UNDERSTANDING THE IMPORTANCE OF MASSIVE ACTIVATIONS THROUGH ADVERSARIAL ROBUSTNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

A common occurrence in Large Language Models (LLMs) is the emergence of so-called *massive activations*. This phenomenon is well known in the literature and has been connected to attention sinks, representational compression, as well as signal propagation. However, the reasons and consequences for their appearance are not yet well understood. In this paper, we explain the role of massive activations in deep learning by showing that they are connected to the *region migration* phenomenon in delayed robustness. Under the spline theoretic interpretation of deep networks, we show that massive activations push the representations away from the nonlinearity boundaries of the network, decreasing the complexity of its input-output map around data points. Furthermore, we uncover for the first time that this phenomenon is not unique to LLMs, and actually appears in a number of common architectures performing simple supervised learning tasks. We validate this connection through extensive experiments on a diverse set of architectures including residual MLP, ResNet18, Vision Transformer and GPT and datasets including MNIST, CIFAR10, Imagenette and Shakespeare Text. We consistently observe that massive activations and region migration co-occur with adversarial robustness. Furthermore, we illustrate that this connection holds in the large scale pretraining of LLMs, by showing that the raise of massive activations in LLMs closely tracks the adversarial accuracy of the model under perturbations of the token embeddings.

1 INTRODUCTION

Despite their remarkable success, the inner workings of neural networks remain poorly understood, giving rise to surprising phenomena that challenge our understanding of how they learn. One such phenomenon that recently gained traction in the context of LLMs are the so-called *massive activations* (Sun et al., 2024). These are a few activations in the hidden representation of neural networks that grow multiple orders of magnitude larger than the others. These activations rise in early layers and persist through the network via the residual stream, until disappearing near the output. Recent work has linked massive activations to attention sinks and pre-norm architectures Sun et al. (2026b), traced their formation to interactions between normalization and feed-forward layers Shi et al. (2026), and interpreted them as structural bias vectors Chen et al. (2026). Related work on activation outliers has also implicated normalization, signal propagation, and optimization in their emergence He et al. (2024). However, their broader functional role remains poorly understood.

This work is, to the best of our knowledge, the first to empirically and theoretically show the connection between massive activations, *delayed robustness* and *adversarial grokking* (Humayun et al., 2024), (the emergence of adversarial robustness (Szegedy et al., 2014; Goodfellow et al., 2015; Madry et al., 2018) long after generalization). Our central finding is that massive activations, emerging in the late phases of the training, simplify the local input-output map of neural networks, inducing the phenomenon called *region migration* (Humayun et al., 2024) in delayed robustness, which empirically coincides with increased adversarial robustness.

To study this connection in a principled manner we view neural networks through the lens of Spline Theory of Deep Networks (Balestriero et al., 2018; Balestriero & Baraniuk, 2020). Under this interpretation, a neural network partitions its input space into convex polytopal regions, and applies an affine mapping on each region. We show that massive activations displace the data points onto

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

large linear regions and away from nonlinearity boundaries. This displacement of representations is the *region migration* phenomenon observed in delayed robustness (Humayun et al., 2024).

We provide extensive experimental support for this connection, on a diverse mix of architectures, like MLPs (LeCun, 1998) with residual connections, ResNet18 (He et al., 2016), Vision Transformers (ViTs) (Dosovitskiy et al., 2021) and GPTs and datasets, like MNIST (LeCun, 1998), CIFAR10 (Krizhevsky, 2009), Imagenette (Howard, 2019) and Shakespeare Text (Karpathy, 2015). Across all the experiments we consistently observe that massive activations and region migration co-occur with adversarial robustness, providing strong empirical support for our proposed connection. Furthermore, we investigate this connection in LLMs and show that during the pretraining of LLMs the rise of massive activations closely tracks the onset of region migration and robustness to perturbations in the token embedding space.

We summarize our contributions as follows:

- We show that massive activations are more widespread in deep learning than previously known (they do not appear only in LLMs, as was originally found in Sun et al. (2024)).
- With the spline theoretic view of deep networks, we theoretically show that massive activations displace the representations away from nonlinearity boundaries, simplifying the neural network input-output map.
- We provide extensive empirical evidence showing that massive activations cause region migration and co-occur with adversarial robustness, providing a new perspective of the role of massive activations and why they are useful in deep learning.
- We demonstrate that our finding is also relevant for LLMs, by showing that the massive activations emerging during pretraining induce region migration and closely follow the robustness to perturbations of the token embeddings.

2 Preliminaries and Related Work

2.1 Massive Activations

Massive activations are a surprising empirical phenomenon, observed in the hidden representations of large neural networks (Sun et al., 2024). Sun et al. (2024) conducts a systematic study of these activations and report that massive activations have two defining characteristics: first, there are very few of them, typically fewer than ten in the hidden representation; second, they are a few orders of magnitude larger than the other activations.

Sun et al. (2024) observed that massive activations emerge as early as after the computation of the first layer in the network, remain roughly constant in value throughout most of the intermediate layers and diminish in the final few layers. Massive activations reside in a fixed location of the hidden representations, are typically input independent, and often occur in the beginning of a sequence, period or newline token, which do not have semantic meaning.

2.1.1 What are Massive Activations For?

Fixed Biases. Sun et al. (2024) show that massive activations are not redundant and act as fixed but important biases in the network. Notably, when they are set to zero, the model performance degrades significantly; on the other hand, setting massive activations to their mean value computed across different samples, causes negligible performance change.

Attention Sinks. (Sun et al., 2024; Queipo-de-Llano et al., 2026; Sun et al., 2026a) claimed that massive activations cause *attention sinks* (Xiao et al., 2024). The connection is due to the fact that massive activations appear in the attention key vectors of their associated tokens and the dot product between any query and these key vectors produces a large activation logit. As a result, the large activation logit dominates the softmax operation, since the softmax output must sum to one and nearly all the probability mass is allocated to the token with the massive activation.

Mitigating Over-mixing. Barbero et al. (2025) show that attention sinks, which are a consequence of massive activations on the first token (Sun et al., 2024; Queipo-de-Llano et al., 2026), are useful

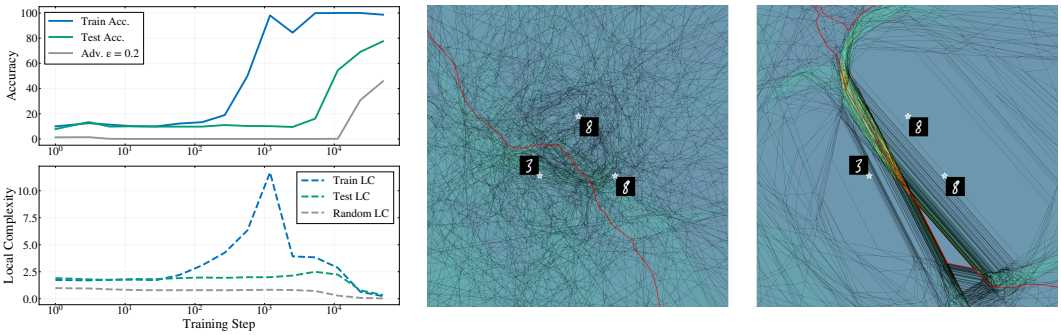


Figure 1: Residual MLP training on MNIST digit classification dataset. **Left:** train, test and adversarial accuracies along with the LC on train, test and random points. **Middle and Right:** SplineCam visualization of the input space partitioning of the residual MLP that passes through three data points. The back lines in in the SplineCam plots represent the nonlinearity boundaries of the model, where on each induced cell or region the neural network is an affine map. Stars represent the data points, red line is the decision boundary and colors of the regions correspond to the norm of their affine mappings. We provide more detailed explanations about SplineCam visualizatino in A.4

in transformers, because they prevent over-mixing of the representations and enable faithful information propagation through the layers, providing an additional interpretation of their role and why they might be useful in deep learning.

Compression Valleys. (Queipo-de-Llano et al., 2026) showed that massive activations compress the hidden representations, which is captured by the low entropy of the hidden representations in the middle layers. Massive activations cause the representation norm of a single token to dominate the others, creating a dominant singular value in the hidden representation matrix. The dominant singular value suppresses the contributions of all the tokens, causing the matrix based entropy of the representation to drop.

2.2 Delayed Robustness and Region Migration

Humayun et al. (2024) found a novel, previously unknown form of grokking (Power et al., 2022; Liu et al., 2023; Kumar et al., 2024), in which adversarial accuracy always exhibits the delayed transition, after the saturation of train and test accuracy, and termed this *delayed robustness*. Humayun et al. (2024) showed that across a wide range of architectures and training settings, deep networks become adversarially robust long after generalization. This means that there is a third phase beyond memorization and generalization where the network reorganizes its parameters to become adversarially robust to input perturbations, which (Humayun et al., 2024) measures with Projected Gradient Descent (PGD) attack (Madry et al., 2018) with different radius ϵ. This phenomenon has been established to be a general occurrence in deep learning and consistently observed in MLPs trained on MNIST (LeCun, 1998), CNNs (LeCun, 1998) and ResNets (He et al., 2016) trained on CIFAR10, CIFAR100 (Krizhevsky, 2009) and Imagenette (Howard, 2019) and GPT trained on Shakespeare Text (Karpathy, 2015).

2.2.1 Local Complexity

To track the emergence of delayed robustness, Humayun et al. (2024) propose a *local complexity* (LC) progress measure (Barak et al., 2022; Nanda et al., 2023), which quantifies the density of the nonlinearity boundaries around data points, that serves as a measure of local expressivity. More concretely, LC is computed by first constructing a small cross-polytopal frame around an input point $\boldsymbol{x}$. Let $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_P\}$ be P orthonormal directions, $r > 0$ be a neighborhood radius and $\mathcal{V}_{\boldsymbol{x}}^{(0)} = \{\boldsymbol{x} \pm r\boldsymbol{v}_p : p = 1, \ldots, P\}$ be the input cross-polytopal frame. The convex hull of this frame approximates a local neighborhood of the data point. The frame is constructed once in the input space and then propagated through the network layer by layer. At each layer, the metric asks how many neuron nonlinearity boundaries cut through the propagated neighborhood. Operationally, this is checked by evaluating each neuron's preactivation on the vertices of the propagated frame.

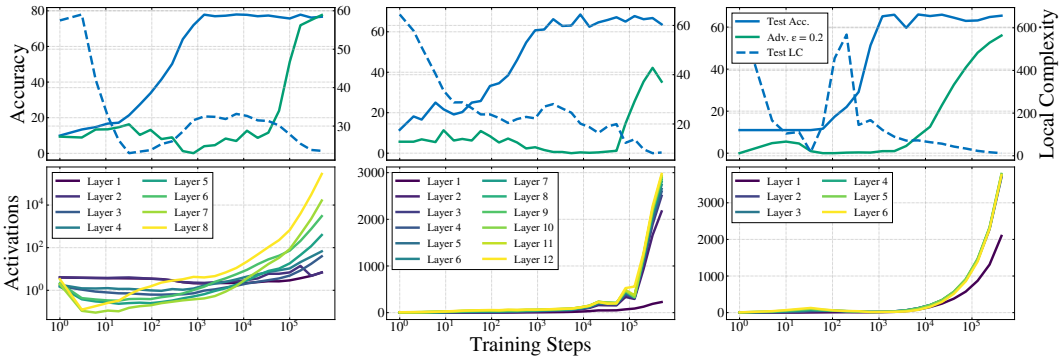


Figure 2: Joint emergence of massive activations, region migration and adversarial robustness in Resnet18 on CIFAR10 (left), ViT Small on Imagenette (middle) and GPT on Shakespeare Text (right). The top row shows the accuracy and adversarial accuracy on the test set along with the LC (on the mirrored axis) throughout the training. The bottom row shows the sum of the three largest activations in the representation across all the layers.

If different vertices fall on different sides of the same nonlinearity boundary, then the frame has crossed that boundary and the neuron contributes to the layer's LC.

The total LC is obtained by summing these boundary-crossing counts over all layers. Thus, LC should be understood geometrically as the number of nonlinear boundaries intersected by a small neighborhood of the data point as that neighborhood travels through the network. A large LC indicates that the neighborhood is cut by many activation boundaries and can move between many affine maps under small perturbations, while a small LC indicates that the propagated neighborhood lies inside larger affine regions.

Humayun et al. (2024) study the LC throughout the training of deep networks and finds that it undergoes a double descent (first decreases, then increases and then decreases). Humayun et al. (2024) shows that the network starts becoming robust against PGD perturbations during the second descent of LC, also called *region migration*, and this is when it reaches the *delayed robustness* state. We provide an illustration of delayed robustness for MLP with residual connections on MNIST dataset in Figure 1. Figure 1 (left) shows the train, test and adversarial accuracy on test set across steps along with the LC metric on train, test and random points. The adversarial accuracy increases with the decrease of the LC metric after generalization. Figure 1 (middle and right) shows SplineCam (Humayun et al., 2023) visualization of the input space partitioning of the network before region migration (middle), where the nonlinearity boundaries are dense around data points (marked with stars) and after region migration (right), where the nonlinearities migrated from data points.

3 Massive Activations Induce Region Migration

In this section, we claim that massive activations are the mechanistic cause for region migration in deep networks. The following sections detail this mechanistic connection along with extensive experimental support. Additionally, we show that this connection is not specific to any architecture or dataset and can be observed in ResNet18 on CIFAR10, ViT Small on Imagenette and GPT on Shakespeare Text as depicted in Figure 2. Additionally, Appendix D.1 provides further experiments that include residual MLP trained on MNIST and CIFAR10 and ViT Tiny on CIFAR10. All the experimental details and computation of the metrics are provided in Appendix C.

3.1 Early Emergence and Persistence Through Residual Stream

Massive activations typically emerge in early layers of the network depending on the architecture. For instance in ViT Small on Imagenette and GPT on Shakespeare Text massive activations emerge in early layers as show in Figure 2. On the other hand, for ResNet18 on CIFAR10, they do not have an exact emergence point, but gradually increase with each layer as depicted in Figure 2. Once they emerge, massive activations live in the representation from that point on and pass through the network via the residual stream. Figure 3, visualizes this mechanism by providing the massive activations in the hidden representations of residual MLP on MNIST (left), the CLS token repre-

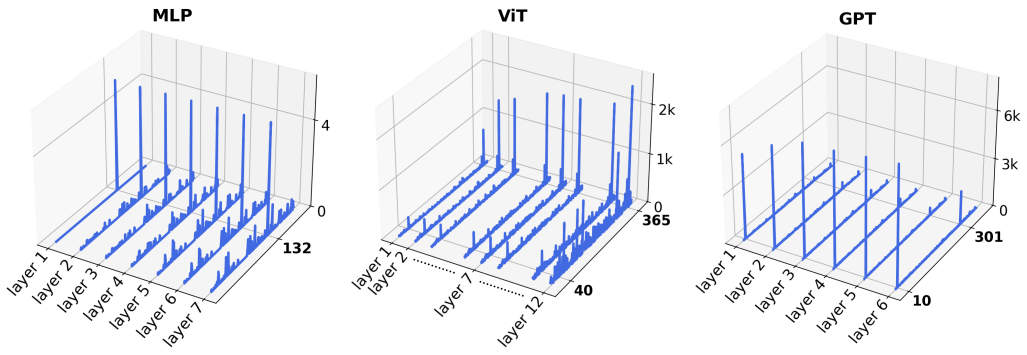


Figure 3: Visualization of massive activations in the hidden representations of residual MLP trained on MNIST (left), the CLS token representations of ViT Small trained on Imagenette (middle) and the first token representations of GPT trained on Shakespeare Text (right). Massive activations emerge as early as in the first layer and propagate through the network via the residual stream. In the case of ViT Small the massive activations amplify during the first few layers after the emergence.

sentation of the ViT Small on Imagenette (middle) and the first token representation of GPT on Shakespeare text (right) across the network layers. In Appendix D.4 we discuss the setting when residual connections are removed and show that massive activations emerge in early layers, however their propagation through the network largely disappears.

3.2 Theoretical Characterization of Region Migration

This section develops a theoretical characterization of region migration through massive activations. We first consider region migration within a single hidden layer. Let $\boldsymbol{W}^{(\ell)} = [\boldsymbol{w}_1^{(\ell)}, \dots, \boldsymbol{w}_{D^{(\ell)}}^{(\ell)}]$ and $\boldsymbol{b}^{(\ell)} = [b_1^{(\ell)}, \dots, b_{D^{(\ell)}}^{(\ell)}]$ be the parameters of layer ℓ and $\boldsymbol{x}^{(\ell-1)}$ be its input representation, where $D^{(\ell)}$ is the output dimensionality of layer ℓ. For clarity we assume a ReLU activation function in the network, however our results can be extended to any continuous affine spline activation function. Given this setup, in the input space partition of layer ℓ, the neuron i induces the following hyperplane $\mathcal{H}_i^{(\ell)} = \{\boldsymbol{z} : {\boldsymbol{w}_i^{(\ell)}}^\top \boldsymbol{z} + b_i^{(\ell)} = 0\}$. The LC of this layer is quantified by how many such hyperplanes lie close to the input representation $\boldsymbol{x}^{(\ell-1)}$. This dependence can be described through the distance from the representation to each neuron hyperplane given by

$$\operatorname{dist}(\boldsymbol{x}^{(\ell-1)}, \mathcal{H}_i^{(\ell)}) = \frac{|{\boldsymbol{w}_i^{(\ell)}}^\top \boldsymbol{x}^{(\ell-1)} + b_i^{(\ell)}|}{\|\boldsymbol{w}_i^{(\ell)}\|}. \tag{1}$$

Therefore, LC and region migration in a single layer can be characterised as the growth of these distances. The increase of these distances, would mean that representations move away from non-linearity boundaries and reside inside large regions.

Now suppose that the input representation $\boldsymbol{x}^{(\ell-1)}$ has large activations. These activations would increase the distance between the representation and the neuron hyperplanes, hence displacing them into large regions and decreasing LC. To formalize this mechanism, Proposition 3.1 derives the activation magnitude required to ensure any desired distance $\delta > 0$ from nonlinearity boundaries.

Proposition 3.1. *Let $0 < \eta < 1$. Assume that the massive activation at kth entry dominates the representation, $\|\boldsymbol{x}_{-k}^{(\ell-1)}\| < \eta |\boldsymbol{x}_k^{(\ell-1)}|$ and assume the neuron hyperplane parameters do not cancel out the dominance of the massive, $|[\boldsymbol{w}_i^{(\ell)}]_k| > \eta \|\boldsymbol{w}_{i,-k}^{(\ell)}\|$. Then $\boldsymbol{x}^{(\ell-1)}$ is more than $\delta > 0$ distance away from every hyperplane in the layer whenever*

$$|\boldsymbol{x}_k^{(\ell-1)}| > \max_{i=1,\dots,D^{(\ell)}} \frac{\delta \|\boldsymbol{w}_i^{(\ell)}\| + |b_i^{(\ell)}|}{|[\boldsymbol{w}_i^{(\ell)}]_k| - \eta \|\boldsymbol{w}_{i,-k}^{(\ell)}\|}. \tag{2}$$

Proof. Given in B.2. □

As massive activations emerge early in the network and travel through the residual stream, which supplies it to the inputs of the subsequent layers, they push the data points away from neuron hyperplanes in each layer, resulting in decreased LC and region migration in all layers, and hence for the whole network.

Theorem 3.2. *Let* $\alpha = \max_{\ell=1,\dots,L} \|\boldsymbol{W}^{(\ell)}\|$ *and* $0 < \eta < 1$. *Assume that the massive activation at entry* k *dominate the representation at every layer* $\|\boldsymbol{x}_{-k}^{(\ell-1)}\| < \eta|\boldsymbol{x}_{k}^{(\ell-1)}|$ *and every neuron hyperplane parameter in the network satisfies* $|[\boldsymbol{w}_i^{(\ell)}]_k| > \eta\|\boldsymbol{w}_{i,-k}^{(\ell)}\|$. *Then the propagated ball of radius* $\delta > 0$ *centered at* $\boldsymbol{x}^{(1)}$ *does not intersect any neuron hyperplane whenever*

$$\boldsymbol{x}_k^{(0)} > \max_{\ell,i} \frac{(1+\alpha)^{(\ell-1)}\delta\|\boldsymbol{w}_i^{(\ell)}\| + |b_i^{(\ell)}|}{|[\boldsymbol{w}_i^{(\ell)}]_k| - \eta\|\boldsymbol{w}_{i,-k}^{(\ell)}\|}$$

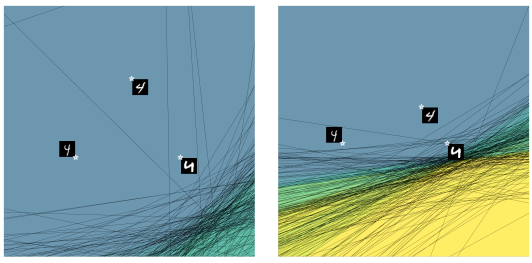
Proof. Given in B.2. □

Under the stated assumptions, Theorem 3.2 illustrates, that for a give distance $\delta > 0$ a large enough activation magnitude would ensure that the nonlinearity boundaries in the input space are further that δ from the input point. We claim that this is the mechanistic connection between massive activations and region migration. We provide additional discussion about the theoretical setup and the proofs in Appendix B.

3.3 Suppressing Massive Activations

To validate the causal link between massive activations and region migration, in this section we study the impact of suppressing the massive activations on the LC. If the causal claim is true, suppressing massive activations would undo region migration and increase LC.

Specifically, we take a fully trained residual MLP on MNIST dataset and visualizing the input space partition of the last layer with SplineCam (explained in Appendix A.4), as we suppress the dominant activation in its input. Figure 4 middle and right present the outcome of this experiment. The Figure shows that when the dominant activation is explicitly removed, region migration gets undone as depicted by the increased density of nonlinearity boundaries around data points and the number of regions in the partitioning. We provide additional example of the input space partitioning in MLPs, when the dominant activation is decreased and analysis of LC in ViT Tiny when the massives are explicitly removed from the representation in Appendix D.3. We conclude that these analysis provides supporting evidence about the causality between massive activations and region migration.



Number of Regions: 3083 Number of Regions: 4558

Figure 4: Removing the dominant activation from the input representation of the last layer of a 4 depth and 200 width residual MLP on MNIST undoes the region migration. **Left:** Without any changes, serving as a reference. **Right:** The dominant activation set to zero. This Figure follows the same format as Figure 1 (middle and right).

3.4 Which Tokens?

Figure 5 illustrates that in ViT Small and GPT the massive activation structure is fundamentally different. In ViT Small's hidden representations they reside on the CLS token embedding, while in GPT they are present across all tokens in the hidden representation. We believe that this is not a structural coincidence, but a reflection of what region migration requires for different architectures. Recall that ViTs utilize the CLS token representation for classification, and since the MLPs (where the nonlinearities reside) in the transformer block are applied pointwise to the tokens, having the massive activation in the CLS token is sufficient for region migration. In contrast GPT would require them in all token representations because each token representation is used to make a prediction on the subsequent token.

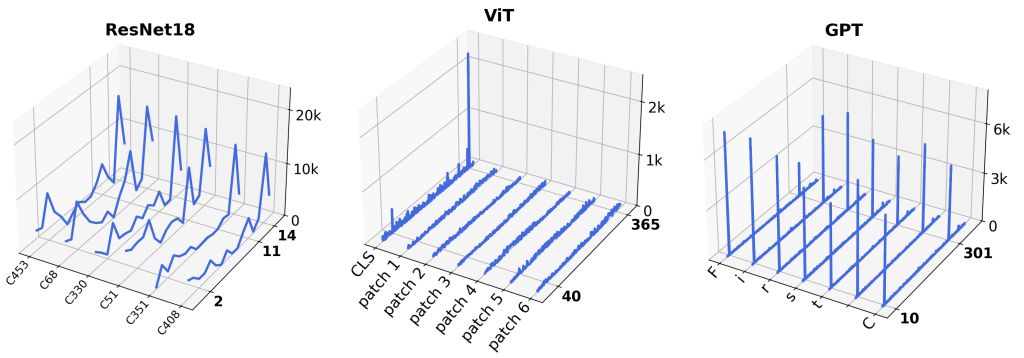


Figure 5: Visualization of the massive activations in the residual streams of ResNet18, ViT Small and GPT. **Left:** Activations of ResNet18 on CIFAR10, which contains the channels where the top 6 (by magnitude) massive activations reside. Importantly we note that the massive activations in ResNet18 are only present in a few of the channels (which are highlighted in the plot), and do not exist in most of them. **Middle:** Activations of ViT Small on Imagenette of the CLS token, followed by the activations of the next 6 patch tokens. **Right:** Activations of GPT on Shakespeare Text for the first 6 tokens in the sequence.

Key Finding: We find that massive activations are more widespread than previously known, by showing that they emerge in a wider range of deep learning architectures and not only in LLMs, where they have been initially observed (Sun et al., 2024). We illustrate that massive activations emerge in residual MLPs (Figure 9 left and middle), ResNets (Figure 2 left), ViTs (Figure 2 middle and Figure 9 right) and GPTs (Figure 2 right) and under the spline theoretic interpretation of deep learning they induce region migration and co-emerge with adversarial robustness.

4 Region Migration and Adversarial Robustness in LLMs

In Section 3 we established a connection between massive activations and adversarial robustness through region migration and provided extensive experimental evidence across diverse training settings. In this section we investigate if this connection holds in the context of LLMs, containing millions and often billions of parameters, trained on extensive datasets. Massive activations have initially been observed in LLMs (Sun et al., 2024), hence it is a natural question raising from our framework, whether or not the massive activations observed in LLMs are a sign of region migration and emergent robustness.

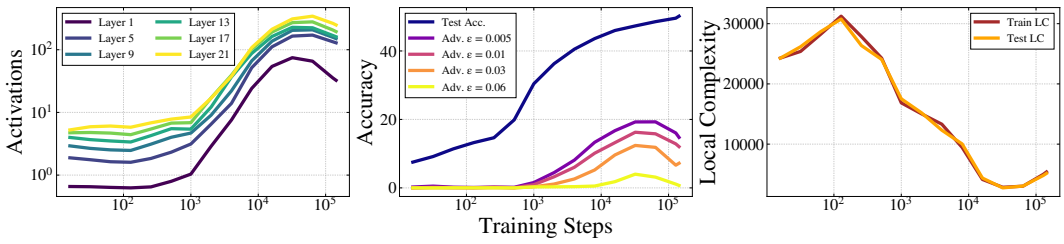


Figure 6: Evolution of top activations, adversarial accuracy and LC in Pythia 1.4B. Massive activations (left) co-occur with region migration, manifested by the decreasing LC (right) and robustness to adversarial perturbations (middle). The perturbations are constructed via the PGD attack on the token embeddings of the input.

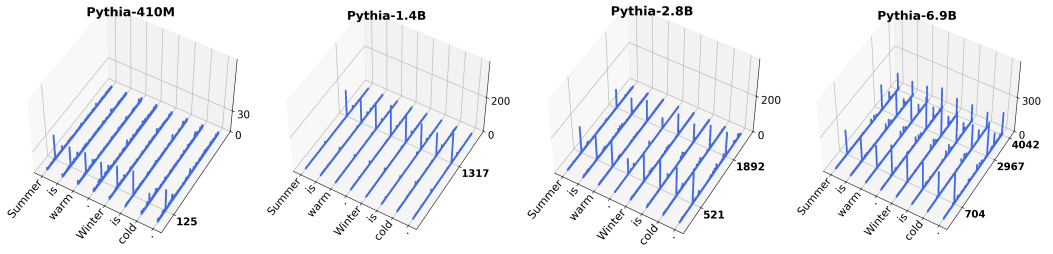


Figure 7: The activation magnitudes in the hidden representations of the Pythia suite across different model sizes.

4.1 Pythia Suite

We study the Pythia suite of LLMs (Biderman et al., 2023), a family of decoder only transformers pretrained on The Pile (Biderman et al., 2023) dataset. We use the checkpoints available throughout pretraining to monitor the emergence of massive activations and their relation to LC and adversarial robustness in the same dataset.

Experiment Setup. The sum of the largest three activations in the hidden token representation is recorded and averaged across tokens and samples across all layers in the network. We use the PGD attack in the token embedding space and measure the adversarial accuracy on the next-token prediction task on the perturbed embeddings, similar to the GPT on Shakespeare Text setting (discussed in Section 3). For PGD attack we use $\alpha = 0.001$ step size, $T = 10$ number of steps and $\epsilon = \{0.005, 0.01, 0.03, 0.06\}$ adversarial radius. LC is computed using $P = 25$ orthonormal directions and $r = 0.005$ neighborhood radius.

Results. Figure 6 presents the results of this experiment on Pythia 1.4B and illustrates that the emergence of massive activations (left) closely tracks the second descent of LC (right) and robustness to adversarial perturbation (middle). We observe the same trend across different model sizes in the Pythia suite. These results demonstrate that massive activations, originally observed in LLMs, are indeed connected to the emergence of adversarial robustness. In Figure 7, we provide the massive activations in the residual stream of the Pythia models across different model scales. Notably, all token representations possess massive activations, which is consistent with the discussion for GPT on Shakespeare Text setting detailed in Section 3.4. Appendix D.5 provides additional evidence for the emergence of per-token massive activations for Qwen 2.5 (Yang et al., 2024) and Gemma 3 (Kamath et al., 2025) pretrained model families.

4.2 Why Is Delayed Robustness Less Evident in LLMs?

In Figure 6, we observe that region migration occurs relatively earlier during training, compared to the experiments in Section 3, where adversarial accuracy starts increasing long after the train and test accuracies plateau. We hypothesise that this is due to the fact that the dataset size of the Pythia suite is significantly larger than in any of the training settings in Section 3 and it has been observed in Humayun et al. (2023) that this factor accelerates the occurrence of delayed robustness.

Key Finding: We find that massive activations that were observed in LLMs actually induce region migration and emerge alongside robustness to perturbations of the input sequence embeddings during pretraining.

5 Discussion and Future Work

In this work, we have uncovered a connection between massive activations and adversarial robustness in deep learning. We have shown that massive activations push the representations of the network away from the neuron hyperplanes, which define the input space partitioning boundaries, inducing region migration, which is connected to adversarial robustness. We demonstrated that this connection is not specific to any particular architecture or dataset configuration and can be observed

in residual MLPs, ResNet18, ViTs, GPTs, and can even be extended to LLMs pretrained on diverse text data.

Are Massive Activations Useful? The existing literature offers no concrete explanations for the role of massive activations in deep learning and whether they are useful or undesirable. Sun et al. (2024) showed that they act as fixed biases in the model and their removal results in significant performance degradation. Massive activations also induce attention sinks (Xiao et al., 2024), which have been argued to prevent over-mixing in the attention operation (Barbero et al., 2025). On the other hand, Xiao et al. (2023); Dettmers et al. (2022) show that large magnitude activations make quantising neural networks more challenging. Our work adds a new perspective to this discussion, suggesting that massive activations act as a mechanism to simplify the local input-output mapping of neural networks, as quantified by the LC progress measure, which empirically accompanies increased robustness.

Delayed Robustness in LLMs. We found that delayed robustness occurs in LLM pretraining, with the raise of massive activations. This finding shows that delayed robustness holds beyond the standard training settings, where it was originally discovered (Humayun et al., 2024), expanding its applicability to large-scale deep learning.

Massive Activations in LLMs. In Section 4 we showed that massive activations emerging in the Pythia suite induce region migration. Our framework suggests that per-token massive activations are associated with robust next token prediction, illustrated in Pythia, Qwen 2.5 and Gemma 3 model families. However, we highlight that Sun et al. (2024) found that not all model families have a per-token massive activation structure (i.e. LLaMA2 (Touvron et al., 2023)). We believe that follow-up research could investigate if these LLM families experience delayed robustness and their vulnerability to adversarial attacks. Additionally, we believe an interesting direction of future study is to develop pretraining prescriptions that better exploit delayed robustness and improve adversarial robustness.

6 Limitations

While this work sheds a new light on the role of massive activations and why they are useful in deep learning, we identify the following limitations. It is not clear what drives the emergence of massive activations in the training and why deep networks construct a small number of exceptionally large activations, rather than distributing it across many moderately large activations. The theoretical setup analysing the connection between massive activations and region migration is based on a fully connected neural network with residual connections and does not cover for operations like attention and normalization layers, which are commonplace in deep learning.

7 Conclusion

In this paper, we showed why massive activations (Sun et al., 2024) are useful in deep learning, via connecting them to delayed robustness (Humayun et al., 2024). Under the spline theoretic view of neural networks, we showed that massive activations reduce the network's LC, by displacing representations away from the network's nonlinearity boundaries, inducing region migration, which coincides with emergent robustness. We supported this claim through extensive experimental evidence and derived theoretical arguments that guarantee the simplification of the input space partitioning around data points if the activations in the representations are sufficiently large. We further showed that our findings are also relevant in the context of LLMs, by uncovering that the emergence of massive activations in LLMs closely tracks region migration and robustness to perturbations of the input token embeddings.

AI use statement

We declare that generative AI was used for polishing the writing, spotting and correcting grammatical errors and typos in the manuscript. Generative AI was used to identify relevant literature. It was used to help check the mathematical claims of the manuscript. It was used to write code for some of the visualization and experiments in the manuscript. All the outputs from the generative AI tools were reviewed and revised by the authors.

REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Randall Balestriero and Richard G Baraniuk. Mad max: Affine spline insights into deep learning. *Proceedings of the IEEE*, 109(5):704–727, 2020.

Randall Balestriero and Richard G Baraniuk. Batch normalization explained. *arXiv preprint arXiv:2209.14778*, 2022.

Randall Balestriero, Romain Cosentino, Behnaam Aazhang, and Richard Baraniuk. The geometry of deep networks: Power diagram subdivision. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/0801b20e08c3242125d512808cd74302-Paper.pdf.

Randall Balestriero, Ahmed Imtiaz Humayun, and Richard G Baraniuk. On the geometry of deep learning. *Notices of the American Mathematical Society*, 72(4), 2025.

Randall Balestriero et al. A spline theory of deep learning. In *international conference on machine learning*, pp. 374–383. PMLR, 2018.

Boaz Barak, Benjamin L. Edelman, Surbhi Goel, Sham M. Kakade, eran malach, and Cyril Zhang. Hidden progress in deep learning: SGD learns parities near the computational limit. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=8XWP2ewX-im.

Federico Barbero, Alvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Petar Veličković, Razvan Pascanu, and Michael M. Bronstein. Why do LLMs attend to the first token? In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=tu4dFUsW5z.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Yung-Chin Chen, Chung Peng Lee, Ze-Wei Liou, and Naveen Verma. Massive spikes in llms are bias vectors: Mechanistic uncovering and spike-free quantization. *arXiv preprint arXiv:2606.02288*, 2026.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=dXiGWqBoxaD.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6572.

Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

Lauren A Hannah and David B Dunson. Multivariate convex regression with adaptive partitioning. *The Journal of Machine Learning Research*, 14(1):3261–3294, 2013.

Bobby He, Lorenzo Noci, Daniele Paliotta, Imanol Schlag, and Thomas Hofmann. Understanding and minimising outlier features in transformer training. *Advances in Neural Information Processing Systems*, 37:83786–83846, 2024.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Jeremy Howard. Imagenette. https://github.com/fastai/imagenette, 2019.

Ahmed Imtiaz Humayun, Randall Balestriero, Guha Balakrishnan, and Richard G. Baraniuk. Splinecam: Exact visualization and characterization of deep network geometry and decision boundaries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3789–3798, June 2023.

Ahmed Imtiaz Humayun, Randall Balestriero, and Richard Baraniuk. Deep networks always grok and here is why. In *High-dimensional Learning Dynamics 2024: The Emergence of Structure and Reasoning*, 2024. URL https://openreview.net/forum?id=NpufNsg1FP.

Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ram'e, Morgane Rivière, et al. Gemma 3 technical report. *ArXiv*, 2025. URL https://arxiv.org/abs/2503.19786.

Andrej Karpathy. char-rnn. https://github.com/karpathy/char-rnn, 2015.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. URL https://arxiv.org/abs/1412.6980.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Tanishq Kumar, Blake Bordelon, Samuel J. Gershman, and Cengiz Pehlevan. Grokking as the transition from lazy to rich training dynamics. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vt5mnLVIVo.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Ziming Liu, Eric J Michaud, and Max Tegmark. Omnigrok: Grokking beyond algorithmic data. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=zDiHoIWa0q1.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Alessandro Magnani and Stephen P Boyd. Convex piecewise-linear fitting. *Optimization and Engineering*, 10(1):1–17, 2009.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9XFSbDPmdW.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

Enrique Queipo-de-Llano, Alvaro Arroyo, Federico Barbero, Xiaowen Dong, Michael M. Bronstein, Yann LeCun, and Ravid Shwartz-Ziv. Attention sinks and compression valleys in LLMs are two sides of the same coin. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=c5TFhCJ6fs.

Zeru Shi, Zhenting Wang, Fan Yang, Qifan Wang, and Ruixiang Tang. A single layer to explain them all: Understanding massive activations in large language models. *arXiv preprint arXiv:2605.08504*, 2026.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. Massive activations in large language models. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=F7aAhfitX6.

Shangwen Sun, Alfredo Canziani, Yann LeCun, and Jiachen Zhu. Anatomy of massive activations and attention sinks. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=VUiMZBfHBe.

Shangwen Sun, Alfredo Canziani, Yann LeCun, and Jiachen Zhu. The spike, the sparse and the sink: Anatomy of massive activations and attention sinks. *arXiv preprint arXiv:2603.05498*, 2026b.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6199.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengpeng Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint*, 2024.

Biao Zhang and Rico Sennrich. Root Mean Square Layer Normalization. In *Advances in Neural Information Processing Systems 32*, Vancouver, Canada, 2019. URL https://openreview.net/references/pdf?id=S1qBAf6rr.

A Review of Spline Theory of Deep Networks

This section provides the preliminaries of the Spline Theory of Deep Networks (Balestriero et al., 2018; Balestriero & Baraniuk, 2020), which builds a mathematical bridge between neural networks and splines, offering a powerful gateway to analyse their inner workings (Balestriero et al., 2025; Balestriero & Baraniuk, 2022; Humayun et al., 2024). Section A.1 introduces splines, with the emphasis on the max affine splines, which will serve as the building blocks to establish the connection to neural networks. Section A.2 presents the precise connection between the composition of max affine spline operators and deep networks and Section A.3 presents the geometric properties of deep networks composed of splines.

A.1 SPLINES

This section presents the terminology and definitions for splines. Specifically, we will discuss affine splines (also called linear or first order splines). The following discussion defines affine splines and max affine splines in the multivariate input-output setting.

A.1.1 AFFINE SPLINES

We start by defining multivariate *affine spline functions*.

Definition A.1 (Multivariate Affine Spline Functions)**.** *Let $\Omega = \{\omega_1, \ldots, \omega_R\}$ be a partition of a domain $\mathbb{R}^D$ and let $\Phi = \{\phi_1, \ldots, \phi_R\}$ be mappings from $\mathbb{R}^D$ to $\mathbb{R}$ for each partition ω_r via $\phi_r(\boldsymbol{x}) = \alpha_r^\top \boldsymbol{x} + \beta_r$, where $\boldsymbol{x} \in \omega_r$ and $\alpha \in \mathbb{R}^{R \times D}$ and $\beta \in \mathbb{R}^R$ are the parameters of the affine mappings. Under this setup an affine spline function is defined by*

$$s[\alpha, \beta, \Omega](\boldsymbol{x}) = \sum_{r=1}^{R} (\alpha_r^\top \boldsymbol{x} + \beta_r) \mathbf{1}(\boldsymbol{x} \in \omega_r) \tag{3}$$

where $\mathbf{1}(\boldsymbol{x} \in \omega_r)$ is the indicator function.

When approximating functions with splines, a substantial challenge is the need to jointly optimize for the partitioning of the input space $\mathbb{R}^D$ and for the affine mapping parameters α and β (Balestriero et al., 2018). Next section discusses how to resolve the issue of joint optimization, by enforcing convexity on the affine spline functions.

A.1.2 MAX AFFINE SPLINES

A simplification of the joint optimization arises when affine spline functions are constrained to be convex, in which case the function can always be written as *max affine spline* (Balestriero et al., 2018; Hannah & Dunson, 2013; Magnani & Boyd, 2009).

Definition A.2 (Max affine spline functions)**.** *Let $\alpha \in \mathbb{R}^{R \times D}$ be a matrix of hyperplane "slopes" and $\beta \in \mathbb{R}^R$ be a vector of "offsets", then a max affine spline function is defined as*

$$s_{\alpha,\beta,\Omega}(\boldsymbol{x}) = \max_{r=1,\cdots,R} \alpha_r^\top \boldsymbol{x} + \beta_r \tag{4}$$

A useful property of max affine spline functions is that they are completely determined by their affine mapping parameters α and β, and the selection of the partition Ω is automatically determined by the choices of α and β (Balestriero et al., 2018). Hence a max affine spline function can be denoted as $s_{\alpha,\beta}$, as Ω is inferred by α and β. This provides a resolution for the joint optimization of the partition and the affine mapping parameters discussed in the previous section.

Next we present *max affine spline operators* (MASOs), which are a natural extension of max affine splines into the multivariate outputs, by concatenating K independent max affine spline functions.

Definition A.3 (Max Affine Spline Operators)**.** *Let $A \in \mathbb{R}^{K \times R \times D}$ and $B \in \mathbb{R}^{K \times R}$ be the parameters, then max affine spline operator is defined as*

$$S_{A,B}(\boldsymbol{x}) = \begin{bmatrix} \max_{r=1,\cdots,R} A_{1,r}^\top \boldsymbol{x} + B_{1,r} \\ \vdots \\ \max_{r=1,\cdots,R} A_{K,r}^\top \boldsymbol{x} + B_{K,r} \end{bmatrix} \tag{5}$$

A MASO is always a piecewise affine and convex with respect to each output dimension (Balestriero & Baraniuk, 2020).

A.2 DEEP NETWORKS AS COMPOSITIONS OF MASOS

In this section we state the connection between neural networks and spline theory. The central result in (Balestriero et al., 2018; Balestriero & Baraniuk, 2020) is that a broad range of layers used in neural networks can be represented as MASOs, and hence a deep neural network is a composition of MASOs.

A.2.1 Fully Connected Layers and Activation Functions

Proposition A.4 (Fully Connected/Convolutional Layers)**.** *Any fully connected or convolutional layer is an affine mapping and hence a MASO with $R = 1$.*

A fully connected layer with parameters $\boldsymbol{W}$ and $\boldsymbol{b}$ is represented by $\boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{b}$, which is a single affine mapping, that is it is equivalent to $S_{\boldsymbol{W},\boldsymbol{b}}(\boldsymbol{x})$ with $R = 1$. This reasoning naturally extends to the convolutional layer.

Proposition A.5 (Activation Functions)**.** *Any piecewise affine and convex activation function (such as ReLU, Leaky ReLU or Absolute Value) is a MASO.*

To exemplify the intuition behind this result, consider the ReLU activation $\text{ReLU}(\boldsymbol{x}) = \max\{\boldsymbol{x}, 0\}$, that is applied elementwise to $\boldsymbol{x}$. The output at kth position is $\max\{\mathbf{e}_k^\top \boldsymbol{x}, 0\}$, which is a max affine spline with $R = 2$, with $A_{k,1} = 0$, $A_{k,2} = \mathbf{e}^k$ and $B_{k,1} = B_{k,2} = 0$, where $\mathbf{e}_k$ is the kth canonical basis vector. The mapping for Leaky ReLU follows the same reasoning, except $A_{k,1} = v\mathbf{e}_k$, where v is the negative slope of the Leaky ReLU activation function. For absolute value $A_{k,1} = -\mathbf{e}_k$ and $A_{k,2} = \mathbf{e}_k$.

A.2.2 Deep Networks

This section presents the main result of (Balestriero & Baraniuk, 2020) connecting deep networks to splines.

Proposition A.6. *A deep network layer constructed from any composition of fully connected or convolutional layers followed by a single activation operator is a MASO.*

Proposition A.6 shows that a deep network layer composed of a fully connected or convolutional layer followed by an activation function is a MASO. Subsequently the following theorem establishes that a deep network is a composition of MASOs.

Theorem A.7. *A deep neural network composed of fully connected/convolutional layers, activation functions (defined according to A.5) is a composition of MASOs.*

We note that while a layer of a network is a MASO, as illustrated by Propositions A.4, A.5 and A.6, a deep network, that is, a composition of MASOs, is not a MASO in general and this is what introduces the non-convexity and the expressive power of neural networks (Balestriero et al., 2018; Balestriero & Baraniuk, 2020). We do not provide the proofs for these theoretical results in this project for brevity and refer the reader to Balestriero & Baraniuk (2020) for the detailed derivations.

General Activation Functions. While the central building block in the Spline Theory is the MASO, which covers activation functions like ReLU, LeakyReLU and the absolute value, it does not naturally extend to smooth activation functions like sigmoid. However, since any continuous function can be approximated with arbitrary precision via affine splines, smooth activation functions can be approximated via MASO, then the spline theoretic view of deep networks would approximately hold Balestriero et al. (2018).

Extension to General Neural Networks. In addition to the discussed connection between splines and neural networks, Balestriero & Baraniuk (2020) shows that various operations that are commonplace in deep learning, such as residual connections and pooling layers, can also be expressed as splines. On top of that Balestriero & Baraniuk (2020) presents the detailed derivations for the neural network to spline mapping for convolutional neural networks (CNNs) (LeCun, 1998), ResNets (He et al., 2016), and recurrent neural networks (RNNs) (Graves, 2013), illustrating that spline theory perspective is applicable in a wide range of architectures.

A.3 Input Space Geometry

A direct consequence of the max affine spline structure of deep networks is its specific geometric structure. As discussed in previous sections, each MASO layer partitions its input space and applies a region dependent affine mapping on it. (Balestriero et al., 2019) provides a comprehensive analysis about the geometry of the partition, revealing that it has a polytopal structure.

Theorem A.8. *Let f be a MASO layer, then it partitions its input space into a finite collection of convex polytopes.*

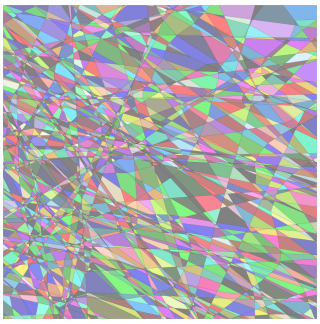


Figure 8: Output of a SplineCam visualization. The input space partitioning on a 2D slice of the input space from a 4 layer ReLU MLP. On each region the network is an affine mapping. Colors are randomly selected.

Theorem A.8 shows that a MASO layer or a network layer divides its input space into convex polytopal regions. A composition of L MASO layers produces an increasingly subdivided partition of the input space, where each next layers further divides the partition induced by the previous layers (Balestriero et al., 2018). Next we provide the result about the geometry of the input space partitioning of a deep network.

Theorem A.9 (Deep Network Input Space Partitioning)**.** *Let $f = f^{(L)} \circ \cdots \circ f^{(1)}$ be a composition of L MASO layers, then the input space partitioning consists of finite convex polytopes given by $\{\omega_r\}_r$ and moreover on each cell ω_r the deep network is an affine mapping*

$$f(\boldsymbol{x}) = \sum_{r=1}^{R} (A_{\omega_r}^{\top} \boldsymbol{x} + B_{\omega_r}) \mathbf{1}(\boldsymbol{x} \in \omega_r) \tag{6}$$

Theorem A.9 states that an affine spline network partitions its input space into a finite number of convex polytopal regions, where within each polytopal region, the network is an affine map. That is let $\boldsymbol{x}$ be in a partition $\omega_{\boldsymbol{x}}$, then the network input-output map reduces to

$$f(\boldsymbol{x}) = A_{\omega_{\boldsymbol{x}}}^{\top} \boldsymbol{x} + B_{\omega_{\boldsymbol{x}}} \tag{7}$$

where $A_{\omega_{\boldsymbol{x}}}$ and $B_{\omega_{\boldsymbol{x}}}$ are the affine mapping parameters for partition $\omega_{\boldsymbol{x}}$. Figure 8 provides a visualization of the input space polytopal partitioning on a toy 4 layer ReLU multilayer perceptron. We do not provide the proofs of Theorems A.9 and A.7, as they have extensive derivations and refer the reader to Balestriero & Baraniuk (2020) for the details.

An important consequence of Theorem A.9 concerns the Jacobian of the network.

Remark A.10. *The Jacobian of a deep network f on region $\omega_{\boldsymbol{x}}$ is given by $J_{\boldsymbol{x}}(f) = A_{\omega_{\boldsymbol{x}}}$.*

A.4 SplineCam

The spline theoretic framing of deep networks provides insights into the geometry of the input space partition, however computing and visualizing these partitions is not trivial. SplineCam (Humayun et al., 2023) was introduced to solve this problem, which is an exact and sampling free method for computing the geometry of deep network input space partitioning.

SplineCam is applicable for any affine spline network (as defined in Section A.2). Since the neural network input space is high dimensional and cannot be easily visualized, Humayun et al. (2023) proposes to visualize the partition on a two dimensional slice of the input space. The output of a SplineCam visualization is a 2D image, where each contiguous line represents a neuron hyperplane boundary and each region enclosed by the lines is a partition region where the network input-output map is an affine transformation. An example of a SplineCam visualization of a 4 layer ReLU MLP is provided in Figure 8.

We note that the original implementation of the SplineCam did not support models with residual connections and for this project we implemented this support in order to be able to visualize residual MLPs as we will see in the upcoming sections. The code for the SplineCam visualization tool along with our added support of the residual is made available with the source code of this project.

B Region Migration Guarantees

Section 3 provided mechanistic description with extensive experimental support of how massive activations emerge, propagate in the network and cause region migration. In this section we develop a theoretical understanding of region migration induced by massive activations to further support our main claim. We show that, large enough massive activations in the input representations of the network layer guarantee low LC.

B.1 Setup and Assumptions

The aim of our analysis is to study the behaviour of the region migration in deep networks under massive activations. We study a fully connected neural network with residual connections with activation ReLU activation function (which is consistent with the definition in Proposition A.5) for ease of analysis. Specifically, we consider the following network

$$\boldsymbol{x}^{(\ell)} = \text{ReLU}\left(\boldsymbol{W}^{(\ell)}\boldsymbol{x}^{(\ell-1)} + \boldsymbol{b}^{(\ell)}\right) + \boldsymbol{x}^{(\ell-1)}, \quad \ell = 1, \ldots, L \tag{8}$$

where $\boldsymbol{x}^{(0)}$ is the input, $\boldsymbol{W}^{(\ell)} = [\boldsymbol{w}_1^{(\ell)}, \ldots, \boldsymbol{w}_{D^{(\ell)}}^{(\ell)}]$, where $D^{(\ell)}$ is the width of the network.

Quantifying Region Migration

To measure region migration and the complexity of the network around data points, we quantify the distance of the representations from the nonlinearities of the network. For a ReLU-like piecewise affine function, the nonlinearity occurs at the point where the pre-activations cross zero. In particular for a neuron i in a layer ℓ the nonlinearity occurs when $\boldsymbol{h}_i^{(\ell)} = \boldsymbol{w}_i^{(\ell)\top}\boldsymbol{x}^{(\ell-1)} + \boldsymbol{b}^{(\ell)} = 0$, hence we define the neuron hyperplane

$$\mathcal{H}_i^{(\ell)} = \left\{\mathbf{z} : \boldsymbol{w}_i^{(\ell)\top}\mathbf{z} + \boldsymbol{b}^{(\ell)} = 0\right\} \tag{9}$$

As we are interested in computing distances of the representations from such hyperplanes to assess how far is the representation from changing the preactivation sign and falling into another affine mapping, which is the definition of LC, Remark B.1 highlights the formula of the distance of a point from a hyperplane.

Remark B.1. *Let $\boldsymbol{w}^\top\boldsymbol{x}+b = 0$, be a hyperplane, then the distance of a point $\boldsymbol{x}_0$ from the hyperplane is given by $\frac{|\boldsymbol{w}^\top\boldsymbol{x}_0+b|}{\|\boldsymbol{w}\|}$.*

Hence, the distance of a representation $\boldsymbol{x}^{(\ell)}$ at layer ℓ from the nonlinearity boundaries $\mathcal{H}_i^{(\ell)}$ for neuron i is given by

$$\text{dist}(\boldsymbol{x}^{(\ell-1)}, \mathcal{H}_i^{(\ell)}) = \frac{|\boldsymbol{w}_i^{\ell\top}\boldsymbol{x}^{(\ell-1)} + \boldsymbol{b}^{(\ell)}|}{\|\boldsymbol{w}_i^{(\ell)}\|} \tag{10}$$

To measure the LC of the whole network we need to quantify distance for all neurons in all layers in the network as the input $\boldsymbol{x}$ propagates through the network. If the distances for all the nonlinearity boundaries are larger, when the representations are located deep inside the linear regions of the network, resulting in a lower LC, while if the distances are smaller, the representations are at the edge of changing the preactivations and crossing a nonlinearity boundary, resulting in a higher LC.

B.2 Bounds for Activation Magnitudes

Building on the discussion from previous sections, this section shows that massive activations if large enough guarantee low LC of the network around the data points. To satisfy this condition, we should be able to show that for any desirable distance $\delta > 0$, there exists a massive activation magnitude such that all the distances between the representations and nonlinearity boundaries given in equation 10 are larger than $\delta > 0$. Proposition 3.1 provides the derivation for the activation magnitude required to migrate the regions in a single layer and Theorem 3.2 extends the layerwise bound to the full network.

Proposition 3.1. *Let $0 < \eta < 1$. Assume that the massive activation at kth entry dominates the representation, $\|\boldsymbol{x}_{-k}^{(\ell-1)}\| < \eta|\boldsymbol{x}_k^{(\ell-1)}|$ and assume the neuron hyperplane parameters do not cancel*

out the dominance of the massive, $|[\boldsymbol{w}_i^{(\ell)}]_k| > \eta\|\boldsymbol{w}_{i,-k}^{(\ell)}\|$*. Then* $\boldsymbol{x}^{(\ell-1)}$ *is more than* $\delta > 0$ *distance away from every hyperplane in the layer whenever*

$$|\boldsymbol{x}_k^{(\ell-1)}| > \max_{i=1,\dots,D^{(\ell)}} \frac{\delta\|\boldsymbol{w}_i^{(\ell)}\| + |b_i^{(\ell)}|}{|[\boldsymbol{w}_i^{(\ell)}]_k| - \eta\|\boldsymbol{w}_{i,-k}^{(\ell)}\|}. \tag{2}$$

Proof. For any neuron i in layer ℓ, decompose its preactivation as

$$\boldsymbol{w}_i^{(\ell)\top}\boldsymbol{x}^{(\ell-1)} + b_i^{(\ell)} = [\boldsymbol{w}_i^{(\ell)}]_k \boldsymbol{x}_k^{(\ell-1)} + \boldsymbol{w}_{i,-k}^{(\ell)\top}\boldsymbol{x}_{-k}^{(\ell-1)} + b_i^{(\ell)}. \tag{11}$$

By the reverse triangle and Cauchy–Schwarz inequalities,

$$\left|\boldsymbol{w}_i^{(\ell)\top}\boldsymbol{x}^{(\ell-1)} + b_i^{(\ell)}\right| \geq |[\boldsymbol{w}_i^{(\ell)}]_k||\boldsymbol{x}_k^{(\ell-1)}| - \|\boldsymbol{w}_{i,-k}^{(\ell)}\|\|\boldsymbol{x}_{-k}^{(\ell-1)}\| - |b_i^{(\ell)}| \tag{12}$$

$$> |\boldsymbol{x}_k^{(\ell-1)}|\left(|[\boldsymbol{w}_i^{(\ell)}]_k| - \eta\|\boldsymbol{w}_{i,-k}^{(\ell)}\|\right) - |b_i^{(\ell)}|. \tag{13}$$

The weight assumption makes the term in parentheses positive. Therefore, the assumed lower bound on $|\boldsymbol{x}_k^{(\ell-1)}|$ gives

$$\left|\boldsymbol{w}_i^{(\ell)\top}\boldsymbol{x}^{(\ell-1)} + b_i^{(\ell)}\right| > \delta\|\boldsymbol{w}_i^{(\ell)}\|. \tag{14}$$

Dividing by $\|\boldsymbol{w}_i^{(\ell)}\|$ shows that

$$\operatorname{dist}\left(\boldsymbol{x}^{(\ell-1)}, \mathcal{H}_i^{(\ell)}\right) > \delta. \tag{15}$$

Since the lower bound is maximized over all neurons, this holds for every hyperplane in the layer. □

Theorem 3.2. *Let* $\alpha = \max_{\ell=1,\dots,L}\|\boldsymbol{W}^{(\ell)}\|$ *and* $0 < \eta < 1$*. Assume that the massive activation at entry* k *dominate the representation at every layer* $\|\boldsymbol{x}_{-k}^{(\ell-1)}\| < \eta|\boldsymbol{x}_k^{(\ell-1)}|$ *and every neuron hyperplane parameter in the network satisfies* $|[\boldsymbol{w}_i^{(\ell)}]_k| > \eta\|\boldsymbol{w}_{i,-k}^{(\ell)}\|$*. Then the propagated ball of radius* $\delta > 0$ *centered at* $\boldsymbol{x}^{(1)}$ *does not intersect any neuron hyperplane whenever*

$$\boldsymbol{x}_k^{(0)} > \max_{\ell,i} \frac{(1+\alpha)^{(\ell-1)}\delta\|\boldsymbol{w}_i^{(\ell)}\| + |b_i^{(\ell)}|}{|[\boldsymbol{w}_i^{(\ell)}]_k| - \eta\|\boldsymbol{w}_{i,-k}^{(\ell)}\|}$$

Proof. Consider any point $\widetilde{\boldsymbol{x}}^{(1)}$ satisfying

$$\|\widetilde{\boldsymbol{x}}^{(1)} - \boldsymbol{x}^{(1)}\| \leq \delta, \tag{16}$$

and propagate it through the same network. Since ReLU is 1-Lipschitz, every residual layer satisfies

$$\|\widetilde{\boldsymbol{x}}^{(\ell)} - \boldsymbol{x}^{(\ell)}\| \leq \|\widetilde{\boldsymbol{x}}^{(\ell-1)} - \boldsymbol{x}^{(\ell-1)}\| + \|\boldsymbol{W}^{(\ell)}(\widetilde{\boldsymbol{x}}^{(\ell-1)} - \boldsymbol{x}^{(\ell-1)})\| \tag{17}$$

$$\leq (1+\alpha)\|\widetilde{\boldsymbol{x}}^{(\ell-1)} - \boldsymbol{x}^{(\ell-1)}\|. \tag{18}$$

Applying this recursively gives

$$\|\widetilde{\boldsymbol{x}}^{(\ell-1)} - \boldsymbol{x}^{(\ell-1)}\| \leq (1+\alpha)^{\ell-1}\delta. \tag{19}$$

The massive activation also persists through the residual stream. Indeed, because ReLU is nonnegative and $\boldsymbol{x}_k^{(0)} > 0$,

$$\boldsymbol{x}_k^{(\ell)} \geq \boldsymbol{x}_k^{(\ell-1)}. \tag{20}$$

Hence $\boldsymbol{x}_k^{(\ell-1)} \geq \boldsymbol{x}_k^{(0)}$ at every layer.

The lower bound on $\boldsymbol{x}_k^{(0)}$ and Proposition 3.1, applied with distance $(1+\alpha)^{\ell-1}\delta$, therefore give

$$\operatorname{dist}\left(\boldsymbol{x}^{(\ell-1)}, \mathcal{H}_i^{(\ell)}\right) > (1+\alpha)^{\ell-1}\delta \tag{21}$$

for every neuron i in every layer ℓ. Meanwhile, every propagated point $\widetilde{\boldsymbol{x}}^{(\ell-1)}$ lies within this distance of $\boldsymbol{x}^{(\ell-1)}$. It therefore cannot belong to $\mathcal{H}_i^{(\ell)}$. Since the initial point $\widetilde{\boldsymbol{x}}^{(0)}$ was arbitrary, the propagated ball intersects no neuron hyperplane in the network. □

These results show that, with sufficiently large activations, the layer representations lie at least distance δ away from the neuron hyperplanes. This provides a sufficient condition for low LC.

C Experimental Details

C.1 Training Details

This section details the training setup for the experiments. The following sections provide information about the optimizer and its parameters, loss functions, the setting and parameters used for measuring adversarial robustness and LC. We use the `PyTorch` deep learning library (Paszke et al., 2019) for all the experiments in this project.

Optimizer and Weight Decay. Following Humayun et al. (2024), we use the Adam optimizer (Kingma & Ba, 2015) across all the experiments. Liu et al. (2023) claims that the delayed generalization or grokking in the network can be accelerated with weight decay, however Humayun et al. (2024) found that in delayed robustness weight decay has the opposite effect and postpones the LC double descent phenomenon. Hence, we choose weight decay consistent with Humayun et al. (2024) and unless specified otherwise use a weight decay of zero for all the experiments, except for the MLP architecture, where the weight decay is set to 0.01. We use a learning rate of 10^{-2} for MLPs, ResNets and GPTs and 3×10^{-4} for ViTs. For MLPs and ResNets we use a batch size of 256 and for ViTs and GPTs 64 across all the experiments.

Loss Functions. Following the experimental setup in Humayun et al. (2024), we use the mean squared error loss function for training our residual MLPs and the cross-entropy loss function for ResNets, ViTs and GPTs. No label smoothing is used for any of the experiments.

C.2 Metrics

Adversarial Accuracy. Following the adversarial evaluation setup in Humayun et al. (2024) computation we quantify adversarial accuracy with PGD attack (Madry et al., 2018). Adversarial accuracy is defined as the fraction of the data points for which the network maintains its correct prediction under the PGD attack. Unless specified otherwise we follow the hyperparameter choices made in Humayun et al. (2024) and use $T = 10$ steps, $\alpha = 0.01$ step size, $\epsilon = 0.2$ adversarial radius and a single random start for the PGD attack. For the MLP, ResNet18 and ViT the adversarial attack is run on the input and for the GPT the attack is run on the input token embeddings.

Local Complexity. Following the LC metric hyperparameter choices in (Humayun et al., 2024) the cross-polytopal frame $\boldsymbol{V_x}$ is constructed with $P = 25$ orthonormal vectors and radius $r = 0.1$ for MLPs, $r = 0.001$ for ResNets and $r = 0.005$ for ViTs and GPTs across all the experiments. The LC at each layer is computed by counting the number of intersections of the embedded frame with the nonlinearity boundaries, and the total LC obtained by summing across the layers is reported.

Massive Activations Magnitudes. We track the sum of the top-3 activations in the residual stream across all the layers for all architectures. For ViTs we track the massive activations on the CLS token only, while for GPT the activations are tracked in all token representations and the averaged value is reported.

D Additional Experiments

D.1 Massive Activations and Delayed Robustness

We provide additional experiments on the emergence of massive activations and delayed robustness in residual MLP on MNIST and CIFAR10 and ViT Tiny on CIFAR10 illustrated in Figure 9, where region migration and adversarial robustness emerges alongside massive activations.

D.2 Massive Activation Patterns

Different classes exhibit different massive activation patterns, while samples from the same class have activations in the same positions as depicted in Figure 10. This indicates that the network sends different classes into separate parts of the space, that are free of nonlinearities. As a result, the

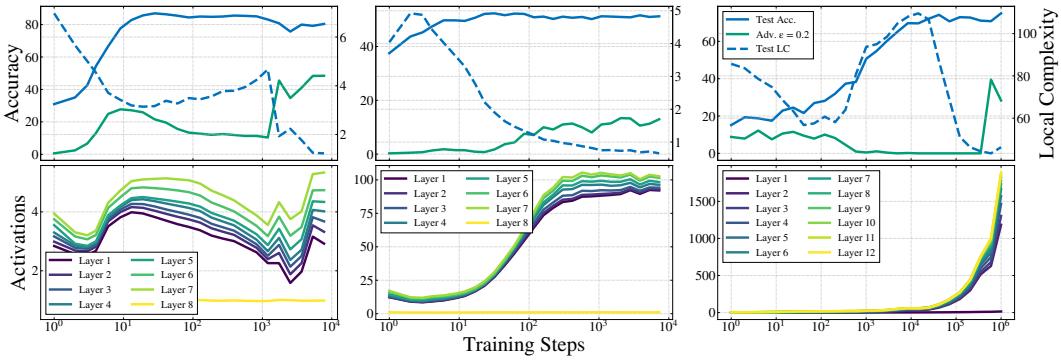


Figure 9: Emergence of the massive activations, region migration and adversarial robustness, in MLP on MNIST (left), MLP on CIFAR10 (middle) and ViT Tiny on Imagenette (right). Plots follow the same format as Figure 5.

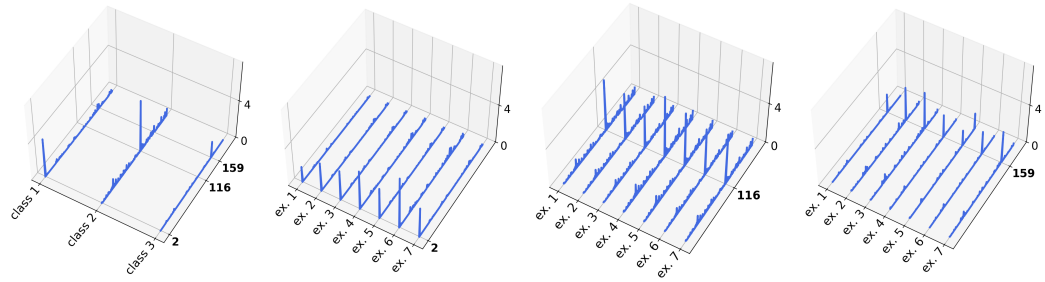


Figure 10: The activations have different patterns for different classes and activate on the same positions for samples within the same class. This figure represents the activation patterns prior to entering the last layer for the 4 layer residual MLP with width 200 trained MNIST dataset. **Left** figure shows that the activation patterns are different from three distinct classes in MNIST. **Middle-left**, **middle-right** and **right** figures show the activation patterns of seven examples from the first, second and third classes outlined on the left plot respectively, showing that the indices where the large activations appear are consistent.

nonlinearity boundaries accumulate in the part of the space, which separates the different classes, consistent with the migration towards decision boundary observation in (Humayun et al., 2024) as illustrated in Figure 1 (right).

D.3 Suppressing Massive Activations Undoes Region Migration

This section supplements Section 3.3, providing more examples of the input space geometry when suppressing the dominant activations. Figure 15 show the SplineCam visualizations of an MLP on MNIST for different classes, as the dominant activations are manually suppressed. We consistently observe that the lowering in these activations pushes the input representations back to the regions where nonlinearity boundaries are densely accumulated. Importantly, the consequences get more severe with more suppression of the dominant activations as exemplified in Figure 15.

Additionally, we conduct similar analysis on ViT Tiny trained on CIFAR10, by manually removing the massive activation from the representation. Figure 11 illustrates that the more massive activation is suppressed in the representations, the higher LC gets, supporting our proposed causal connection.

D.4 Massive Activations in Non-Residual MLPs

This section studies the massive activation dynamics in MLP without residual connections. We use 8 layer and 200 width MLP and train it on MNIST and track the accuracy, adversarial accuracy, LC and activation magnitudes according to the methodology described in Appendix C. Figure 12 shows the result of the experiment, where massive activations emerge in the first layer however they

slowly fade away in the following layers, confirming that residual stream is the primary enabler for the propagation of massive activations through the network.

D.5 Massive Activations in Qwen 2.5 and Gemma 3

Figure 13 and 14 show the massive activations in the hidden representations of Qwen 2.5 and Gemma 3 family of models across different sizes and similar to the Pythia suite they reside in each token representation. Additionally we observe that the activation magnitudes increase with the size of the model for each model family, which matches the trend in the Pythia suite, presented in Figure 7.

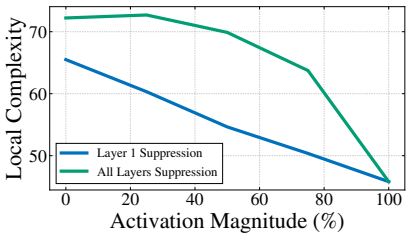


Figure 11: The LC of ViT Tiny trained on CIFAR10 when the massive activation is suppressed immediately after the emergence in the first layer (blue) and when it is suppressed after the emergence in every subsequent layer (green). We observe that the more suppression of the activation magnitude yields a more increased LC.

E Further Discussion

E.1 Massive Activation Magnitudes

We highlight that massive activation magnitudes differ in different architectures. Figure 5 shows that they are much larger in ResNet18 ($\approx$ 15k) compared to ViT Small ($\approx$ 2k) and GPT ($\approx$ 6k). Additionally, we notice this trend in the Pythia, Qwen 2.5 and Gemma 3 LLMs as depicted in Figures 7, 13 and 14, where the magnitudes of the largest activations increase with the scale of the model. We conjecture that these magnitudes have a dependence on the architectural configurations of the network, such as the width and depth. This intuition is also consistent with the theoretical analysis that is presented in Section B, where the required massive activation magnitudes depend on quantities of the layer weight norm, which scales with the width of the network. We believe that these questions can be explored in future works with more extensive empirical and theoretical developments.

E.2 The Impact of Normalization Layers

This section provides a discussion on the effect of the normalization layers on the massive activations, as they are commonplace in deep learning, such as Layer Normalization (Ba et al., 2016) and Root Mean Squared Normalization (Zhang & Sennrich, 2019).

Layer Normalization performs the following operation on each entry of a d dimensional input $\boldsymbol{x}$

$$\gamma \frac{\boldsymbol{x}_i - \mu}{\sqrt{\sigma^2}} + \beta \tag{22}$$

where $\mu = \frac{1}{d}\sum \boldsymbol{x}_i$ and $\sigma^2 = \frac{1}{d}\sum \boldsymbol{x}_i^2 - \left[\frac{1}{d}\sum \boldsymbol{x}_i\right]^2$ are the sample mean and variance of the representation entries respectively, while γ and β are learnable parameters. While the normalization operation divides the input signal by its standard deviation, which may suppress the dominant activations, we empirically observe that it does not remove their dominance from the representation and the flow in the network as described in Section 3.

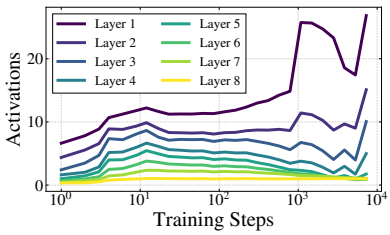


Figure 12: MLP without residual connections on MNIST. Massive activations emerge in the first layer, but gradually fade away with each layer unlike in residual MLP, where they are propagated through the layers via the residual stream.

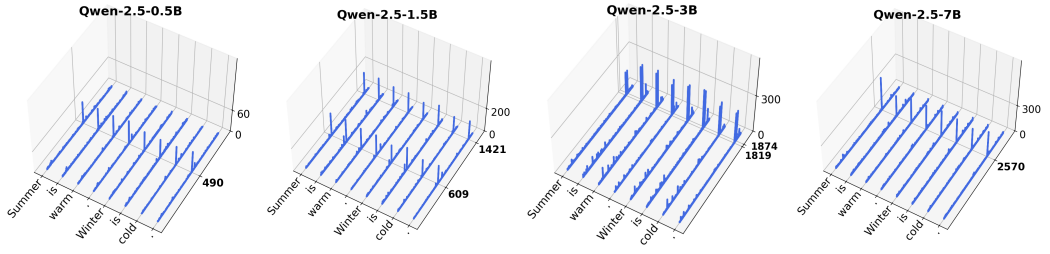


Figure 13: Massive activations in Qwen 2.5 family of pretrained models across different sizes.

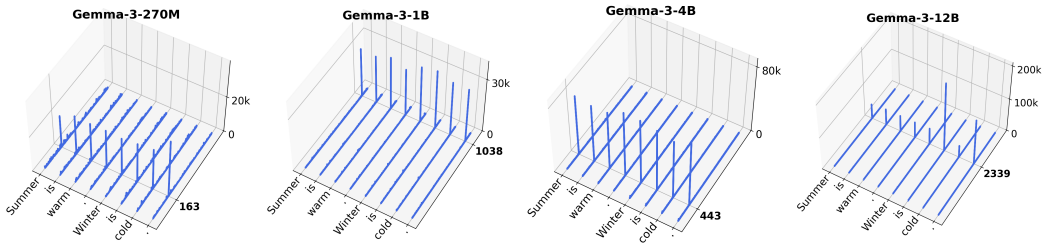


Figure 14: Massive activations in Gemma 3 family of pretrained models across different sizes.

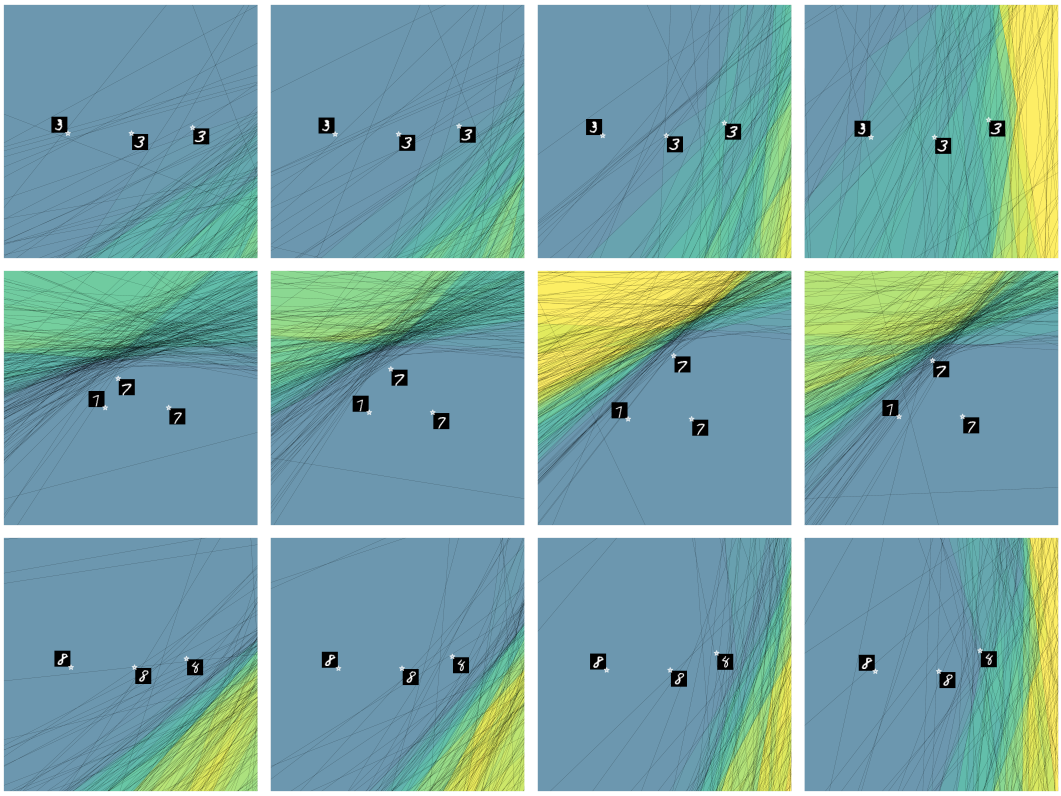


Figure 15: SplineCam visualization of the input space partitioning of the last layer of a depth 4 and 200 width residual MLP trained on MNIST as the dominant activations in the input are suppressed from left to right. **Left** represents the partitioning without any input modifications, serving as a reference, **right** with the dominant activation set to zero, **middle left** and **middle right** with the dominant activation set to 66% and 33% of its original value respectively.