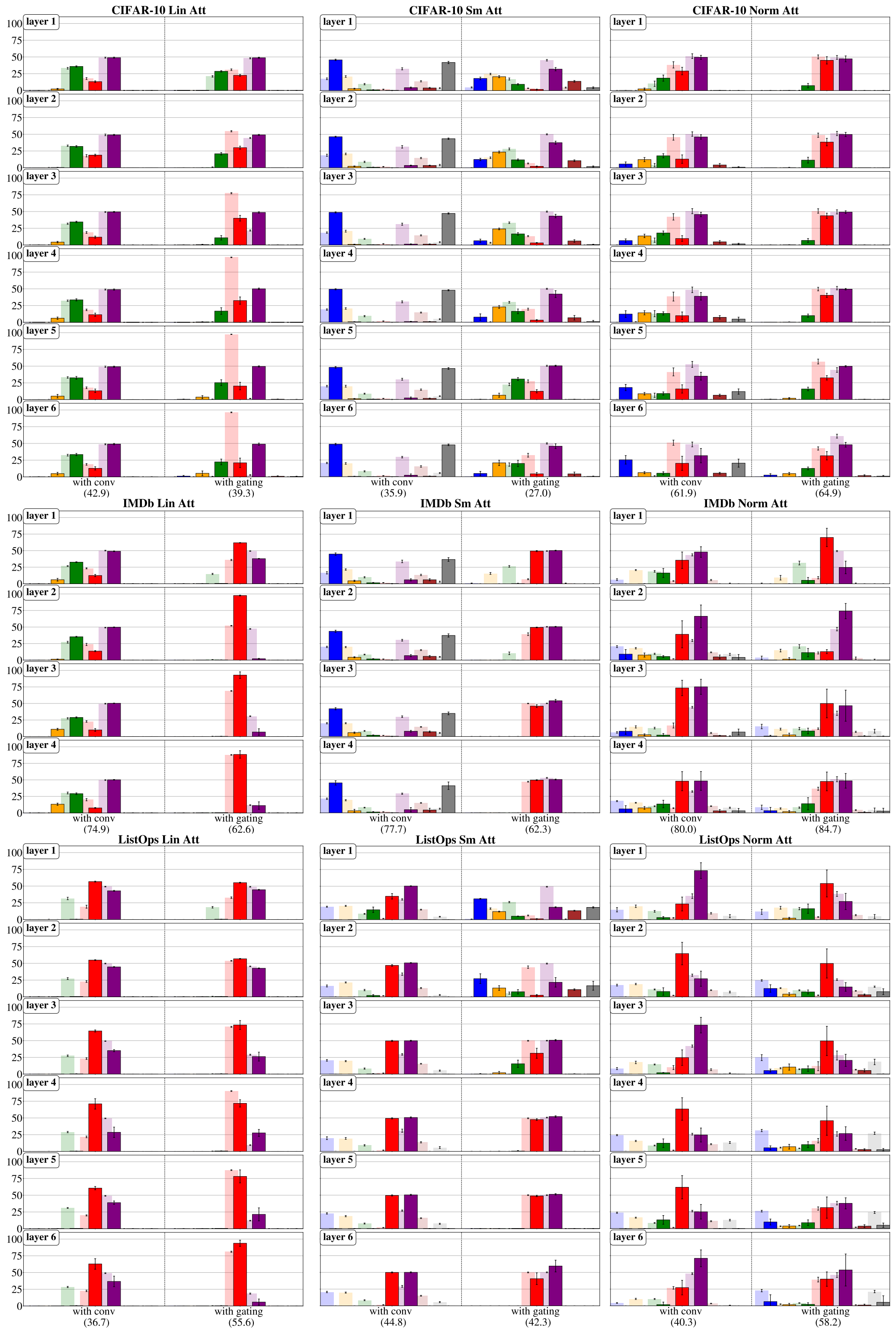


Figure 20: Comparison of the effects of gating and convolution on the eigenvalue spectra for one head, across CIFAR-10, IMDb, and ListOps, the three investigated attention models, and all layers.

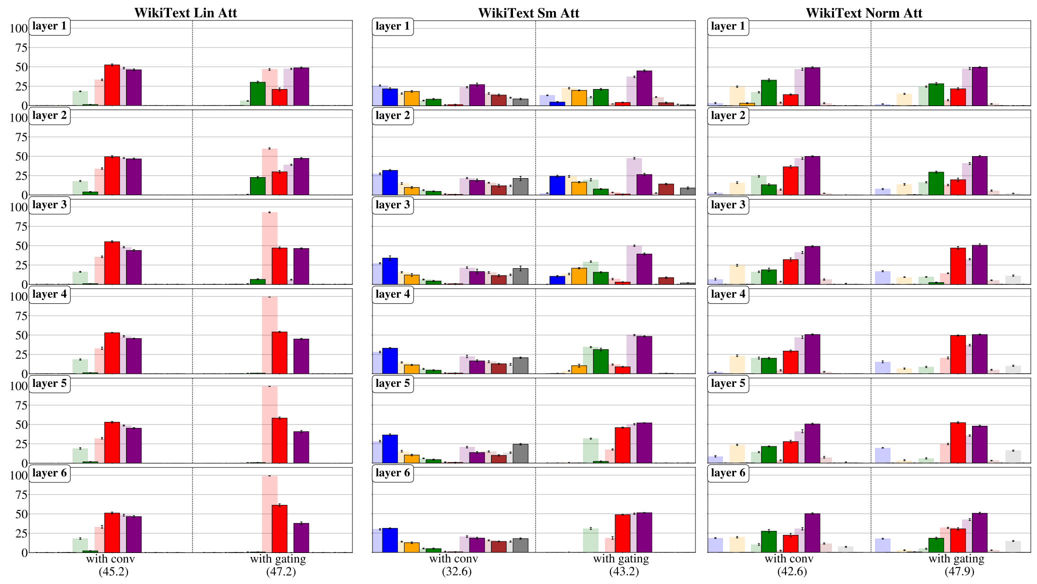


Figure 21: Comparison of the effects of gating and convolution on the eigenvalue spectra for one head, WikiText, the three investigated attention models, and all layers.

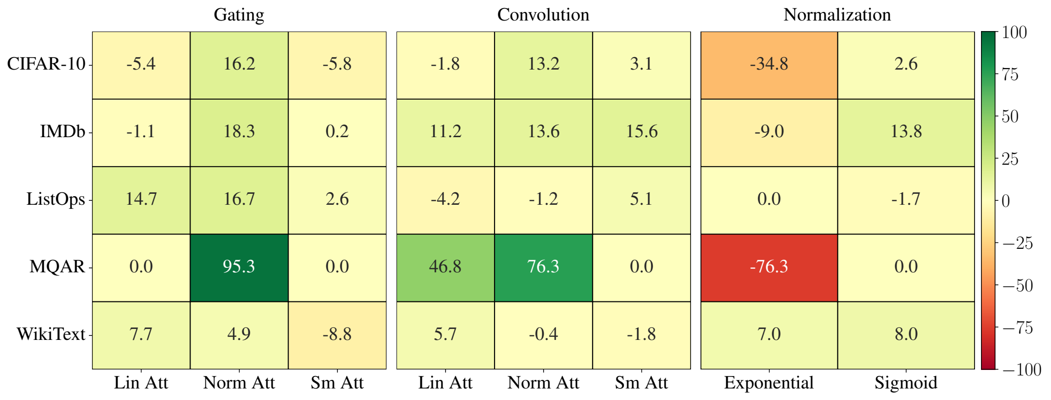


Figure 22: The absolute change in performance after each architectural change across models and tasks. For gating and convolution, the difference is computed in comparison to the nominal models shown in Figure 1, whose performance is summarized in Table 3. For normalization, the difference is computed with respect to the performance of norm attention with convolution and softplus as normalization function, shown in Figure 23.

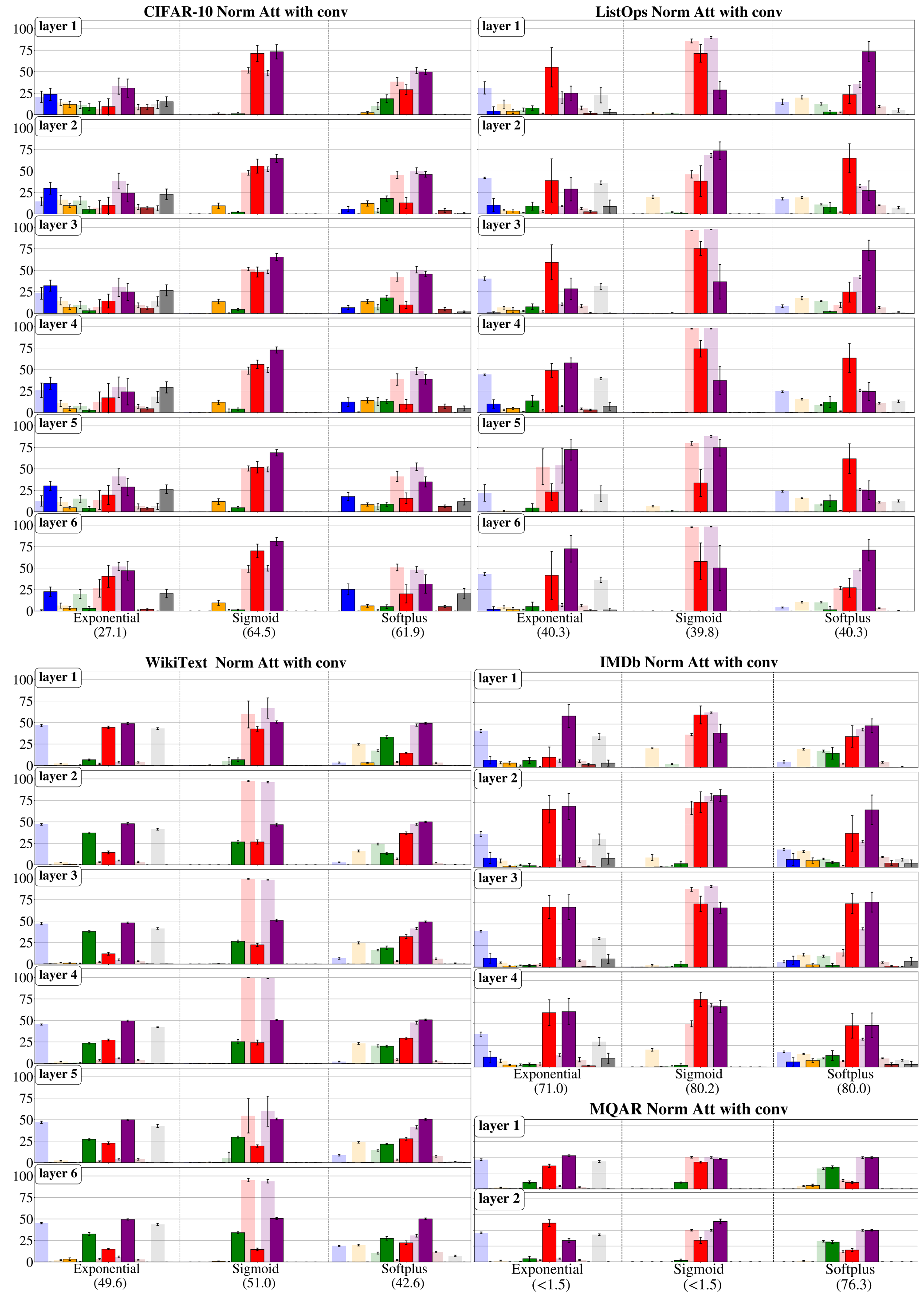
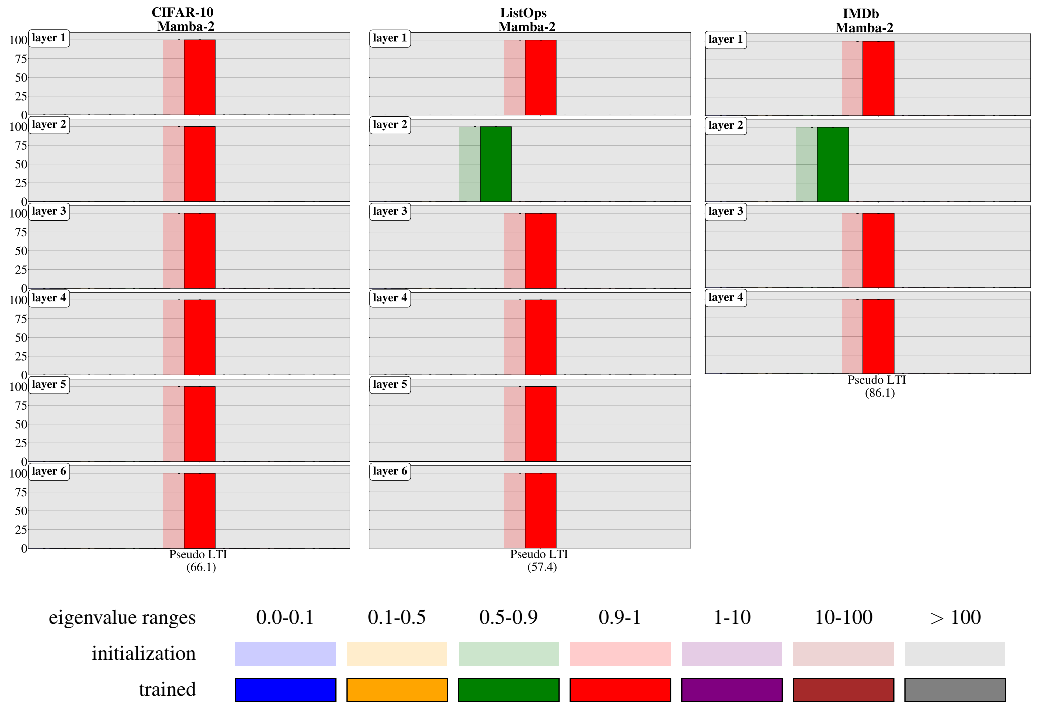


Figure 23: Eigenvalue distributions for all tasks and one of the heads for norm attention with convolution, using different normalization functions.



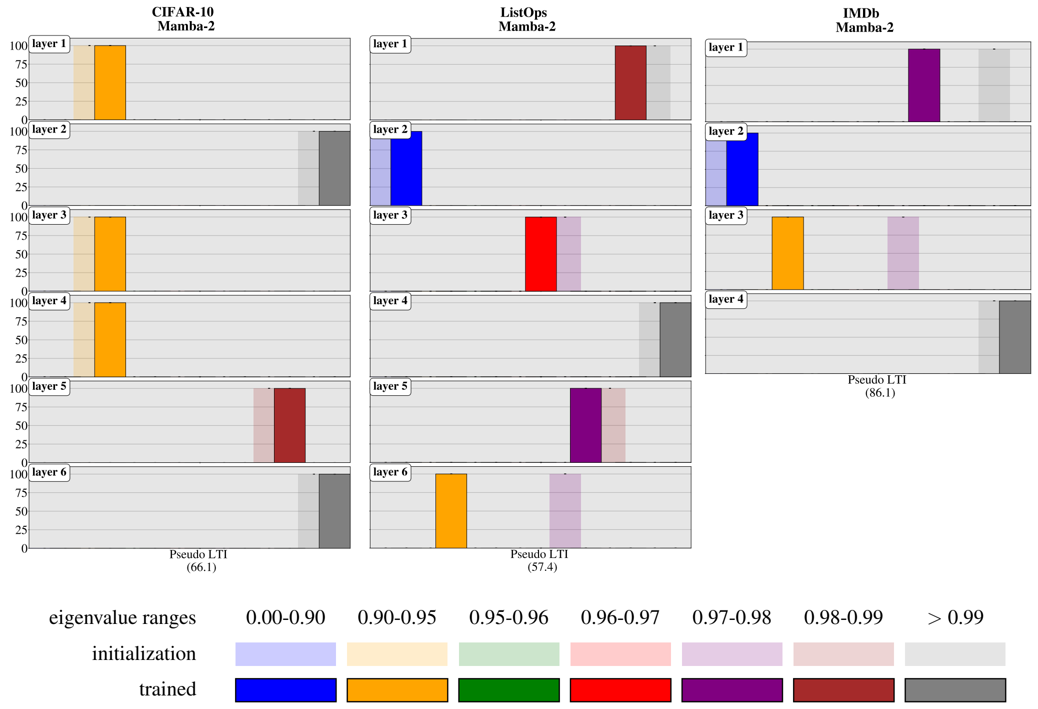


Figure 24: Eigenvalue spectra of Mamba-2 Pseudo LTI, one head, across all layers and LRA models. Standard binning is presented in the top row of plots, followed by a more fine-grained visualization around one. The corresponding legends are provided below the respective plots.

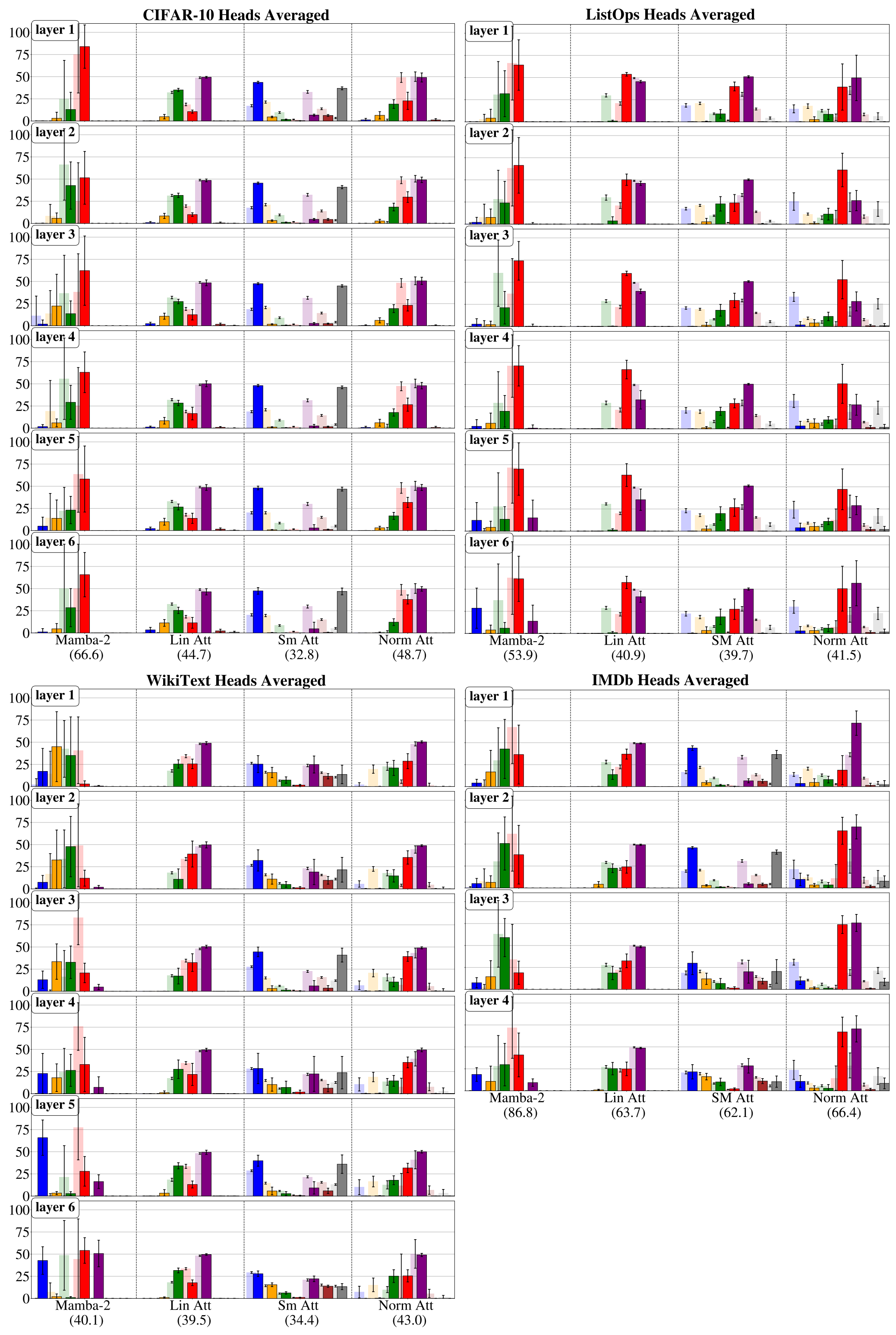
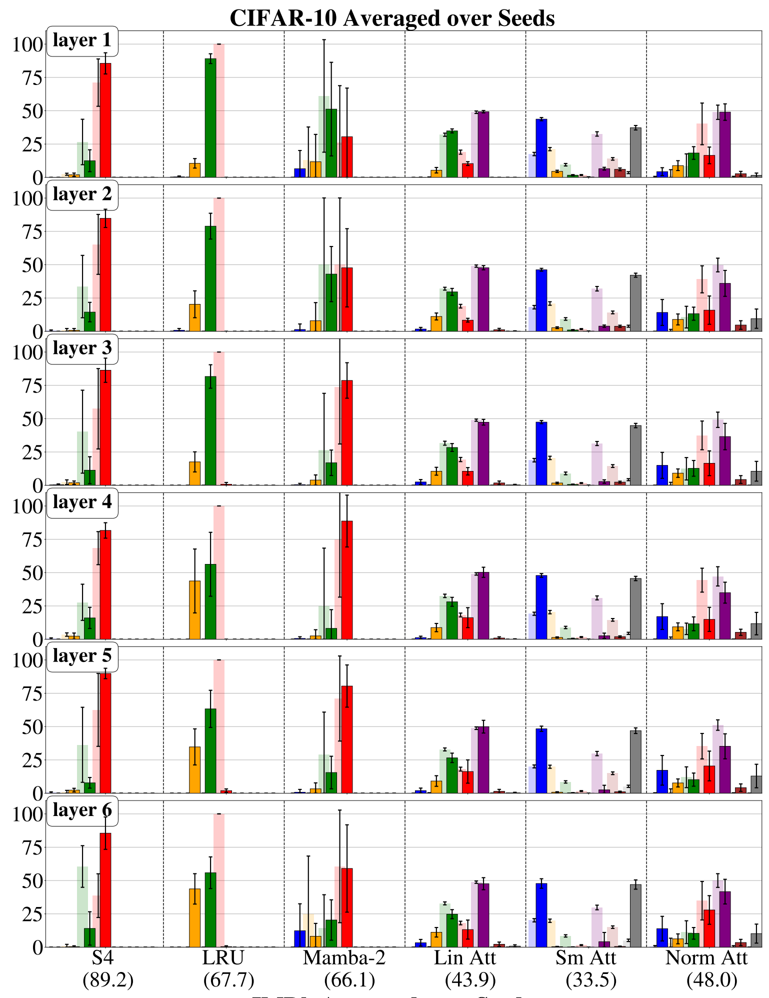


Figure 25: Distributions of eigenvalues averaged across heads for all layers and tasks.



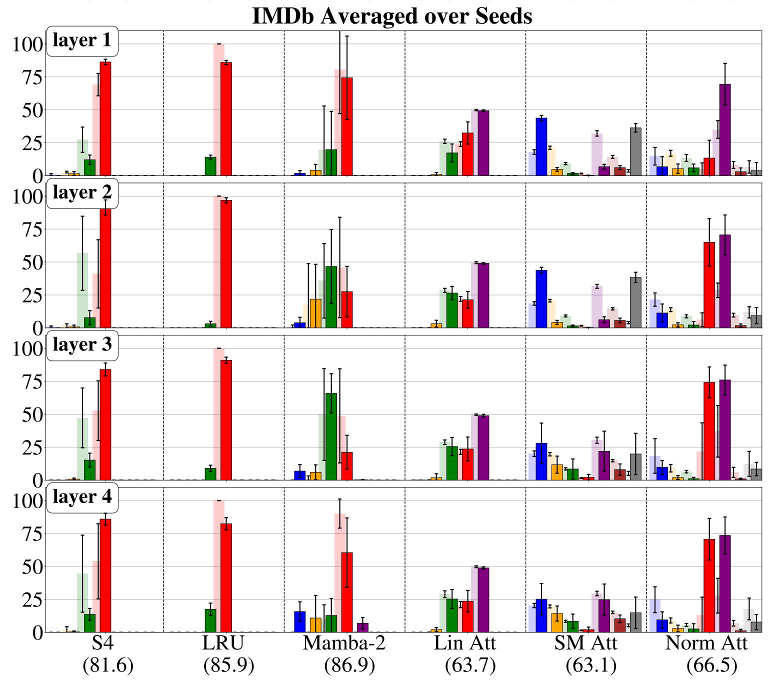
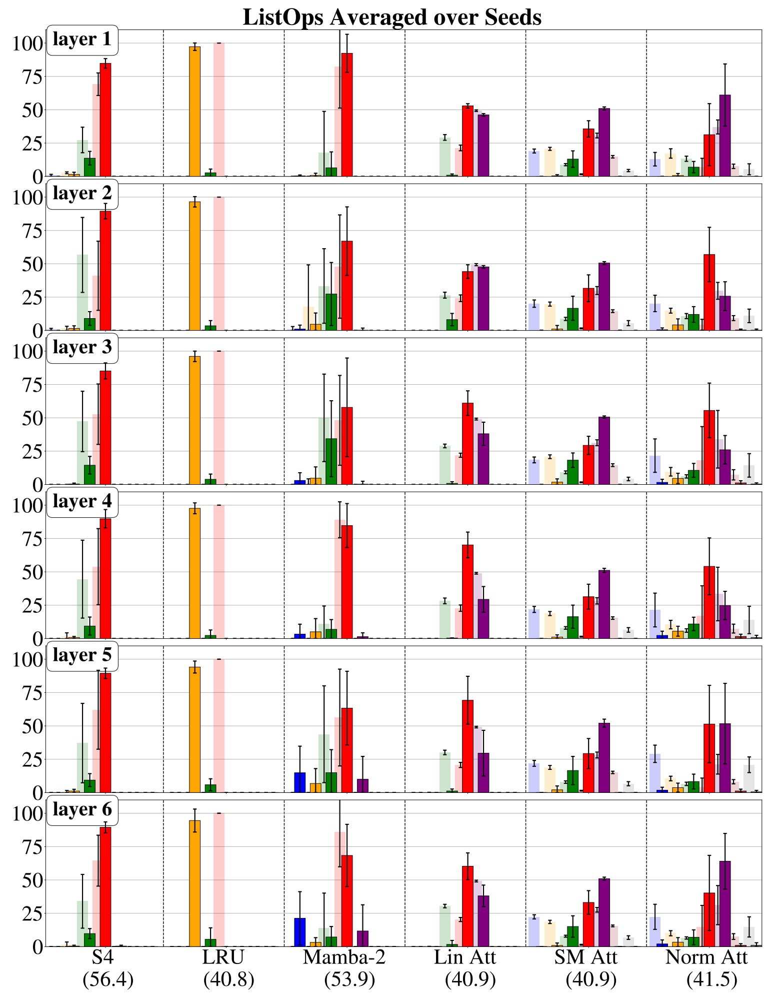


Figure 26: Distributions of eigenvalues averaged across seeds for one of the heads on CIFAR and IMDb tasks.



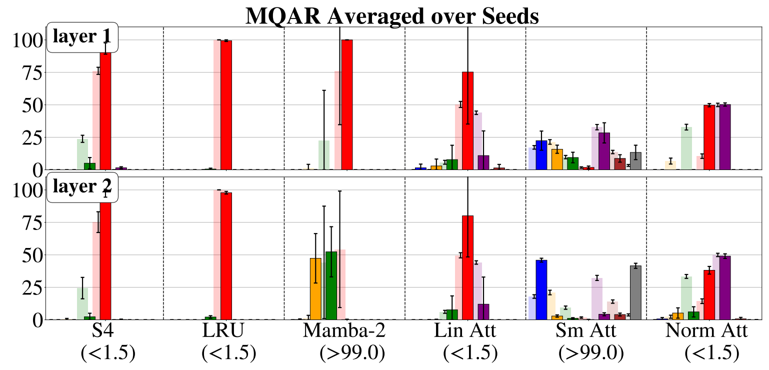


Figure 27: Distributions of eigenvalues averaged across seeds for one of the heads on ListOps and MQAR tasks.

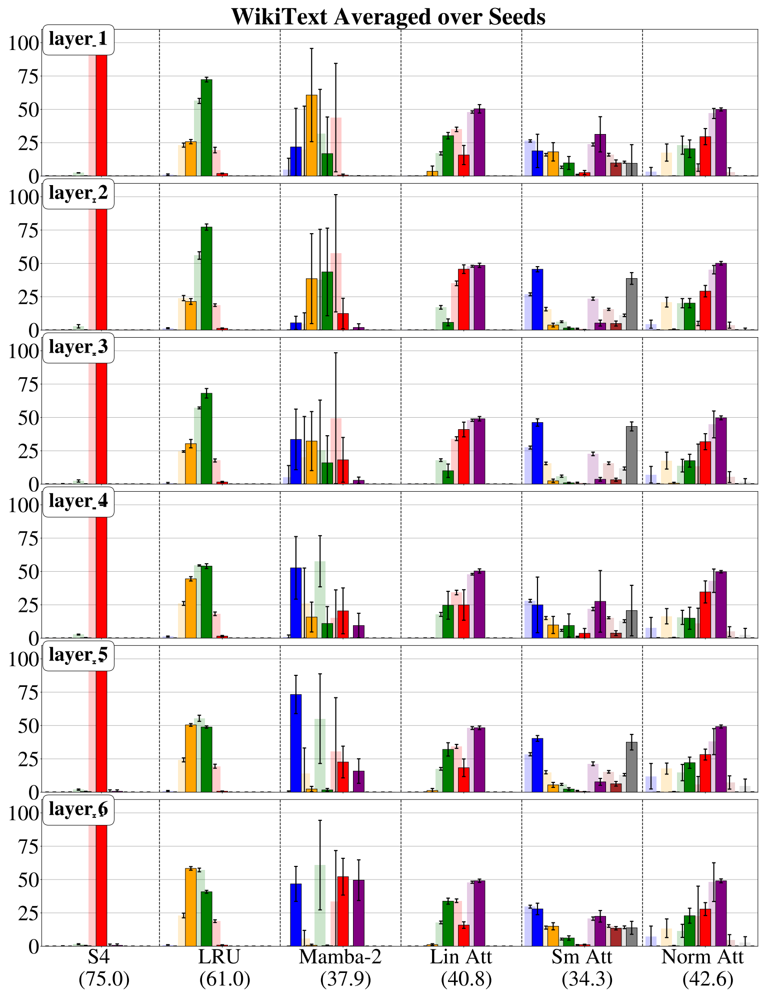


Figure 28: Distributions of eigenvalues averaged across seeds for one of the heads on WikiText task.

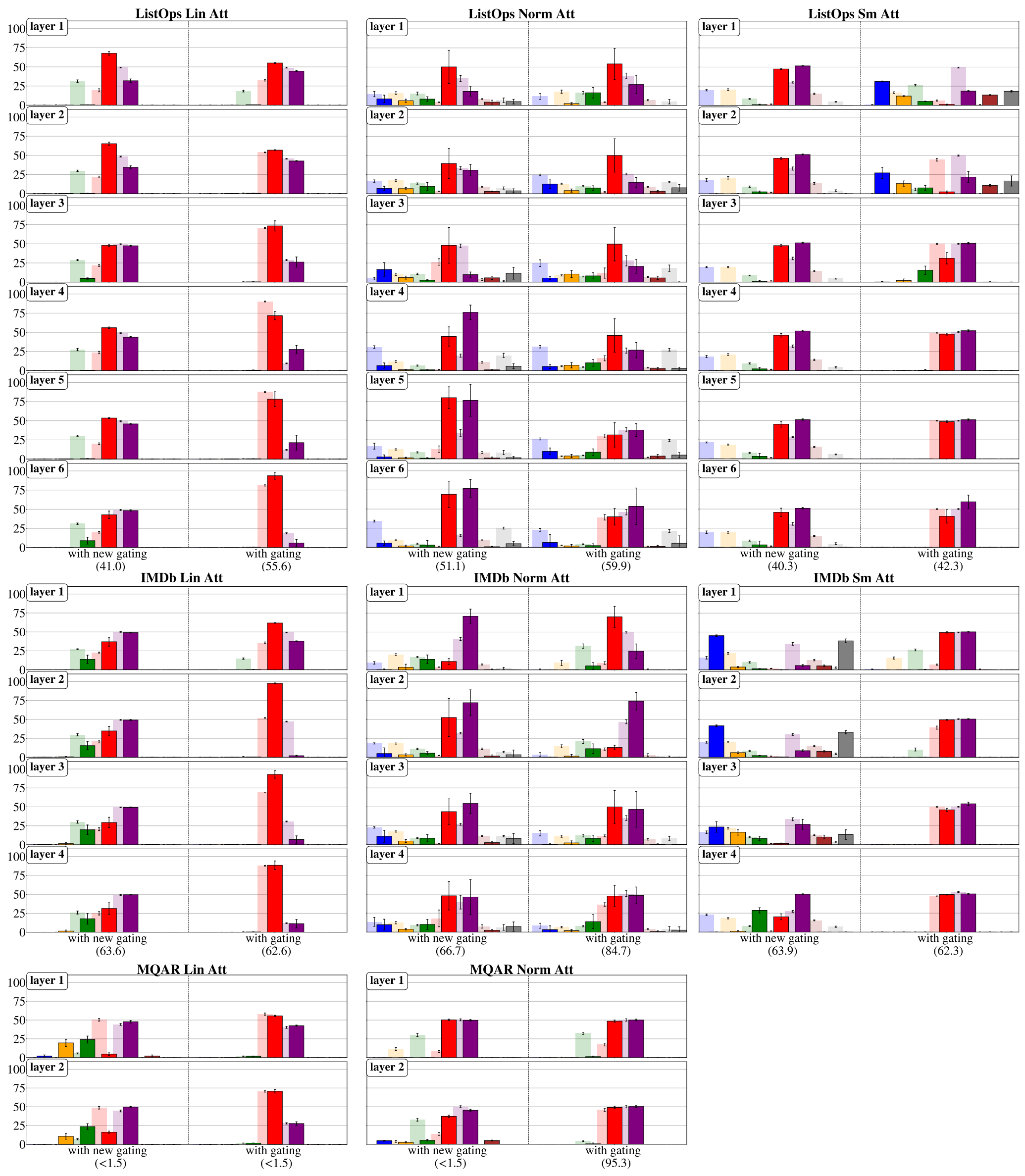


Figure 29: Comparison of two different gating placements on the eigenvalue spectra for one head, across ListOps, IMDb, and MQAR, attention models, and all layers. With "new gating," we refer to a gating placement right before the output projection. We observe that the "new gating" implementation results in overall worse performing models, and based on the eigenvalue spectra, does not seem to alleviate the task of gating from the dynamical system in the same manner as our initial implementation. The observations are thereby assumed to be correlated.

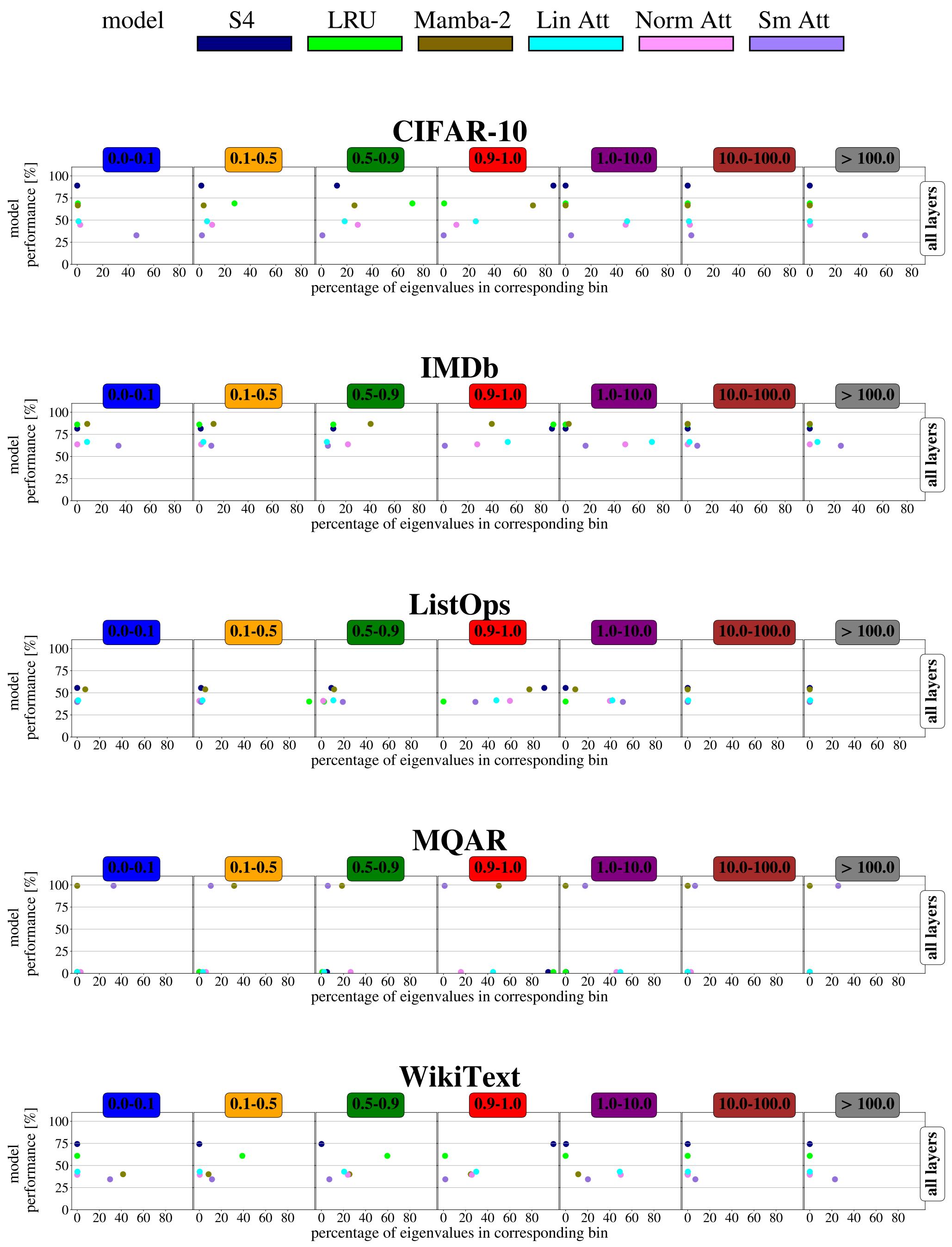


Figure 30: Correlation plots of model performance against the percentage of eigenvalues contained in the corresponding bin averaged across layers.

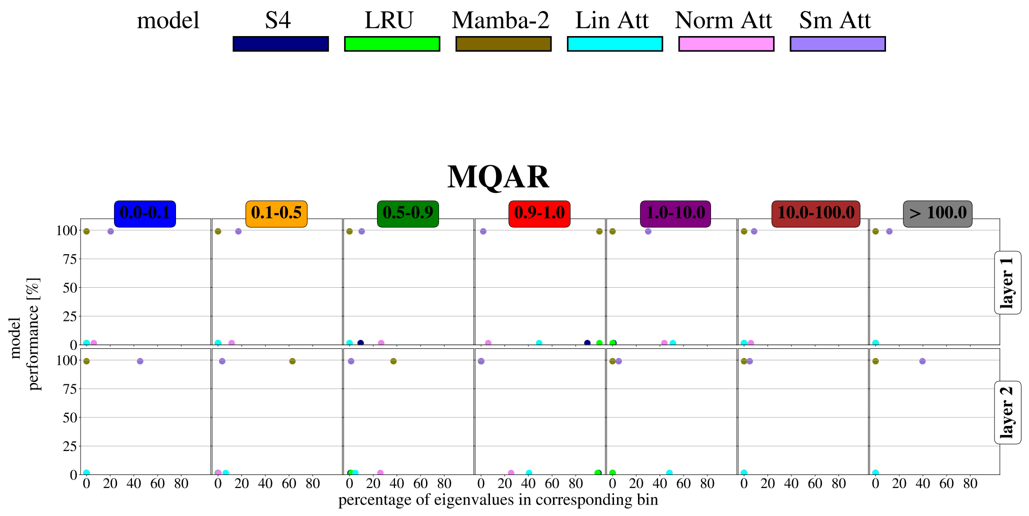
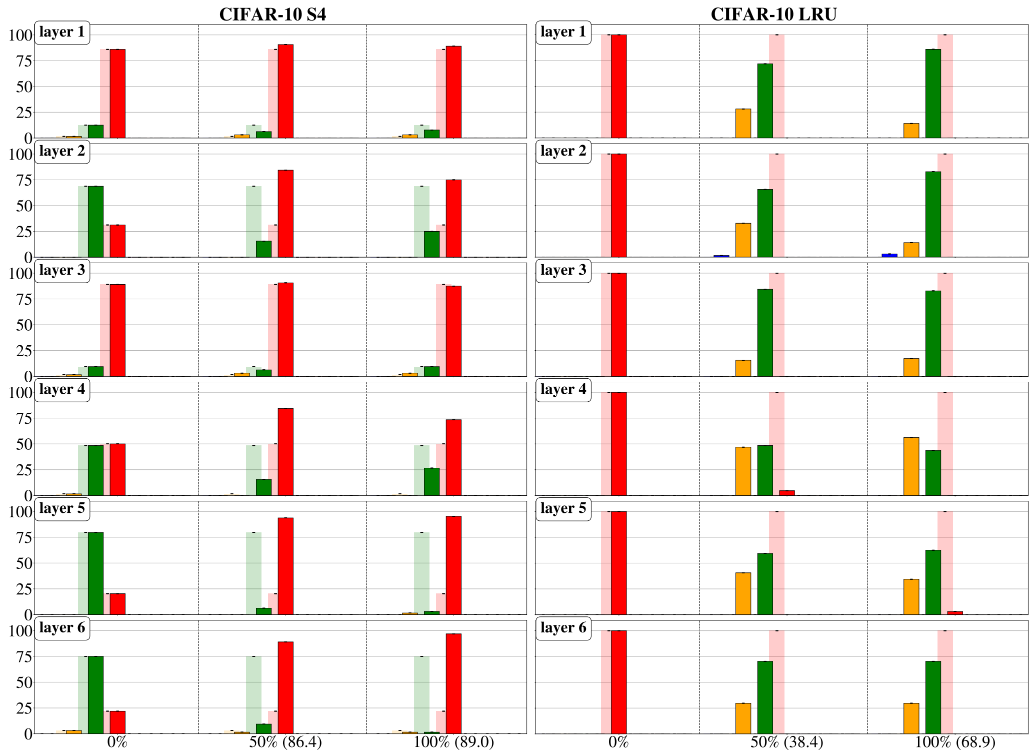


Figure 31: Correlation plots of model performance against the percentage of eigenvalues contained in the corresponding bin.



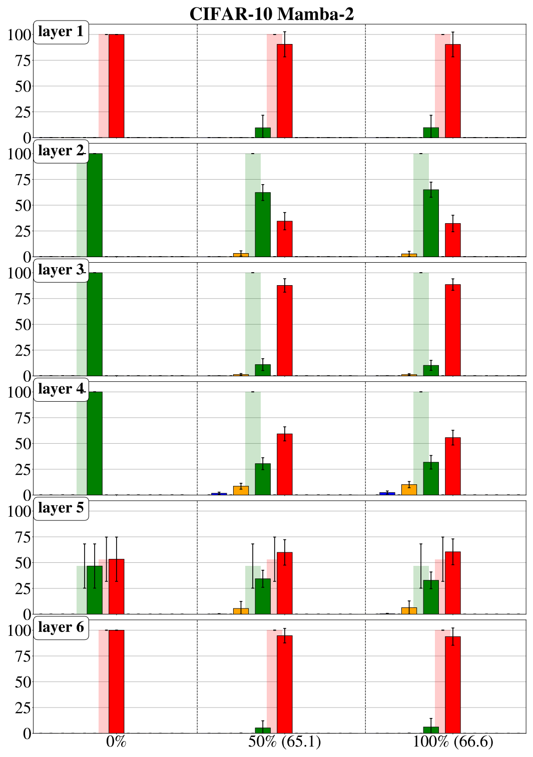
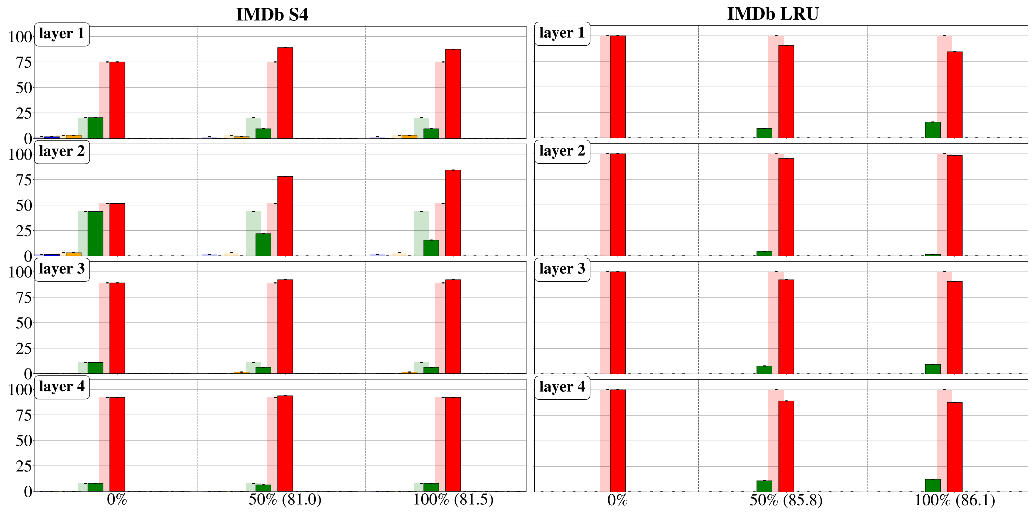


Figure 32: Evolution of eigenvalues from initialization, over training half-time, to the final distribution for CIFAR-10.



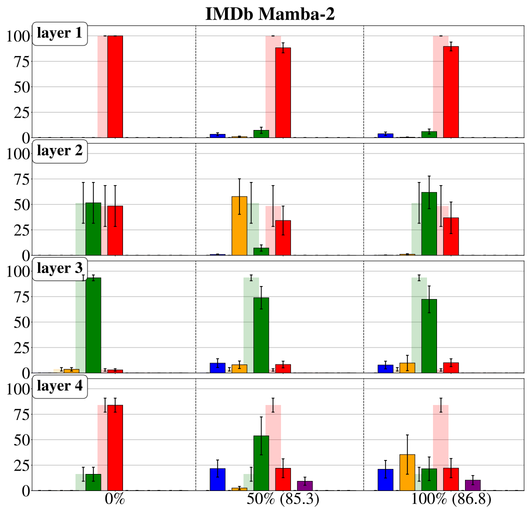


Figure 33: Evolution of eigenvalues from initialization, over training half-time, to the final distribution for IMDb.

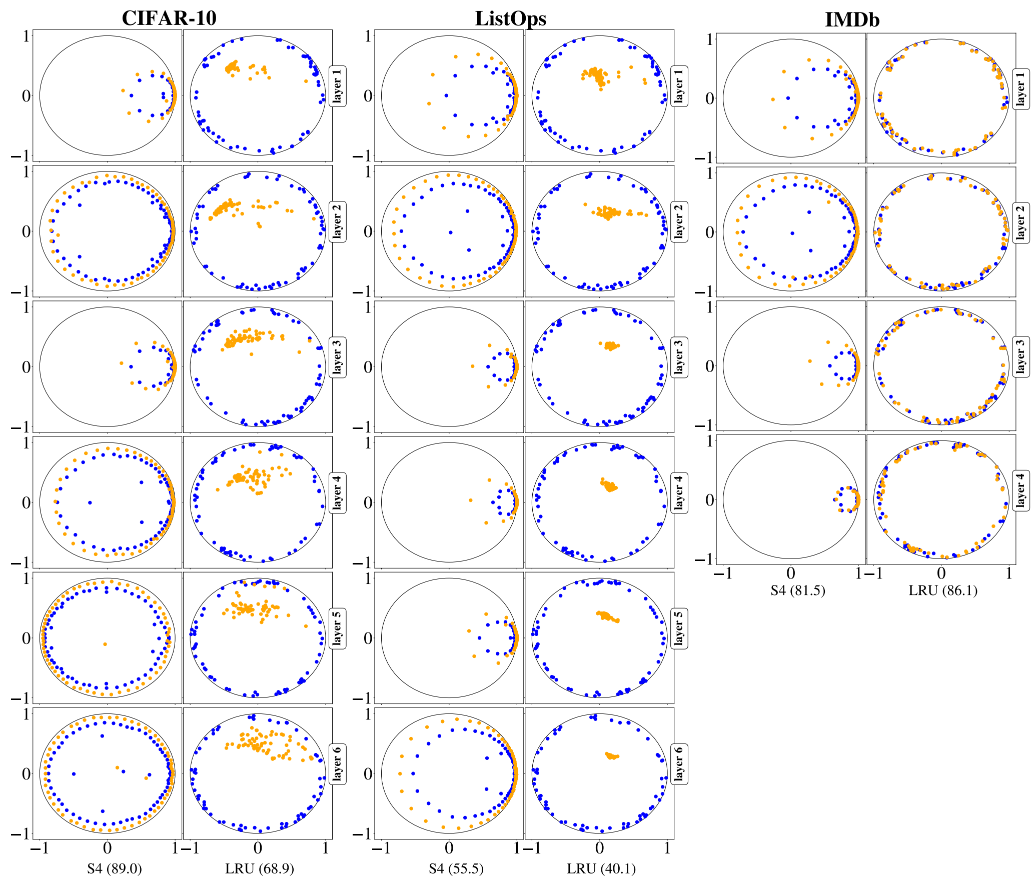


Figure 34: Visualization of complex eigenvalues across LRA task, SSMs, and all layers. The trained values are indicated in orange and the initialization in blue. While for LRU there appears to be a shift from initialization to the trained pattern, which is consistent across layers, this behaviour is not as pronounced in S4. There, conversely, eigenvalues remain spread around a circle with a specific radius. While the results are presented for completeness, they would require further investigation for more concrete statements.