Linear Transformers with Learnable Kernel Functions are Better In-Context Models

Yaroslav Aksenov[♣♦], **Nikita Balagansky**[♣♥], **Sofia Maria Lo Cicero Vaina**[♠*], **Boris Shaposhnikov**[♣], **Alexey Gorbatovski**[♣], **Daniil Gavrilov**[♣]
[♣]Tinkoff [♦]Higher School of Economics
[♥]Moscow Institute of Physics and Technology [♠]Innopolis University
n.n.balaganskiy@tinkoff.ru

Abstract

Advancing the frontier of subquadratic architectures for Language Models (LMs) is crucial in the rapidly evolving field of natural language processing. Current innovations, including State Space Models, were initially celebrated for surpassing Transformer performance on language modeling tasks. However, these models have revealed deficiencies in essential In-Context Learning capabilities – a domain where the Transformer traditionally shines. The Based model emerged as a hybrid solution, blending a Linear Transformer with a kernel inspired by the Taylor expansion of exponential functions, augmented by convolutional networks. Mirroring the Transformer's in-context adeptness, it became a strong contender in the field. In our work, we present a singular, elegant alteration to the Based kernel that amplifies its In-Context Learning abilities evaluated with the Multi-Query Associative Recall task and overall language modeling process, as demonstrated on the Pile dataset.

1 Introduction

Large Language Models (LLMs) are revolutionizing the field of natural language processing and establishing new benchmarks across various tasks (Touvron et al., 2023; Jiang et al., 2023). Nevertheless, despite their triumphs, most of these models are built on Transformer frameworks that employ attention mechanisms. These mechanisms scale poorly with long text sequences, leading to impractical computational complexity for extending contextual processing (Vaswani et al., 2017; Tay et al., 2021).

To address this constraint, several alternatives to Transformers were proposed. Katharopoulos et al. (2020) suggested replacing the exponential function in the attention mechanism with the kernel function to change the order of computations and thus move away from quadratic complexity of the sequence length. However, when compared to vanilla Transformers, this approach leads to a drop in performance. Furthermore, the kernel function selection is a topic still in need of consideration. An alternative way to define a linear model is to utilize State Space Models (SSMs) (Gu et al., 2022; Smith et al., 2023; Gu and Dao, 2023), which are capable of producing quality that is comparable to Transformers when measured with perplexity on language modeling.

Notably, both Linear Transformers Katharopoulos et al. (2020) and SSMs can be described as Recurrent Neural Networks (RNNs) (Chung et al., 2014; Hochreiter and Schmidhuber, 1997), which have their limitations when it comes to managing lengthy dependencies within texts since memory capacity can be overrun as the volume of information increases. Additionally, while the hidden state of RNNs is larger for Linear Transformers than for SSMs, the latter showed higher text modeling quality. The introduction of the Based model (Arora et al., 2024) attempted to address the abovementioned challenges by utilizing a hybrid architecture (Fu et al., 2023a) based on a Linear Transformer with a novel kernel function derived from a Taylor expansion of an exponential function. Arora et al. (2024) demonstrated that the Based model was less prone to performance issues when working with longer content than other models when assessed on the Multi-Query Associative Recall (MQAR) task. Nonetheless, even the Based model experiences a drop in performance when faced with extensive contexts relative to the conventional transformer architecture.

A profound comprehension of the processes occurring within the Based architectures is essential for their advancement. Upon examining how attention scores are distributed, we argue that the kernel function previously adopted in Based cannot be considered optimal, resulting in limitations when dealing with lengthy context and small model

*Work done while at Tinkoff.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9584–9597

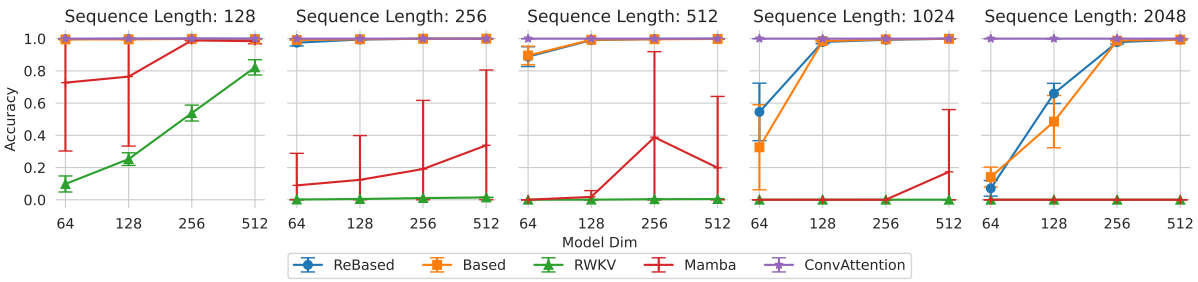


Figure 1: Results on the MQAR dataset, designed to measure In-Context Learning capabilities of an architecture Arora et al. (2023). ReBased outperforms all baselines except Attention across different sequence lengths and model sizes. See Section 5.2 for more details.

capacity. To address this issue, we introduce **ReBased** (Revisited Based), a novel variation of the Linear Transformer model that improves the use of attention kernels. The crux of our development lies in addressing the inability of Based to disregard specific tokens with zero probability during the attention process. By refining the kernel function and incorporating new architectural modifications, we have created a model that improves accuracy on tasks involving retrieving information from long sequences of tokens while simplifying the calculation of the attention mechanism.

When testing our enhanced architecture on the MQAR task, we found that ReBased surpasses the original Based model across a variety of contexts and model sizes. Additionally, after training with the Pile dataset (Gao et al., 2020), we observed that ReBased performs better than its predecessor at In-Context Learning and excels at modeling associative dependencies measured through improved perplexity metrics.

2 Recent Work

The Vanilla Transformer architecture (Vaswani et al., 2017), although widely used in NLP (Radford et al., 2019; Touvron et al., 2023; Devlin et al., 2019; Jiang et al., 2023), suffers from growing computational and memory demands ($O(d*N^2)$ as sequence lengths (N) and head size (d) increase). While this is not much of a problem when it comes to shorter sequences, it becomes a significant bottleneck when working with longer ones.

Several alternative architectures were proposed to address this issue. Katharopoulos et al. (2020) suggested substituting the attention mechanism's exponential function, which is meant to measure the similarity between queries and keys, with a product of kernel functions that can be separately evaluated for queries and keys. This kernel-based approach reshapes the computation within the attention mechanism, cutting the time and memory complexity to $O(d^2 * N)$. Additionally, during inference, it supports sampling sequences with linear complexity regarding length, similar to RNNs (Hochreiter and Schmidhuber, 1997; Chung et al., 2014).

In a different approach, State Space Models (SSMs) borrow from control theory to offer a simplified structure akin to RNNs, but without activation functions across time (Gu et al., 2022; Smith et al., 2023; Gu et al., 2023). The Mamba model, also known as S6 (Gu and Dao, 2023), stands out in this category, displaying enhanced learning of short-term dependencies in texts compared to existing pre-trained LLMs (Jiang et al., 2023; Touvron et al., 2023).

Despite these advancements, there is no standard way to fully evaluate these innovative architectures to assess their performance limits. One standard evaluation method is to pre-train a language model and assess its perplexity with a given dataset, but this may not truly reflect the model's ability to manage long context dependencies. Another option is to use the Long Range Arena (LRA) benchmark, which involves classification tasks with long input sequences. Though some new models have outperformed Transformers in the LRA, it is believed that the benchmark is capable of introducing bias in the comparison (Amos et al., 2023).

One promising evaluation approach is to test an architecture's In-Context Learning abilities. Olsson et al. (2022) introduced the concept of Associative Recall (AR), a task where the model learns to copy a token from a sequence after a certain point. However, while in Fu et al. (2023a) the associative recall task was implemented with a goal to retrieve only one token, Arora et al. (2023) noted that this task could be considered overly simplistic. This led to

the creation of the Multi-Query Associative Recall (MQAR) task, which requires retrieving multiple tokens from the context. Findings on MQAR indicate that while newer models may compete with Transformer in terms of perplexity, they can still struggle with long contexts at small model sizes because of their limited In-Context Learning capabilities. Meanwhile, Transformers remain robust against such factors. Lastly, Arora et al. (2024) introduced Linear Transformer with a new kernel function (namely Based), showcasing enhanced performance on the MQAR task when compared to Mamba.

Despite this improvement, compared to traditional Transformers, the problem of decline in performance when handling long sequences with smaller models still remains. Addressing this challenge is the primary goal of our paper.

3 Background

3.1 Linear Transformers

To fully grasp the Based architecture, it is vital to first discuss the original Transformer model. The attention mechanism, which is central to the Transformer's functionality, evaluates the output y_i for each position i as follows

$$y_i = \frac{\sum_{j=0}^{i} \text{sim}(Q_i, K_j)V_j}{\sum_{n=0}^{i} \text{sim}(Q_i, K_n)},$$

where the term $\text{sim}(Q_i, K_j) = \exp\left(\frac{Q_i^T K_j}{\sqrt{d}}\right)$ represents the similarity between the query Q_i and the key K_j using an exponential function. Despite its effectiveness, the original Transformer's reliance on this attention mechanism incurs a quadratic increase in both computational time and memory use as the sequence length grows, which becomes impractical for processing long sequences.

To address this scalability problem, Katharopoulos et al. (2020) suggested replacing the direct computation of similarity between Q and K with a transformation through a non-linear kernel function $\phi(\cdot)$. This allows for the following approximation: $\text{sim}(Q_i, K_j) \approx \phi^T(Q_i)\phi(K_j)$. By implementing this kernel, the Linear Transformer computes y_i as

$$y_i = \frac{\sum_{j=0}^{i} \phi^T(Q_i)\phi(K_j)V_j}{\sum_{n=0}^{i} \phi(Q_i)\phi^T(K_n)}.$$

By rearranging the operations, we can express the computation as

$$y_i = \frac{\phi^T(Q_i)\sum_{j=0}^{i} \phi(K_j)V_j^T}{\phi^T(Q_i)\sum_{n=0}^{i} \phi(K_n)}.$$

By calculating $\phi(K_j)V_j^T \in \mathbb{R}^{d\times d}$ upfront, the complexity of the attention mechanism transitions to linear with the sequence length, addressing the inefficiencies of the original model.

3.2 Based

Selecting an appropriate kernel function $\phi(\cdot)$ is critical to a Linear Transformer's performance. Various kernel functions have been proposed (Peng et al., 2021; Schlag et al., 2021; Qin et al., 2022), but on language modeling tasks, none have surpassed the original attention mechanism. However, a breakthrough was achieved by Arora et al. (2024), who introduced a novel kernel function inspired by the Taylor series expansion of the exponential function, defined as

$$\text{sim}(q, k) = 1 + q^T k + \frac{(q^T k)^2}{2}.$$

The choice of this kernel is motivated by its ability to approximate the exponential function over a specific range of $q^T k$ values. In addition, Arora et al. (2024) utilized a hybrid architecture by combining linear attention with convolutional layers since doing so was shown to help models handle short non-associative dependencies in the sequences (Fu et al., 2023a; Poli et al., 2023; Fu et al., 2023b)

In doing so, when evaluated on the MQAR task, the Based model demonstrated that it was capable of outperforming the Mamba model (Gu and Dao, 2023) under circumstances of substantial context length and constrained model capacity due to smaller sizes. Nevertheless, compared to the original Transformer, a discernible drop-off in performance remains, indicating room for further improvement.

4 Revisiting Based

In our study, we explore the fundamental requirements for kernel functions. We examine the exponential function and its approximate representation, as depicted in Figure 2. We observe a limitation in the approximation since its minimal value is fixed at 0.5. This is problematic for handling long sequences, as it is difficult to assign a near-zero attention score to specific token pairs. Ideally, we want to be able to diminish the attention scores

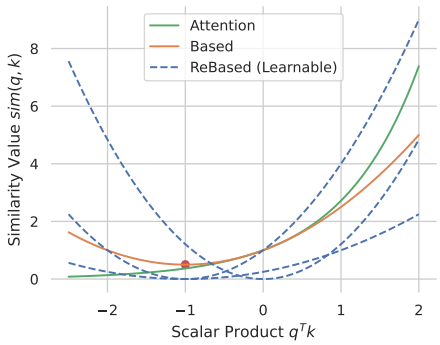


Figure 2: Similarity between q and k with respect to scalar product. Note that the Based model has a minimal $\text{sim}(q, k)$ value of 0.5, which can lead to suboptimal performance. We propose to learn the scale and shift of the parabola jointly with the model and make it possible to zero out the similarity value. See Section 4 for more details and Section 5.1 for experimental setup description.

to zero, which would require significantly larger values elsewhere in the normalization process with the Based model.

To rectify this issue, a straightforward approach would be to adjust the lowest point of the kernel function to zero. However, this solution prompts us to ask why the minimum value of the kernel function should occur precisely at $q^T k = -1$. As used in the original Transformer, the traditional exponential similarity function increases monotonically, but the quadratic kernel has an optimal value to which it decreases and then ascends from. Therefore, to decrease attention in the Transformer, one would aim to minimize $q^T k$. In contrast, the ideal $q^T k$ should be exactly -1 for the Based method. Otherwise, the attention score would increase. This condition may induce less-than-ideal training outcomes and degrade the model's accuracy.

These challenges lead us to conjecture that if the quadratic kernel is used to calculate the similarity between q and k, we must consider the range of potential $q^T k$ values and create adjustable parameters for the parabolic function to align with these values during training. Simplifying for clarity, let us look at a one-dimensional scenario. We can express the trainable parameters of the kernel function in relation to the affine transformation of q and k as such

$$q' = \gamma_Q \cdot q + \beta_Q, \quad k' = \gamma_K \cdot k + \beta_K;$$
$$\text{sim}(q', k') = \phi^T(q')\phi(k').$$

Here, $\phi(\cdot)$ represents a quadratic function. The model can learn any quadratic function with a determined minimum value by adjusting its parameters. We can, therefore, simplify the kernel function to

$$\phi(x) = x^2.$$

Incorporating the affine transformation into the kernel function, we obtain

$$\phi(q') = (\gamma_Q \cdot q + \beta_Q)^2 = \gamma_Q^2 q^2 + 2\gamma_Q \beta_Q q + \beta_Q^2,$$
$$\phi(k') = (\gamma_K \cdot k + \beta_K)^2 = \gamma_K^2 k^2 + 2\gamma_K \beta_K k + \beta_K^2.$$

where q and k have their unique parameters γ_Q, γ_K, β_Q, and β_K, enabling the model to learn any quadratic function that is non-negative and has a single real root.

Interestingly, our transformation resembles the application of Layer Normalization (Ba et al., 2016), minus the normalization itself. We hypothesize whether normalizing q and k before the kernel function could improve the model's performance. Our suspicion is confirmed when normalization enhances results, as demonstrated in a later Ablation study. Consequently, our refined *ReBased* model incorporates Layer Normalization.

In the following sections, we provide an in-depth analysis and conduct comprehensive experiments to validate the effectiveness of these modifications.

5 Experiments

5.1 Experimental Setup

We applied the first evaluation of our ReBased model on the **MQAR task**, for which we trained a model to perform associative recall with varying numbers of retrieved tokens. Arora et al. (2023) suggested that for a comprehensive assessment, models need to be tested across different sequence lengths, model sizes, and number of query-key pairs to be retrieved. However, those experiments were limited, only exploring sequence lengths up to 512. These constraints resulted in the Based model displaying performance comparable to the traditional attention mechanism.

Longer sequence lengths can be explored to gain a deeper understanding of how improvements in associative recall are affected by changes in model configurations. This is why we extended our training to include models capable of handling sequence lengths of $[128, 256, 512, 1024, 2048]$. We tested a range of hidden sizes from 64 to 512. For our

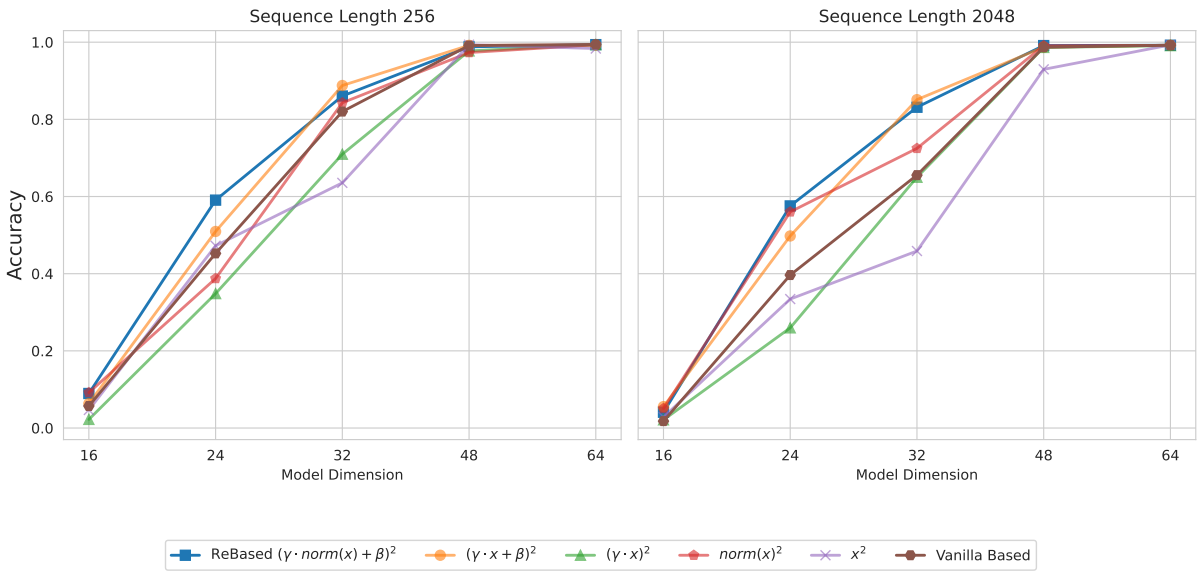


Figure 3: Ablation study for the proposed modifications. For sequence length 256, the difference is not very significant. Nevertheless, the ReBased model performs best on all model dimensions. With a sequence length of 2048, the difference becomes more evident. Unlike Based, the ReBased model retains performance across long and short sequences. See Section 5.3 for the experiment setup and extended description of our results and Section 5.1 for experimental setup description.

ablation study to yield more precise insights, we also employed smaller models with hidden sizes as modest as $[16, 24, 32, 48]$.

In order to tailor our approach to varied sequences, we used different query-key (qk) pairs for each length. The specifics of these configurations are detailed in Appendix A.

We also put other sub-quadratic architectures to the test, including Mamba (SSM family) (Gu and Dao, 2023), Hyena (the long convolutions family) (Poli et al., 2023), the vanilla attention method, and RWKV (Peng et al., 2023). By comparing a diverse range of models, our goal was to present a well-rounded evaluation of how our ReBased model stands out in the field. For Based, we utilized Triton kernels published by Yang and Zhang (2024), and for ReBased, we modified it so that $\phi(x) = x^2$.

We used a hybrid architecture with short convolution and kernel size 3 in the first layer, and specified a mixer in the second. We found that this setup was more stable on longer sequence lengths, especially when using an attention mixer. However, we did not modify the Mamba model since convolutions were already present inside the Mamba block. We put the full results and model architecture details in Appendix A.

In **language modeling**, our second experimental setup leveraged the extensive Pile dataset (Gao et al., 2020) to train a language model (LM). We opted for a sequence length of 4096, a slight increase from the standard value while still ensuring the replication of the architectural framework as presented by Arora et al. (2024)[1]. Note that some hyperparameters such as model dimension and the number of layers were set in order to match the number of model parameters in the initial experiment. Detailed model configuration can be found in Appendix C.

The MQAR task provided insights into In-Context Learning proficiencies across various architectures, while the language modeling assessment allowed us to appraise short-term dependency modeling capacities. Beyond traditional perplexity metrics on validation data, we also scrutinized the Associative (AR) and Non-Associative (Non-AR) variants of perplexity. Here, AR corresponds to token positions necessitating associative recall, while Non-AR refers to other tokens. When tokens recur within a text, the subsequent appearances are categorized as AR, highlighting the model's capability to recall from context.

5.2 MQAR experiment

In Figure 1, we present the capability of various models to handle the MQAR task as the sequence length increases. One key observation is that, at a

[1]The experiment details can be found in a blog post and WandB report associated with the main paper.

Architecture	Sequence Length 256					Sequence Length 2048				
	16	24	32	48	Mean	16	24	32	48	Mean
Based	0.06 ± 0.02	0.45 ± 0.15	0.82 ± 0.06	0.99 ± 0.00	0.58	0.02 ± 0.02	0.40 ± 0.18	0.66 ± 0.08	0.99 ± 0.01	0.51
x^2	0.05 ± 0.05	0.47 ± 0.08	0.64 ± 0.42	0.99 ± 0.00	0.54	0.03 ± 0.03	0.33 ± 0.21	0.46 ± 0.42	0.93 ± 0.09	0.44
$norm(x)^2$	0.09 ± 0.04	0.39 ± 0.24	0.84 ± 0.09	0.97 ± 0.02	0.57	0.05 ± 0.05	0.56 ± 0.10	0.72 ± 0.17	0.99 ± 0.00	0.58
$(\gamma \cdot x)^2$	0.02 ± 0.02	0.35 ± 0.22	0.71 ± 0.09	0.98 ± 0.03	0.51	0.02 ± 0.03	0.26 ± 0.45	0.65 ± 0.37	0.99 ± 0.01	0.48
$(\gamma \cdot x + \beta)^2$	0.06 ± 0.01	0.51 ± 0.08	0.89 ± 0.03	0.99 ± 0.00	0.61	0.06 ± 0.03	0.50 ± 0.08	0.85 ± 0.04	0.99 ± 0.01	0.60
ReBased $_{(\gamma \cdot norm(x) + \beta)^2}$	0.09 ± 0.05	0.59 ± 0.06	0.86 ± 0.08	0.99 ± 0.00	**0.63**	0.04 ± 0.03	0.58 ± 0.01	0.83 ± 0.04	0.99 ± 0.00	**0.61**

Table 1: Ablation study for proposed modifications with standard deviation across 5 seeds. See Figure 3 for a visual presentation of the results, Section 5.3 for experiment setup and extended result description and Section 5.1 for the description of our experimental setup.

sequence length of 2048, all models, except for the Attention model, struggled to perform effectively when limited to a model dimension of 64. As we expanded the model dimensions, the performance of the ReBased model matched or surpassed the Based model. The RWKV and Mamba architectures failed on the MQAR task across all tested model sizes.

This experiment highlights the significance of utilizing more sophisticated setups, as the performance discrepancy between the Attention model and the other models (Based and ReBased) becomes pronounced only when the sequence length exceeds 512. These results suggest that the efficacy of attention alternatives like ReBased becomes particularly important when processing long sequences. Therefore, more consideration should be devoted to configurations involving lengthy sequences to leverage the full potential of such models.

5.3 Ablation Study

We comprehensively examined the individual elements of our ReBased model to understand how each of them contributes to its overall effectiveness, and ensure the transparency of our findings. Our experiments were meticulously designed to evaluate the model by assessing the influence of its separate components on performance. The experimental configurations were as follows:

- x^2 – substituting the original kernel function with a simple element-wise squaring operation, $\phi(x) = x^2$.
- $norm(x)^2$ – integrating a normalization step without an affine transformation before applying the squaring operation.
- $(\gamma \cdot x)^2$ – introducing an affine transformation solely in terms of scaling (without bias) for the queries and keys.
- $(\gamma \cdot x + \beta)^2$ – incorporating affine transformation with both scaling and bias for the queries and keys.
- ReBased $(\gamma \cdot norm(x) + \beta)^2$ – our comprehensive model, which involves normalization and affine transformation, including bias, for queries and keys.

Note that for q and k, there are different scaling parameters $\gamma_Q, \beta_Q, \gamma_K$, and β_K for each experiment involving affine transformation.

Our goal is to highlight the effect of sequence length variability in the MQAR task on model performance. For this evaluation, we standardized the number of retrieval pairs to 32. Theoretically, no impact on performance should be observed, as the amount of information required to be stored in the hidden states is sequence-length agnostic. We investigated the effects on sequences of lengths 256 and 2048 and illustrated our findings in Figure 3 (also available in Table 1 with a standard deviation of accuracy across 5 seeds). We must emphasize the significance of long context setups evaluated in our experiments. Its characteristics are vital, as successful performance on long sequences highlights the capability of the model to make full use of its architectural innovations. It also translates into notable practical advantages in real-world applications where handling extensive context efficiently can be crucial.

The proposed ReBased model performs better than every other modification. Performance on the short 256 length is less noticeable than on the long 2048 sequence length. We see a performance drop from simply replacing the original kernel function with x^2. We presume that this is caused by suboptimal scale of features, since by placing normalization before the kernel function, we can notice a performance increase even in comparison to the Based model. Affine transformations $(\gamma \cdot x)^2$ and

$(\gamma \cdot x + \beta)^2$ also show favorable performance compared to the x^2 model, which does not significantly decrease with sequence length.

5.4 Language Modeling

Architecture	Perplexity All	AR	Non-AR
Attention	11.98	3.07	33.95
Based	12.99	3.27	37.02
ReBased	**12.90**	**3.25**	**36.73**

Table 2: Perplexity results on Pile (Gao et al., 2020) dataset. ReBased improves the result on AR tokens. However, there is still a small gap between Attention and ReBased. See Section 5.4 for more details and Section 5.1 for experimental setup description.

We conducted experiments with language modeling following the setup described in Section 5.1. See Table 2 for the results.

We note that ReBased model performs better than Based on both AR and non-AR tokens leading to lower overall perplexity. This can be considered as a sign of better In-Context Learning performance. In the next section we consider few-shot setup on several tasks to validate

When considering AR perplexity, we observe that there is still a gap between the vanilla Transformer architecture and alternative models, which is aligned with the results on the MQAR dataset. However, we note that ReBased still performed better than Based. Regarding Non-AR perplexity, ReBased outperformed both Based architectures, leading to better overall perplexity value. Note that attention has slightly more trainable parameters, see Appendix C for more details.

These results suggest that, despite language modeling perplexity being lower for an alternative to Transformer architectures (Arora et al., 2024; Gu and Dao, 2023), this may be achieved due to better short-term dependency modeling, which does not require learning associative operations necessary to perform In-Context Learning (Olsson et al., 2022). The vanilla Transformer still performs best in terms of its ability to attend to some token in-context.

5.5 Few-Shot Performance

To further explore the ability of the model to improve results on real-world scenarios we validate trained Based and ReBased models on a common few-shot benchmarks from the LM Evaluation Harness (Gao et al., 2023) benchmark and SuperGLUE (Wang et al., 2019). Results are presented in Table 3 and Table 4. ReBased outperforms Based on the most of the tasks.

5.6 Analysis

In this section, we delve into the internal dynamics of the ReBased model by examining attention matrices, which are commonly used to elucidate the decision-making of models and the flow of information between tokens. Notably, we can use the parallel mode with both Based and ReBased models to construct these matrices.

For our analysis, we employ the MQAR dataset (Arora et al., 2023) and train a compact model configured with a sequence length of 128 and 32 retrieval pairs. To ensure clear interpretation of the attention maps, we used fixed weights in the first layer, which consists of a short convolution with a kernel that attends to the previous token. Following the training phase, we compute the Intersection over the Union (IoU) metric between the attention matrix and the actual positions of the tokens that are to be retrieved. The correct positions are crucial, as they represent the locations from which the model must copy the hidden states in order to successfully resolve the task. This copying mechanism is particularly vital and is implemented via focused attention in the second layer of the network (Olsson et al., 2022). Consequently, the IoU provides a quantitative measure of how well our model has learned to replicate this crucial pattern of token retrieval. A visualization of this phenomenon using IoU on a randomly selected example from the dataset is shown in Figure 4. Note that we cropped attention matrix to examine only a region where qk-pairs stored.

Our results are presented in Table 5. In our experiment, the Attention model yielded a superior IoU score compared to both the Based and ReBased models. However, the ReBased model shows promise in narrowing the performance divide that exists between sub-quadratic methods and the attention-based model. This suggests that, despite the relative simplicity of the method, it could serve as an informative metric for the MQAR dataset, particularly when the accuracy score is close to one, making it challenging to discern the performance differences between models in more intricate testing scenarios.

Architecture	Winogrande	Piqa	Hellaswag	Arc-E	Arc-C	Macro
Based	50.4	62.1	**30.8**	40.4	**22.9**	41.3
ReBased	**54.6**	**62.8**	30.7	**41.0**	21.5	**42.1**

Table 3: 1-shot performance on tasks from LM evaluation harness benchmark (Gao et al., 2023). See Section 5.5 for more details.

Architecture	WSC	WIC	RTE	Record (F1/EM)	MultiRC	Copa	BoolQ
Based	55.8	46.5	47.6	62.7/62.1	51.5	66.0	48.3
ReBased	**56.7**	**46.9**	**53.1**	**62.8/62.2**	**51.9**	**67.0**	**52.0**

Table 4: 1-shot performance on SuperGLUE benchmark (Wang et al., 2019). See Section 5.5 for more details.

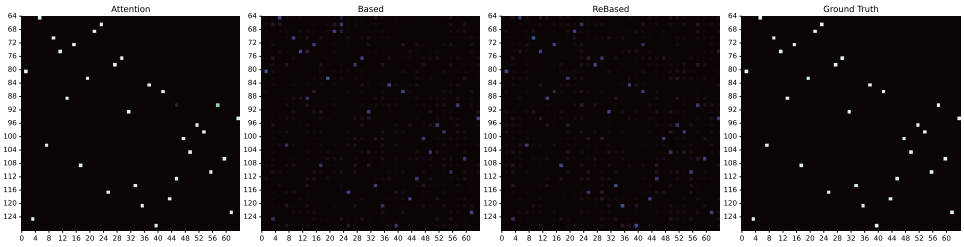


Figure 4: Attention matrix for the different models, and ground truth positions for the query. We measure IoU between model's attention and ground truth matrix for 10000 examples. Illustration of the experiment is described in Section 5.6 Results are presented in Table 5.

Architecture	IoU	Accuracy
Attention	**0.999**	1
Based	0.157	0.956
ReBased	*0.173*	0.957

Table 5: IoU with attention matrix and ground truth position to retrieve on the MQAR task for 10000 examples. Detailed experiment setup can be found in 5.6.

6 Conclusion and Future Work

In this paper, we present ReBased, a novel architecture for sub-quadratic attention computation. For our model, we analyzed the Base architecture and proposed to develop it even further by using polynomial kernels with learnable parameters and adding normalization before the kernel evaluation. While incorporating layer normalization into model training was attempted previously (Henry et al., 2020), our method integrates this normalization directly into the kernel function. With this simple architectural change, we achieved results that outperformed Based on MQAR and language modeling with the Pile dataset tasks.

We analyzed the internal representations of ReBased, Based, and vanilla attention modules, and concluded that ReBased resembles attention more than Based. Notably, while Based uses a Taylor expansion of an exponential function, a ReBased kernel function is different from the exponent but shows better performance. Our research suggests that using a second-order polynomial might be insufficient for the best performance, and indicates that more sophisticated learnable kernels could be utilized to improve the performance of trained models. Normalization could further improve various kernel functions. This highlights a need for researchers to revisit kernel-based methods with the goal of enhancing their adaptability and efficiency.

Our findings reveal a disparity in handling the MQAR task between attention-based models and others such as Based, specifically as sequence lengths increase. Attention models excel on longer sequences, significantly outperforming their non-attention counterparts. These results highlight the necessity of further research into strategies that could bridge this gap in order to reach the performance of attention-based methods. Perhaps the superior aspects of attention mechanisms could be matched or exceeded by other models, especially on tasks that require associative recall, such as machine translation (Vardasbi et al., 2023). Future research could give insight into this, leading to improved models for processing long sequences.

7 Limitations

While our proposed method demonstrates applicability to a wide range of tasks typically addressed by Transformers, its effectiveness in handling tasks involving intensive copying or recalling previous context remains unclear (see Table 2 and Jelassi et al. (2024)). Successfully addressing these tasks is crucial for fully mitigating inference problems associated with attention mechanisms.

It is also worth noting that our experiments are limited to academic-scale models. This does pose certain limitations, particularly in extrapolating the findings to larger models. However, given the resource constraints, our results still provide valuable insights into the potential efficacy of our method.

References

Ido Amos, Jonathan Berant, and Ankit Gupta. 2023. Never train from scratch: Fair comparison of long-sequence models requires data-driven priors. *CoRR*, abs/2310.02980.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. 2023. Zoology: Measuring and improving recall in efficient language models. *arXiv:2312.04927*.

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, Dylan Zinsley, James Zou, Atri Rudra, and Christopher Ré. 2024. Simple linear attention language models balance the recall-throughput tradeoff. *arXiv:2402.18668*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. Cite arxiv:1607.06450.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. Cite arxiv:1412.3555Comment: Presented in NIPS 2014 Deep Learning and Representation Learning Workshop.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2021. Expected validation performance and estimation of a random variable's maximum. *CoRR*, abs/2110.00613.

Daniel Y. Fu, Tri Dao, Khaled K. Saab, Armin W. Thomas, Atri Rudra, and Christopher Ré. 2023a. Hungry Hungry Hippos: Towards language modeling with state space models. In *International Conference on Learning Representations*.

Daniel Y. Fu, Elliot L. Epstein, Eric Nguyen, Armin W. Thomas, Michael Zhang, Tri Dao, Atri Rudra, and Christopher Ré. 2023b. Simple hardware-efficient long convolutions for sequence modeling. *International Conference on Machine Learning*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces.

Albert Gu, Karan Goel, and Christopher Re. 2022. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*.

Albert Gu, Isys Johnson, Aman Timalsina, Atri Rudra, and Christopher Re. 2023. How to train your HIPPO: State space models with generalized orthogonal basis projections. In *International Conference on Learning Representations*.

Alex Henry, Prudhvi Raj Dachapally, S. Pawar, and Yuxuan Chen. 2020. Query-key normalization for transformers. *FINDINGS*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. 2024. Repeat after me: Transformers are better than state space models at copying. *arXiv preprint arXiv: 2402.01032*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are RNNs: Fast autoregressive transformers with linear attention.

In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5156–5165. PMLR.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan Wind, Stanisław Woźniak, Zhenyuan Zhang, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. 2023. RWKV: Reinventing RNNs for the transformer era. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14048–14077, Singapore. Association for Computational Linguistics.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A. Smith, and Lingpeng Kong. 2021. Random feature attention. *ArXiv*, abs/2103.02143.

Michael Poli, Stefano Massaroli, Eric Q. Nguyen, Daniel Y. Fu, Tri Dao, S. Baccus, Y. Bengio, Stefano Ermon, and Christopher Ré. 2023. Hyena hierarchy: Towards larger convolutional language models. *International Conference on Machine Learning*.

Zhen Qin, Weixuan Sun, Huicai Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. 2022. cosformer: Rethinking softmax in attention. *ArXiv*, abs/2202.08791.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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

Jimmy T.H. Smith, Andrew Warrington, and Scott Linderman. 2023. Simplified state space layers for sequence modeling. In *The Eleventh International Conference on Learning Representations*.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021. Long range arena : A benchmark for efficient transformers. In *International Conference on Learning Representations*.

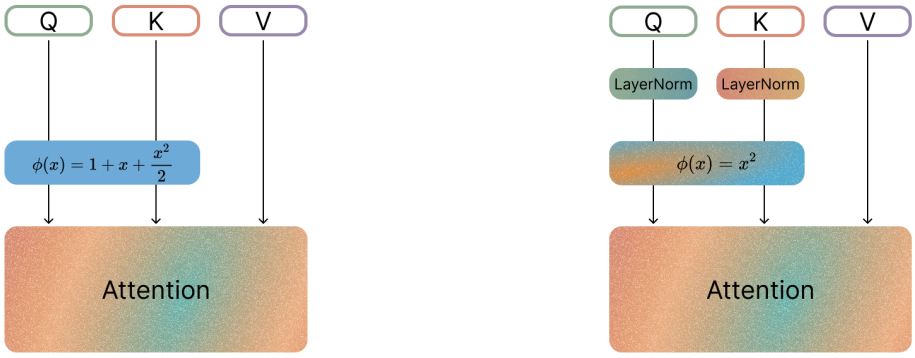
Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, D. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, A. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Ali Vardasbi, Telmo Pessoa Pires, Robin M. Schmidt, and Stephan Peitz. 2023. State spaces aren't enough: Machine translation needs attention. In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation, EAMT 2023, Tampere, Finland, 12-15 June 2023*, pages 205–216. European Association for Machine Translation.

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

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Songlin Yang and Yu Zhang. 2024. Fla: A triton-based library for hardware-efficient implementations of linear attention mechanism.



(a) Based architecture

(b) ReBased architecture.

Figure 5: Architectures visualization.

Model Dimension	Attention	ConvAttention	RWKV	ConvRWKV	Mamba	Based (Rebased)
64	623744	578752	623872	677120	655360	577984 (+768)
128	1313024	1149312	1313280	1395200	1413120	1179520 (+768)
256	2888192	2462464	2888704	2561024	3235840	2459392 (+768)
512	6824960	5580288	6825984	5777408	7847936	5307904 (+768)

Table 6: Number of model parameters in MQAR dataset. See Appendix A.

A Details for the MQAR dataset experiments

In our experiments, we use the code from the official MQAR repository (Arora et al., 2024)[2]. However, we modify the attention model from the one reported in Arora et al. (2024), as we found it more stable (see Figure 6). We can see that replacing the first attention layer is beneficial for performance. RWKV performs better when we do not replace the first layer, which is why we use we use two RWKV layers in our main experiment (see Figure 1). We report the number of trainable parameters in Table 6.

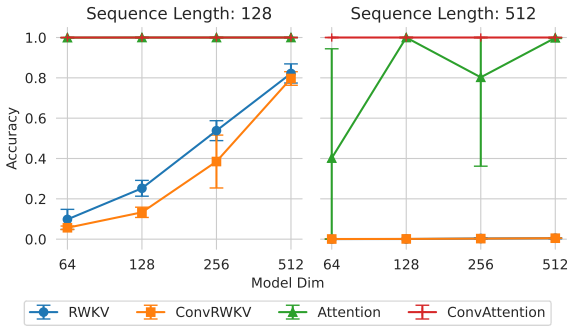


Figure 6: Performance of the hybrid architecture with convolutions on the first layer and the vanilla architecture.

[2]https://github.com/HazyResearch/zoology

We also modify the data configs to be more challenging for the model. You can see the adjusted parameters in Table 7.

seq_length	qk_pairs
128	16
256	64
512	128
1024	256
2048	512

Table 7: Sequence lengths and number of QK pairs in dataset.

We use a batch size of 512 for all experiments. In cases where there is not enough GPU memory, we use the gradient accumulation technique. For the learning rate, we use hyperparameter search with the following grid: 5e-4, 1e-3, 3e-3, 1e-2. We use five different seeds for all reported results.

B Stability

In our experiments, we found ReBased to be more stable during training with various hyperparameters. To demonstrate this, we utilize an Expected Validation Performance (EVP) plot (Dodge et al., 2021). We treat the average across five seeds as the final accuracy. Our results are presented in

Appendix Figure 7. We noticed that even in cases where the model dimension is sufficiently large to store all necessary information, our modifications lead to 100% accuracy for every hyperparameter set and every seed we use, in contrast with the Based model, where we observe degradation for certain learning rates.

C Pile Dataset Experiment Details

Model	# Parameters
Attention	151 880 448
Based	147 542 016
ReBased	147 548 928

Table 8: Parameters count for the pile experiment. See Appendix C.

We train our model on the tokenized Pile dataset published on huggingface hub[3]. Note that this tokenization differs from the one used in Based[4]. We also use our pipeline, which we plan to release to the public soon. We do not use rotary positional embeddings (Su et al., 2024) or other tricks, as we copy our models from the Based repository. Hyperparameters can be found in Table 9.

Hyper-Parameter	Value
warmup steps	200
max grad norm	1
num steps	20000
seq len	4096
lr	1e-3
weight decay	0.1
num heads	12
d model	768
effective batch size	1024

Table 9: Hyperparameters used for training.

As in Arora et al. (2023), we use more hyperparameters in the Based/ReBased models then in the Attention baseline. The number of layers and head dim reported in Tables 10 and 11. We use a hybrid architecture for the Based/ReBased models where we use short convolution as a mixer for every odd-numbered layer.

Hyperparameters	Value
layers	12
head dim	64

Table 10: Attention hyperparameters.

Hyper-Parameter	Value
layers	18
head dim	16

Table 11: Based/ReBased hyperparameters.

D Additional Analysis

In this section, we provide additional results and experiments to further understand how the ReBased model learns dependencies. First, we offer more examples for our experiments with attention matrices, as detailed in Section 5.6. Attention matrices for random examples from the test set are presented in Figure 8. Generally, we can observe that attention "fires" more intensely at retrieving tokens compared to Based/ReBased. This result suggests that there may still be a flaw in our kernel function that distributes attention to irrelevant tokens. We further investigate this phenomenon by analyzing the distribution of attention on the last token, as shown in Figure 9. The number of noisy tokens for the ReBased model is smaller compared to Based, but Attention exhibits superior results.

Layer normalization is the main difference between the Based and ReBased models. Therefore, it is important to analyze the parameters obtained during the training process. We logged the mean and standard deviation of the parameters across different sequence lengths. Our results can be found in Figure 10. Notably, the final parameter value is independent of the training sequence length, which can indicate that we may not need to train the model for all possible lengths. Both γ and β parameters have high standard deviation values compared to the mean absolute value. Consequently, we can assume that it is important to provide features with different scales.

[3] https://huggingface.co/datasets/EleutherAI

[4] See report

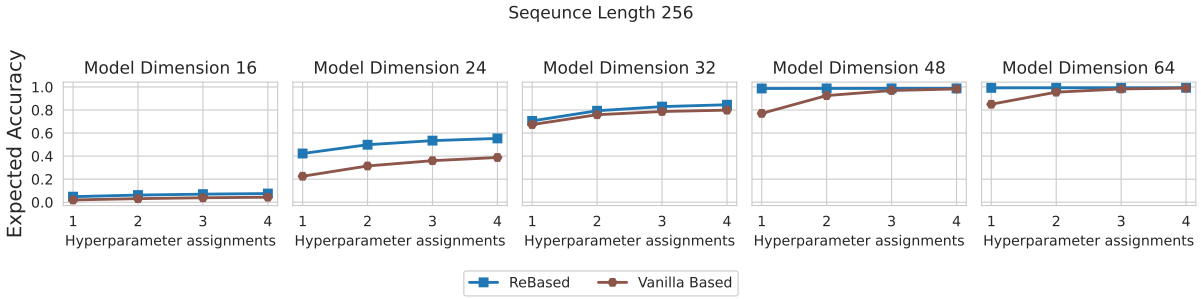


Figure 7: Expected validation accuracy across different hyperparameters. The ReBased model works best across all hyperparameters, budgets, and model dimensions. See Section 5.3 for more details.

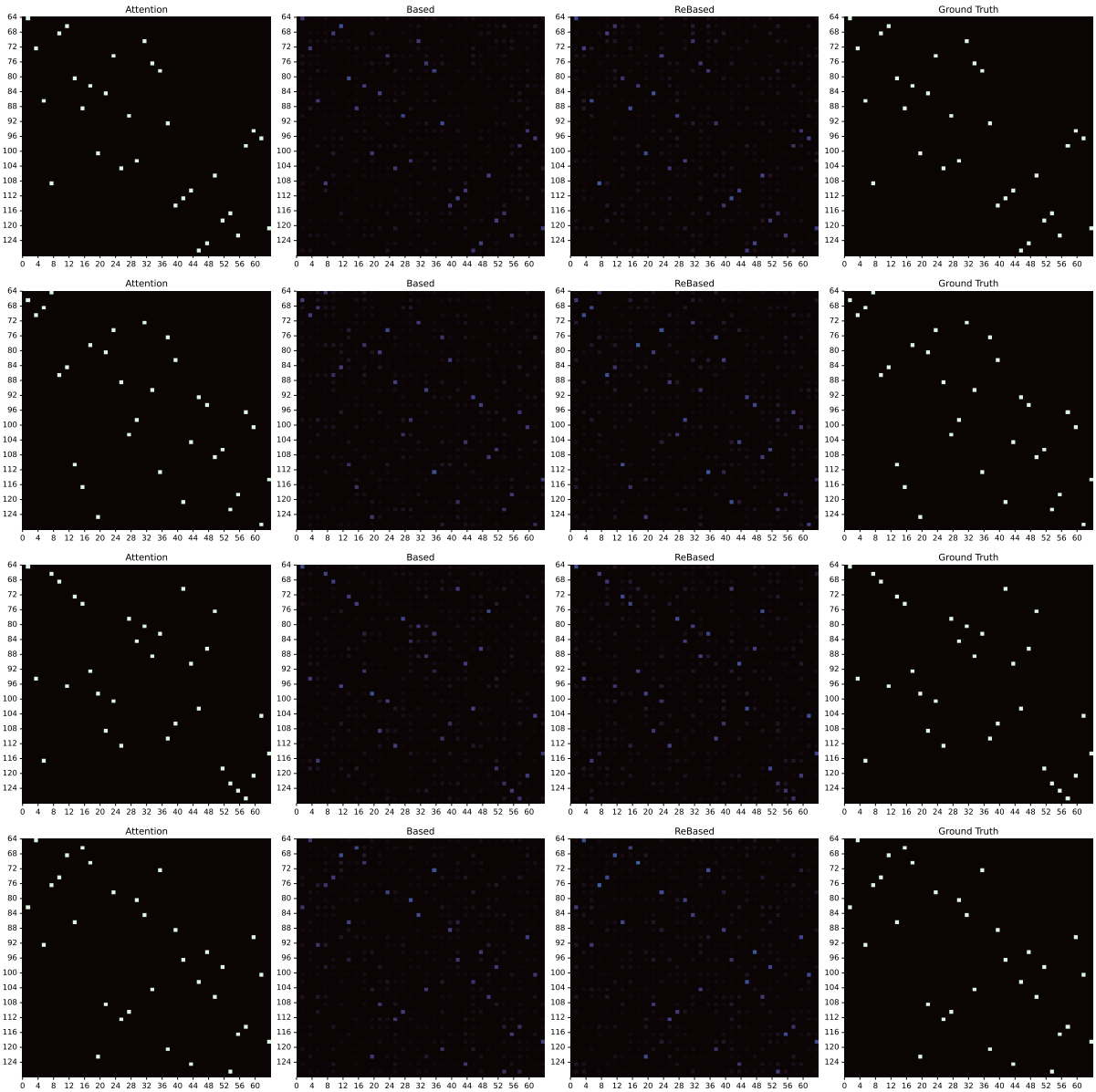


Figure 8: Attention matrix for the different models, and ground truth positions for the query. We measure IoU between the model's attention and ground truth matrix for 10000 examples. llustration of the experiment is described in Section 5.6 Results are presented in Table 5.

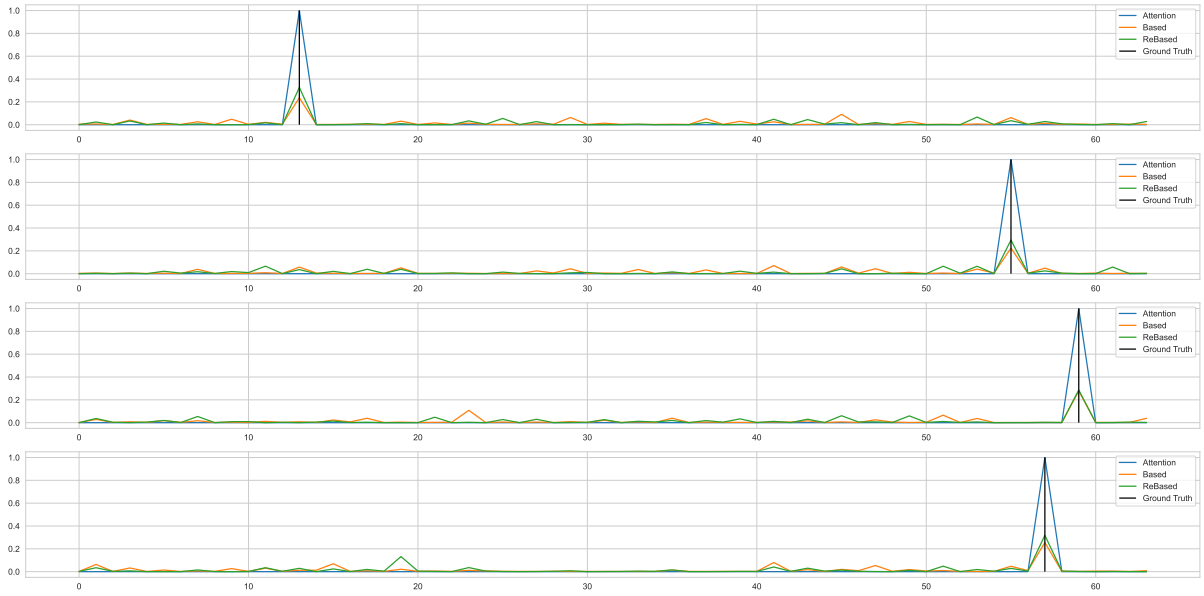


Figure 9: Attention scores for a random example. Based and Rebased scores are noisy, while attention has one peak at the ground truth position. See Appendix A.

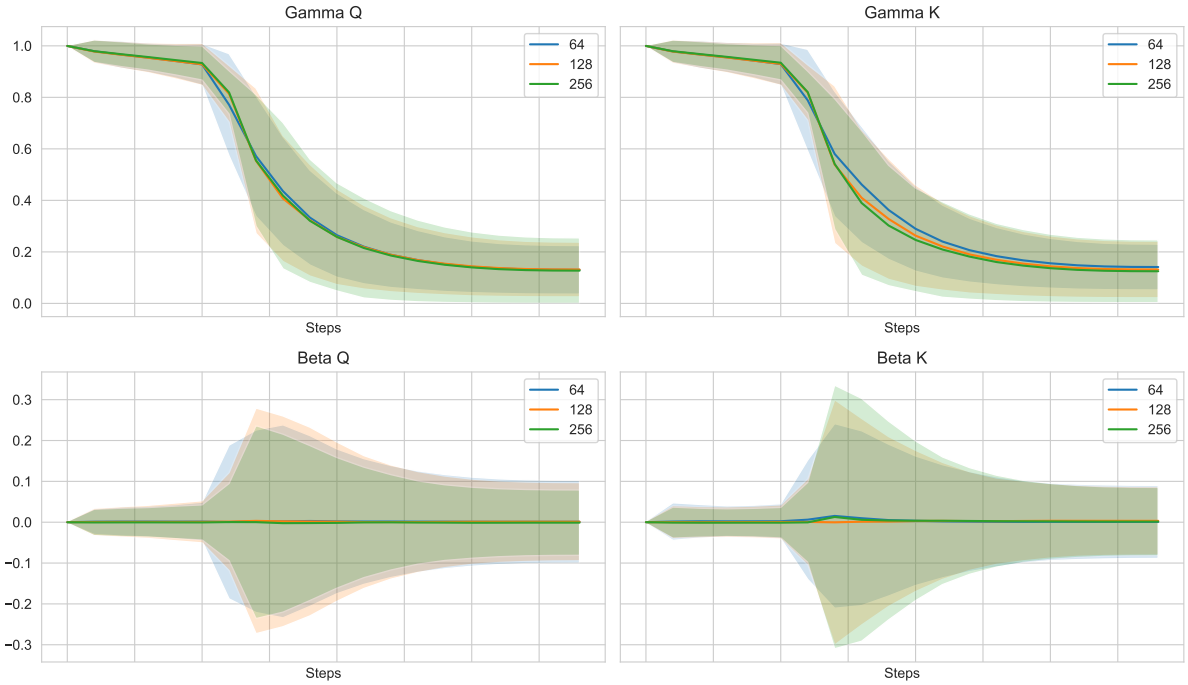


Figure 10: Analysis of the layer normalization parameters. Mean value of the scale parameter (gamma) tends to the same value of about 0.13, and the bias parameter (beta) tends to 0. See Section D.