Hidden Dynamics of Massive Activations in Transformer Training

Jorge Gallego-Feliciano[1, +], S. Aaron McClendon[1, +], Juan Morinelli[1], Stavros Zervoudakis[2], and Antonios Saravanos[2, *]

[1]Aimpoint Digital Labs, Atlanta, GA, USA
[2]New York University, New York, NY, USA
[+]these authors contributed equally to this work
[*]please direct correspondence to: Dr. Antonios Saravanos (saravanos@nyu.edu)

ABSTRACT

Massive activations are scalar values in transformer hidden states that achieve values orders of magnitude larger than typical activations and have been shown to be critical for model functionality. While prior work has characterized these phenomena in fully trained models, the temporal dynamics of their emergence during training remain poorly understood. We present the first comprehensive analysis of massive activation development throughout transformer training, using the Pythia model family as our testbed. Through systematic analysis of various model sizes across multiple training checkpoints, we demonstrate that massive activation emergence follows predictable mathematical patterns that can be accurately modeled using an exponentially-modulated logarithmic function with five key parameters. We develop a machine learning framework to predict these mathematical parameters from architectural specifications alone, achieving high accuracy for steady-state behavior and moderate accuracy for emergence timing and magnitude. These findings enable architects to predict and potentially control key aspects of massive activation emergence through design choices, with significant implications for model stability, training cycle length, interpretability, and optimization. Our findings demonstrate that the emergence of massive activations is governed by model design and can be anticipated, and potentially controlled, before training begins.

Introduction

Transformers have become the dominant architecture for large-scale language models, in no small part due to their powerful decoder-only implementations for generative tasks [1, 2]. A decoder-only transformer consists of a stack of blocks, with self-attention and feed-forward components, as well as residual connections and normalization operations. Each transformer layer processes a d-dimensional hidden state (the "residual stream"), applying self-attention and feed-forward operations, with residual connections and normalization to support stable, deep information flow [3]. Self-attention contextualizes each token's representation, while feed-forward networks apply position-wise transformations—together, these build complex, high-dimensional features that are crucial for model predictions. An interesting phenomenon in these models is the emergence of massive activations. These are individual neuron activations in transformer architectures [4] that dramatically exceed typical magnitudes within their layer—often by factors of 1,000 to 10,000 times the median activation value [5].

Unlike normal activations that vary with input content, these extreme values remain essentially constant across different inputs and act as implicit bias terms that concentrate self-attention onto particular tokens. These activations have significant practical implications for quantization [6, 7], inference optimization [7, 8, 9], and training stability [10]. Previous work has demonstrated that removing massive activations can lead to model failure, while setting massive activations to their mean values preserves model functionality [5], and augmenting certain high impact activations can encourage Chain-of-Thought (CoT) reasoning capabilities without the use of Reinforcement Learning [11]. In [12, 13], authors perform an extensive root cause analysis to trace the origin of MAs to particular architecture components. However, fundamental questions about their emergence during training remain unanswered.

Recognizing these challenges, recent research has proposed a variety of architectural and algorithmic techniques to mitigate or manage the effects of massive activations. For example, alternatives to the softmax function in the attention mechanism—such as Softpick and softmax-1—have been introduced to prevent or lessen the occurrence of extreme activations [14, 15]. Other work has explored training-time interventions like MacDrop (Massive Weights Curriculum Dropout), which applies progressively decreasing dropout to pre-trained massive weights, encouraging the model to rely on a broader set of parameters. This helps reduce overdependence on sparse components and improves generalization and robustness, particularly in low-data or parameter-efficient settings [16]. Simple modifications to the attention mechanism, including clipped softmax and gated attention, have also been shown to effectively suppress extreme activations, enabling models to be quantized to full

INT8 precision without loss in floating-point accuracy [17]. Outlier management techniques, such as categorizing activation outliers into "normal" and "massive" and redistributing them using methods like DuQuant (which employs channel-wise rotations and permutations), have further improved quantization performance [18].

The placement of normalization layers has been identified as another key factor in controlling massive activations. Standard Pre-LayerNorm architectures allow unchecked growth of hidden-state variance, which can destabilize training. Newer approaches, such as normalizing both before and after each sub-layer (Peri-LayerNorm), prevent this runaway growth and lead to more stable activation variances and gradients [19]. More broadly, recent analyses across a range of transformer architectures suggest that not all massive activations are detrimental, and that combining multiple mitigation strategies can balance the suppression of harmful outliers with the preservation of downstream performance [20]. In the domain of Vision Transformers, it has been observed that high-norm "artifact" tokens—activations of unusually large magnitude in background regions—can emerge and distort attention patterns [21, 22]. Introducing a few trainable "register" tokens to the input sequence allows the model to assign these high activations to dedicated locations, thereby smoothing feature and attention maps and improving performance.

Despite these advances, most existing interventions remain fundamentally reactive, addressing the symptoms of massive activations only after they have emerged, rather than predicting or proactively controlling their origins. Yet understanding the underlying dynamics of massive activations is not just of academic interest, rather a mechanistic understanding of massive activation dynamics is central in illuminating internal representation formation [23], information propagation, and implicit biases in deep networks at scale. Practically, predicting or controlling massive activations could inform training diagnostics, guide architectures optimized for quantization [8, 24, 25, 26], and enable principled approaches to model regularization and interpretability. While previous research has deepened understanding of massive activations in fully trained transformer models, much less is known about their temporal dynamics during training. Most studies focus exclusively on final checkpoints, leaving open questions about when massive activations first appear, how they evolve across layers and training stages, and whether their properties can be anticipated from model architecture.

To address these gaps, this work systematically examines the development of massive activations across training using the EleutherAI Pythia model suite [2], a collection of 16 decoder-only transformers ranging from 14 million to 12 billion parameters. The availability of over 150 training checkpoints per model, together with consistent data and architectural controls, enables a detailed investigation of when and how massive activations arise, how their trajectories depend on model scale, and the extent to which they can be predicted from architectural parameters such as depth, hidden size, and attention head count. The following sections provide a quantitative characterization of these trajectories, introduce a unified mathematical framework for modeling massive activation emergence, and discuss the practical implications for transformer design, interpretability, and optimization.

Preliminary

This section establishes the mathematical framework and key definitions necessary for analyzing massive activation dynamics during transformer training. We focus on decoder-only transformer architectures, as exemplified by the Pythia model family studied in this work.

Transformer architecture and hidden states

We consider decoder-only transformer models composed of L residual blocks. Each layer $\ell \in L$ receives a hidden state $h_{\ell-1} \in \mathbb{R}^{S\times d}$ and produces an updated hidden state:

$$h_\ell = h_{\ell-1} + \mathscr{F}_\ell(h_{\ell-1}) \tag{1}$$

where $\mathscr{F}_\ell$ includes both multi-head self-attention and MLP submodules. Throughout this paper, we denote by h_ℓ the post-residual hidden state, i.e., the output after the residual summation. As in [5], we do not consider intermediate computations within $\mathscr{F}_\ell$ unless explicitly stated.

An *activation* refers to a specific scalar element of a hidden state tensor h_ℓ. For a model processing a sequence of S tokens with d hidden dimensions, each layer's output $h_\ell \in \mathbb{R}^{S\times d}$ contains $S \cdot d$ scalar activations. In this work, we focus exclusively on the scalar values in h_ℓ, rather than weights, attention logits, or intermediate MLP states.

Massive activations

Following the definition introduced in [5], we refer to certain rare, abnormally large activations as *massive activations* (MAs). They propose a loose rule of thumb, and consider a scalar activation $a \in h_\ell$ to be massive if:

$$|a| > 100 \quad \text{and} \quad \frac{|a|}{\text{median}(|h_\ell|)} \geq 1000. \tag{2}$$

These activations have been observed to occur consistently at a small set of fixed feature dimensions and are often associated with the initial token or delimiter tokens in the input sequence (e.g., "." or "\n"). While small in number, they are disproportionately large—often exceeding the median activation by four or more orders of magnitude—and have been shown to function as implicit bias terms in the model's computation.

The original definition in [5], which includes a hard threshold of $|a| > 100$, does not generalize well to smaller models. For instance, in Pythia-14M (Figure 2), the top activation magnitudes at layer 3 clearly dominate all others in the model, exhibiting the characteristic sharp spike associated with massive activations. Yet their absolute values remain well below 100, and their ratio to the median is also far less than 10^3.

Despite this, the behavioral pattern is qualitatively similar to that observed in larger models: a small number of activations attain disproportionately high values, persist across tokens and inputs, and concentrate in specific feature dimensions. To better capture this effect across model scales, we relax the definition and focus instead on large-magnitude outliers that dominate their respective layers, echoing conclusions reached independently in [12].

Specifically, we define a *massive activation candidate* as one of the top-k activations in a hidden state that exceeds the layer's median activation by a significant margin. In practice, we identify massive activation candidates as the top activations in each layer that satisfy:

$$\frac{|a|}{\text{median}(|h_\ell|)} > \text{threshold} \tag{3}$$

where the threshold is typically set to 50 for our analysis, although this value can vary based on model size. This relaxed definition allows us to study the emergence patterns of disproportionately large activations across the full range of Pythia model sizes while maintaining the core conceptual framework established in prior work.

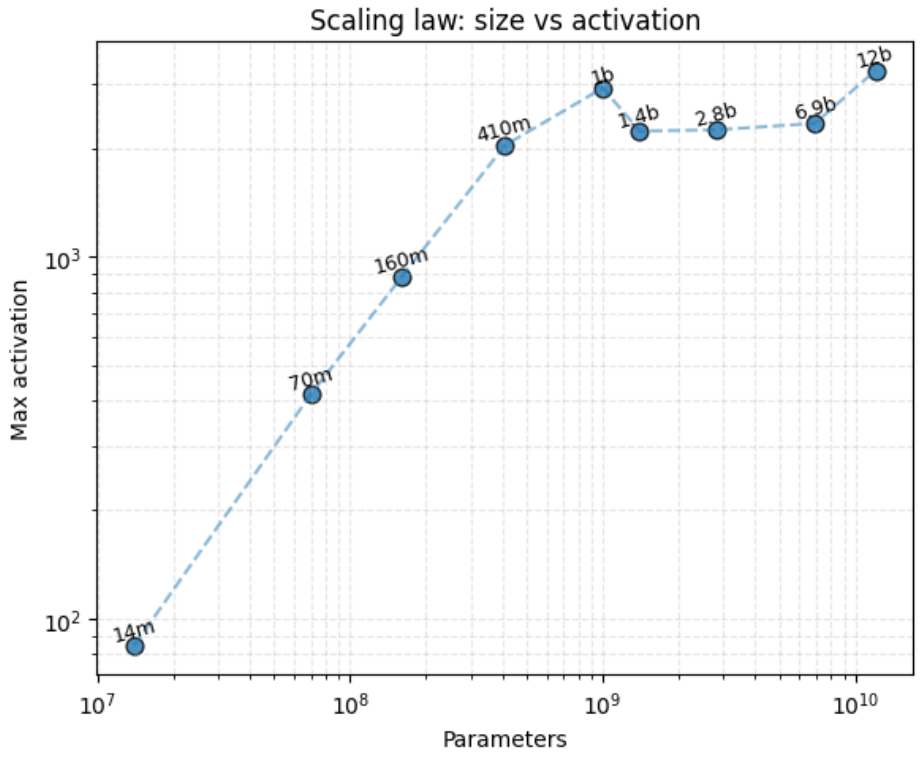


Figure 1. Plot of transformer parameter count vs value of the top activation to median ratio per model, in each respective final model checkpoint.

Figure 1 illustrates the scaling relationship between model size and the maximum observed activation magnitude (averaged across samples in X). We observe a steep rise in activation magnitudes from 14M to 1B parameters, followed by a plateau and a secondary increase beyond 6B. This suggests that massive activations emerge gradually with scale and stabilize in prevalence or intensity past a certain model size.

We study a transformer, along with its training checkpoints, denoted as M_t, for $0 \leq t \leq T$, where each M_t is a transformer model with the GPT-NeoX architecture [2]. The index t represents the training step, ranging from 0 to 143,000. EleutherAI released 154 checkpoints at regular intervals: multiples of 1000 steps for the full training duration, with additional higher-resolution checkpoints at powers of 2 up to step 512 for detailed analysis of early training dynamics. Each model M_t consists of L decoder layers. Passing an input sequence x through M_t yields a series of hidden states:

$$h_\ell(M_t, x) \in \mathbb{R}^{S \times d} \tag{4}$$

where $\ell \in \{1, \ldots, L\}$ is the layer index, S is the sequence length, and d is the hidden dimension. For brevity, we denote activations as $h_{\ell,t}(x) := h_\ell(M_t, x)$.

Computing massive activations during training

Massive activations are defined based on the *top* values in each layer relative to the *median*. To clarify, we characterize the activations being measured as the final hidden state output from each decoder layer, which represents the post-residual activations after both the self-attention and MLP (feed-forward) components have been applied. These correspond to the h_ℓ values in our notation and are the inputs to the subsequent layer. Let the following denote scalar quantities:

$$h_{\ell,t}^{\text{median}}(x): \text{ the median value of } |h_{\ell,t}(x)| \tag{5}$$

$$h_{\ell,t}^{\max}(x): \text{ the largest value in } |h_{\ell,t}(x)| \tag{6}$$

$$r_{\ell,t}(x) := \frac{h_{\ell,t}^{\max}(x)}{h_{\ell,t}^{\text{median}}(x)}: \text{ ratio of the largest activation to the median} \tag{7}$$

Since activations depend on the input, we evaluate them over a distribution $\mathscr{X}$ of realistic inputs. We define $\mathscr{X}$ to contain natural language sentences representative of real-world usage, excluding out-of-distribution inputs that would yield unpredictable behavior. We define the expected activation over $\mathscr{X}$ as:

$$H_{\ell,t}(\mathscr{X}) := \mathbb{E}_{x \sim \mathscr{X}}[h_{\ell,t}(x)] \quad \text{and approximate it with} \quad \tilde{h}_{\ell,t}(X) := \frac{1}{|X|} \sum_{x \in X} h_{\ell,t}(x) \tag{8}$$

In practice, X is a random sample of 10 sequences from the RedPajama dataset [27]. The sample size can be kept to a relatively low size, justified by prior work [5], which found low variance in massive activation patterns across similar inputs. We similarly define $\tilde{h}_{l,t}^{\text{median}}$, $\tilde{h}_{l,t}^{\max}$, and $\tilde{r}_{l,t}$.

Mathematical modeling of massive activation evolution

To track the development of MAs over time, we construct a time series:

$$r_\ell := \left(\tilde{r}_{\ell,t}(x)\right)_{t \in T} = \left(\frac{\tilde{h}_{\ell,t}^{\max}(x)}{\tilde{h}_{\ell,t}^{\text{median}}(x)}\right)_{t \in T} \tag{9}$$

for each layer l. These series are smooth and exhibit consistent patterns across model sizes and layers, suggesting the potential for a generalizable predictive model. Throughout the paper, we set the variable i in r_l to be 1, or, equivalently, we measure the max activation value as in Equation 7.

Results

This section reports two main findings on MAs. First, we trace how MA magnitudes rise and fall throughout training and fit these curves with an accurate predictive model. Second, we show how key architectural choices—layer depth, hidden width, and head count—shape those trajectories, revealing design-level predictors of when and how large MAs will become.

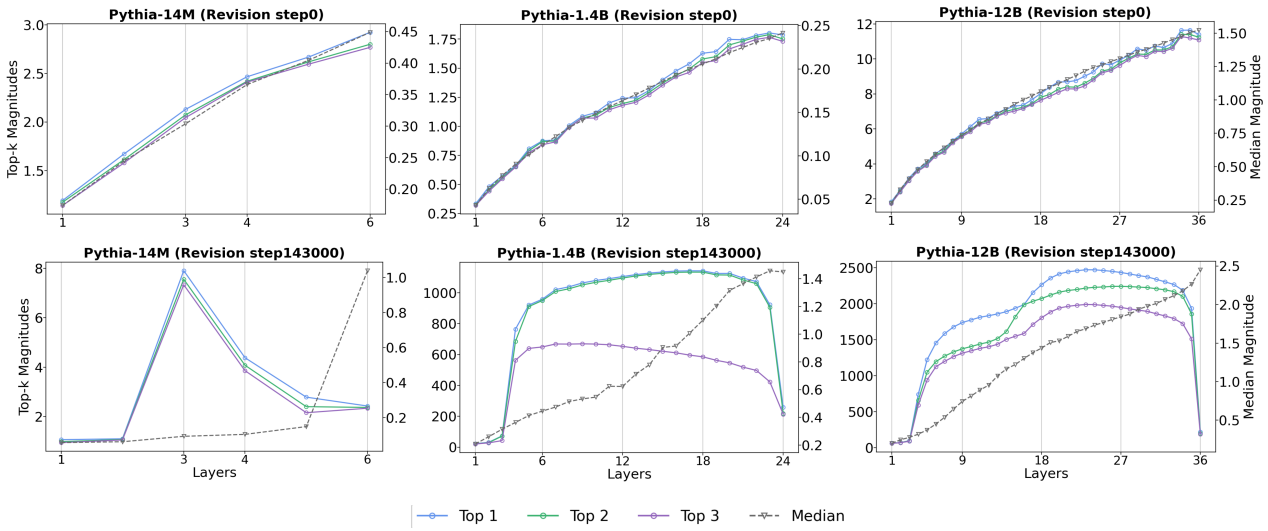


Figure 2. Top activation magnitudes per layer in models Pythia-14M, Pythia-1.4B and Pythia-12B at revision step 0 and 143000, which correspond to the start and end of training. Pythia-14M reaches a top 1 to median ratio of 83, Pythia-1.4B reaches 2350, and Pythia-12B reaches 3200.

Evolution of massive activations during training

We now focus on the evolution of the ratio of the top 1 activation to the median (Equation 9), a magnitude that characterizes massive activations. For convenience, throughout this section, we refer to this quantity simply as 'massive activations'.

Massive activations are learned throughout training, as they are not present at model initialization (see Figure 2), which motivates our work in discovering exactly when and how they develop. We thus plot how they evolve during training and analyze the resulting data, see Figure 3.

We discover several clear patterns:

1. **Layer differentiation** - The evolution of shallow, middle, and deep layers exhibit starkly different shapes.
2. **Strong Predictability** - An exponentially decaying, log-modulated function predicts MA evolution very accurately, scoring an average coefficient of determination of 0.984.
3. **Stage-wise Development** - MAs often peak early on during training and then monotonically decrease from there, showcasing two clear stages.

Layer differentiation

In terms of MA *magnitude*, the first few shallow and last few deep layers overall have significantly smaller MAs than the middle layers. Observing Figure 4a, we note that systematically, between 1 and 3 shallow layers, and 1 or 2 deep layers have significantly different MA patterns than the rest of the layers. We observe that bigger models tend to support this pattern more strongly, with the exception of size 2.8B that has noisier data. This is seen in Figure 6b where we observe bigger models have less noisy MA trajectories that can be modeled with Equation 10 with higher confidence.

In terms of MA *temporal dynamics*, most model sizes and layers display smooth MA trajectories with very similar shapes, with only a reduced number of model sizes and layers displaying noisier time series. Across all model sizes, we observe two broad classes of MA trajectories, largely determined by layer depth:

- **Early peak**: Shallow and deep layers exhibit a rapid rise, reach a clear maximum early in training, then decay toward an asymptote.
- **Log increase**: Middle layers follow a smooth logarithmic climb with no apparent peak during the training window.

These patterns are exemplified in Figure 3, which shows layers 1, 2, 3 and 16 to follow an "early peak" pattern, with layers 4 to 15 displaying a "log increase" curve. More systematically, Figure 4b shows the stark change in pattern in early and late, that peak during training vs middle layers, that peak at step 143k, which is the end of training. Middle layers peak at the end of training as they are monotonically increasing, do to their logarithmic shape. This pattern is particularly stronger for larger models, with 410M or more parameters.

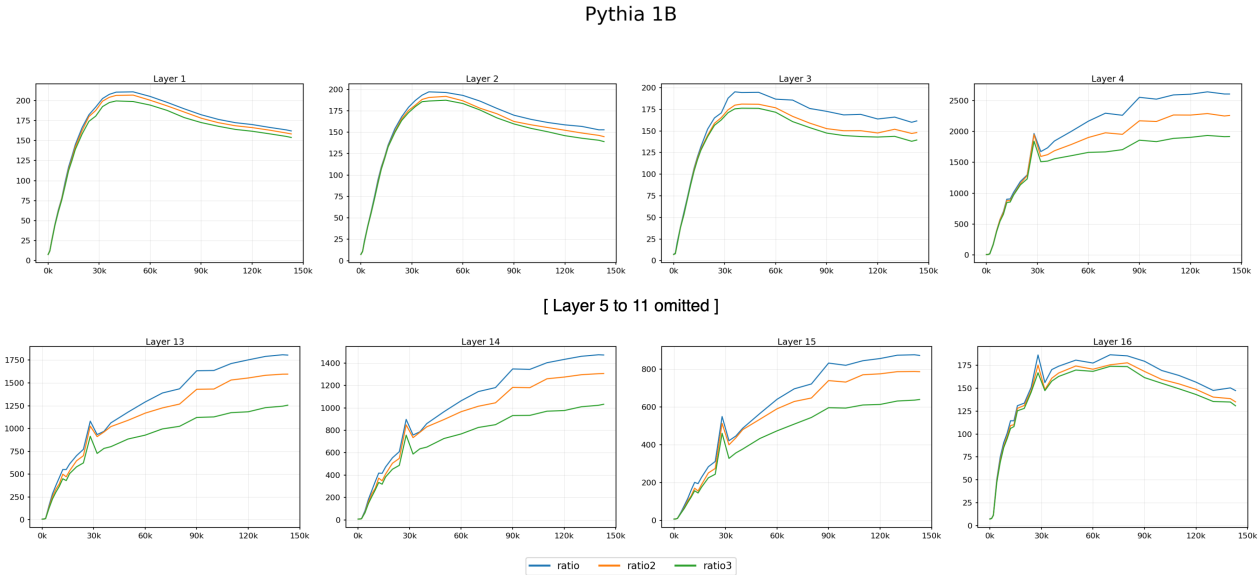


Figure 3. Evolution of the ratio of top activations to median (Equation 9) during training for Pythia 1B. It is a linear interpolation of 37 data points corresponding to different training checkpoints. Apart from the highest activation which is the focus of our study, we also plot ratios corresponding to the top 2 and 3 for comparison. The plots show the training steps on the x-axis, and the ratio of the top magnitudes to median activations on the y-axis.

Strong Predictability

Based on our observations of the MA trajectories, we sought to find a low-dimensional functional form hypothesis that could describe the two main observed dynamics: a logarithmic shape, and an initial peak with eventual decay. Our experiments overwhelmingly support an exponentially decaying, log-modulated function:

$$f(t) = A\, e^{-\lambda x_t} \log(x_t) + K, \qquad \text{where} \quad x_t = \gamma t + t_0 \tag{10}$$

This unified model fits both the "early peak" and "log increase" regimes across all model sizes and depths with high fidelity, by adjusting the influence of decay with the λ parameter, which can make the curve purely logarithmic when $\lambda = 0$, or decaying if $\lambda \gg 0$.

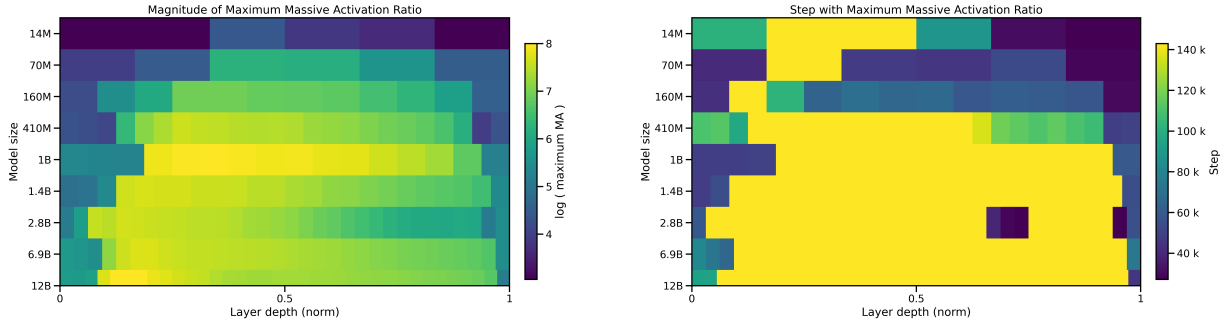
Equation 10 forms the core of our following analysis. The fitting process is described further in the Methodology portion. It has 5 parameters: $\{A, \lambda, \gamma, t_0, K\}$, where A controls the amplitude, λ the decay rate, γ the time scaling, t_0 the time offset, and K the asymptotic baseline. The proposed equation was based upon the observation of the massive activation trajectories of each layer, and seeks to unify all models and layers under a single general formula. Note that for each model and each layer, these 5 parameters take different values. The dynamics of the MA trajectories are very similar across models and layers, but their magnitudes and curves vary from model to model.

Model size	14M	70M	160M	410M	1B	1.4B	2.8B	6.9B	12B
R^2	0.9307	0.9681	0.9831	0.9937	0.9922	0.9956	0.9686	0.9954	0.9829

Table 1. Average layer-wise best R^2 scores — quality of fit of Equation 10 to the time series in Equation 9 — for each Pythia model size.

Our proposed log-modulated exponential model predicts the MA evolution with outstanding accuracy, achieving a mean coefficient of determination of 0.984 across 188 layers from nine model sizes[1]. Table 1 reports the average R^2 for each model. Notably, smaller models (14M & 70M) already reach $R^2 > 0.93$, while larger sizes (160M and above) are generally above $R^2 = 0.98$, indicating that the MA pattern becomes even more pronounced and regular as model size grows.

[1] Average taken over all fitted layers.



(a) Middle-depth layers display significantly higher MAs than shallow and deep layers.

(b) Note the very stark change from shallow and deep layers to middle ones, particularly in the bigger (>410M) models.

Figure 4. Heatmaps showing the location and magnitude of peak MAs by layer depth and model size. Training ends at 143k for the Pythia family, so the yellow middle layers in 4b show that MAs would continue to rise monotonically if training continued, where as the darker layers, generally shallow and deep layers, peak and start decreasing before training ends.

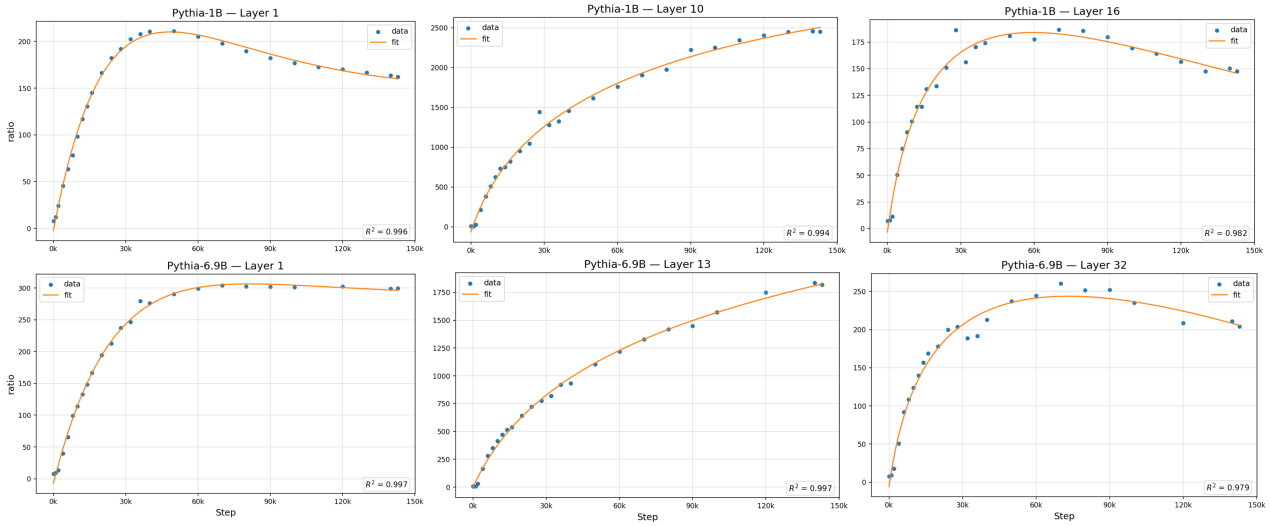


Figure 5. Example fits for two model sizes: 1B and 6.9B and an example shallow, middle and deep layer from each. The plot shows the best fit for Equation 10, and data points corresponding to Equation 9. The last training step for the Pythia family is 143k.

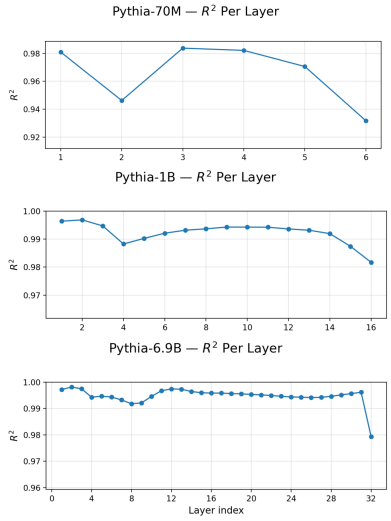
Figure 5 showcases representative fits for shallow, middle, and deep layers in both Pythia 1B and Pythia 6.9B, illustrating how the same exponentially decaying log function captures the full range of training dynamics. In turn, Figure 6b breaks down the full layer-wise R^2 distribution across all nine sizes, revealing that—even at the extremes—no layer falls below $R^2 \approx 0.93$, and most cluster tightly around $R^2 \approx 0.99$. This robustness underlines the universality of our fitted form across depth and scale.

The strength of these results is two-fold. Firstly, the magnitude we are modeling is a result of LLM training - a very noisy process - so an average R^2 value greater than 0.98 is surprising. Secondly, the curve hypothesis is a very simple 5-parameter model, which is able to generalize to 9 model sizes and 188 layers.

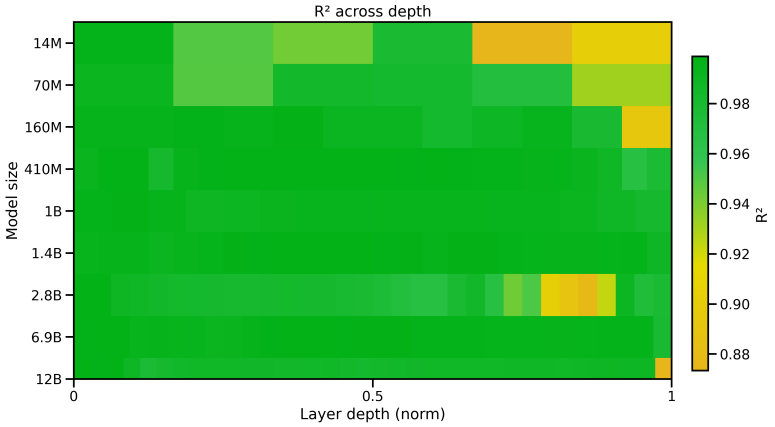
Stage-wise Development

We have found evidence that in early and late layers across all model sizes, MAs quickly develop in the first 60k steps before starting to monotonically decrease. This occurs in every model and layer that is colored anything but yellow in Figure 4b. To the best of our knowledge, this is the first time this phenomenon has been recorded, and it opens up an exciting path of deeper understanding of LLMs. The critical point at which MAs start decreasing suggests there is an underlying two-stage development process governing LLM learning that is poorly understood. Proof of the existence and uniqueness of this critical point is discussed in the next section.

We hope this discovery motivates future work to understand these developmental stages in the field of mechanistic



(a) Example R^2 values for various models.



(b) Full map of all models and all layers and the strength or ease of fit. A greener color means that massive activation trajectories in that coordinate can be modeled with high confidence with Equation 10. Even the lower scoring locations still show evidence of reasonable fits.

Figure 6. Coefficients of Determination for the MA trajectory fits.

interpretability focusing on *training-dynamics*.

Predicting Massive Activation Trajectories from Transformer Architecture

Parameters within Equation 10 are highly influential in the overall shape and size of the curve of massive activation ratios (Equation 9). We now analyze the mathematical behavior of the equation itself. Let us look at steady-state behavior of the equation. In the limit of $t \longrightarrow \infty$, the equation reduces via L'Hôpital's rule for indeterminate forms:

$$\lim_{t\to\infty} f(t) = \lim_{t\to\infty} Ae^{-\lambda(\gamma t+t_0)}\log(\gamma t+t_0)+K = K \tag{11}$$

So the parameter K can be seen to be related to the final steady state value of the ratio. Indeed, in the limit as train steps go to infinity it mathematically is the steady state value, however in practice it is also impacted by the value of the exponential term. Clearly the A parameter affects the overall height of the peak, but let's investigate where the peak occurs. We can do this by first taking the derivative of Equation 10:

$$\frac{d}{dt}f(t) = A\left[\lambda\gamma\log(\gamma t+t_0)e^{-\lambda(\gamma t+t_0)} + e^{-\lambda(\gamma t+t_0)}\left(\frac{\gamma}{\gamma t+t_0}\right)\right] \tag{12}$$

Setting this derivative to zero gives us:

$$\log(\gamma t+t_0)+\frac{\lambda}{\gamma t+t_0} = 0 \tag{13}$$

Rearranging this critical point equation yields $(\gamma t+t_0)\log(\gamma t+t_0) = -\lambda$, which can be solved exactly using the Lambert W function. The solution is:

$$t_{peak} = \frac{e^{W(-\lambda)}-t_0}{\gamma} \tag{14}$$

where W is the Lambert W function. This analytical solution reveals several key insights about peak behavior. First, a real valued peak exists only when $\lambda \leq 1/e \approx 0.368$. This is because the critical point Equation 13 can only be solved in $\mathbb{R}$ using the Lambert W function when $W(-\lambda)$ has real solutions, which requires $-\lambda \geq -1/e$, or equivalently $\lambda \leq 1/e \approx 0.368$. Second, the peak location scales inversely with γ (smaller γ shifts the peak to much larger training steps) and is offset by t_0. This mathematical analysis suggests that γ and λ are the most critical parameters for controlling curve shape and location, with γ having particularly strong influence on peak timing (shown in Figure 7) due to its position in the denominator. Figure 7 illustrates the relationship between the number of training steps needed to see a peak in Equation 9 and parameters γ and λ. Note that the Pythia model family did not reach training steps greater than $\approx 143k$, so for any fitted equation parameters that

predict a peak at location greater than the maximum number of training steps, a peak would not have been seen for a particular layer during the training cycle.

As seen in our previous results section, certain layers within each model exhibit monotonically increasing ratio (9) behavior as training progresses, and some layers exhibit a sharp peak early in training and then decay to a steady state value (see Figure 5). Overall, the layer depth (along with training length) is correlated with peak observation, but we notice that there are breaks in this pattern. For example, the layers (or more specifically, the depth of a layer within a model relative to its overall size) in which quantity 9 exhibits a peak is different for Pythia 12B vs Pythia 1B.

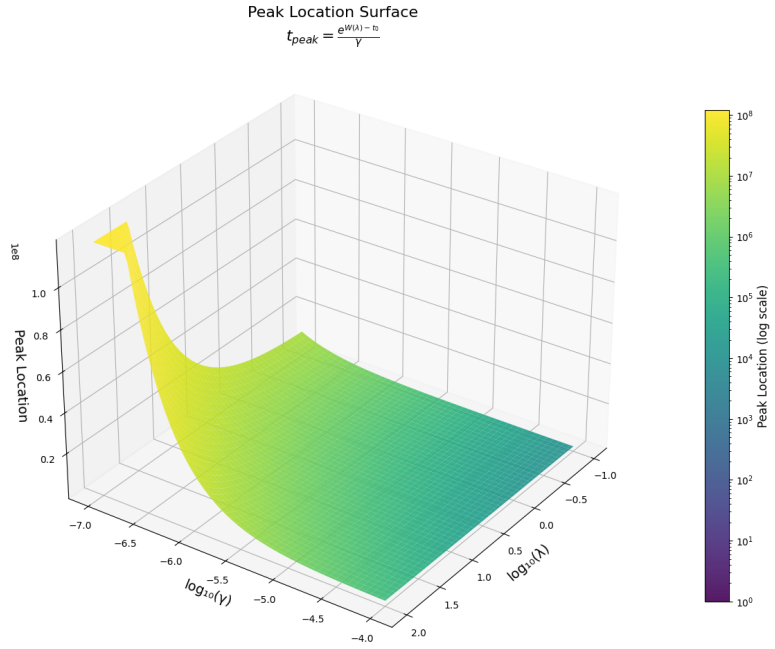


Figure 7. Plot illustrating the change in the location of a peak relative to changes in (log) parameters of Equation 14. Note that Only real valued solutions to the Lambert W function are pictured. t_0 set to a fixed, typical value found across the Pythia model family. Small changes in γ can have large effects in location of a peak.

The predictability analysis presented below uncovers a relationship between transformer architecture and massive activation dynamics. We demonstrate that the complex, seemingly chaotic emergence of massive activations during training is actually governed by predictable mathematical relationships that can be controlled through architectural design. Specifically, we show how architectural parameters play a role in whether peaks in Equation 9 will emerge, their magnitude, and their steady-state behavior—enabling practitioners to architect models with desired massive activation properties from the outset for both further study of massive activations and potentially optimized training dynamics. For instance, controlling γ could theoretically allow peaks to occur sooner in the training cycle, allowing steady states to be reached sooner. However, the relationship between massive activation timing and overall training efficiency requires further investigation. Note that the λ parameter does effectively control whether a peak exists, but in practice within the Pythia model family, λ always takes a value which allows peak existence, therefore γ or t_0 would play the largest roles in peak observation.

The fitted models in the predictability analysis showed strong fit characteristics for certain variables (see Table 2), such as A, λ, and K, while the variables γ and t_0 showed a slightly weaker performance across all models. Table 2 shows the performance of various standard machine learning algorithms in predicting the values of parameters in Equation 10 using only features constructed using architectural specifications for various models within the Pythia family. In most cases we fit the models to transformed features due to large outliers which can easily skew predictions for most model types. Additionally, target variables were isolated such that other variables in the equation were not used as predictors to help isolate the effects of the architecture from the fitted equation. Below, we offer an in-depth analysis of key features and their fits, as well as key explainability measures the top models (shown in bold in Table 2) were able to determine using a SHAP and PDP feature analysis. This interpretability analysis gives us a way to control the value of the behaviors of our MA ratio (Equation 9) using purely architectural design patterns. We focus on parameters γ and K due to their large impact in determination of the massive activation evolution during a training cycle. As noted above, parameter λ does determine theoretic existence of a peak, but in practice (at least within the Pythia family) it is always within the correct range for real solutions, so discussion is not centered

Table 2. Machine Learning Model Performance for Predicting Massive Activation Parameters. Values show test set R^2 scores. Best performing model for each parameter is shown in bold. Negative R^2 values indicate worse-than-baseline performance. Dataset: 188 samples (80% train, 20% test), 5-fold cross-validation.

Parameter	Transform	Ridge	Lasso	Random Forest	Gradient Boosting	XGBoost
A (Amplitude)	log1p	0.077	0.195	**0.476**	0.274	0.244
λ (Peak Occurrence)	log1p	0.031	-0.015	0.643	0.506	**0.664**
γ (Peak Location)	log1p	**0.056**	-0.006	0.055	-1.571	-0.089
t_0 (Time Offset)	log1p	0.100	0.017	**0.447**	0.266	0.387
K (Asymptotic Baseline)	yeo-johnson	0.405	0.316	0.803	0.824	**0.847**

on its relationship to the model architecture. A and t_0 form amplitudes and fixed offsets of the peak and are less impactful, omit their discussion for brevity. Table 3 provides interpretable definitions of various features used in the predictive ML models. Note that raw features are not used to avoid having direct dependency on specific models with the Pythia family; we primarily focus on normalized features to increase applicability more widely across the model family, in general. We also note that the architectural design choices within the Pythia model family do not have high degrees of variation, and often a particular feature scales proportionally to model size. So machine learning models fit using this architectural distribution could have trouble generalizing.

Table 3. Key Architectural Features Used in Predictive Models. **Note:** ℓ = layer index, L = total layers, d = hidden dimension, H = attention heads, d_{ff} = intermediate (MLP) dimension.

Feature Name	Formula	Interpretation
Layer Position	ℓ/L	Relative depth within model (0 = first layer, 1 = last layer)
Layer Position²	$(\ell/L)^2$	Quadratic position effects for non-linear layer behavior
Layer Position³	$(\ell/L)^3$	Cubic position effects capturing complex depth dependencies
Layer Position$^{1/2}$	$\sqrt{\ell/L}$	Square root position effects for early-layer emphasis
Attn. Heads/Hidden Size	H/d	Number of attention heads per hidden dimension
Intermediate Ratio	d_{ff}/d	MLP expansion factor (in the Pythia family this a fixed value of 4x)
Width/Depth Ratio	d/L	Model width relative to depth (architectural shape)
Attn. Heads/Num. Layers	H/L	Attention head budget distributed across layers
log(Hidden Size)	$\log(d)$	Logarithm of hidden dimension (proxy for model size)
Layer DepthxModel Depth	ℓL	Interaction between layer location and total length

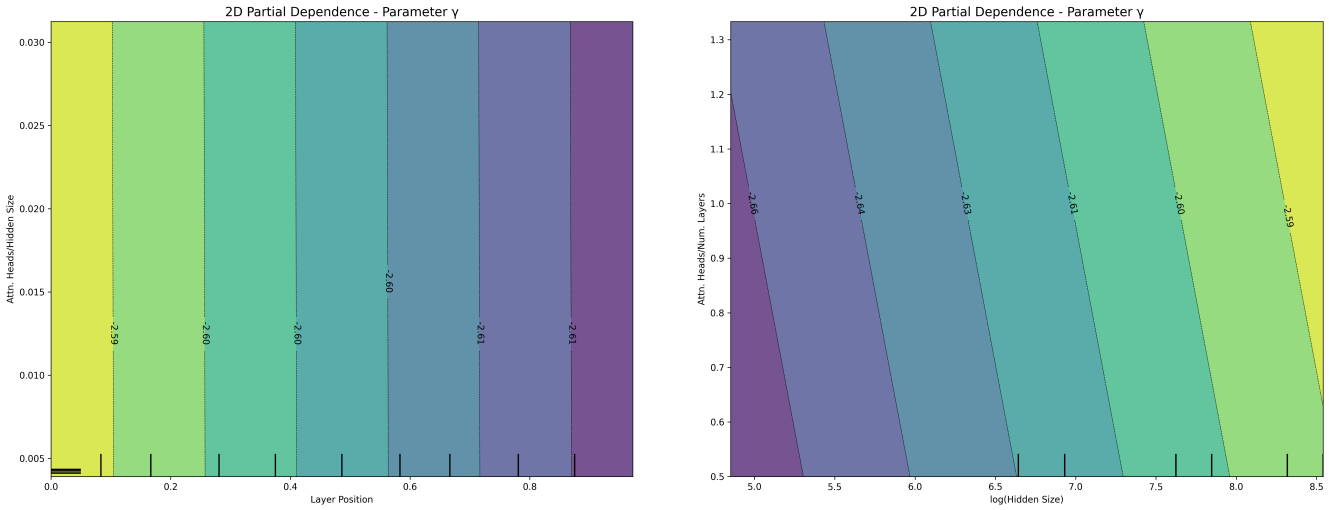
Parameter γ

Although the fitted relationship determined for γ was weaker overall, we can still pull out some valuable insights from the machine learning models. The 2D partial dependence plot in Figure 8a reveals the complex relationship between layer position, attention density, and parameter γ, which controls peak timing in massive activation dynamics. The plot demonstrates clear phase transitions across the architectural design space, with distinct regions corresponding to different peak timing behaviors.

Several key insights emerge from this analysis. First, the relationship exhibits strong stratification by layer position, with early layers (position < 0.2) showing different γ responses to attention density changes compared to later layers (position > 0.6). Second, within each layer position regime, attention density modifications can provide a slight, minimal architectural

control over peak timing. For instance, in early layers, adjusting the attention density from approximately 0.005 to 0.020 results in measurable shifts toward earlier peak occurrence, as evidenced by the transition from yellow to green regions in the plot.

This finding has important implications for transformer design, as it suggests that architects could implement layer-specific attention configurations to create controlled massive activation peak timing schedules. Rather than using uniform attention density across all layers, designers could strategically vary the number of attention heads per hidden dimension to influence when different layers reach their massive activation peaks during training. The clear phase boundaries visible in the plot provide specific guidance for these architectural choices, enabling more precise control over the temporal dynamics of massive activations throughout the model.



(a) 2D PDP plot relating parameter γ, attention density, and layer position. Values in PDP shown in original input feature scale.

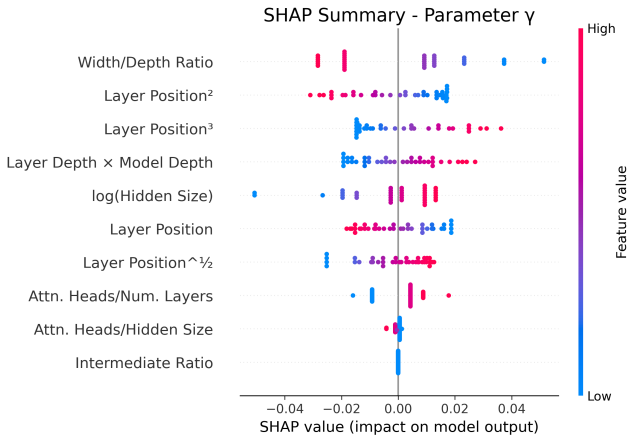
(b) 2D PDP plot relating parameter γ, hidden layer size relative to attention head count. Values in PDP shown in original input feature scale.

Figure 8. 2D PDP plots for parameter γ

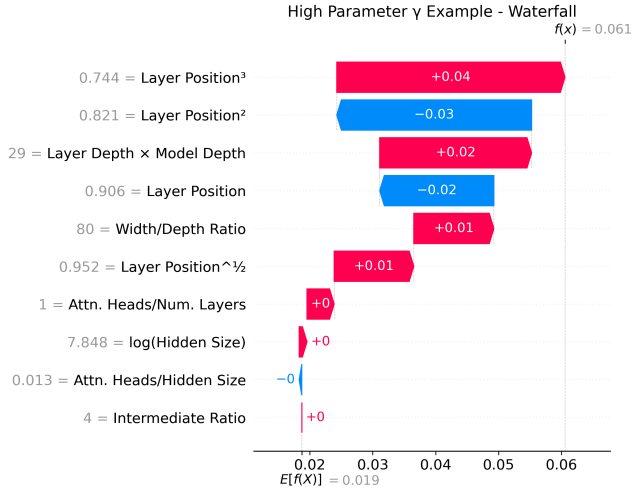
Figure 8b demonstrates the architectural trade-offs between attention head configuration and model shape in controlling parameter γ. The 2D partial dependence plot shows the relationship between the number of attention heads on the y-axis and hidden dimension size on the x-axis. The diagonal contours reveal that these two architectural choices interact to determine peak timing behavior. The plot indicates that for any fixed hidden size, designers can systematically tune peak timing through attention head allocation. Moving vertically at a constant hidden size, configurations with higher head counts produce higher γ values corresponding to earlier peak occurrence, while smaller head counts result in lower γ values and delayed peaks. This relationship provides practical architectural control within fixed computational constraints. For instance, a model with a certain hidden dimension could implement either fewer, larger attention heads to achieve earlier peaks or more, smaller heads to delay peak timing, without changing the overall model size. The consistent diagonal patterns indicate this trade-off holds across different model scales.

The SHAP analysis in Figure 9 corroborates these findings. The summary plot shows that width/depth ratio has strong influence on γ, with higher ratios (wider, shallower models) consistently decreasing γ and pushing peak timing later in training. Conversely, more attention heads increase γ, promoting earlier peaks. The waterfall plot provides a concrete example of how these architectural choices combine: a high γ prediction results from the interaction of layer location (+0.04) along with smaller contributions from the count of attention heads and size of the hidden dimension. We can also see that the overall shape of the model (Width/Depth) plays a strong role.

Together, these analyses establish that transformer architects can control massive activation peak timing through two primary mechanisms: adjusting the overall width/depth ratio and configuring the attention head size within each layer, enabling targeted training dynamics optimization within fixed computational budgets. Additionally, future work could experiment with transformer architectures with varying values of these key features per block.



(a) SHAP Summary Plot showing directional relationships between model architecture features and predicted values of γ



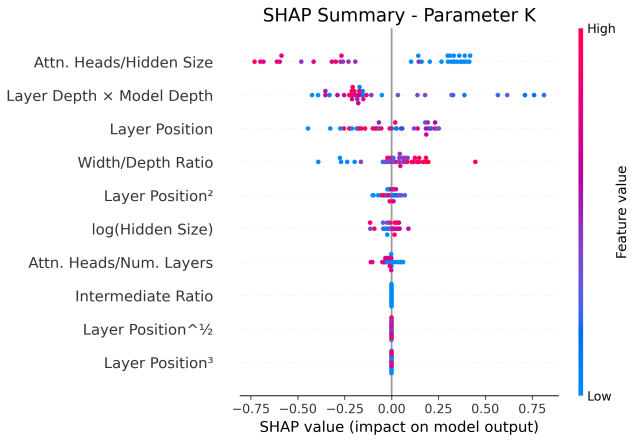
(b) High Parameter γ Example - Waterfall

Figure 9. SHAP Analysis for Parameter γ (Peak Timing). Expected value and predicted value shown in transformed scales.

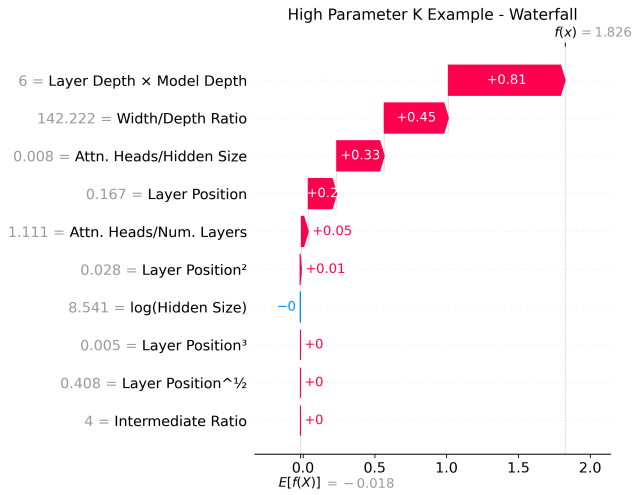
Parameter K

Parameter K represents the steady-state value of massive activation ratios in the limit as training steps approach infinity, making it a critical architectural design target. The analysis reveals that this asymptotic behavior is highly predictable from architectural choices, with the model achieving an R^2 of 0.8186 on the test set.

The SHAP feature importance analysis identifies attention density (attention heads per hidden dimension) as the dominant architectural control for steady-state behavior, followed by the layer depth interaction term and layer position (as seen in Figure 10a). This hierarchy indicates that attention architecture design has the strongest influence on long-term activation dynamics, while layer-specific effects provide secondary modulation.



(a) SHAP Summary Plot for Parameter K revealing directional relationships between architectural features and steady-state predictions. Colors indicate feature values (red = high, blue = low).



(b) SHAP Waterfall Plot for highest Parameter K prediction showing individual feature contributions. Predicted value f(x) = 1.811 in transformed space corresponds to approximately 5.1 in original scale.

Figure 10

The SHAP summary plot in Figure 10a reveals the directional relationships governing steady-state control. Most notably, decreasing attention density—either by reducing the number of attention heads or increasing the hidden dimension—consistently increases Parameter K, leading to higher steady-state massive activation ratios. This relationship suggests that models with

fewer, larger attention heads will exhibit elevated baseline activation levels in the long term compared to models with many smaller heads. Notably, the attention density feature shows consistent directional effects across both γ and K parameters. While attention density is a weaker predictor for γ, the SHAP analysis reveals that changes in this architectural choice create coordinated effects: decreasing attention density simultaneously increases both γ (bringing peaks closer in time) and K (elevating the final steady-state ratio of largest to median activations). This coupling suggests that attention architecture modifications provide unified control over both the timing and magnitude of massive activation dynamics.

The waterfall plot in Figure 10b demonstrates these effects in practice, showing how a high Parameter K prediction results from specific architectural choices. The largest contribution (+0.81) comes from the layer depth interaction term, indicating this is a deep layer within a deep model. Additional positive contributions from low attention density (+0.33) and high width/depth ratio (+0.45) further elevate the steady-state prediction, while early layer position (+0.01) provides additional upward pressure.

These findings establish that transformer architects can systematically control steady-state massive activation behavior through attention architecture design, with attention density serving as the primary control mechanism and layer depth providing amplification effects for deeper models.

Parameter λ

Lastly, we provide analysis for parameter λ, which determines whether peaks occur at all through the critical threshold $\lambda \leq 1/e \approx 0.368$. Although this parameter is less influential in determining precise peak timing compared to γ, its strong predictive performance ($R^2 = 0.664$) indicates that architects may be able to control peak occurrence through architectural choices.

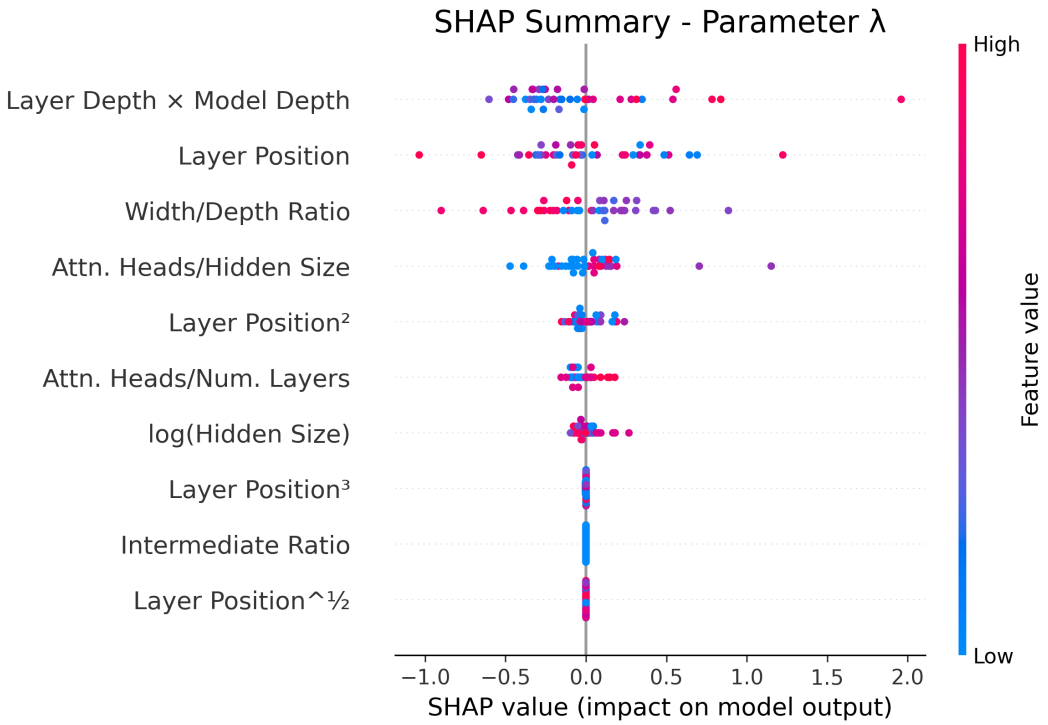


Figure 11. SHAP summary plot for parameter λ, indicating directional relationships between architectural choices and values of λ predicted by the top performing machine learning model.

The SHAP analysis in Figure 11 reveals that layer depth characteristics dominate λ predictions, with the layer depth interaction term showing the strongest and most consistent effects. Deeper layers within deeper models (high feature values, red points) consistently push λ toward higher values, effectively suppressing peak behavior. This relationship suggests that peak occurrence follows a systematic architectural pattern, with early layers in shallow models most likely to exhibit peaks, while deeper layers in larger models tend toward monotonic decay. This is exactly what is seen, for example, in 3.

The attention density (Attn. Heads/Hidden Size), aligns perfectly with findings from the SHAP analysis for parameter γ; where, in λ, the feature relationship is reversed. Increasing this ratio increases λ, and decreases γ, both causing an increase in the predicted location of a ratio peak during a training cycle.

These findings establish a clear architectural hierarchy for controlling peak occurrence: layer depth and model shape provide the primary controls, while attention architecture offers secondary modulation. The high predictive accuracy suggests that architects can systematically design models to either encourage or suppress massive activation peaks across different layers, providing a new dimension of control over internal model dynamics during training.

Discussion

This study presents a new framework for understanding the emergence and dynamics of MAs in transformer models. Our core contributions can be summarized as follows. First, we introduce a five-parameter, exponentially-decaying log-modulated function that accurately models how the top-magnitude activations evolve throughout training within each transformer layer. Each parameter: amplitude (A), decay rate (λ), time scaling (γ), time offset (t_0), and asymptotic baseline (K), captures an interpretable aspect of the activation emergence curve. This mathematical model provides, for the first time, a quantitative and interpretable description of the temporal development of MAs across model scales and layers.

Second, we develop a machine learning-based predictive framework that can forecast the values of these five parameters solely from architectural features such as the number of layers, hidden dimension size, attention head count, and layer position. By leveraging tree-based ensemble models (Random Forest [28], XGBoost [29]) and modern explainability techniques (SHAP values [30], partial dependence plots [31]), we show that it is possible to predict, with high accuracy, key aspects of massive activation dynamics—notably the steady-state baseline K ($R^2 = 0.847$) and curvature features λ and γ—based purely on a model's static architecture. This demonstrates that MAs are not random artifacts of training, but follow systematic, architecture-dependent rules.

Third, our interpretability analyses reveal that specific architectural choices serve as master controls for MA dynamics. In particular, the ratio of attention heads to hidden dimension ("Attn. Heads/Hidden Size") and layer position ("Attn. Heads/Num. Layers") within the network are the dominant drivers of MA emergence and amplitude. We identify that changing these ratios within a transformer architecture, for example, changing "Attn. Heads/Layer" from 0.7 to 1.0, can have direct impact in controlling the shape of the development of MA ratio curves during the training cycle, and change the distribution of which layers have a visible peak and which do not. This non-monotonic relationship suggests that careful tuning of attention architecture offers a concrete handle for modulating high-norm activation behavior.

Taken together, these findings establish that the emergence of massive activations is a predictable, quantifiable, and interpretable phenomenon rooted in architectural design—not merely a quirk of optimization or dataset. MAs appear abruptly during training, often at specific layers, hidden dimensions, and token positions (e.g., beginning-of-sequence or delimiter tokens [5, 20]), and then stabilize to input-agnostic, nearly constant values that act as implicit bias terms within the network.

This new understanding of MA dynamics opens several avenues for both theory and practice. For model designers, our predictive framework enables MA-aware architecture by providing precise control over when and where massive activations emerge. Rather than treating MAs as unpredictable training artifacts, architects can now systematically design models to either suppress or accentuate MAs as desired. For example, our findings show that adjusting attention density can create sharp phase transitions in steady-state behavior, while modifying width/depth ratios provides coordinated control over both peak timing and magnitude. This offers a principled way to navigate architectural tradeoffs while maintaining desired MA properties.

While our study is comprehensive within the EleutherAI Pythia family of decoder-only transformers, several limitations remain. Our results may not fully generalize to encoder-based models (see BERT, [32]), sequence-to-sequence architectures (see orginal transformer variant outlined in [4]), or other architectures such as LLaMA-based models [33]. The temporal resolution of our checkpoint data means we capture the emergence of MAs at coarse granularity; more frequent checkpoints might uncover finer dynamics or more gradual onset. Our input sampling for activation measurement was necessarily limited for computational reasons; broader input distributions may reveal additional or rare MAs not captured here. Finally, while our framework robustly predicts several parameters (especially K, λ), timing-related parameters (γ and t_0), remain harder to predict, suggesting that some aspects of MA emergence depend on optimization dynamics or data ordering not captured in architecture alone.

These limitations point toward several promising directions for future research. First, our observation that some layers require extended training beyond 143k steps to reach their predicted peaks raises intriguing questions about the relationship between MA dynamics and grokking phenomena [34, 35]. Since grokking often occurs after hundreds of thousands of training steps, future work could investigate whether this extended training provides sufficient time for "slow-peaking" layers to complete their internal reorganization, potentially revealing MA peak timing as a predictor or correlate of delayed learning transitions. Additionally, our predictive framework suggests the possibility of designing quantization-aware architectures that intentionally delay MA peak emergence well beyond typical training horizons. Since many applications require quantization for deployment, architectures that can maintain performance while keeping MAs suppressed during standard training durations could offer significant practical advantages for efficient inference.

Second, validation across diverse model families would strengthen the generalizability of our framework. While Pythia provides an excellent controlled environment, expanding to encoder-decoder architectures [4], different training objectives [36], and alternative attention mechanisms [37, 38] would test the universality of our architectural control principles. Of particular interest would be investigating whether our predictive relationships hold across models with different positional encoding schemes [39, 40], normalization strategies [41, 42] and non-normalization strategies [43], or activation functions.

Third, our current analysis is constrained by the limited architectural diversity within existing model families. For example, Pythia models consistently use a 4× MLP expansion ratio, making it impossible to predict how variations in this parameter affect MA dynamics. Future work could involve training custom model families with systematic variations in currently fixed ratios—such as MLP expansion factors ranging from 2× to 8×, or models with heterogeneous attention head configurations across layers. Such experiments would provide the architectural diversity needed to validate and extend our predictive framework to the full space of transformer design choices.

In summary, our results provide the first quantitative, predictive, and interpretable model of massive activation emergence in transformers, with immediate implications for theory, design, and deployment. We hope these findings will serve as a foundation for future MA-aware model development and inspire new techniques for harnessing or controlling this fundamental phenomenon.

Methodology

In this section, we outline the experimental setup used to both fit a mathematical model to the evolution of massive activations during a training cycle, and also to detail the framework used to extract explainable predictive insights from the model.

Experimental setup

MAs were estimated by setting X in Equation 8 to be a sample of 10 random sequences from the Red Pajama dataset [44]. This dataset represents data from the training distribution, or more generally, from the target distribution. For each model, we elicited MAs for at least 37 regularly spaced steps.

Parameters for Equation 10 were estimated with SciPy 1.15.2 (`scipy.optimize.curve_fit`; [45]) using the Trust-Region Reflective algorithm [46], with analytic Jacobians supplied and bounds enforced on the λ parameter to keep it positive, as negative values produce an exponential curve, much different from the observed trajectories. The `curve_fit` algorithm seeks to find a set of parameters that minimize the fit error, and does so iteratively. To speed up convergence, we provide an analytic Jacobian of our function, and an initial guess. Data points are normalized first, fitted in the normalized space, and then the parameters are scaled back. Each model and layer had a minimum of 27 data points, corresponding to regularly spaced training checkpoints.

Other hypothesis were tested, such as *first* and *second degree step function*, but were discarded due to lower R^2 and Akaike Information Criterion (AIC) score [47]. We opted for AIC score because we sought to compare not only accuracy of the models, but also simplicity, when comparing the 5 parameter hypothesis in Equation 10, versus the 3 parameter second degree step function. The strength of performance from Equation 10 compensated the extra couple of parameters.

Parameter analysis and architectural relationships

After fitting our mathematical models to the temporal evolution data, we investigate how the learned parameters relate to architectural properties of the transformer models. This predictive analysis enables us to understand which architectural design choices most strongly influence massive activation emergence patterns, and more importantly, provides us with a way to directly control the training dynamics purely through initial model architectural choices.

Feature engineering and data preparation

We construct a comprehensive feature set from the architectural specifications of each Pythia model variant, including:

- **Core architectural features**: hidden dimension size (d), number of layers (L), number of attention heads, feed-forward network width, rotary embedding base, and rotary percentage
- **Derived features**: layer position normalized by total depth, ratios between architectural components (e.g., attention heads per hidden dimension), and interaction terms
- **Polynomial features**: quadratic and cubic terms for layer position to capture non-linear positional effects
- **Logarithmic transformations**: log-scaled versions of large architectural parameters to handle wide dynamic ranges

Given the diverse scales and distributions of our fitted parameters, we apply appropriate transformations to improve model performance. For parameters with positive values and high skewness, we use log transformations. For parameters with mixed signs or extreme outliers, we employ a Yeo-Johnson [48] power transformation to achieve approximate normality. Features are transformed with a standard scaler to allow usage across a range of model types; plots showing relationships between targets and features, therefore, often represent the relationship between the transformed target and transformed feature.

Machine learning model selection

We evaluate multiple regression algorithms to identify the best predictor for each fitted parameter:

- **Linear models**: Ridge and Lasso regression with L2 and L1 regularization respectively
- **Tree-based ensembles**: Random Forest and Gradient Boosting regressors to capture non-linear relationships
- **Advanced boosting**: XGBoost with careful hyperparameter tuning for optimal performance

Model selection is based on 5-fold cross-validation performance, with the final evaluation conducted on a held-out test set (20% of data). We use coefficient of determination (R^2) as our primary metric, supplemented by mean absolute error (MAE) and root mean squared error (RMSE).

Model interpretability and feature importance

To understand which architectural factors most strongly influence massive activation dynamics, we employ multiple interpretability techniques:

- **Feature importance analysis**: For tree-based models, we extract built-in feature importance scores.
- **SHAP (SHapley Additive exPlanations) analysis**: Provides model-agnostic explanations of individual predictions and global feature importance.
- **Partial dependence plots**: Visualize the marginal effect of individual features on predicted parameter values
- **Residual analysis**: Examine prediction errors to identify potential model limitations or data quality issues.

This comprehensive analysis enables us to develop predictive relationships that can forecast massive activation emergence patterns based solely on architectural specifications, providing insights into how design choices influence these critical phenomena during training.

Ethical Approval

This study was reviewed by the Institutional Review Board of New York University and was determined to be exempt from further review (IRB protocol number: IRB-FY2025-10500).

Funding

This work was not supported by any specific funding.

References

1. Brown, T. *et al.* Language models are few-shot learners. *Adv. neural information processing systems* **33**, 1877–1901 (2020).
2. Biderman, S. *et al.* Pythia: A suite for analyzing large language models across training and scaling (2023). 2304.01373.
3. Zhang, A., Lipton, Z. C., Li, M. & Smola, A. J. *Dive into deep learning* (Cambridge University Press, 2023).
4. Vaswani, A. *et al.* Attention is all you need. *Adv. neural information processing systems* 6000–6010 (2017).
5. Sun, M., Chen, X., Kolter, J. Z. & Liu, Z. Massive activations in large language models. In *First Conference on Language Modeling* (2024). ArXiv preprint: https://arxiv.org/abs/2402.17762.
6. Nrusimha, A. *et al.* Mitigating the impact of outlier channels for language model quantization with activation regularization. *arXiv preprint arXiv:2404.03605* (2024). Applies kurtosis-based regularization to mitigate activation outliers for W4A4 quantization.

7. Dettmers, T., Thoppilan, R. *et al.* Gptq and llm.int8(): 1-bit quantization for large language models. *NeurIPS* (2022). Isolates activation outliers into high-precision for 8-bit inference.

8. Ma, C. *et al.* First activations matter: Training-free methods for dynamic activation in large language models (2024). 2408.11393.

9. Szatkowski, F., Wójcik, B., Piórczyński, M. a. & Scardapane, S. Exploiting Activation Sparsity with Dense to Dynamic-k Mixture-of-Experts Conversion. In Globerson, A. *et al.* (eds.) *Advances in Neural Information Processing Systems*, vol. 37, 43245–43273 (Curran Associates, Inc., 2024).

10. Team, D. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437* (2024). Details FP8 training for a 671B MoE model, using tile-wise scaling to handle activation outliers and maintain stability.

11. Zhao, Z. *et al.* Activation control for efficiently eliciting long chain-of-thought ability of language models (2025). Under review, 2505.17697.

12. An, Y., Zhao, X., Yu, T., Tang, M. & Wang, J. Systematic outliers in large language models. In *The Thirteenth International Conference on Learning Representations* (2025).

13. He, B., Noci, L., Paliotta, D., Schlag, I. & Hofmann, T. Understanding and minimising outlier features in transformer training. *Adv. Neural Inf. Process. Syst.* **37**, 83786–83846 (2024).

14. Zuhri, Z. M. K., Fuadi, E. H. & Aji, A. F. Softpick: No attention sink, no massive activations with rectified softmax (2025). 2504.20966.

15. Kaul, P., Ma, C., Elezi, I. & Deng, J. From attention to activation: Unravelling the enigmas of large language models. *arXiv preprint arXiv:2410.17174* (2024).

16. Oh, J., Shin, S. & Oh, D. House of cards: Massive weights in llms (2025). 2410.01866.

17. Bondarenko, Y., Nagel, M. & Blankevoort, T. Quantizable transformers: Removing outliers by helping attention heads do nothing. In *NeurIPS* (2023).

18. Lin, H. *et al.* Duquant: Distributing outliers via dual transformation makes stronger quantized llms. *Adv. Neural Inf. Process. Syst.* **37**, 87766–87800 (2024).

19. Kim, J. *et al.* Peri-ln: Revisiting normalization layer in the transformer architecture. In *Forty-second International Conference on Machine Learning* (2025).

20. Owen, L., Chowdhury, N. R., Kumar, A. & Güra, F. A refined analysis of massive activations in llms (2025). 2503.22329.

21. Darcet, T., Oquab, M., Mairal, J. & Bojanowski, P. Vision transformers need registers. *arXiv preprint arXiv:2309.16588* (2023).

22. Gan, C. *et al.* Unleashing diffusion transformers for visual correspondence by modulating massive activations (2025). Under Review, 2505.18584.

23. Xu, Y., Huang, H., Wang, Y. & Wang, H. Tracking the feature dynamics in llm training: A mechanistic study (2024). 2412.17626.

24. Jin, M. *et al.* Massive values in self-attention modules are the key to contextual knowledge understanding (2025). 2502.01563.

25. Yue, Y. *et al.* Wkvquant: Quantizing weight and key/value cache for large language models (2024). 2402.12065.

26. Yang, J., Kim, H. & Kim, Y. Mitigating quantization errors due to activation spikes in glu-based llms (2024). 2405.14428.

27. Computer, T. Redpajama: An open-source recipe to reproduce the llama training dataset (redpajama-data-1t-sample) (2023).

28. Breiman, L. Random forests. *Mach. learning* **45**, 5–32 (2001).

29. Chen, T. & Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794 (2016).

30. Lundberg, S. M. & Lee, S.-I. A unified approach to interpreting model predictions. *Proc. 31st Int. Conf. on Neural Inf. Process. Syst.* 4768–4777 (2017).

31. Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Annals statistics* 1189–1232 (2001).

32. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

33. Touvron, H. *et al.* Llama 2: Open foundation and fine-tuned chat models (2023). 2307.09288.

34. Power, A., Burda, Y., Edwards, H., Babuschkin, I. & Misra, V. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177* (2022).

35. Liu, Z. *et al.* Towards understanding grokking: An effective theory of representation learning. *Adv. Neural Inf. Process. Syst.* 34651–34663 (2022).

36. Radford, Narasimhan & Salimans, S. Improving language understanding by generative pre-training. *OpenAI Tech. Rep.* (2018).

37. Liu, Z. *et al.* Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 9992–10002 (2021).

38. Choromanski, K. *et al.* Rethinking attention with performers. *arXiv preprint arXiv:2009.14794* (2020).

39. Su, J. *et al.* Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing* **568**, 127063 (2024).

40. Press, O., Smith, N. A. & Lewis, M. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409* (2021).

41. Ba, J. L., Kiros, J. R. & Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

42. Zhang, B. & Sennrich, R. Root mean square layer normalization (2019). 1910.07467.

43. Zhu, J., Chen, X., He, K., LeCun, Y. & Liu, Z. Transformers without normalization (2025). CVPR 2025, 2503.10622.

44. Computer, T. RedPajama-Data-1T-Sample: a 1 b-token open corpus for llm pre-training (2023). Accessed 28 Jul 2025.

45. Virtanen, P. *et al.* SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **17**, 261–272, DOI: 10.1038/s41592-019-0686-2 (2020).

46. Coleman, T. F. & Li, Y. An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM J. on Optim.* **6**, 418–445, DOI: 10.1137/0806023 (1996).

47. Akaike, H. A new look at the statistical model identification. *IEEE Transactions on Autom. Control.* **19**, 716–723, DOI: 10.1109/TAC.1974.1100705 (1974).

48. Yeo, I.-K. & Johnson, R. A. A new family of power transformations to improve normality or symmetry. *Biometrika* **87**, 954–959 (2000).