MOYU: Massive Over-activation Yielded Uplifts in LLMs

Anonymous ACL submission

Abstract

Massive Over-activation Yielded Uplifts(MOYU) is the inherent properties of large language models and dynamic activation (DA) based on MOYU property is a clever but under-explored method designed to accelerate inference in large language models. Existing approaches to utilize MOYU typically face at least one major drawback, whether in maintaining model performance, enhancing inference speed, or broadening applicability across different architectures. This paper introduces two Sequential DA methods called sTDA and sRIDA that leverage sequence information while utilizing MOYU property, effectively overcome the "impossible triangle" that bothers current DA approaches. Our two schemes have improved generation speeds by 20-25% without significantly compromising the model's task performance. Additionally, given the blur of theoretical studies of MOYU, this paper also explains its root cause, then outlines the mechanisms of two main limitations (i.e. history-related activation uncertainty and semantic-irrelevant activation inertia) faced by existing DA methods.

1 Introduction

Large language models (LLMs), such as LLaMA, GPT and OPT series, have demonstrated impressive performance and in-context learning abilities by leveraging a vast number of parameters. Nevertheless, their computational and memory requirements during inference, particularly in latency-sensitive scenarios, are significant. To mitigate these challenges, several techniques based on Massive Over-activation Yielded Uplifts(MOYU) have been suggested to cut down the latency of these models by reducing the massive over-activated heads, neurons or weights during inference.

Existing MOYU-based techniques can be classified into *static* and *dynamic activation* methods. Static activation (SA), such as pruning, trims the over-activated surplus weights within LLMs based on metrics such as magnitude, implemented either once or iteratively. These structures remain fixed across all subsequent inputs and are fully activated during inference. However, a limitation of SA is that once SA is complete, the inactive weights cannot be restored without a recovery phase, potentially leading to performance degradation and the loss of in-context learning ability. Additionally, the iterative SA process entails significant additional training efforts, yet it may not result in a corresponding enhancement in speedup.

On the other hand, MOYU-based dynamic activation (DA) offers adaptability by selectively activating certain heads or neurons during inference, thereby enhancing computational efficiency. This approach leverages the inherent property of massive over-activation present in LLMs to optimize resource utilization. Existing researches on DA can be categorized as follows:

1. **Threshold Dynamic Activation (TDA)**: TDA employs a predefined threshold to decide which activation units to retain or discard. Units with activation values falling below this threshold are either set to zero or eliminated during the current forward propagation, thereby reducing computational overhead.

2. **Router-off-the-loop Dynamic Activation (RODA)**: This approach utilizes a pre-trained *router* block to dynamically determine which activation units are essential during the model's forward propagation. The router is trained using the model's historical data. DejaVu(Liu et al., 2023b) utilizes a predictive router that consists of a two-layer linear network.

3. **Router-in-the-loop Dynamic Activation (RIDA)**: Unlike RODA, the router in this method makes dynamic decisions based on

the current input and contextual information. RIDA also allows the router to adjust its routing strategy in real-time, catering to the difficulties of the task at hand, and thereby enhancing the overall efficiency and accuracy.

Despite significant progress, current research on MOYU and DA still lacks a comprehensive theoretical framework that explains MOYU phenomena across various architectures and activation functions, as well as the underlying mechanisms of MOYU within sequences.

Therefore, besides of our two sequential MOYU-based strategies named **sTDA** and **sRIDA**, we have also developed a mathematical rationale that explains the origins of MOYU phenomenon. From this point of view, we have analyzed the cause of two major limitations of existing DA methods: 1) restriction to ReLU activation functions; 2) failure to identify active neurons based on semantic similarity.

- Firstly, we suggests that in *token-level*, history-information-related activation uncertainty (in Section 3.2.1) makes non-ReLU model's weight importance hard to predict, which in turn restricts token-level RODA methods to ReLU models.
- Secondly, we suggests that in *sequence-level*, neuron activation is semantic-irrelevant (in Section 3.2.2). In other words, neurons are more likely to be activated by the heavy hitter within the same sequence rather than by the semantic information in the input itself, which in turn restricts sequence-level DA to RIDA instead of RODA.
- In short, it is despairing that technically we only have 3 DA strategies: token-level RODA for ReLU models(Liu et al., 2023b), token-level RIDA (MoE), and sequence-level TDA and RIDA in this paper.

The rest of the paper is organized as follows. Related works are reviewed in Section 2. We introduce our universal theoretical framework in Section 3, and conduct extensive experiments in Section 4. Finally, in Section 5, conclusions are drawn.

2 Related Works

2.1 Massive Over-activation

In the study of LLMs, "massive over-activation" describes the excessive activation of numerous neurons during task execution, potentially leading to computational waste and decreased efficiency(et.al, 2022; Yuan et al., 2024). Research(Liu et al., 2023a) indicates that dense deep neural networks often exhibit massive over-activation, and by treating the discrete sparse process as a continuous problem, it becomes feasible to optimize the model architecture and end-to-end. The Lottery Hypothesis(Frankle and Carbin, 2019; Malach et al., 2020) also underscores the importance of pruning techniques in eliminating unnecessary connections and mitigating over-activation in dense models. Another research(Shazeer et al., 2017) address this issue by introducing "sparse activation" concept through a "sparsely-gated mixture-of-experts(MoE) layer", which enhances model capacity while reducing computational costs. Furthermore, MC-SMoE(Li et al., 2024) tackles the issue of massive over-activation in MoEs by streamlining the model architecture through the merging and low-rank decomposition of redundant experts, guided by the router's information.

2.2 TDA and RODA

Research(Liu et al., 2023a; Mirzadeh et al., 2023) elucidates the capacity of the ReLU to introduce activation sparsity and proposes the concept of dynamic activation. DejaVu(Liu et al., 2023b) identifies that the sparsity introduced by ReLU can be predicted and thus proposes the first viable RODA scheme. On the OPT series, DejaVu can facilitate a 2-6x acceleration in inference latency at 75% sparsity. Building upon the DejaVu approach, ReLU^2(Zhang et al., 2024) first uses TDA on non-ReLU models and achieved nearly 70% of sparsity with almost no loss to model performance. ProSparse(Song et al., 2024) proposed a practical DA inference framework and, based on ReLU^2, achieved only a 1-percent increase in perplexity at approximately 80% of sparsity by replacing the activation function and continuing to induce sparsity.

2.3 RIDA

Router-in-the-loop is the predominant method within the Mixture of Experts (MoE) framework. Unlike TDA and RODA methods, most RIDA approaches depend on training an expert router to facilitate dynamic activation.

MoE(Team, 2023) transforms feed-forward networks (FFNs) into MoEs. This approach involves constructing experts and training an additional gating network for expert routing. DS-MoE(Pan et al.,

2024) introduces a framework that employs dense computation during training and switches to sparse computation during inference. It showcases improved parameter efficiency over traditional sparse MoE methods and significantly cuts down the total parameter count. Learn-To-be-Efficient(Zheng et al., 2024) achieves a superior balance between sparsity and performance by activating fewer neurons and it is applicable to models with both ReLU and non-ReLU activation functions. Lory(Zhong et al., 2024) retains the autoregressive properties of language models by adopting a causally segmented routing strategy and a similarity-based data batching method, which enables efficient expert merging operations and promotes specialization among experts in processing similar documents during training sessions.

3 MOYU

Section 2 provided a review of the literature pertinent to MOYU. This section begins with outlining the theoretical foundations of MOYU and then presents evidence of the limitations inherent in the RODA method when applied to non-ReLU activation architecctures, as well as the necessity of incorporating sequence information in the RIDA method. Building upon these insights, this section then introduces our two methods: sTDA and sRIDA.

3.1 Unveiling MOYU

Following literature(Li et al., 2023), we can demonstrate through the following derivation how massive over-activation arises and why SwiGLU cannot produce greater sparsity than ReLU.

Assuming a neural network as in Equation 1:

$$f(x) = \boldsymbol{V}\sigma(p(\boldsymbol{x};\boldsymbol{\theta})) \tag{1}$$

,where $\boldsymbol{V} = [v_1, ..., v_{d_{ff}}]$ is network parameter for the last layer drawn from a random distribution, $\sigma()$ is the SwiGLU activation function, and $p(\boldsymbol{x};\boldsymbol{\theta})$ denotes all other layers with parameter θ. We write $p = p(\boldsymbol{x};\boldsymbol{\theta})$ for simplicity.

Consider the cross-entropy (CE) loss with function $\ell_{CE}(f(\boldsymbol{x}), \boldsymbol{y})$, where $\boldsymbol{y}$ is an arbitrary vector that sums up to one and independent of $\boldsymbol{V}$. Assume that the entries of $\boldsymbol{V}$ are drawn from independent distributions, the probability of any entry of $\boldsymbol{V}$ being 0 is less than 1, and $E[\boldsymbol{V}] = 0$. If there exist an i^* such that $p_{i^*} > 0$, then we have Equation 2:

$$\frac{\partial \ell}{\partial p_{i*}} = \left\langle \frac{\partial \ell}{\partial f}, \frac{\partial f}{\partial p_{i*}} \right\rangle = \left\langle \frac{\partial \ell}{\partial f}, v_{i^*} \right\rangle \tag{2}$$

Substituting CE loss function into Equation 2 yields Equation 3:

$$\begin{aligned} \frac{\partial \ell_{CE}}{\partial f} &= \frac{exp(f(x))}{\langle exp(f(x)), \mathbf{1}\rangle} - y \\ &= \frac{exp(\sum_i \sigma(p_i) \cdot \boldsymbol{v_i})}{\langle exp(\sum_i \sigma(p_i) \cdot \boldsymbol{v_i}), \mathbf{1}\rangle} - y \end{aligned} \tag{3}$$

By substituting Equation 3 back into Equation 2, we can obtain Equation 4:

$$\frac{\partial \ell_{CE}}{\partial p_{i^*}} = \frac{\langle exp(\sum_i \sigma(p_i) \cdot \boldsymbol{v_i}), \boldsymbol{v_{i^*}}\rangle}{\langle exp(\sum_i \sigma(p_i) \cdot \boldsymbol{v_i}), \mathbf{1}\rangle} - \langle \boldsymbol{v_{i^*}}, y\rangle \tag{4}$$

Expanding the numerator of Equation 4 yields Equation 5. In Equation5, we assume that parameter θ and τ have no negative features. If we have $p^0_{i^*} = Swish_1(x\theta) \odot (x\tau)$ and $p^1_{i^*} = ReLU(x)$ respectively, it is easy to get $Swish_1(x\theta) < x\theta$ when $x > 0$, and $p^0_{i^*} < x\theta = p^1_{i^*}$ and $p^0_{i^*} < x\tau$ holds true.

Similar to literature(Li et al., 2023), we also have $\mathrm{E}[\frac{\partial \ell_{CE}}{\partial p_{i^*}}] > 0$ holds true since the expectation of $\mathbf{V}$ is zero and the transformation of the activation function does not change the non-negative property of the loss expectations.

$$\mathrm{E}[\frac{C_1 V \cdot exp(pV)}{C_2\ exp(pV) + C_3}] = \mathrm{E}[\frac{C_1 V}{C_2 + C_3 exp(-pV)}] \tag{6}$$

The first term on the right-hand side(RHS) of the loss function(in Equation 4)'s expectation can be simplified to the form of Equation 6, while the expectation of the second term on the RHS is zero. With respect to $p^0_{i^*} < p^1_{i^*}$, we have Equation 6 demonstrates that when the activation function is switched from ReLU to SwiGLU, the expected value of the loss function will decrease.

That is to say: if there exist an i^* such that $p_{i^*} > 0$, the gradient of CE loss with respect to any positive activation $p_{i^*} > 0$ is positive in expectation. Therefore, any training algorithm based on negative gradient directions tends to *reduce the magnitude* of such *positive activation*, since it will lead to a smaller training loss, and thus causes *sparsity*. And ReLU activation function will cause a bigger magnitude reduction that SwiGLU in this process.

3.2 Sequencing MOYU

In Section 3.1, this paper has theoretically deduced the root causes of the MOYU phenomenon and explored how non-ReLU activation functions might mitigate it. The literature(Georgiadis, 2019; Kurtz

$$\left\langle exp(\sum_i \sigma(p_i)\cdot \boldsymbol{v_i}), \boldsymbol{v_{i^*}} \right\rangle = \sum_m (v_{i^*,m}\cdot exp(\sum_i \sigma(p_i)\cdot v_{im}) \\ = \sum_m (v_{i^*,m}\cdot exp(p_{i^*}\cdot v_{i^*m})\cdot exp(\sum_{i\neq i^*}\sigma(p_i)\cdot v_{im}) \tag{5}$$

et al., 2020; Zhu et al., 2023) has also highlighted that the current level of activation map sparsity is not sufficient to fully unlock the performance of DA methods. In this section, we figure out two limitations when choosing DA methods in Section 3.2.1 and 3.2.2, and then introduce two viable sequential MOYU-based methods called sTDA and sRIDA as simple and training-free methods for dynamic activation.

3.2.1 History-related Activation Uncertainty

RODA schemes excels in models that utilize ReLU as the activation function(Mirzadeh et al., 2023; Liu et al., 2023b; Zhang et al., 2024; Song et al., 2024). However, in models employing non-ReLU activation functions, the offline-trained router struggles to accurately select which heads and neurons will be activated(Ma et al., 2024; Dong et al., 2024).

We suggest that the failure of the RODA in non-ReLU scenarios is closely linked to the shifts in weight importance under different history inputs: a router trained on different historical activation data may find it difficult to accurately identify the weights that are most crucial for the current input.

Similarly, we assume the presence of a ReLU-activated model as described in Equation 1. And the simplified current loss of input token x_i can be described as (Equation 7):

$$L_i = (\frac{\partial f}{\partial x_i}\mathrm{d}x_i + \frac{\partial f}{\partial \theta_i}\mathrm{d}\theta_i)^T(\frac{\partial f}{\partial x_i}\mathrm{d}x_i + \frac{\partial f}{\partial \theta_i}\mathrm{d}\theta_i) \tag{7}$$

Weight change sensitivity (gradients) in model training is as Equation 8:

$$\frac{\partial L_i}{\partial \mathrm{d}\theta_i} = 2(\frac{\partial f}{\partial x_i}\mathrm{d}x_i + \frac{\partial f}{\partial \theta_i}\mathrm{d}\theta_i)\frac{\partial f}{\partial \theta_i} \tag{8}$$

By summing gradients, we have Equation 9:

$$\nabla_{\mathrm{d}\theta_i} L = \sum_i 2(\frac{\partial f}{\partial x_i}\mathrm{d}x_i + \frac{\partial f}{\partial \theta_i}\mathrm{d}\theta_i)\frac{\partial f}{\partial \theta_i} \\ = \nabla_{\mathrm{d}\theta_i} L_i + \sum_{j=0:i-1} \nabla_{\mathrm{d}\theta_j} L_j \tag{9}$$

And the importance of model weights can be described in Equation 10:

$$\Theta_i = \sum_i |V\cdot \nabla_{\mathrm{d}\theta_i} L_i| \\ = |V|\cdot \sum_i |\nabla_{\mathrm{d}\theta_i} L_i| \\ = |V|\cdot (\nabla_{\mathrm{d}\theta_i} L_i + \sum_{j=0:i-1} \nabla_{\mathrm{d}\theta_j} L_j) \\ = |V|\cdot \nabla_{\mathrm{d}\theta_i} L_i + \Theta_{i-1} \tag{10}$$

, which means weight importance of a model are not only related to current input along the direction of θ, but also to the cumulative gradient information from all previous data.

For models utilizing ReLU activation, Equation 10 can be simplified to the sum of the weights corresponding to positive inputs, which linearly correlates with the magnitude of the current weights themselves. However, for models employing non-ReLU activations, the significance of the current weights becomes considerably more complex.

3.2.2 Semantic-irrelevant Activation Inertia

By using simplified loss function, Section 3.2.1 demonstrated that models with non-ReLU activations rely on historical information to accurately decide which neurons will be activated. This section reveals that historical information is significantly influenced by the Heavy Hitter(H_2) and the occurrence of H_2 is not related to semantics(Sun et al., 2024).

Following literature(Zhang et al., 2023) we have $H_2 : S^* \subset [m]$, and $k = |S^*|$, $\tau \in (0, 1)$ denote a threshold. $\alpha \in (0, 1)$ denote a fraction of mass (larger than τ) outside S^*.

It is natural that attention with H_2 is a (α, τ, k)-good mapping since for all $x \in \mathbb{R}^d$, $S^* \subset supp_\tau(Att(x))$, and $|supp_\tau(Att(x)) \setminus S^*| \leq \alpha \cdot k$. Then we have $S^* \subseteq \cap_{i\in[n]} supp_\tau(x_i)$, and $|(\cup_{i\in[n]} supp_\tau(Att(x))) \setminus S^*| \leq \alpha kn$ for x_i draw from (α, τ, k)-good distribution uniformly at random. That is to say, H_2 in a sequence significantly decides the activation pattern. Figure 1 to Figure 4 demonstrate the existence of activation inertia and its irrelevance to semantics. Figures 1 and 2 illus-

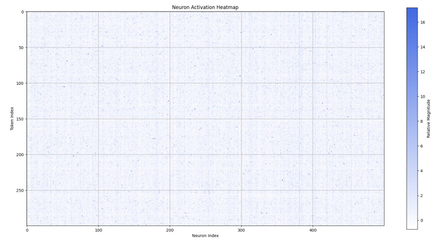

Figure 1: Active neuron of sentence tokens in paralell

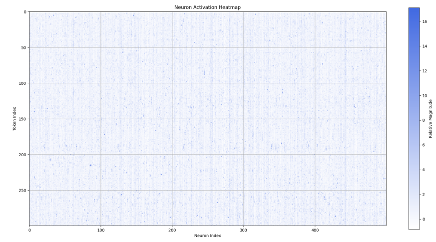

Figure 2: Active neuron of sentence tokens in sequence

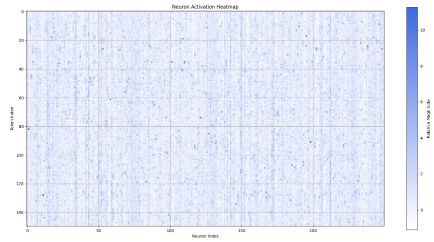

Figure 3: Active neuron of random tokens in paralell

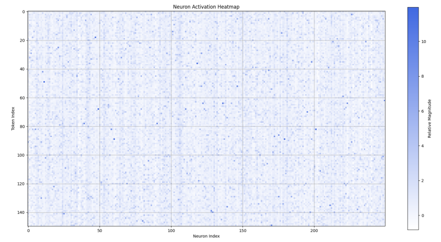

Figure 4: Active neuron of sentence tokens in sequence

trate the active neurons when tokens from a sentence are input either separately or as a sequence. Figures 3 and 4, on the other hand, display the active neurons when tokens from a random word list are fed in the same manner. It is observed that during sequential input, neuronal activation becomes more focused. Furthermore, random words tends to intensify this trend of concentrated activation.

3.2.3 Sequential MOYU

Using our insight on sequence activation, we introduce sequential TDA and RIDA methods called Sequential MOYU as a simple and training-free method for dynamic activation. Shortly, we activate neurons in generation based on sequential information.

MOYU-based Sequential TDA. As previously mentioned in section 2, TDA leverages an offline-decided thresholds to determine which LLMs heads or weights under different inputs should be retained. TDA offers the advantage of having minimal impact on the model's performance. However, a notable drawback is its dependency on the online computation of some values of neurons or heads and the threshold, typically requiring multiple network projections. But sMOYU addresses this issue by shifting the computation from a token-by-token basis to a sequence-based approach.

Following the approach outlined in DejaVu(Liu et al., 2023b) and ReLU^2(Zhang et al., 2024), sMOYU can be represented as follows.

The formula for LLaMA's MLP block can be described in Equation 11 given an input x:

$$MLP(x) = W^{out}\left[\sigma(W^{in}x) \odot (V^{in}x)\right] \quad (11)$$

, where the output of the i-th neuron can be defined as Equation 12:

$$n_i(x) = \left[\sigma(W^{in}_{i,:}x) \odot (V^{in}_{i,:}x)\right] W^{out}_{:,i} \quad (12)$$

From Equation 11 and Equation 12, it can be easily obtained that (Equation 13):

$$MLP(x) = \textstyle\sum_{i=1}^{d_h} n_i(x) \quad (13)$$

, where d_h is the dimension of the hidden layer in MLP block. Therefore, the formula for CETT(cumulative errors of tail truncation) is as follows in Equation 14:

$$\begin{gathered} CETT(x) = \frac{||\sum_{i\in\mathcal{D}} n_i(x)||_2}{||MLP(x)||_2}, \\ \mathcal{D} = \{i|\ ||n_i(x)||_2 < \epsilon\} \end{gathered} \quad (14)$$

, where ϵ represents the threshold, $\mathcal{D}$ is the set of neurons with magnitudes less than the threshold ϵ, and n_i denotes the output of the i-th neuron from Equation 12. Generally, the CETT is empirically set at 0.2, after which the maximum ϵ achievable is calculated to determine the threshold.

MOYU-based Sequential RIDA. Literature on MoE(Shazeer et al., 2017; Team, 2023; Pan et al., 2024; Zheng et al., 2024; Guo et al., 2024) serve as practical examples of general RIDA as in Figure 5. For MOYU-based sRIDA in Figure 6, computations are executed on the initial phase of the sequence, referred to as "prompt" in literature(Dong et al., 2024), but note the tokens within this sequence may not have semantic connections. Based on these initial calculations and sampling strategy of DA, the router determines which neurons(or heads) to activate. The information from the prompt is then relayed through these activated neurons(or heads) to generate subsequent content.

The router, a crucial component in dynamic activation, adjusts the model's activation path in real-time based on the input data. This DA mechanism enables the model to allocate computational resources more flexibly and dynamically select the most suitable neurons or experts for processing varying input data. While the concepts of RODA and RIDA are both implemented in the dynamic activation of attention heads, such as MoA(Wang et al., 2024), the routing of heads often involves complex management of KV Cache and can significantly impair model performance(Ma et al., 2024), which is not discussed in this paper at present.

4 Experiments

4.1 Setups

Our approach, along with the baseline models, is implemented using the PyTorch framework, and we leverage the Hugging Face Transformers library for model and dataset management. Our experiments are powered by eight NVIDIA A100 GPUs, each with 80 GB of memory. Adhering to the methodologies outlined in Section 3.2.3, we sequentially applied our methods for each Transformer layers, which reduces inference latency while preserving model performance. All experiments are conducted in a single phase, without any post-training or fine-tuning stages.

Models, Datasets. In this paper, we conducted a comprehensive series of experiments using the LLaMA-2-7B and LLaMA-3-8B models. These models represent a significant advancement in language modeling capabilities, providing a spectrum of scales to meet various computational needs and performance benchmarks.

Our experimentation focused on subset of two of the most commonly used language datasets: Wikitext-2 and the XSum. Wikitext-2 is renowned for its collection of high-quality, well-structured textual data, predominantly comprising Wikipedia articles. XSum is a comprehensive text summarization corpus that includes approximately 400,000 extensive articles and their corresponding summaries, primarily sourced from CNN and Daily Mail. This dataset challenges summarization models to comprehend the text thoroughly, capture essential information, and produce accurate and coherent summaries. Our experiments were designed to assess the data-based TDA or RIDA activation and related performance of the models on these datasets under sTDA and sRIDA settings.

Our experiments focused on two of the most widely used language datasets: Wikitext-2 and XSum. Wikitext-2(Merity et al., 2016) is known for its high-quality, well-structured textual data, primarily consisting of Wikipedia articles. XSum(Narayan et al., 2018), meanwhile, is a comprehensive text summarization corpus featuring approximately 400,000 extensive articles and their corresponding summaries, mainly sourced from CNN and the Daily Mail. This dataset poses a significant challenge to summarization models, requiring them to thoroughly understand the text, capture essential information, and generate accurate and coherent summaries. Our experimental design aimed to evaluate the performance of TDA, sTDA, RIDA and sRIDA activation in processing these datasets under the MOYU background.

Baselines. In our analysis, we evaluate the standard TDA(Zhang et al., 2024), sTDA, and sRIDA approaches. Unless specified otherwise, each technique is applied in a layer-wise manner, enhancing scalability even when dealing with exceptionally large models.

Sparsity. In our evaluation, we specifically focus on the MLP blocks of LLaMA models, which constitute approximately 67% of the parameters of model's two main blocks, making them a crucial target for dynamic activation. We investigate three distinct types of Dynamic Activation (DA): TDA, sTDA and sRIDA. This approach facilitates a more comprehensive comparison and deeper understanding of how different DA methods affect the performance of LLMs.

Evaluation Metrics. In this study, we concentrate on the impact of Dynamic Activation (DA) on model performance, assessing it through two primary metrics: classification accuracy and generative performance using the Rouge metric family.

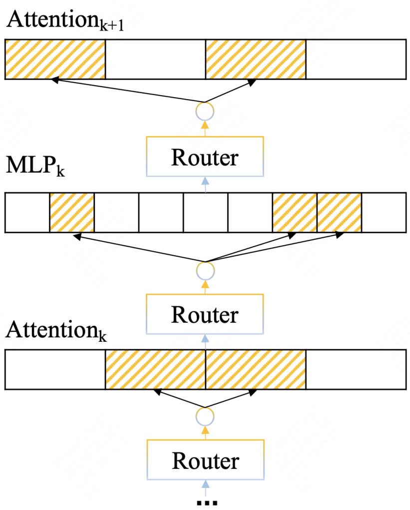

Figure 5: MoE RIDA

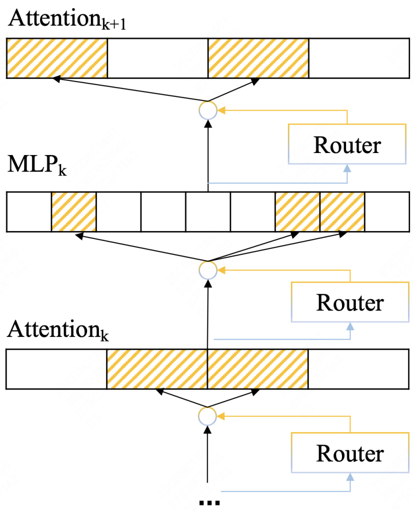

Figure 6: MOYU-based sRIDA

4.2 Performance

Table 1 displays the performance of LLaMA-2-7B and LLaMA-3-8B across four distinct datasets: MMLU, TruthfulQA, Winogrande, and GSM8K. It compares the efficacy of four different Dynamic Activation (DA) settings: the original dense model, TDA, sTDA, and sRIDA. The effectiveness of each method is evaluated based on classification accuracy, which is denoted as "acc(%)" and expressed as a percentage.

Methods	MMLU	TruthfulQA	Winogrande	GSM8K
LLaMA-2-7B	45.83	61.04	74.11	13.95
TDA	45.62	60.66	73.88	13.65
sTDA	43.59	59.26	73.21	12.31
sRIDA	42.28	56.92	70.64	10.00
LLaMA-3-8B	66.60	56.11	76.64	49.13
TDA	63.89	55.64	75.37	44.66
sTDA	61.37	50.81	75.18	43.39
sRIDA	60.74	49.02	74.29	40.81

Notes: The sparsity of TDA and sTDA methods for LLaMA-2-7B is 67.12%, for LLaMA-3-8B is 45.84%. The sparsity of sRIDA methods for LLaMA-2-7B is 56.17%, for LLaMA-3-8B is 40.25%.

Table 1: Classification acc(%) across different methods

From this table, it is evident that for the LLaMA-2 and LLaMA-3 models, the layer-wise TDA method best preserves model accuracy. However, as highlighted in the previous chapter, the token-level layer-wise TDA method involves calculating the values for all neurons initially and then comparing these results to a predetermined threshold. This computationally intensive process significantly diminishes the benefits of DA provided by the TDA method. In contrast, the sequence-level sTDA and sRIDA methods only require calculations for selected tokens, thereby mitigating this computational burden. Nevertheless, implementing DA at the sequence level also slightly compromises model performance. Additionally, since Table 1 presents classification results and the final response involves only one token, the advantages of the sTDA and sRIDA methods are not fully demonstrated in this scenario. In Table 2, mild drops in generation metrics on XSum can also be witnessed for LLaMA-2-7B models. It is evident that under the 1-shot scenario, there is a noticeable decline in model performance. This decline occurs because the maximum length set for this experiment is shorter than the prompt length for the XSum 1-shot, resulting in the overwhelming of effective information, which leads to suboptimal model performance. However, the performance in the 0-shot scenario aligns with expectations.

Methods	ROUGE-1	ROUGE-2	ROUGE-L	1-shot R-1
LLaMA-2-7B	25.81	8.24	21.83	27.15
TDA	24.46	7.99	21.01	13.65
sTDA	22.13	6.92	18.32	12.31
sRIDA	23.92	7.19	20.00	10.17

Notes: The sparsity of TDA and sTDA methods for LLaMA-2-7B is 67.12%. The sparsity of sRIDA methods for LLaMA-2-7B is 56.17%.

Table 2: Generation rouge on XSum

4.3 Efficiency

In Table 3, a batch size of 1 is used for these experiments. Utilizing Hugging Face implementations of LLaMA-2-7B at FP16 precision, we measure latency across various scenarios on a single NVIDIA A100 GPU. Table 3 reveals that al-

Model	Setup	Sparsity	Latency(s)
LLaMA-2-7B	1024+128	67.12%	4.11
	1024+1024	67.12%	132.88
TDA	1024+1024	67.12%	126.23
sTDA	1024+1024	67.12%	91.25
sRIDA	1024+128	56.17%	3.18

Notes: TDA and sTDA methods here is conducted in a model-wise manner.

Table 3: Generation latency across different methods

though the sRIDA method exhibits lower sparsity, it records the lowest latency, suggesting a potential advantage in terms of generation speed.

4.4 Ablations and Analysis

Ablation of Different Input Length. We suppose that both sTDA and sRIDA methods become less robust for generation tasks when the prompt is shorter. Building on the 0-shot scenario in XSum, we further reduced the prompt length to examine changes in evaluation metrics. Notably, the 1-shot scenario in Table 2 was compromised by prompt length limitations, leading to an underestimation of the model's generative capabilities. In Table 4, we

Model	Setup	ROUGE-1	1-shot R-1
LLaMA-2-7B	512+128	18.29	20.31
TDA	512+512	17.27	20.07
sTDA	512+512	14.72	15.83
sRIDA	512+128	16.93%	15.72

Table 4: Generation rouge on XSum with shorter prompt

employed a truncation strategy to ensure that the 1-shot and content each occupy half of the prompt space. From this table, we arrived at a conclusion similar to that of Table 2: TDA remains the most accurate method. Additionally, the sTDA method demonstrates the most significant performance improvement when transitioning from 0-shot to 1-shot.

Ablations of Heavy Hitters. In Table 5, this paper follows the methodology of (Zhang et al., 2023) by eliminating heavy hitters and assessing their impact on classification metrics. The data in Table 5 illustrates that after the removal of heavy hitters, the classification accuracy of all model-wise DA methods declined significantly, with the TDA method experiencing the most substantial decrease. This decline is attributed to the TDA method's direct influence on the selection of the most critical neurons once heavy hitters are eliminated. Conversely,

Methods	MMLU	TruthfulQA	Winogrande	GSM8K
LLaMA-2-7B	38.83	52.04	66.11	-
TDA	33.94	55.00	63.18	-
sTDA	29.83	48.17	51.11	2.16
sRIDA	39.22	50.72	63.84	8.00

Table 5: Classification acc(%) without heavy hitter

the sRIDA method exhibits a smaller reduction in accuracy compared to the other methods, making the underlying reasons for this discrepancy worthy of further investigation.

5 Conclusion

Massive Over-activation Yielded Uplifts (MOYU) are intrinsic characteristics of large language models, and leveraging these properties through Dynamic Activation (DA) is a promising yet underutilized strategy to enhance inference speeds in these models. Traditional methods that exploit MOYU often encounter significant challenges, including maintaining model performance, speeding up inference, or extending their use to various architectures. This paper introduces two novel Sequential DA techniques, sTDA and sRIDA, which utilize sequence data to effectively address the challenges faced by existing DA methods, often referred to as the "impossible triangle." These methods have successfully increased generation speeds by 20-25% without substantially degrading task performance.

In addition to our sequential strategies based on MOYU, named sTDA and sRIDA, we have developed a mathematical framework that elucidates the origins of the MOYU phenomenon. Through this framework, we have identified two primary limitations of current DA methods: 1) their reliance on ReLU activation functions; 2) their inability to detect active neurons based on semantic similarities.

Limitations

Firstly, the mathematical rationale and implementation of the proposed DA methods could introduce complexities that might impede their practical application. Additionally, this paper highlights that sequence-level activation is predominantly influenced by heavy hitters within the same sequence; however, due to length constraints, this ablation experiment was not conducted. Lastly, the datasets and the volume of data utilized in this study are relatively limited. It is anticipated that future research will undertake more extensive experiments.

References

Harry Dong, Beidi Chen, and Yuejie Chi. 2024. Prompt-prompted mixture of experts for efficient llm generation. *Preprint*, arXiv:2404.01365.

Rishi Bommasani et.al. 2022. On the opportunities and risks of foundation models. *Preprint*, arXiv:2108.07258.

Jonathan Frankle and Michael Carbin. 2019. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *Preprint*, arXiv:1803.03635.

Georgios Georgiadis. 2019. Accelerating convolutional neural networks via activation map compression. *Preprint*, arXiv:1812.04056.

Yongxin Guo, Zhenglin Cheng, Xiaoying Tang, and Tao Lin. 2024. Dynamic mixture of experts: An auto-tuning approach for efficient transformer models. *Preprint*, arXiv:2405.14297.

Mark Kurtz, Justin Kopinsky, Rati Gelashvili, Alexander Matveev, John Carr, Michael Goin, William Leiserson, Sage Moore, Nir Shavit, and Dan Alistarh. 2020. Inducing and exploiting activation sparsity for fast inference on deep neural networks. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5533–5543. PMLR.

Pingzhi Li, Zhenyu Zhang, Prateek Yadav, Yi-Lin Sung, Yu Cheng, Mohit Bansal, and Tianlong Chen. 2024. Merge, then compress: Demystify efficient smoe with hints from its routing policy. *Preprint*, arXiv:2310.01334.

Zonglin Li, Chong You, Srinadh Bhojanapalli, Daliang Li, Ankit Singh Rawat, Sashank J. Reddi, Ke Ye, Felix Chern, Felix Yu, Ruiqi Guo, and Sanjiv Kumar. 2023. The lazy neuron phenomenon: On emergence of activation sparsity in transformers. *Preprint*, arXiv:2210.06313.

Ziang Liu, Genggeng Zhou, Jeff He, Tobia Marcucci, Li Fei-Fei, Jiajun Wu, and Yunzhu Li. 2023a. Model-based control with sparse neural dynamics. *Preprint*, arXiv:2312.12791.

Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, and Beidi Chen. 2023b. Deja vu: Contextual sparsity for efficient llms at inference time. *Preprint*, arXiv:2310.17157.

Chi Ma, Mincong Huang, Chao Wang, Yujie Wang, and Lei Yu. 2024. Dynamic activation pitfalls in llama models: An empirical study. *Preprint*, arXiv:2405.09274.

Eran Malach, Gilad Yehudai, Shai Shalev-Shwartz, and Ohad Shamir. 2020. Proving the lottery ticket hypothesis: Pruning is all you need. *Preprint*, arXiv:2002.00585.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

Iman Mirzadeh, Keivan Alizadeh, Sachin Mehta, Carlo C Del Mundo, Oncel Tuzel, Golnoosh Samei, Mohammad Rastegari, and Mehrdad Farajtabar. 2023. Relu strikes back: Exploiting activation sparsity in large language models. *Preprint*, arXiv:2310.04564.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *Preprint*, arXiv:1808.08745.

Bowen Pan, Yikang Shen, Haokun Liu, Mayank Mishra, Gaoyuan Zhang, Aude Oliva, Colin Raffel, and Rameswar Panda. 2024. Dense training, sparse inference: Rethinking training of mixture-of-experts language models. *Preprint*, arXiv:2404.05567.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *Preprint*, arXiv:1701.06538.

Chenyang Song, Xu Han, Zhengyan Zhang, Shengding Hu, Xiyu Shi, Kuai Li, Chen Chen, Zhiyuan Liu, Guangli Li, Tao Yang, and Maosong Sun. 2024. Prosparse: Introducing and enhancing intrinsic activation sparsity within large language models. *Preprint*, arXiv:2402.13516.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. 2024. Massive activations in large language models. *Preprint*, arXiv:2402.17762.

LLaMA-MoE Team. 2023. Llama-moe: Building mixture-of-experts from llama with continual pre-training.

Kuan-Chieh Wang, Daniil Ostashev, Yuwei Fang, Sergey Tulyakov, and Kfir Aberman. 2024. Moa: Mixture-of-attention for subject-context disentanglement in personalized image generation. *Preprint*, arXiv:2404.11565.

Zhihang Yuan, Yuzhang Shang, Yang Zhou, Zhen Dong, Zhe Zhou, Chenhao Xue, Bingzhe Wu, Zhikai Li, Qingyi Gu, Yong Jae Lee, Yan Yan, Beidi Chen, Guangyu Sun, and Kurt Keutzer. 2024. Llm inference unveiled: Survey and roofline model insights. *Preprint*, arXiv:2402.16363.

Zhengyan Zhang, Yixin Song, Guanghui Yu, Xu Han, Yankai Lin, Chaojun Xiao, Chenyang Song, Zhiyuan Liu, Zeyu Mi, and Maosong Sun. 2024. Relu^2 wins: Discovering efficient activation functions for sparse llms. *Preprint*, arXiv:2402.03804.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023. H_2O: Heavy-hitter oracle for efficient generative inference of large language models. *Preprint*, arXiv:2306.14048.

Haizhong Zheng, Xiaoyan Bai, Xueshen Liu, Z. Morley Mao, Beidi Chen, Fan Lai, and Atul Prakash. 2024. Learn to be efficient: Build structured sparsity in large language models. *Preprint*, arXiv:2402.06126.

Zexuan Zhong, Mengzhou Xia, Danqi Chen, and Mike Lewis. 2024. Lory: Fully differentiable mixture-of-experts for autoregressive language model pre-training. *Preprint*, arXiv:2405.03133.

Zeqi Zhu, Arash Pourtaherian, Luc Waeijen, Egor Bondarev, and Orlando Moreira. 2023. Star: Sparse thresholded activation under partial-regularization for activation sparsity exploration. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4554–4563.